

Statistical Analysis of Massive Data Streams: Proceedings of a Workshop



Committee on Applied and Theoretical Statistics,
National Research Council

ISBN: 0-309-59302-6, 395 pages, 8 1/2 x 11, (2004)

**This PDF is available from the National Academies Press at:
<http://www.nap.edu/catalog/11098.html>**

Visit the [National Academies Press](http://www.nap.edu) online, the authoritative source for all books from the [National Academy of Sciences](http://www.nap.edu), the [National Academy of Engineering](http://www.nap.edu), the [Institute of Medicine](http://www.nap.edu), and the [National Research Council](http://www.nap.edu):

- Download hundreds of free books in PDF
- Read thousands of books online for free
- Explore our innovative research tools – try the “[Research Dashboard](#)” now!
- [Sign up](#) to be notified when new books are published
- Purchase printed books and selected PDF files

Thank you for downloading this PDF. If you have comments, questions or just want more information about the books published by the National Academies Press, you may contact our customer service department toll-free at 888-624-8373, [visit us online](#), or send an email to feedback@nap.edu.

This book plus thousands more are available at <http://www.nap.edu>.

Copyright © National Academy of Sciences. All rights reserved.
Unless otherwise indicated, all materials in this PDF File are copyrighted by the National Academy of Sciences. Distribution, posting, or copying is strictly prohibited without written permission of the National Academies Press. [Request reprint permission for this book](#).

STATISTICAL ANALYSIS OF MASSIVE DATA STREAMS

Proceedings of a Workshop



Committee on Applied and Theoretical Statistics
Division on Engineering and Physical Sciences
NATIONAL RESEARCH COUNCIL

OF THE NATIONAL ACADEMIES

THE NATIONAL ACADEMIES PRESS

Washington, D.C.

www.nap.edu

THE NATIONAL ACADEMIES PRESS 500 Fifth Street, N.W. Washington, DC 20001

NOTICE: The project that is the subject of this report was approved by the Governing Board of the National Research Council, whose members are drawn from the councils of the National Academy of Sciences, the National Academy of Engineering, and the Institute of Medicine. The members of the committee responsible for the report were chosen for their special competences and with regard for appropriate balance. This study was supported by the National Security Agency (Grant #MDA904-02-1-0114), the Office of Naval Research (Grant #N00014-02-1-0860), and Microsoft (Grant #2327100). Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the organizations or agencies that provided support for the project.

International Standard Book Number 0-309-09308-2 (POD)

International Standard Book Number 0-309-54556-0 (PDF)

Additional copies of this report are available from the National Academies Press, 500 Fifth Street, N.W., Lockbox 285, Washington, DC 20055; (800) 624-6242 or (202) 334-3313 (in the Washington metropolitan area); Internet, <http://www.nap.edu>

Copyright 2004 by the National Academy of Sciences. All rights reserved.

Printed in the United States of America

COVER ILLUSTRATIONS: The terms “data streams” and “data rivers” are used to describe sequences of digitally encoded signals used to represent information in transmission. The left image is of the Oksrukuyik River in Alaska and the right image is an example of a crashing wave, similar to the largest recorded tsunami on Siberia’s Kamchatka Peninsula. Both images illustrate the scientific challenge of handling massive amounts of continuously arriving data, where often there is so much data that only a short time window’s worth is economically storable. The Oksrukuyik River photo is courtesy of Karie Slavik of the University of Michigan Biological Station; the tsunami photo is courtesy of the U.S. Naval Meteorology and Oceanography Command and was obtained from its Web site. Both images are reprinted with permission.

THE NATIONAL ACADEMIES

National Academy of Sciences
National Academy of Engineering
Institute of Medicine
National Research Council

The **National Academy of Sciences** is a private, nonprofit, self-perpetuating society of distinguished scholars engaged in scientific and engineering research, dedicated to the furtherance of science and technology and to their use for the general welfare. Upon the authority of the charter granted to it by the Congress in 1863, the Academy has a mandate that requires it to advise the federal government on scientific and technical matters. Dr. Bruce M. Alberts is president of the National Academy of Sciences.

The **National Academy of Engineering** was established in 1964, under the charter of the National Academy of Sciences, as a parallel organization of outstanding engineers. It is autonomous in its administration and in the selection of its members, sharing with the National Academy of Sciences the responsibility for advising the federal government. The National Academy of Engineering also sponsors engineering programs aimed at meeting national needs, encourages education and research, and recognizes the superior achievements of engineers. Dr. Wm. A. Wulf is president of the National Academy of Engineering.

The **Institute of Medicine** was established in 1970 by the National Academy of Sciences to secure the services of eminent members of appropriate professions in the examination of policy matters pertaining to the health of the public. The Institute acts under the responsibility given to the National Academy of Sciences by its congressional charter to be an adviser to the federal government and, upon its own initiative, to identify issues of medical care, research, and education. Dr. Harvey V. Fineberg is president of the Institute of Medicine.

The **National Research Council** was organized by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and advising the federal government. Functioning in accordance with general policies determined by the Academy, the Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in providing services to the government, the public, and the scientific and engineering communities. The Council is administered jointly by both Academies and the Institute of Medicine. Dr. Bruce M. Alberts and Dr. Wm. A. Wulf are chair and vice chair, respectively, of the National Research Council.

www.national-academies.org

COMMITTEE ON APPLIED AND THEORETICAL STATISTICS

EDWARD J. WEGMAN, *Chair*, George Mason University
DAVID BANKS, Duke University
ALICIA CARRIQUIRY, Iowa State University
THOMAS COVER, Stanford University
KAREN KAFADAR, University of Colorado at Denver
THOMAS KEPLER, Duke University
DOUGLAS NYCHKA, National Center for Atmospheric Research
RICHARD OLSON, Stanford University
DAVID SCOTT, Rice University
EDWARD C. WAYMIRE, Oregon State University
LELAND WILKINSON, SPSS, Inc.
YEHUDA VARDI, Rutgers University
SCOTT ZEGER, Johns Hopkins University School of Hygiene and Public Health

Staff

BMSA Workshop Organizers
Scott Weidman, BMSA Director
Richard Campbell, Program Officer
Barbara Wright, Administrative Assistant
Electronic Report Design
Sarah Brown, Research Associate
Meeko Oishi, Intern

ACKNOWLEDGEMENT OF REVIEWERS

This report has been reviewed in draft form by individuals chosen for their diverse perspectives and technical expertise, in accordance with procedures approved by the Report Review Committee of the National Research Council (NRC). The purpose of this independent review is to provide candid and critical comments that will assist the institution in making its published report as sound as possible and to ensure that the report meets institutional standards for objectivity, evidence, and responsiveness to the study charge. The review comments and draft manuscript remain confidential to protect the integrity of the deliberative process. We wish to thank the following individuals for their review of this report:

Amy Braverman, Jet Propulsion Laboratory
Ron Fedkiw, Stanford University
David Madigan, Rutgers University
Jennifer Rexford, AT&T Laboratories

Although the reviewers listed above have provided many constructive comments and suggestions, they were not asked to endorse the conclusions or recommendations, nor did they see the final draft of the report before its release. Responsibility for the final content of this CD report rests entirely with the authoring committee and the institution.

Preface and Workshop Rationale

On December 13 and 14, 2002, the Committee on Applied and Theoretical Statistics of the National Research Council conducted a two-day workshop that explored methods for the analysis of streams of data so as to stimulate further progress in this field. To encourage cross-fertilization of ideas, the workshop brought together a wide range of researchers who are dealing with massive data streams in different contexts. The presentations focused on five major areas of research: atmospheric and meteorological data, high-energy physics, integrated data systems, network traffic, and mining commercial streams of data.

The workshop was organized to allow researchers from different disciplines to share their perspectives on how to use statistical methods to analyze massive streams of data, so as to stimulate cross-fertilization of ideas and further progress in this field. The meeting focused on situations in which researchers are faced with massive amounts of data arriving continually, making it necessary to perform very frequent analyses or reanalyses on the constantly arriving data. Often there is so much data that only a short time window's worth may be economically stored, necessitating summarization strategies.

The overall goals of this CD report are to improve communication among various communities working on problems associated with massive data streams and to increase relevant activity within the statistical sciences community. Included in this report are the agenda of the workshop, the full and unedited text of the workshop presentations, and biographical sketches of the speakers. The presentations represent independent research efforts on the part of academia, the private sector, federally funded laboratories, and government agencies, and as such they provide a sampling rather than a comprehensive examination of the range of research and research challenges posed by massive data streams. In addition to these proceedings, a set of more rigorous, technical papers corresponding to the workshop presentations has also been published separately as a 2003 special issue of the *Journal of Computational and Graphical Statistics*.

This proceedings represents the viewpoints of its authors only and should not be taken as a consensus report of the Board on Mathematical Sciences and Their Applications or the National Research Council.

Committee on Applied and Theoretical Statistics
STATISTICAL ANALYSIS OF MASSIVE DATA STREAMS

National Research Council

Washington, D.C.

December 13 and 14, 2002

December 13

Welcome and Overview of Sessions

Sallie Keller-McNulty, *Los Alamos National Laboratory* Chair, Committee on Applied and Theoretical Statistics

James Schatz, *National Security Agency*

Session 1. Atmospheric and Meteorological Data

Douglas Nychka, Session Chair, *National Center for Atmospheric Research* Introduction

John Bates, *National Climatic Data Center* Exploratory Climate Analysis Tools for Environmental Satellite and Weather Radar Data

Amy Braverman, *Jet Propulsion Laboratory* Statistical Challenges in the Production and Analysis of Remote Sensing Earth Science Data at the Jet Propulsion Laboratory

Ralph Milliff, *Colorado Research Associates* Global and Regional Surface Wind Field Inferences from Spaceborne Scatterometer Data

Report from Breakout Group

Session 2. High-Energy Physics

David Scott, Session Chair, *Rice University* Introduction

Robert Jacobsen, *Lawrence Berkeley National Laboratory* Statistical Analysis of High Energy Physics Data

Paul Padley, *Rice University* Some Challenges in Experimental Particle Physics Data Streams

Miron Livny, *University of Wisconsin-Madison* Data Grids (or A Distributed Computing View of High Energy Physics)

Report from Breakout Group

Luncheon Keynote Address

Daryl Pregibon, *AT&T Shannon Research Laboratories* Keynote Address: Graph Mining—Discovery in Large Networks

Session 3. Integrated Data Systems

Sallie Keller-McNulty, Session Chair, *Los Alamos National Laboratory* Introduction

J.Douglas Reason, *Los Alamos National Laboratory* Global Situational Awareness

Kevin Vixie, *Los Alamos National Laboratory* Incorporating Invariants in Mahalanobis Distance-Based Classifiers: Applications to Face Recognition

John Elder, *Elder Research* Ensembles of Models: Simplicity (of Function) Through Complexity (of Form)

Report from Breakout Group

After-Dinner Address

Mark Hansen, *Bell Laboratories* Announcement from the Whitney Museum of American Art Untitled Presentation

December 14

Session 4. Network Traffic

Wendy Martinez, *Session Chair, Office of Naval Research* Introduction

William Cleveland, *Bell Laboratories* FSD Models for Open-Loop Generation of Internet Packet Traffic

Johannes Gehrke, *Cornell University* Processing Aggregate Queries over Continuous Data Streams

Edward Wegman, *George Mason University* Visualization of Internet Packet Headers

Paul Whitney, *Pacific Northwest National Laboratory* Toward the Routine Analysis of Moderate to Large-Size Data

Report from Breakout Group

Session 5. Mining Commercial Streams of Data

Leland Wilkinson, *Session Chair, SPSS, Inc.* Introduction

Lee Rhodes, *Hewlett-Packard Laboratories* A Stream Processor for Extracting Usage Intelligence from High-Momentum Internet Data

Pedro Domingos, *University of Washington* A General Framework for Mining Massive Data Streams

Andrew Moore, *Carnegie Mellon University* Kd-R-Ball-and Ad-Trees: Scalable Massive Science Data Analysis

Report from Breakout Group

Sallie Keller-McNulty

Welcome and Overview of Sessions

Transcript of Presentation

BIOSKETCH: Sallie Keller-McNulty is group leader for the Statistical Sciences Group at Los Alamos National Laboratory. Before she moved to Los Alamos, Dr. Keller-McNulty was professor and director of graduate studies at the Department of Statistics, Kansas State University, where she has been on the faculty since 1985. She spent 2 years between 1994 and 1996 as program director, Statistics and Probability, Division of Mathematical Sciences, National Science Foundation. Her on-going areas of research focus on computational and graphical statistics applied to statistical databases, including complex data/model integration and related software and modeling techniques, and she is an expert in the area of data access and confidentiality. Dr. Keller-McNulty currently serves on two National Research Council committees, the CSTB Committee on Computing and Communications Research to Enable Better Use of Information Technology in Government and the Committee on National Statistics' Panel on the Research on Future Census Methods (for Census 2010), and chairs the National Academy of Sciences' Committee on Applied and Theoretical Statistics. She received her PhD in statistics from Iowa State University of Science and Technology. She is a fellow of the American Statistical Association (ASA) and has held several positions within the ASA, including currently serving on its board of directors. She is an associate editor of *Statistical Science* and has served as associate editor of the *Journal of Computational and Graphical Statistics* and the *Journal of the American Statistical Association*. She serves on the executive committee of the National Institute of Statistical Sciences, on the executive committee of the American Association for the Advancement of Science's Section U, and chairs the Committee of Presidents of Statistical Societies. Her Web page can be found at <http://www.stat.lanl.gov/people/skeller.shtml>

TRANSCRIPT OF PRESENTATION

MS. KELLER-MCNULTY: Okay, I would like to welcome everybody today. I am Sallie Keller-McNulty. I am the current chair of the Committee on Applied and Theoretical Statistics. This workshop is actually sponsored by CATS. That is the acronym for our committee. It is kind of a bit of a déjà vu looking out into this room, back to 1995, the nucleus of people who held the first workshop, or at least attended the first workshop that CATS had, on the analysis of massive data sets. It has taken us a while to put a second workshop together. In fact, as CATS tried to think about what makes sense for a workshop today, that really deals with massive amounts of data, is where we decided we would really try to actually jump ahead a bit and try to look at problems of streaming data, massive data streams.

Now, the workshop committee, which consisted of David Scott, Lee Wilkinson, Bill DuMouchel and Jennifer Widom, when they started planning this, they were pretty comfortable with the concept of massive data streams.

I think that, by the time that this actually got together, they debated whether, instead of data streams, it should be data rivers. Several of you have asked me what constitutes a stream, how fast does the data have to flow. I am not qualified to answer that question, but I think our speakers throughout the day should be able to try to address what that means to them.

We need to give a really good thank you to our sponsors for this workshop, which is the Office of Naval Research and the National Security Agency. Now I will turn it over to Jim Schatz from NSA. He will give us an enlightening, boosting talk for the workshop.

James Schatz

Welcome and Overview of Sessions

[Transcript of Presentation](#)

James Schatz is the chief of the Mathematics Research Group at the National Security Agency.

TRANSCRIPT OF PRESENTATION

MR. SCHATZ: Thanks, Sallie. I am the chief of the math research office at NSA. The sponsorship of the conference here comes from an initiative that we have at the agency called Advanced Research and Development Activity, and Dr. Dean Collins is the head of that, whom I think some of you probably met at one of these conferences last year. We are very happy and proud to be part of the sponsorship and so forth.

I am only going to talk for a few minutes, but interrupt with questions as needed, in the spirit of what I think you are here for, which is not lecturing to each other, but having a dialogue on things.

Of course, I don't think it is a big secret why the National Security Agency is interested in massive data sets. I don't know what the stream rate for massive data sets is either, but I think we are going to make it at our place.

Let me dwell on the obvious here just for a few minutes. As we look back over this past year, of course, a big question for us, not only for us as individuals, but for the government in the form of official commissions and so forth is, could we have prevented 9/11.

We look back on that at a kind of obvious point. There is a question, was there a message in our collection somewhere that said, attack at dawn, with all the details and so forth? While we certainly have done a due diligence search of everything we can lay our hands on that was available to us prior to 9/11 and we haven't found such a transmission, another type of question, though, that probably bothers us a lot more is if we got one last night that said, attack at dawn, would we have noticed it?

We have so much data to look at, and we don't look at the data. Our computers do first. So, if the computers don't find that message, and that means if the algorithms don't find that message, the analysts aren't going to read that message. That is just sort of the beginning part, of course, the most obvious. Already, it gets us into huge questions about what is the nature of our databases, how do we store data, how do we retrieve data.

Of course, in the past year of really being thrown into a whole new paradigm of intelligence analysis and so forth, we are more in the situation of asking the question, okay, we are probably not going to be fortunate enough to have a message that says, attack at dawn. What we are going to have to do is find clues in a massive data set and put that together and, to do that, that there is something happening tomorrow morning.

It has really been a huge change for us. It is not that we weren't thinking about massive data sets before; of course we were. When you are traditionally, after decades and decades, looking at well-defined nation-state targets, like Iraq, and you would—your way of approaching the analysis is sort of dictated by the fact that there is a country with national boundaries and a military and diplomats and various things like that to worry about.

We were certainly aware of terrorist targets and studying them and worried about them and taking action long before 9/11, but of course, an event like that pumps up the level of urgency just beyond anything else that you could do. The postmortem stuff of analyzing what we could have done or would have done will, of course, go on for decades, just like today you still hear people talking about, could we have prevented Pearl Harbor. I think, 50 years from now, those same questions will be being asked. Of course, we are here now and we have to worry about the future, for sure.

I looked over the topics here and the group. It is a wonderful group of people. I am even starting to recognize lots of names and people who are good friends like David Scott, of course, who has been working with us for so long.

I am not one of the experts in the group, but I know that we have got a good pile of the experts here. Even if you are interested in environmental data or physics data, of course, there is one thing that you don't have to worry about with that data, I hope, which is that a good portion of it is encrypted. Even if we get past the encryption problem and say, supposed that all of our data is unencrypted, you probably do have to deal with some amount of garbling and that sort of stuff in your data, too.

I imagine that we are all looking at the same kinds of questions, which are, there are certain events in our data that we would like to be able to spot and just pull out, because we know what they are going to be, and we just want to have rapid ways to do that. I think the real sense of where the science is going, at least for us, and I think for you is, how do we take a massive data set and see pieces of what we need to see in many different places and pull it together and make conclusions based on that, and how do we find patterns?

For us, a key aspect of this problem, of course, is we don't have a homogeneous type of a data set. We have got any kind of communications medium that you can imagine, we will have that data. A key problem for us is kind of combining information from all these different things, and database structures for how you would actually set things up to make the algorithms run in a situation like that.

Certainly, Kay Anderson and Dave Harris, who are from our group are here today, were working on these types of problems long ago. It didn't start a year ago, but post 9/11, we have ramped up dramatically our efforts in these areas. S&T in the research area alone, there are dozens of people working on this who weren't working on it a year ago.

We have certainly got tons to learn about this stuff. It just seems, with the data explosion that everybody is going through, we are all kind of new at it, in a sense.

I hope, in the spirit of these conferences, our guys will find ways to share technical things with you as best they can, and that even with all your proprietary information that you have to worry about, you can have a good technical exchange back with us. It is certainly an area where there are a lot of economic issues, and companies have ways of doing things and so forth, but hopefully the in-crowd here can get down to the mathematics and the statistics and share ideas.

We need a lot of help in this area. What we are doing is dramatically good. We have had some amazing success stories just in the past year that were an absolute direct result of what I would call data mining on massive data sets.

I can assure you, for us, it is not just an academic exercise. We are right in the thick of it. We utilize it every day. It has done wonderful stuff for us in the past year, and we are getting a lot out of these conferences. I popped into one last year, and I am glad to see a lot of the same faces.

AUDIENCE: [Question off microphone.]

MR. SCHATZ: Probably not, but I do want you to know that I am not just saying that to make you feel good. We really have had some dramatic successes in terms of techniques we didn't have a year ago for looking for patterns in massive data, drawing conclusions and taking some known attributes of a situation and mining through the data

to find new ones, and very algorithmic based, and really providing tools for our analysts.

Of course, however many analysts we have—and I wouldn't know what that number is, it is finite, and any given human being can only look at so much text and pictures in one day.

For us, it is all about teaching the machines how to work for us, and teaching the machines is teaching the algorithms. I can't think of an example that we could share with you, but real examples, real intelligence, real impact, plenty of it, just in this past year, based on the kinds of techniques we are learning with you.

Anyway, I don't want to overstay my welcome, because there is some real work to do, but if there are a couple more questions, I would be happy to talk.

AUDIENCE: [Question off microphone.]

MR. SCHATZ: Certainly, gigabytes on a daily basis and so forth. Maybe our experts will find a way they can give you a better sense of that. I don't really know.

The thing is, we have lots of channels coming in at lots of rates, and if you put it all together, it would be something astronomical. We probably span every range of problems you could think of. It is not as though we have the mother lode coming in through one pipe every minute. We have lots of ways of collecting lots of sources.

I am sure some of our most important stuff is very slow speed compared to the things you are talking about, and some of it is very high speed.

There isn't any kind of one technique that we are looking for, and any range of techniques here—you know, something that takes longer and has to work at slower speeds is probably just as interesting to us as something that has to work at the speed of light, we are going to have all kinds of problems to apply this stuff to.

Anything else I could give a vague kind of government answer to? Okay, Sallie, you are back, or maybe John is up, and thanks for being here, and we are happy to be part of this, and thanks for the research.

Douglas Nychka, Chair of Session on Atmospheric and Meteorological Data

Introduction by Session Chair

Transcript of Presentation



BIOSKETCH: Douglas Nychka is a senior scientist at the [National Center for Atmospheric Research](#) (NCAR) and is also the project leader for the [Geophysical Statistics Project](#) (GSP). He works closely with many of the postdoctorate fellows at NCAR and his primary goal is to emphasize interdisciplinary research: migrating statistical techniques to important scientific problems and using these problems to motivate statistical research. Dr. Nychka's personal research interests include nonparametric regression (mostly splines), and statistical computing, spatial statistics, and spatial designs. He received his undergraduate degree from Duke University in mathematics and physics and his PhD from the University of Wisconsin under the direction of Grace Wahba. He came to GSP/NCAR in 1997 after spending 14 years as a faculty member in the Statistics Department at North Carolina State University.

TRANSCRIPT OF PRESENTATION

MR. NYCHKA: So, without further ado, our first speaker is going to be John Bates at the National Climatic Data Center. He will be talking about exploratory climate analysis and environmental satellites and weather radar data.

John Bates

Exploratory Climate Analysis Tools for Environmental Satellite and Weather Radar Data

[Abstract of Presentation](#)

[Transcript of Presentation and PowerPoint Slides](#)



BIOSKETCH: John J. Bates is the chief of the Remote Sensing Applications Division of the U.S. National Oceanic and Atmospheric Administration's (NOAA's) National Climatic Data Center. Dr. Bates received a PhD in meteorology from the University of Wisconsin-Madison in 1986 under William L. Smith on the topic of satellite remote sensing of air-sea heat fluxes. Dr. Bates then received a postdoctoral fellowship at Scripps Institution of Oceanography (1986–1988) to work with the California Space Institute and the Climate Research Division. He joined the NOAA Environmental Research Laboratories in Boulder, Colorado, in 1988 and there continued his work in applying remotely sensed data to climate applications. In 2002, Dr. Bates moved to the NOAA National Climatic Data Center in Asheville, North Carolina.

Dr. Bates' research interests are in the areas of using operational and research satellite data and weather radar data to study the global water cycle and studying interactions of the ocean and atmosphere. He has authored over 25 peer-reviewed journal articles on these subjects. He served on the AMS Committee on Interaction of the Sea and Atmosphere (1987–1990) and the AMS Committee on Applied Radiation (1991–1994).

As a member of the U.S. National Research Council's Global Energy and Water Cycle Experiment (GEWEX) Panel (1993–1997), Dr. Bates reviewed U.S. agency participation and plans for observing the global water cycle. He was awarded a 1998 Editors' Citation for excellence in refereeing *Geophysical Research Letters* for "thorough and efficient reviews of manuscripts on topics related to the measurement and climate implications of atmospheric water vapor." He has also been a contributing author and U.S. government reviewer of the Intergovernmental Panel on Climate Change Assessment Reports. He currently serves on the International GEXEX Radiation Panel, whose goal is to bring together theoretical and experimental insights into the radiative interactions and climate

feedbacks associated with cloud processes, including the effects of water vapor within the atmosphere and at Earth's surface.

ABSTRACT OF PRESENTATION

Exploratory Climate Analysis Tools for Environmental Satellite and Weather Radar Data John Bates, National Climatic Data Center

1. Introduction

Operational data from environmental satellites form the basis for a truly global climate observing system. Similarly, weather radar provides the very high spatial and rapid time sampling of precipitation required to resolve physical processes involved in extreme rainfall events. In the past, these data were primarily used to assess the current state of the atmosphere to help initialize weather forecast models and to monitor the short-term evolution of systems (called nowcasting).

The use of these data for climate analysis and monitoring is increasing rapidly. So, also, are the planning and implementation for the next generation of environmental satellite and weather radar programs. These observing systems challenge our ability to extract meaningful information on climate variability and trends. In this presentation, I will attempt only to provide a brief glimpse of applications and analysis techniques used to extract information on climate variability. First, I will describe the philosophical basis for the use of remote sensing data for climate monitoring, which involves the application of the forward and inverse forms of the radiative transfer equation. Then I will present three examples of the application of statistical analysis techniques to climate monitoring: (1) the detection of long-term climate trends, (2) the time-space analysis of very large environmental satellite and weather radar data sets, and (3) extreme event detection. Finally, a few conclusions will be given.

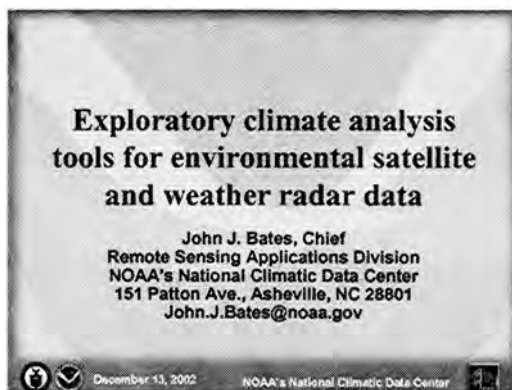
2. Philosophy of the use of remote sensing data for climate monitoring

Remote sensing involves the use of active or passive techniques to measure different physical properties of the electromagnetic spectrum and to relate those observations to more traditional geophysical variables such as surface temperature and precipitation. Passive techniques use upwelling radiation from the Earth-atmosphere system in discrete portions of the spectrum (e.g., visible, infrared, and microwave) to retrieve physical properties of the system. Active techniques use a series of transmitted and returned signals to retrieve such information.

This is done by using the radiative transfer equation in the so-called forward and inverse model solutions. In the forward problem, sample geophysical variables, such as surface temperature and vertical temperature and moisture profiles, are input to the forward radiative transfer model. In the model, this information is combined with specified instrument error characteristics and responsivity to produce simulated radiances. The inverse radiative transfer problem starts with satellite-observed radiances. Because the inverse radiative transfer equation involves taking the inverse of an ill-conditioned matrix, a priori information, in the form of a first guess of the solution, is required to stabilize the matrix prior to inversion. The output of this process is geophysical

retrievals. The ultimate understanding of the satellite or radar data requires full application of the forward and inverse problems and the impact of uncertainties associated with each step in the process.

TRANSCRIPT OF PRESENTATION



MR. BATES: Thank you. I didn't think we would be in the big room. It is nice to be in this building. I am going to mainly talk about some of our larger so-called massive data sets that we acquire now over the wire from both environmental satellites—the ones you see on the television news every night, the geostationary satellites.

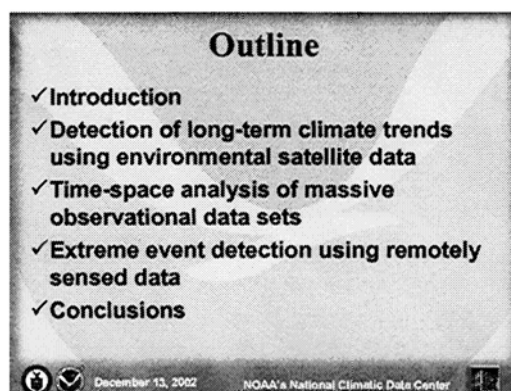
NOAA, as well as Department of Defense, also operate polar-orbiting—that is, satellites that go pole to pole and scan across a swath of data on a daily basis. Also, right now, our biggest data stream coming in is actually the weather radar data, the precipitation animations that you see now nightly on your local news. In talking in terms of what we just heard, in terms of the different data sets that come in, they come in from all different sources.

The National Climatic Data Center is the official repository in the United States of all atmospheric weather-related data. As such, we get things like simple data streams, the automatic observing systems that give you temperature, moisture, cloud height at the Weather Service field offices. Those are mostly co-located now at major airports for terminal forecasting, in particular.

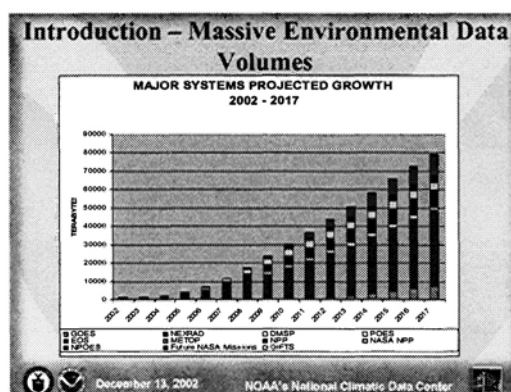
We have, in the United States, a set of what are called cooperative observers, about 3,000 people who have their own little backyard weather station, but it is actually an officially calibrated station. They take reports. Some of them phone them in, and deposit the data, and that is a rather old style way of doing things.

We have data that comes in throughout the globe, reports like that, upper-air reports from radiosondes, and then the higher data now are the satellite and weather radar data. The United States operates nominally two geostationary satellites, one at 75 watts, one at 135 watts. The Japanese satellite, which is at 155 degrees East, is failing. So, we are in the process of actually moving one of our United States satellites out there. Then, of course, these polar-orbiting satellites.

I am mostly going to talk about the polar-orbiting satellites and some techniques we have used to analyze those data for climate signals. Those data sets started in about 1979, late 1978, and then go through the present.



This is what I want to talk about today, just give you a brief introduction of what we are thinking about as massive data is coming in, and we are responsible for the past data as well as the future planning for data coming in from the next generation of satellites. A couple, three examples of how we use some techniques, sometimes rather simplistic, but powerful, to look at the long-term climate trends, some time-space analysis—that is, when you have these very high spatial and temporal data sets, you would like to reduce the dimensionality, but yet still get something meaningful out about the system that we are trying to study. Then, just briefly talk about amplification of the radar data. I just inherited the radar data science there, and so, that is new stuff, and it has just really begun in terms of data mining. So, when you have rare events in the radar such as tornadic thunderstorms, how can we detect those. Then, just a couple of quick conclusions.

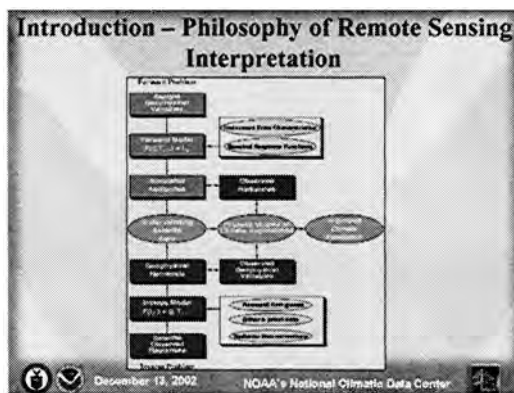


That is what we are talking about in terms of massive here. So, the scale is time from about 2002 projecting out about the next 15 years or so. This is probably conservative because we are re-examining this and looking at more data, probably, more products being generated than we had considered before. On the axis here is terabytes, because people aren't really thinking of pedabytes. Those numbers are really 10, 20, 30 pedabytes. Right now, we have got a little over one pedabyte and daily we are probably ingesting something like a terabyte.

The biggest data set coming in now is that we are getting the next radar data—this is the weather radar data—from about 120 sites throughout the United States. We are getting about a third of that in real time. They used to come in on the little Xabite 8 millimeter cassettes. For years, we used to just have boxes of those because there wasn't

a funded project to put this on mass store. In the last two years, we have had eight PC work stations with each having eight readers, tape readers, on them, to read back through all the data and get it into mass store. So, now we can get our hands on it.

So, the first lesson is accessibility of the data, and we are really just in a position now to be going through the data, because it is accessible. We are looking at data rates by 2010, on the order of the entire archive—this is cumulative archives. So, that is not data read per year, so it is cumulative archive building up, of something over 30 pedabytes by the year 2010 or so. So, that is getting fairly massive.



In terms of remote sensing, there is a philosophical approach, and I am not sure how many of you have worked with remote sensing data. There are two ways of looking at the data, sort of the data in the satellite observation coordinates or the geophysical space of a problem you want to deal with.

These are referred to variously as the forward problem. Just very briefly, the forward problem, you have geophysical variables—temperature and moisture profiles of the atmosphere, the surface temperature, and your satellite is looking down into that system. So, using a forward model—a forward model being a radiative transfer model, physical model for radio transfer in the atmosphere—you can simulate so-called radiances.

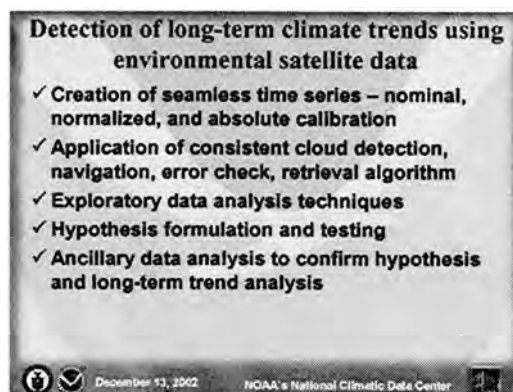
The radiances are what the satellite will actually observe. In the middle are those ovals that we want to actually work on, understanding the satellite data and then understanding the processes of climate, and then, in fact, improving modeling. As an operational service and product agency, NOAA is responsible for not just analyzing what is going on but, foolishly, we are attempting to predict things. Analogous to other businesses, we are in the warning business. The National Weather Service, of course, is bold enough to issue warnings.

However, when you issue warnings, you also want to look at things like false alarm rate. That is, you don't want to issue warnings when, in fact, you don't have severe weather, tornadoes, etc.

The other aspect of the problem, the so-called inverse problem—so, starting from the bottom there—you take the satellite radiances and we have an inverse model that is usually a mathematical expression for the radio transfer equation which is non-linear. We linearize the problem. Then we have a linear set of equations. The inverse model, then, is an inverse set of equations. The matrix is usually ill conditioned. So, we go to those yellow boxes and condition the matrix by adding a priori information, a forecast

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

first guess, other a priori data, and then biases to somehow normalize the data set. We invert that to get geophysical retrieval. Then we can retrieve temperature and moisture profiles. We can retrieve surface properties, surface temperature, ocean wind speed, other geophysical quantities of interest.



So, the first application, detection of long-term climate trends using environmental satellite data, the issue of global warming has really surfaced in the last 10 years. We would like to know, is the Earth warming, how much. Are systems changing? How much? Is there more severe weather? That would just be an issue with the extremes in a distribution. You know, certain weather events are normal distributions. Certain aren't. Precipitation is not normally distributed by any sense of the imagination. We get far fewer events of extreme rainfall—precipitation—than we do of light precipitation. So, it is more of a log normal distribution. We would like to know, are the extremes changing. So, that is a small portion of those distributions.

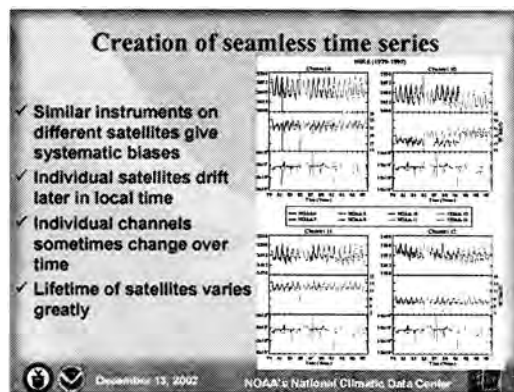
With satellite data, we face a couple of unique problems. First, we are sensing the atmosphere with satellites that have a lifetime of three to five years. So, we need to create a so-called seamless time series so that, when you apply time-space analysis techniques, you are not just picking up artifacts of the data that have to do with a different satellite coming on line. We use a three-step approach to that, something we call the nominal calibration. That is where you take an individual satellite, do the best you can with that satellite in terms of calibrating it, normalizing the satellites, and I will show you what that means. We have different satellites with different biases. Often, different empirical techniques are used to stitch those together in a so-called seamless manner. We would like an absolute calibration. That, of course, is very difficult, because what is absolute, what is the truth?

Then, we would like to apply some consistent algorithm. In the infrared, when you are remote sensing in the infrared, and you are looking down at the atmosphere from space, in the infrared, clouds are opaque. So, in order to send the temperature and moisture profile down to the surface, you have to choose or detect the cloud-free samples. So, you have to have a threshold that tells you, this is cloudy, this is clear. You can base that on a number of different characteristics about the data, usually time and spatial characteristics, time and space variability. Clouds move, the surface tends to be more constant in temperature. Not always, but the oceans certainly do. So, you use those statistical properties about changes in time and space of the data set, to allow you to identify clouds. You have to look at navigation. You have to do all kinds of error

checks, and then build retrievals to go from your radiant space to your geophysical space.

Then, we get, finally, into the fun part, exploratory data analysis. I tend to view this as sort of my tool kit out there in the shop working on a data set where, you know, you throw things at it and see what sticks. Once you get something that looks interesting, you start to formulate hypotheses about the physical system, how it works, and how your data set compares to what physics of the problems say are possible solutions. Then you might go on to look at data analysis and confirm your hypothesis.

Anyway, let's go through the first step here. I am going to take more time with this first example and a little less with the second and just briefly go into the third one.



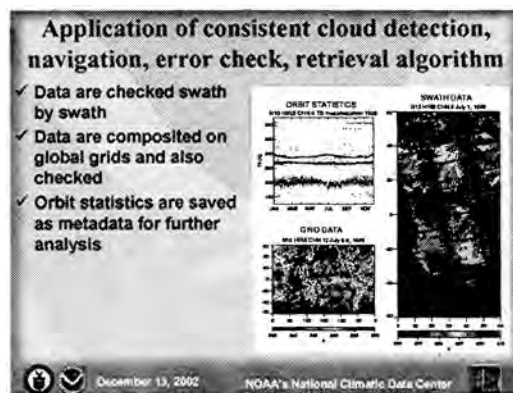
So, creation of seamless time series, you have here three different channels of data from a satellite, channel 8, that is an infrared window, channel 10 is actually a moisture channel in the upper atmosphere, a so-called water vapor channel, and these channels—10, 11, 12—are all water vapor channels. We look at emission lines of water vapor in the atmosphere. Channel 12 in particular we are going to look at because it is involved with a so-called water vapor feedback mechanism in global warming. In global warming, we hear these numbers quoted—atmospheric, oh, the temperature is going to go up two degrees in 100 years. Actually, anthropogenic CO₂ manmade gasses only contribute one of the two degrees there. The extra warming, the other one degree of warming, comes from a so-called water vapor feedback. So, there has been a lot of controversy in the community about, does this water vapor feedback, in fact, work this way or not.

So, the different colors in these three charts, then, I am showing three things. One is the average global temperature over time, and this is a 20-year data set. So, the top line in each one of these is just your monthly mean data point for each of these satellites over time, about a 20-year time series on each one. These are four different channels. The mean—you see the march of the annual cycle up and down—the standard deviation of the data set, and just simply the number of observations, these are something like millions of observations a month—you can't read that scale here, this is times 10⁶. So, on the order of, you know, 5 or 6 million or so observations a month coming down.

This is from the polar-orbiting satellite. So, these have sampled the entire planet. The geostationary only sample that region that they are over. You can see a bit of the problem here, especially in this channel 12 where, number one, there are offsets between the different colors. The different colors are the different satellites. Over this time period, there are eight different satellites. They are designated by this NOAA-7, 8, 9, 10,

etc. These are a series of the NOAA polar-orbiting satellites.

So, there are a couple of things you can pick out right away. There are biases between—this is supposed to be the same channel, but physically we know there are some differences. We can account for some of those, physically, it is just a matter of the system. We would like to seam these time series together. There are offsets and there is another problem here that you can probably see. This yellow one drifts in time, while the satellite crossing time is actually drifting in time later in the day. This can be problematic, depending on what you are trying to study. So, we would like to stitch those time series together to get a seamless time series, and then do time series analyses.



So, it takes a lot of checking. Over here, these are individual swaths of data, so, swath one, two, three. This is the Middle East. This is Saudi Arabia, the Red Sea. This is Africa, South Africa here. The different colors denote the different temperatures that the system is radiating at. Then, we have several other things going on here.

We have already applied the cloud detection algorithm. So, the spotty pixilation of these swaths, the dark spots are where we have detected clouds and then not put a color in. So, you only see color where we have detected already that it is clear. Then you have another process you are banding here. The instrument is calibrated every 40 lines. So, instead of looking at the Earth, it looks inside the housing at black bodies with constant temperatures, so it can get an absolute calibration every 40 lines.

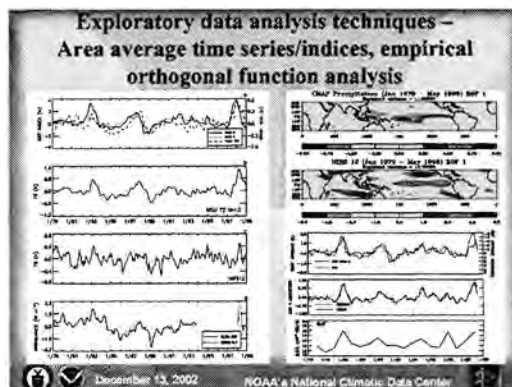
So, this is a couple of swaths. There are 14 swaths a day for each satellite. You start to composite them together in a global view now. So, this is the global area, this is the Americas here, North and South America, the outlines are in white of the continental land boundaries, the Pacific Ocean. Here, again, we have color-coded the radiant temperatures. Dark areas are missing. That is persistent cloudy areas. Then, we have gridded these together for a five-day period where we can start to evaluate that visually also for quality.

Then, this is a long-term sort of diagram of the health of the satellite data set again. These are just simple statistical quantities, but very helpful for scanning out bad data. What we have here plotted are simply the mean in red, of each swath, for an entire year. So, we just compute the mean of each channel, pretty simple, the maximum that we detect and the minimum that we detect.

You can start to see, when you do this, right away you get outliers, and those outliers occur preferentially in different seasons. As this satellite goes around, seasonally, you will see different things, and you tend to have problems. This already

allows us to throw a lot of data out that we have found is out of bounds. On the other hand, if you are looking for abnormal things, this may be the data you are interested in.

For us, we know this is the bad data. We don't want that. So, we composite them, we obtain metadata, and we save that for further analysis. Now that we have seamed everything together, what do we want to do? Well, we want to try to get a handle on what the system is doing.



What we have done here is composite a bunch of different analyses together about the system. The top two panels are the spatial patterns of empirical orthogonal analysis of precipitation, and then this is water vapor. What we have done here is that we have subtracted the annual harmonics from the time series of the data sets, so that we can look at interregional variability. So, it is just a simple filtering technique. What we have done is, we have fit the first three harmonics in the annual cycle to all the data sets, to every point in the Tropics. So, these are 30 degrees North/South now. We subtracted those out. Then, we have done empirical orthogonal analysis on monthly mean data. This is precipitation.

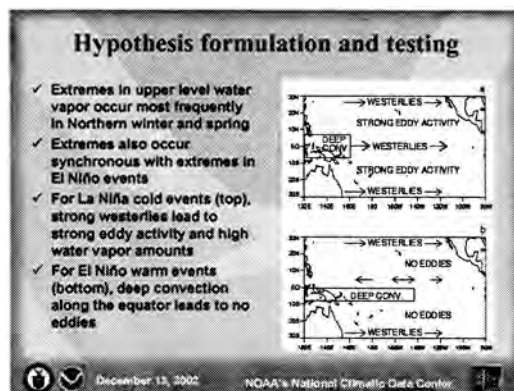
This is the so-called El Niño swing in the Pacific. So, during warm events with El Niño, you have less convection and precipitation in the west Pacific and more in the central and eastern Pacific. That signal shows up much more as a global signal in the upper tropospheric humidity. You have teleconnections, so-called teleconnections, where the specific pattern here, again, is much more moist in the central equatorial Pacific. At the same time, you have much drier areas north and south of that. So, you have a speeding up of the whole atmospheric cycle.

These are typical indices of time series. So, this is a 22-year time series. These are El Niño events. These are 1982–1983, a large event. Then you tend to get a small cold event. This is a modest event in 1986–1987, a big cold event in 1989. Cold events for the United States, in particular, are noted for droughts. It bounces around in the early 1990s, and this is 1997–1998, when it got a lot of publicity. You see it is sort of a four-year periodicity. The other time series that I didn't show you—it is supposed to be over here and magically isn't, I don't think—this is the other time series I wanted to show you. These are just global average time series, 30 degrees South, where we have stitched these things all together, we have gone through about 1.5 terabytes of data. Now, we are just doing some really big-time summarizing of the data.

These are just simple tropical time series, and here you see the speed of El Niño again. It is about every three to four years on these time series. The time series I am

interested in are these guys for the Tropics. This is this upper trop humidity. I have subtracted, still, these harmonics of the annual cycle. So, I was very surprised to see these time series sort of just look like white noise. I said, what in the world is this? I said, I know what these are. This is an El Niño event, this is an El Niño event, this is a cold event, here is the big 1997–1998 El Niño. So, this is sort of easy, when I see these beats of this time series. I know what those guys are. When I saw this I said—of course, the first thing you always say to yourself, did I screw up. Did I really subtract out the annual cycle from here to get interregal variability?

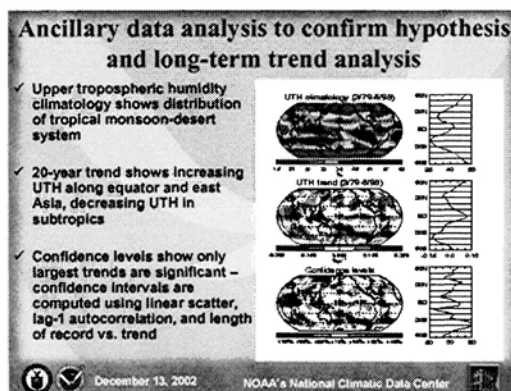
So, I went back. Yes, I did. What in the world is going on with this? Then I started noticing, while some of these peaks here are synchronous with these peaks here and some of these other ones are synchronous, there is a lot more going on. These time series, this one here, and this one which is a global radiation time series, there is a lot more going on. There is much more, if you will, of white noise. This is more red, or this is a period of three or four years. This has a lot of stuff going on. There are some synchronous events going on. Based on that, and some talking with colleagues, we formulated a hypothesis that involves a seasonality and an interregal time scale.



What we came up with is also knowledge of the system. That is, the dynamics of the atmosphere works such that, when you get strong westerlies across the equator in these El Niño cold events, you get westerlies, and this can lead to strong eddies. Big eddies are just big winter storms and they flex moisture up into the upper atmosphere. On the other hand, this leads to the possibility of a dynamical wave duct. In the atmosphere, you can get storms in this configuration of the atmosphere in northern winter and spring, or you can get storms that come down deep into the Tropics, and actually cross the Equator. In summer, you can't get that. That is why you don't see those extremes in summer.

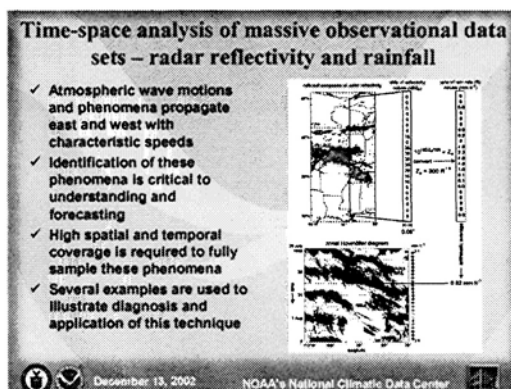
This one, you have the opposite. This is an El Niño warm event condition. You have deep convection extending out deep across the Pacific, along the Equator to the central, and even the eastern Pacific. What happens there is, you have strong westerlies, dynamically, no winds and then westerlies. That means that these storms actually can't go against the gradient of the planet here and against this shear. So, no storms get into the subtropics. It gets very dry, and this helps balance the system in the warm and cold events. So, it is sort of a neat thing.

On the longer time scale, we are interested in longer-term trends of the system. So, we want to take those long time series and, again, do some rather simple things.



Long-term climatology, that is just the long-term monthly mean for 20 years. This is your global pattern. In the Tropics, your blues are your monsoon regions, the reds are your desert regions, basically, and this is just a zonal average of that, that shows you that the Tropics are moist, the subtropics are dry, in midlatitudes, you have more of a constant temperature-moisture relationship.

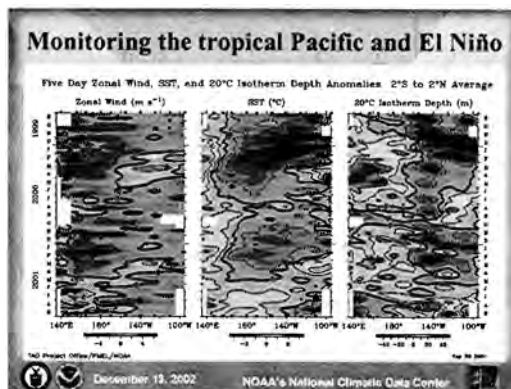
This is your linear trend, pretty simple, just a linear fit to the system. Subtropics, or the deep Tropics, because of those El Niño events of the last 20 years, are trending to tend slightly more moist in the subtropics to lower midlatitudes drying out a little bit. Of course, you would like to assess the statistical significance of any of this. This confidence interval is just computed at each grid point time series, and it is both the fit to the linear trend, plus a red noise persistence term. That is just a simple lag one auto-correlation, and then fit to the significance in the length of the time series.



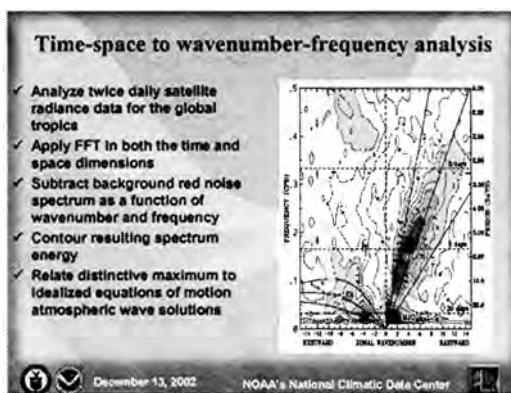
Example number two, I will try and speed up a little bit. We would like to try to reduce the dimensionality of massive time-space data sets. One of the easiest ways to do this is also to take advantage of one of our dynamical systems. Many of our systems propagate west to east basically along latitudes. So, what we can do is, we can take advantage of this by taking a cross section at any longitude, and then averaging data for latitudinal bands. By doing this, we have reduced the dimensionality.

So, here is an example of radar echoes from these weather radar precipitation data sets. We take a particular longitude here. You are about 90 degrees West. Then, what you can do is average five-degree latitude swaths. From that, you end up with a diagram that you see on the bottom. We call these Hovmöller diagrams. The weather pioneers

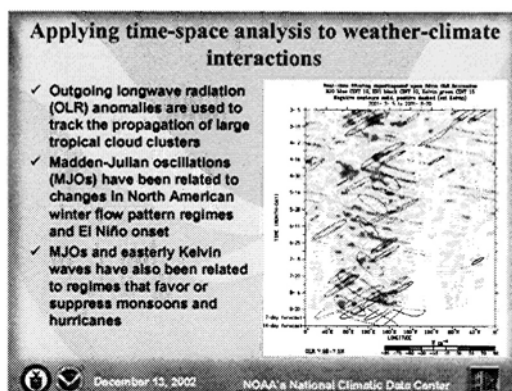
did this. This is how they predicted—the first predictions were not numerical models, but were statistical techniques where we reduced the dimensionality of the data set, and then looked at the propagation speed. Since this is a time-space diagram, this is the degrees of longitude per time steps. Here are days on your ordinate here. We can actually, from these diagrams, just come off with a propagation speed of various phenomena here. These guys tend to propagate slower, and these guys are propagating a little faster. This, although simple, is a very powerful technique.



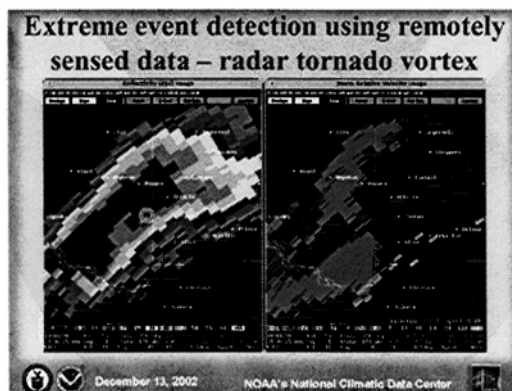
I am going to skip that example and I am going to go right to the end here.



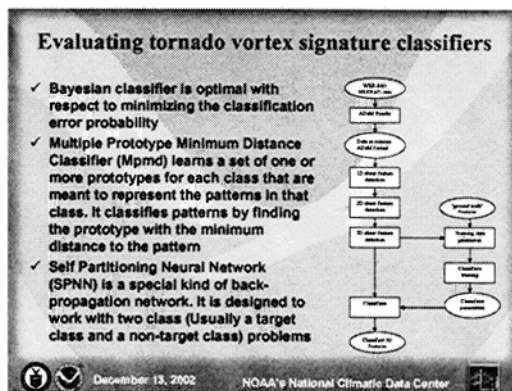
What you would really like to do is analyze this in time-frequency space. You apply FFT to both time and space dimensions and you come out with a frequency wave number diagram that allows you to detect various atmospheric phenomena, so called Madden-Julian Oscillations, Kelvin Waves and other waves, their propagation direction, their wave length and their periodicity.



These are very powerful. We have used those now to look at onsets of changes in the monsoons and regimes that favor or suppress hurricane activity in both the Atlantic and Pacific Oceans.



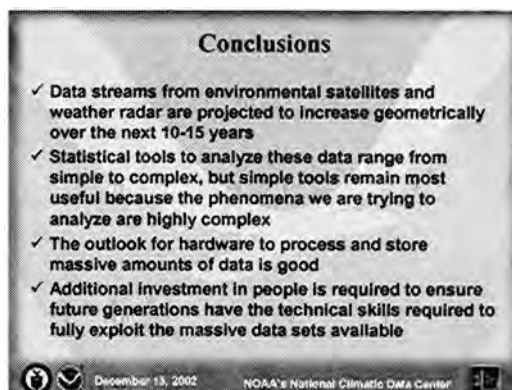
Just real quick, this is just data mining techniques. This is radar. This was a confirmed tornado. This is Doppler velocity shear, small-scale signatures only.



This is the large-scale outflow boundary and techniques are being developed to classify those schemes. As with any classification techniques—I have just inherited this one—the classification depends on your trainer and then your criteria for evaluating whether or not you have success, including probability of detection, false-alarm rate and so forth. So, some of those techniques are applicable to many different situations. Again, with the public, if you are issuing warnings, like the National Weather Service

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

does with severe weather, you want people to take those warnings seriously. So, you want to have not only a good success rate, but a low false-alarm rate. So, you need to balance all those different factors in evaluating any technique for detection.



So, concluding, we have got these massive data streams that are going to continue increasing geometrically in the next 10 to 15 years. The statistical tools range from simple to complex but, because we are dealing with such a difficult phenomenon, I really like a lot of the simple tools to first get a handle on our system. The outlook for hardware is that the hardware will probably keep up with these massive data rates, but our investment, I think, and I think many people are finding this rather obvious, so maybe I am just stating the obvious, that additional investment in the people resource is required, to ensure that the future generations have the technical skills required to fully exploit these massive data sets. Thank you.

Amy Braverman

Statistical Challenges in the Production and Analysis of Remote Sensing Earth Science Data at the Jet Propulsion Laboratory

Transcript of Presentation and PowerPoint Slides

BIOSKETCH: Amy Braverman is a statistician at the Jet Propulsion Laboratory. She received a PhD in statistics from the University of California, Los Angeles, in 1999 and an MA in mathematics in 1992, also from UCLA. From 1999 to 2001 she was a Caltech postdoctoral scholar at the Jet Propulsion Laboratory (JPL) and was hired as permanent staff in late 2001.

Dr. Braverman's research focuses on data reduction techniques for massive data sets. These methods are based on statistical clustering and signal processing algorithms modified for use in data analytic settings. At JPL Dr. Braverman serves on project teams for the Atmospheric Infrared Sounder (AIRS) and the Multi-angle Imaging SpectroRadiometer (MISR). She is responsible for the design of data reduction algorithms. She is also involved in active research collaborations with JPL's Machine Learning Group to develop data mining techniques and tools for data from NASA's Earth Observing System. Dr. Braverman has published in both statistics and geoscience journals, is active in the American Statistical Association and the American Geophysical Union, and is an officer of the Interface Foundation of North America.

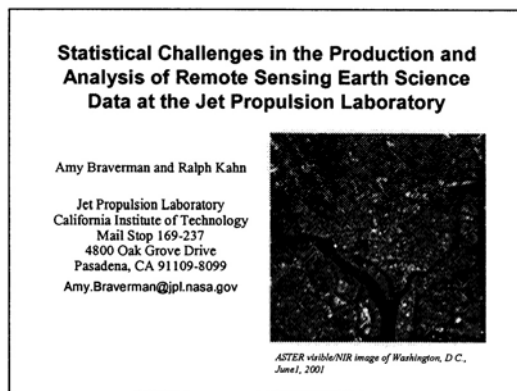
TRANSCRIPT OF PRESENTATION

MS. KELLER-MC NULTY: Our next speaker is Amy Braverman from the Jet Propulsion Lab. Amy has a Mac, so this is going to be a new experience for me.

MS. BRAVERMAN: This is the first time I have used this computer for a presentation, and I also took the plunge and did my presentation in PowerPoint for the first time. So, beware, if I have problems.

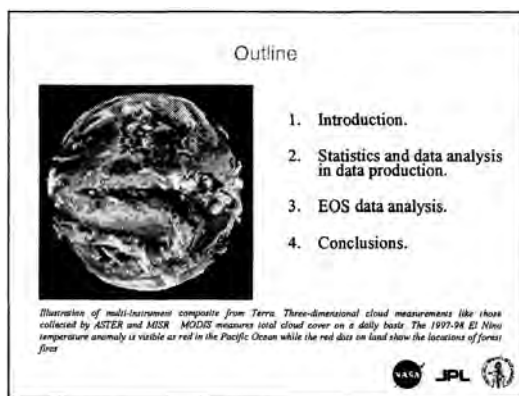
MS. KELLER-MC NULTY: While they are setting this up, I want to remind everybody to look at your programs. We have scheduled some time in the afternoon for some breakout sessions with focused discussion in each of the presentation areas.

So, write your questions down and think about that, so if it isn't covered here and doesn't get covered in the break, you will have some time to talk to each other about some of the problem areas and ideas.



MS. BRAVERMAN: I would like to thank a couple of people. I would like to thank Doug for inviting me to come and talk here. I have been chomping at the bit to get some help, and this seems like the perfect opportunity to, I don't know, to whine for it, let's say. I would also like to thank the organizers for holding this workshop. It was based on the proceedings of the 1995 conference that I got into this problem in the first place, and was able to find Ralph Kahn at the Jet Propulsion Laboratory—that is how I met him, was reading the proceedings and saying to myself, gee, he is right across town there and I need a good application for my dissertation work, that looks like a good one. So, the rest is history, and I finally got a job out of it, just last year. I actually got a job as a graduate student and then as a post doc, and now as a regular bona fide person.

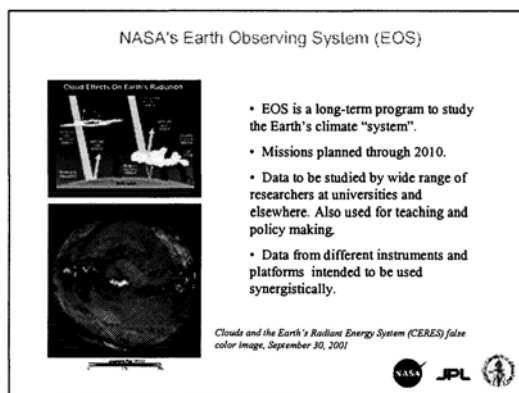
I would like to include some interesting images in the presentation, and I don't necessarily plan to explain all of them in detail. They are just kind of there to liven things up. This is an Aster visible near-IR image of Washington taken on June 1, 2001. Aster is one of the instruments on NASA's Terra satellite which was launched about three years ago now, and I just thought, we are here in Washington, I might as well show you Washington.



Here is the outline for my talk. It is pretty simple. I just want to kind of go through some observations I have made in my five years of various experiences working at JPL. I should tell you that I work in the Earth and Space Sciences Division, which is a little bit unusual. Most of the people who do my kind of work are up in the machine learning group. I have the benefit, actually, of working directly with the scientists who face these problems, and I think that is a real advantage, because I get to hear them talk about their problems, kind of on a daily basis, what it is they need to do.

The image on the left there is fake. It was put together before any of the data that —well, except for the El Niño red blob there—before any of the data that it is trying to depict was actually collected. The El Niño stuff, I don't know where that comes from, exactly which satellite, but the other things are sort of designed to show what sorts of things the Earth Observing System would eventually provide. This is kind of a dream, which is a global picture, easily visualized, of what the world is doing right now.

Anyway, what I was going to do is just run through what I see as the major statistical challenges in what we do at JPL, and then make some recommendations for how I think the statistics community could become more involved in what we do. I asked Doug what he thought would make a good talk and he said, well, some recommendations for how we could become more involved would be good. Statisticians are curiously absent from the scene at NASA right now. I think part of the reason for that is that we are pretty much perceived as not being practical enough to contribute, and that is something I want to come back to later.



So, the Earth Observing System program is a long-term program to study the

Earth's climate system. What that means is looking at the atmosphere, the oceans, the biosphere, and looking at it all as one integrated system, and studying the feedbacks that are involved.

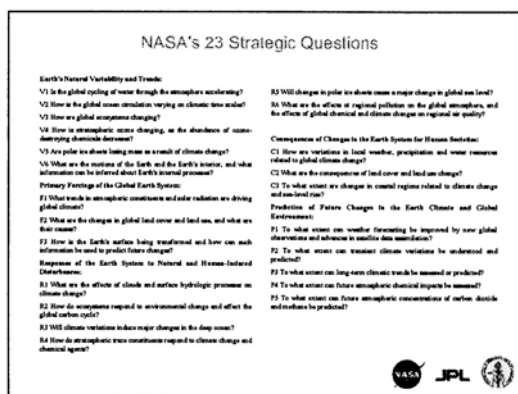
The upper graphic there is kind of a little bit of an illustration of why clouds are important in this system, and it is because all the energy the Earth gets is from the Sun. The question is, how is that radiation budget working out for us? Some of that radiation is reflected back out into space. Some of it gets all the way through to the ground, gets absorbed, goes to power everything we do here, create fossil fuels and what not. That is really one of the major things we are trying to study. So, a very, very important question that we are trying to answer is, what are the radiative effects of clouds, and that necessitates knowing where the clouds are and how they are spatially distributed, how they are changing over time. There are also what we call aerosols in the atmosphere, some of which are manmade—pollution, for example—others of which are natural, like forest fire smoke, and these things, too, have an impact on the radiative balance of the Earth.

The bottom image there is from an instrument called CERES, which I will mention again a little bit later on, which is a global map of the—what does it say, outgoing long wave flux of the Earth, I guess September 30, 2001.

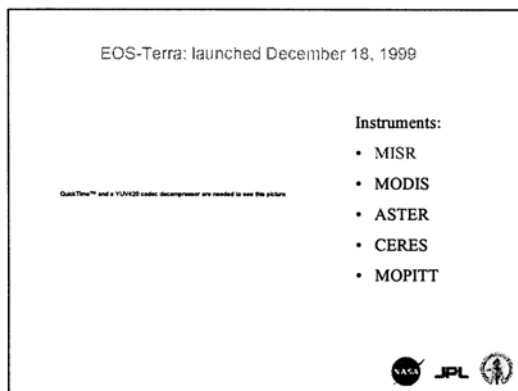
Anyway, we have lots of data at NASA. I am kind of amused by the question of, what is a massive data set. I know that some people don't agree with this but, in my view, a massive data set depends on who you are and what your computational resources are. You will understand why I have that perspective when I get on with what I am about to say about what our job is at NASA. These data that we collect, we are actually in the data production business at NASA, above and beyond everything EOS. We do participate in the analysis of the data that we collect, as you can imagine, but our primary responsibility is to build and fly instruments to collect data to study climate change.

So, we are in the business of providing these data to the community. The community is a very diverse group of individuals with lots of different interests, lots of different opinions about what assumptions are valid, and lots of different resources. Our users range from university or college researchers with desk top computers to people like NOAA, for example, as we just heard, who use some of our data. So, it is a real challenge to try to design a one-size-fits-all sort of way of producing and distributing data that can satisfy everybody's needs.

One of the things I wanted to mention specifically was that the EOS program is a long-term program that involves a number of different satellites and instruments. One of the prime intentions of the EOS program is that these data were supposed to be used synergistically. We were supposed to be able to combine data across instruments and across platforms, and as yet, we haven't really done that, and we don't know of anybody who has, largely because of the huge data volumes we are getting and the very complicated way in which the data are collected.



So, these are NASA's 23 strategic questions. I don't know how easy that is to read from where you sit. NASA has formulated these questions as a way of concretizing what sorts of problems we want to address with the data. I can certainly refer you to the Web page where this came from and let you know that I am not going to dwell on it or read them all, but you get kind of an idea for what kind of the big picture questions are.



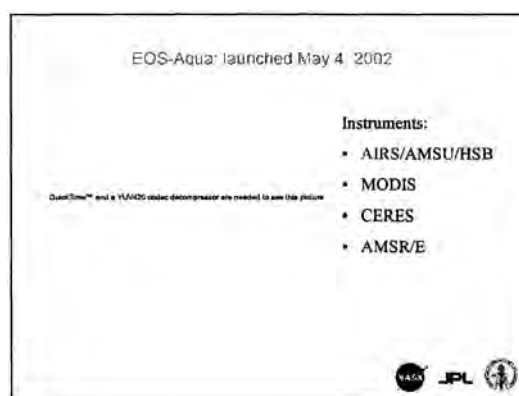
So, the Earth Observing System, right now it has a collection of satellites involved. Two of them are actually in orbit now. The Terra satellite was launched on December 18, 1999, from Vandenberg Air Force Base in California. It carries five instruments: MISR, the Multi-Angle Imaging Spectroradiometer; MODIS, the Moderate Resolution Imaging Spectrometer; ASTER, which I think is the Advanced Spaceborne Thermal Emission Radiometer—you get to be really good as acronyms when you work at NASA; CERES, Clouds and the Earth's Radiant Energy System; and MOPITT, Measurements of Pollution in the Troposphere. The reason MISR is in red there is because that is one of the projects that I work on, so I have drawn many of my examples from what I know about MISR.

NASA is also very good at making pictures and doing animations and doing PR. This is a depiction of the Terra satellite in orbit, and the instruments that it carries, doing what they do. If you have heard me talk before, which some of you have, you will recognize MISR as the multi-colored beams stretching out forward and aft along the direction of flight. This depicts the fact that MISR collects data. So, MISR looks down at the Earth at nine angles and four wavelengths and has a swath width of about 300 kilometers or so. The Terra satellite is in a polar orbit. So, what we do is, we

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

successively see the same spot on the ground at nine angles and four wavelengths simultaneously. So, we have 36 channels' worth of data.

On the other instruments onboard are MODIS, which has a much wider swath width and, therefore, gets a lot more coverage. MISR, with its skinny little swath width, gets global coverage about every nine days. MODIS, with its much wider swath width, gets global coverage every day. The other instrument, CERES, was the red and yellow ones going back and forth like this in the back. I am not sure what its coverage is, nor am I sure about MOPITT. ASTER is what they call the zoom lens of the Terra platform, because it only looks at specific spots when it is told to do so, and it has very high resolution, about a meter, I think. So, I wanted to put that up. The other—now I know I had better wait with this until I finish describing what I want to say.



The second EOS satellite was launched on May 4, 2002. That is called the EOS-Aqua. It carries four instruments. You will notice some of the names appear again. MODIS and CERES, it has a MODIS and CERES as well. It also has the AIRS instrument, which is the other project that I work on at JPL, which is the Atmospheric Infrared Sounder, which looks down at Earth at single view angle, but 2,378 spectral bands, and has a spatial resolution of about 15 kilometers. MISR was about 1 kilometer resolution. So, that is how I knew what CERES looked like, from looking at that.

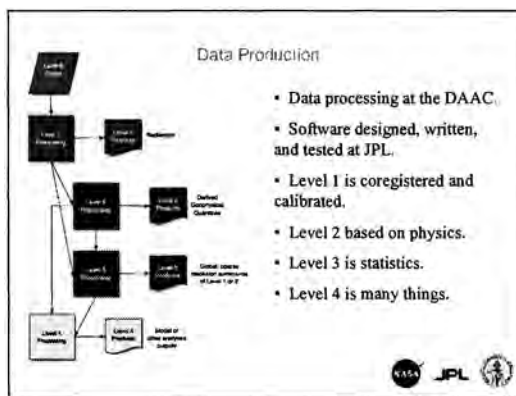
Then we have AIRS, AMSU and HSB, which are actually bundled together in a single package, and are processed together at JPL. The spinning thing on the top is AMSR/E, which is going to come up. There is MODIS. You get some sort of relative idea of what the swath widths are, the relative widths, from the animation. I don't know what AMSR/E does, to tell you the truth.

There is way too much to know at NASA. It took me about four years before I could actually read a document without having to look up every acronym that I came across and therefore kind of understand what was going on. I like this, because you are getting bathed in the glow of the data here. I think that is kind of nice. Choking on it is more like it, as I said.

[Question off microphone from audience.]

It depends on which instrument you are talking about. For MISR, I can tell you it is about 75 gigabytes a day, and for AIRS it is about 28.2 gigabytes a day. It is a little hard to say how big the data are, because the data are processed on successive levels. So, if you quote a big huge number, you are really talking about the same data in different forms.

Pretty much, any way you slice it, it is a lot. I wasn't going to concentrate on the "gee, whiz, how big is it?" statistics, but one of the things people like to talk about is how, in the first six months of operation, the Terra satellite doubled all of NASA's holding from the time that it began. So, there is a lot of data. At the MISR science team meeting this week, we had the people from the data distribution center come and talk. They told us that we now have 33 terabytes—33 terabytes is 11 percent of what we have in storage now for MISR, and that is just one instrument on one platform.



Let me say a few things about what happens to the data when we get it. It comes zipping down from the satellites and gets beamed around, and finally ends up at something called a data processing center called a DAAC, which is a Distributive Active Archive Center which, I think Ed pointed out, is a big oxymoron, all by itself, and it gets processed.

The data, as it arrives at the DAAC, is called Level 0 data. It is just raw and uncalibrated from the spacecraft. It goes through a series of processing steps called Level 1 processing, which geolocates and calibrates the data and yields for you a data product called Level 1-B-2 in the terminology of things, which you can think of conceptually as a great big data set that has a row for every spatial scene on the ground, which would be 1 kilometer for MISR and 15 kilometers for AIRS, and a column for each observed channel, 36 columns for MISR and 2,378 columns for AIRS.

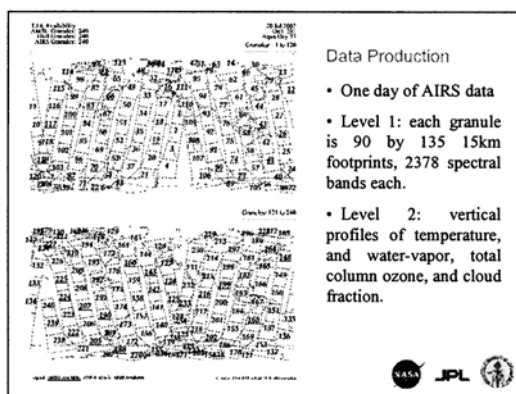
Then comes the good part. So, that Level 1 data is then pumped through what we call Level 2 processing algorithms. John alluded to the word retrieval, which was a mysterious term to me for quite some time. That is where we retrieve the geophysical quantities that, in theory, produce those radiances. So, Level 2 data products are derived geophysical quantities that will form the basis for answering those 23 questions.

Because the data are so large, we have an obligation to provide them in a form that is a little bit easier to use, and that is the so-called Level 3 stage of processing, where we produce global gridded summaries of the data on a monthly or a daily or a weekly or whatever basis. This is supposed to satisfy the needs of people who can't handle 75 gigabytes of data a day, and make the analysis a little bit easier.

Level 4 is sometimes talked about, and it is the analysis stage, where you put the input into a climate model and use it to actually generate something.

In my mind, I make a distinction between—Levels 1, 2 and 3 are what I call data production, and Level 4 is the data analysis stuff, and I think it is important to make that distinction.

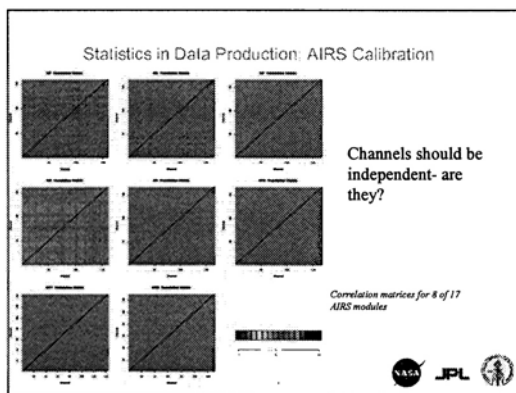
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



This is a granule map for AIRS. I said the data come down in chunks. Every day we get 240 chunks of AIRS data. Each granule is an array of 90 prints across and 135 footprints along for AIRS, and there are 2,378 radiance observations for that.

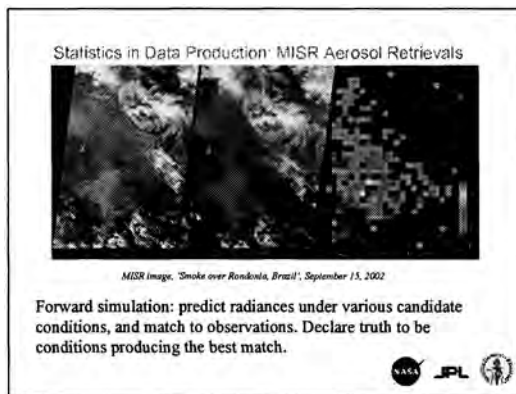
I just wanted to put that up there so you get some idea, if you wanted to order data, and you wanted to order a whole world's worth of data for one day, you would have to order 240 files that look like this. If you wanted a specific spot, you would have to figure out which granule it is in. This changes every day. For AIRS, a granule is defined as six minutes' worth of data. One of the problems that we have is that each instrument defines its granule its own way. MODIS, for example, describes its granule as five minutes' worth of data. The naming conventions for the files are not easy to figure out and they are not standardized. The sampling grids, the grids at which the data are collected, are different for every instrument. So, you are looking at a big, big problem, if you want to compare data across instruments.

Now, I wanted to mention just a few problems that I personally have encountered in working with the folks at JPL on looking at where statistics and data analysis fit in. There is a tremendous amount of statistics that goes on at JPL, and data analysis. Everybody does data analysis at JPL, and nobody is a statistician, except for me.



One problem we had was with AIRS calibration data. The 2,378 channels are broken down into what they call 17 different modules. Channels go into the same modules because they share a certain amount of electronics. The channels are supposed to be independent of one another.

These are covariance matrices for eight of the modules. They asked me to try to determine whether or not the measurements from these channels were, in fact, independent. To a lot of people at JPL, correlation means independence. Zero correlation would mean independence. Never mind the fact that the data don't really look very Gaussian when you try to look at them, but these are just a couple of the covariance matrices that we generated sort of a cohort so you could look at them quickly. It looks like we are doing okay on the last two, and then things get a little dicier as you go back, as you go down to number—this one here.

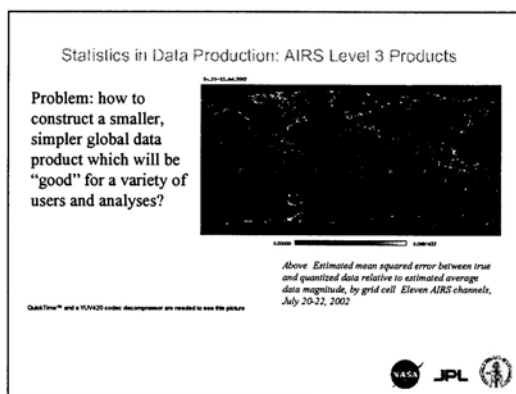


The previous speaker alluded to doing retrievals and doing forward models. The calibration issue I think of as a Level 1 issue. It is something that you do at Level 1 processing.

This is an example of a Level 2 problem. We collect from MISR, for example, these radiances from nine angles and four wavelengths. You see here six of the channels, RGB nadir and RGB 70 degree forward, and the so-called optical depth retrieval that comes from it. As John said, the way this is done almost across the board is by matching the observed radiances to radiances that are predicted, under the assumption that certain conditions are true. Then, wherever you find the best matches, then, aha, that must be it; that is what we are going to call the truth.

This is okay, I guess. It seems like a very rich area for statisticians to help improve how that is done. The thresholds for when they say something fits or doesn't fit are pretty much ad hoc, and could benefit from some good principled statistical thinking.

The other thing is in how they characterize the uncertainties associated with these retrievals. It has nothing to do with variance at all. It has more to do with how many of the candidate models fit, how close they fit. For example, if lots of the models fit, we are pretty uncertain. If none of the models fit, we are even more uncertain, and if just one of the models fits well, then they say we are pretty certain. The uncertainty characterization tends to be certain, not very certain, not certain at all, that sort of thing. It is qualitative, rather than quantitative.



Finally, the third area that I wanted to mention was the creation of Level 3 products, which is how I got involved and what my responsibilities on MISR and AIRS are, to design Level 3 products. Level 3 products tend to be maps of averages and standard deviations. You take all the data for the period of time over which you are summarizing, and you chop it up into little 1 degree by 1 degree bins. If you are creating a monthly summary, then you produce maps of each of the quantities you are interested in, with a mean value in each grid cell, and then maybe another map with the standard deviation in each grid cell. If you are really clever, maybe you make some maps of the correlations or covariances to go along with it.

I think the natural reaction of a statistician is, oh, that is awful. You know, you are throwing away most of the data there. It begins to make more sense when you stop to think about the operational problems that go into doing these things. Doing things like density estimates, fancy stuff is just completely out of the question because of the processing. The processing has to keep up, basically, so it has to be fast.

I will just plug my own thing here and something I have worked with Ed Wegman a little bit about. What I propose to do for both MISR and AIRS is to create a Level 3 product that puts what I call a quantized data product. Instead of simply providing a mean of standard deviation in each grid cell, we provide basically the results of a clustering algorithm.

You have a number of representative vectors and associated weights, which might be the numbers of original data points represented by each of those representatives, and an error measure, which might be the within-cluster mean squared error, and provide this product as a quantitative Level 3 product that retains more of the distributional information about the data than just a simple mean standard deviation would. In particular, it would retain some of the information about outliers which, for science analyses, tend to be among the most important things and the things you don't want to smooth out and throw away.

What this image here is, is just a map of the relative error in one of the products that I created to show at the American Geophysical Union last week. The original data set from which this image was created was about 550 megabytes, and the compressed or quantized product was about 60 kilobytes. So, it is about a 10-fold reduction in data size, and you suffer pretty much, at worst, about a 7 percent error in the data, as measured by mean squared error relative to the average magnitude of the data within each grid cell.

So, that is pretty good and it is quite good for a lot of applications. Sort of the problem here is how do you tell people ahead of time whether it is good enough for their

application, because you don't know what their application is and, of course, if that is good enough or not depends on what their application is.



I wanted to show one other thing here, which is this little zippy animation. So, this is an animation of the Aqua orbit. You can see how it goes. We are in a polar orbit. That little skinny red strip there is not too far off of what a MISR swath would look like. An AIRS swath would be considerably wider than that. You see that what is going on there, is that as you go around the Earth is, of course, turning underneath you. So, how do you make a global summary of data like that.

Someone who has thought about that, to some degree, is Noel Cressey, who has developed some techniques that we have experimented with a little bit, for creating kind of a Level 3 product that would sort of take account of spatial and temporal dependencies, in order to produce kind of a monthly summary of these data, that takes care of sort of the interpolation between swaths and over time. That is a big problem for us, too. I personally put that in the realm of the data analysis rather than the data production.

EOS Data Analysis

- Understanding what's in the data is a necessary precursor to doing inference with it.
- The vast majority of analyses with EOS data right now are exploratory and descriptive.
- Level 2 for regional or process studies; Level 3 for global studies and climate models.

Four images of the Mediterranean obtained concurrently on June 14, 2002 from the three instruments that make up the AIRS experiment. Upper left: visible light; lower left: 900 cm⁻¹ measures actual surface or cloud top temperature; upper right: 150 gigahertz channel from the Humidity Sounder for Brazil is sensitive to moisture, ice particles and precipitation; lower right: the 31.4 gigahertz channel from the Advanced Microwave Sounding Unit is not affected by clouds.



So, data analysis, I will go through this quick because I want to get to the end here. Understanding what is in the data is a necessary precursor for doing anything with it, for inference. The vast majority of what is done with EOS data these days is exploratory and descriptive, because we are still just trying to understand what is in it. Inference is just something that is going to have to wait a little while. So, there are lots of opportunities for descriptive type techniques that need to be brought to bear on these data.

EOS Data Analysis

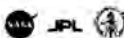
Three major areas where the statistics community could really, really help:

Visualization: techniques that allow us to explore multivariate relationships while retaining spatial and temporal context.

Data mining and analysis of massive data sets: techniques that allow us to find important or unusual patterns and relationships we don't already know about.

Data fusion: statistically sound techniques for combining data from different sources (different orbiting instruments, data acquired on the ground and in field experiments).

There are many interesting applications in all these areas and others.

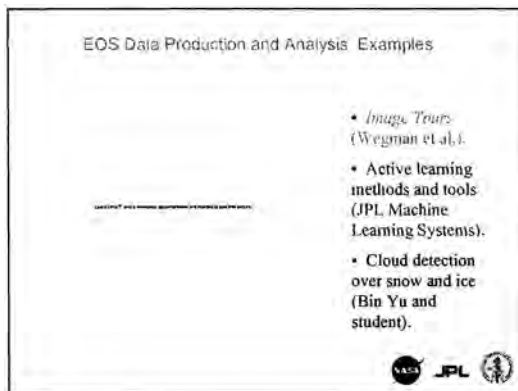


I am going to just zip along. These are three areas where I think we could really benefit from the help of the statistics community, which is multivariate visualization, particularly for data sets where you want to preserve the spatial and temporal context of the data.

In my view, the real problem here is how do you visualize features of joint distributions of very highly multivariate data as they evolve as you move around in space and time.

Data mining and analysis of data sets, it is pretty obvious we need help with that. No one can look at all this data. So, we are doing some things. I like to think of the Level 3 products as kind of a first stab at that.

Finally, data fusion, which I was unable to find too much literature by statisticians on that, and that is really important. That is a big problem for us. We want to be able to combine data from different instruments on the same satellite, from different satellites. We need to be able to combine information from ground sources with the satellite data that we get in order to validate it. So, if anybody has any good ideas about that, or would like to work on that, we would be very happy to have you help us.



As far as analysis is concerned, I wanted to just mention a couple of examples.

Ed Wegman has been real good to us and put together a new incarnation of his image tours ideas. I don't know how well you can see that. It is a little difficult to see here, but I will just run that. This was Ed's answer to our problem about how to visualize 36 channels' worth of information while retaining a spatial context.

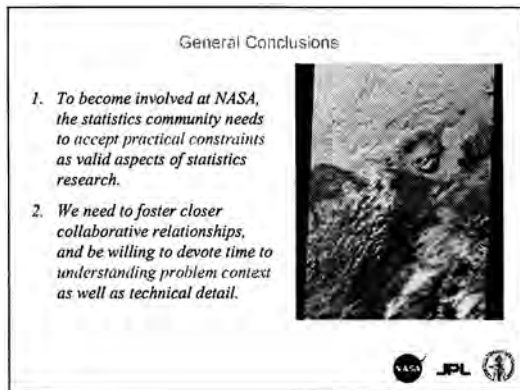
What is going on here is, we have 36 gray-scale images, each representing a different channel's worth of data. It is successively producing the same linear combination as shown by the little thing down in the corner here, of the 36 channels in each pixel of the data, and I will refer you to Ed to explain that a little more carefully. The animation doesn't run very long, and it is hard to see what is going on there, but what you hopefully noticed there was that certain features pop in and out of view as you run that thing. It actually turned out to be pretty useful for finding features that you didn't otherwise know about.

Now, if you already knew those features were there, you could go straight to the image where it was most easy to see, or to the combination of images where it was most easy to see. If you don't know they are there, which we don't know anything ahead of time about this, then you need something that is going to help you look at a lot of data quickly and find these things, and this was very useful for that. We are still working with

this. We need to do more work with it in order to be able to help the scientists interpret what it is showing us.

A second thing that we are heavily engaged in is collaborating with the machine learning systems people at JPL to do data mining, particularly active learning methods and tools. We got some internal money from NASA to pursue this, and we are looking forward to eventually being able to show it at the JSM interface meeting.

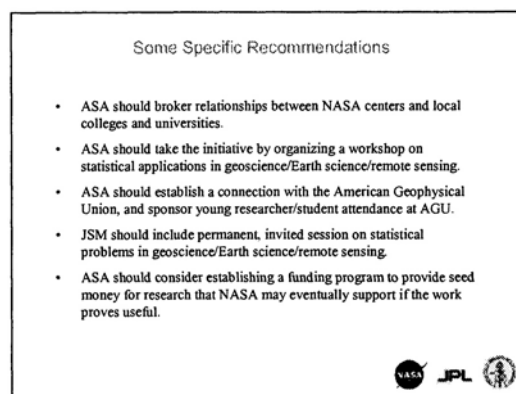
Finally, we were able to suck Bin Yu in. I met Bin at a workshop at MSRI, and she showed some interest in the data. So, we put her up to a particularly vexing problem in the analyses of these data, which is how to detect thin clouds over snow and ice with MISR data. You know, you are looking at a white object over a white background. That is difficult to see. She was down in Pasadena yesterday, gave a talk to our science team, and has some very nice results for us, and we are looking forward to hearing more from her. She has a student working on the problem, actually, and that has been great. Now we have kind of got him learning how to order data, and that looks good for us because it means somebody is using our data, and it is good for him because it is a neat problem to work on.



My general things I want to say—this (by the way) is the RGB nadir image of the previous movie here. I might have called this, like I said, a shameless cry for help, because NASA has so much wonderful data, whether you are interested in massive data set or whatever particular area of analysis you are interested in, there is a wonderful NASA data set that would be really interesting for you, I am sure.

The great prohibition, I think, has been that a lot of statisticians think of practical constraints as kind of a detail that isn't really something they are interested in. What we really need people to do is to devote the time to understanding the problem context and the practical restrictions on the analyses, and to accept those as important research problems in their own right. You know, it is not enough to think of a great way of summarizing data or analyzing data. It has got to be something that can be done, or they won't pay any attention to us.

Thanks to Ben and Ed and a couple other people, we are moving in that direction. So, Doug said to bring some specific suggestions for how we could remedy the situation, and these are the ones that I came up with.



The first one is that I would really love to see ASA broker relationships between statisticians at colleges and universities and local NASA centers. Like I said, the NASA centers are just a tremendous source of really interesting data sets for teaching and research, and it would be great if we could get those into the hands of people who could actually do something with them.

We would love to see ASA organize a workshop specifically devoted to statistical applications and problems in geoscience, Earth science, and remote sensing. Doug and I were just at the American Geophysical Union meeting early this week, where we had a session on model testing and validation in the geosciences that went very well. It was very, very well received, very well attended. Ed, Doug and Di Cook spoke last year at the AGU, and they are very, very happy to have us, and we would like to push forward some more formal relationship with the AGU. In fact, they suggested us having a committee and, they don't have any money, but if we could somehow find a way to come up with some money to send young researchers and students to the AGU to show their work, that would really be great, because that is really how we are going to inject ourselves into that community.

Also, I would like to see geoscience get a little bit higher profile at JSM. Sometimes we have an occasional session on it, but it is nothing like—you know, bioinformatics you see all the time. It would be nice to have a session devoted to just bringing the problems to the statistics community, maybe not the solutions, but just tell us what the problems are.

This was Ralph's suggestion, this last one, speaking like a strapped NASA researcher like he is. We would love to see a funding program to fund work that is directly relevant to NASA's problems. NASA has not traditionally funded statistics, partly because of the problem with being practical and doing things very useful for missions. We think that, if we could get some seed money to prove how useful we could be, that that funding would then come later.

So, those are the suggestions. I hope that we can make some of them ring true, and thank you.

Ralph Milliff

**Global and Regional Surface Wind Field Inferences from
Spaceborne Scatterometer Data**

[Transcript of Presentation](#)

[Technical Paper](#)

[Title Page](#)

[Presentation Summary](#)

[Figure 1](#)

[Figure 2](#)

[Figure 3](#)

[Table 1](#)

BIOSKETCH: Ralph Milliff is a research scientist at the Colorado Research Associates division of NorthWest Research Associates. His expertise is in numerical modeling of the ocean and atmosphere and in the relation of air-sea dynamics to climate. He was an ocean modeling postdoctoral fellow at NCAR (1989–1991), and a staff scientist there until 2001. Dr. Milliff has served as a member of the NASA Ocean Vector Winds Science Team for the NSCAT and QSCAT missions (1991-present). His current research involves the application of global surface vector wind datasets of studies of upper ocean mixing and the ocean's general circulation, the Madden-Julian Oscillation, and the quasistationary waves of the Southern Hemisphere. In addition, Dr. Milliff is adapting methods of Bayesian hierarchical models from probability and statistics to problems of air-sea interaction.

TRANSCRIPT OF PRESENTATION

MR. NYCHKA: Our next speaker is Ralph Milliff from Colorado Research Associates in Boulder, Colorado.

MR. MILLIFF: I, too, would like to thank Doug for inviting me. The work that I am going to talk about began with collaborations in the geophysical statistics project at the National Center for Atmospheric Research, of which Doug is the director. My earliest collaborations, and ongoing, are with Mark Berliner who is now at Ohio State—he was a prior director of GSD—and Chris Wikle, who was then a post-doc at GSD. Also, in addition to Doug and Chris and Mark, I am happy to acknowledge Tim Hoar, who is the staff scientist for the statistics project, and Jan Morzel.

Sustaining research support has come from the NASA Earth Science Enterprise ocean vector wind science team, and I acknowledge and appreciate that very much. Since there is a JPL person in the audience, I am obliged to show a picture of their instrument. This is what the people in the trade call eye candy. It is for you to look at. It isn't really the real system, but the data set that underlies the massive data stream I am going to talk about today is the surface winds over the global ocean. These all have begun to explode in volume and precision since about 1991 when, within the Earth-observing era of satellite data sets, the first global wind data set began with the European Space Agency mission, ERS-1 and 2.

Before I tell you what a scatterometer is and how it works, I should convince you a little bit that the global ocean surface wind field is a worthwhile field to measure. The surface winds transfer momentum between the atmosphere and ocean. They modulate the transfer of heat and material properties, like carbon dioxide and liquid water. These obviously have important implications for inferences on climate processes and the rates of climate change, when taken in a global perspective. On shorter time scales, the surface winds and these same exchanges are very important in predicting weather. So, a scatterometer is a system that continually emits an active microwave pulse at the ocean's surface, where it is backscattered by capillary waves, and the backscatter signal is detected by the same platform in space, and related to a surface wind speed. So, this is a retrieval algorithm, of the kind that John Bates so cleanly described in his presentation.

We are retrieving the surface wind from what the surface wind has done to ripple the surface, and backscatter a pulse of radiation that we know very well its properties. The little waves that backscatter radiation from a scatterometer are called cat's paws by sailors. They are the little ripples that form on the surface when a puff of wind shears the surface of the ocean.

Because we know the polarization frequency, the angles of incidence of the emitted pulse very well, and the backscattered pulse as well, we use several returns to fit model functions for separately the wind speed and wind direction, and retrieve the vector wind. The returns are aggregated over what we call wind vector cells. So, that is going to be my pixel. Instead of the radiance coming out of a particular patch or footprint on the surface of the Earth, I am looking at backscattered returns over an area. So, they are aggregated over an area.

In terms of data volume, the European system that I mentioned first was launched in 1991. It is the longest-lived scatterometer system on our record. It continued until 2000. This is a 12-hour coverage of the globe. Within each swath there in black, there

are 70-kilometer resolution cells that overlap each other. So, the wind vector cell in the ERS system was 70 kilometers, and we got about 41% of global coverage in 24 hours.

In 1996, NASA launched the NASA scatterometer, or N-scat. It about doubled the precision and the volume of the surface vector wind data set. The data organization in this case is a 25-kilometer resolution wind vector cells in two swaths that orient along each side of the satellite ground track. So, these are the polar orbits for a 12-hour coverage. You can see the gap at nadir in each swath, and then two swaths on either side of the satellite.

The N-scat system came to an abrupt end when the solar collector on the satellite bus failed dramatically in 1997. So, we had about nine months worth of data. In response to its failure, NASA launched what is called QuickSCAT. It was a QuickSCATterometer. It has an 1,800 kilometer swath, so about 18 degrees longitude, with 25-kilometer wind vector cell resolution. This is the 12-hour coverage.

Now, we are seeing 92 percent of the globe every 24 hours. As of tonight, at 8:30, a second sea winds system—sea winds is the scatterometer aboard the QuickSCAT satellite, a second sea winds system will launch aboard a Japanese satellite, hopefully, and we will have, for the first time, synoptic coverage of the surface wind field of the global ocean every day. This will be the 12-hour coverage from two sea wind systems. You can see that, in 12 hours, there are only very few gaps in the coverage of the surface wind fields.

When we started to think about this problem, there were very large gaps, and this is one of the problems that Amy brought up in her talk just a minute ago. So, our first statistical model was how to fill those gaps with physically sensible surface winds. Well, what do I mean by physically sensible surface winds? One property of the surface wind field that has emerged from the scatterometer data set, an average property that we can hang our hat on as physicists and use statistical techniques to drive and time our interpolations is the spectral properties in wave number space. If you can permit me, I'll put the whole slide on this way. So, along what should be the ordinate, we have power spectral density or kinetic energy along—the abscissa is the spatial wave number. The spatial scales that correspond to those wave numbers are listed on the top here.

What we observe in the surface wind field for our planet is that they obey an approximate power law. There is almost a constant slope in wave number space for the kinetic energy. That isn't the case, if you look at the other curves on this picture, for surface winds that come from weather center models. This is the climate model and these are forecast models.

They depart from an approximate power law behavior on spatial scales much coarser than the grid resolution and numerical models that generate the weather to begin with. So, we have a spatial constraint now, to use in our interpolation. These spectra are for a box in the North Pacific Ocean, averaged over the calendar year 2000. There are interesting relations to two-dimensional and three-dimensional theoretical turbulence theories that also make this an appealing result, but they are not really relevant to this talk today.

What we do notice now is that this spectral slope, this approximate power law, the slope of that spectrum has a spatial and temporal distribution. It is shallowest in the Tropics, where convective events supply energy at small scales, and they propagate upscale in an inverse cascade to the larger scale. They are steeper as you go to higher

latitudes. The western Pacific and the Indian Oceans are the shallowest. Typical slopes— this is for the zonal wind— there is a similar relationship with the Mariana winds, and there is an annual march in the slope of these spectra. It begins—it is shallowest in the earliest part of the year and steepens slightly in the Tropics and a less evident annual march in the midlatitude storm track regions.

This pattern is repeatable. We can say that now because we have about three years. That was the picture for 2000. This is the picture for 2001. Again, we see shallow slopes in the western Pacific, shallowest slopes early in the year. We use these regional and seasonal properties of the surface wind field to perform that interpolation problem that I mentioned. We need to account for this wave number deficit in the weather center winds. What the weather center winds have going for them is that they are available four times a day everywhere.

The satellite isn't true. It drops out in rain, and the satellite has a non-parametric sampling scheme that has to do with its polar orbit configuration. What we did was use a multi-revolution wave length procedure to blend wave number deficient surface analysis of the surface wind four times a day, with the available scatterometer data. The constraint and the reason why we used the wavelets was because wavelets have a multi-resolution property that allows you to specify the slope of a fractal process. The slopes we were going to use, obviously, are those spectral slopes that distribute with space and time over the globe, as observed by the scatterometer.

So, with an eight degree square on the globe every day, we collect the spectral slope over a 30-degree-long spectrum, and store it, and use that, and sample from the log spike distribution based on that collection, to augment the wave number deficient weather center winds, whenever we don't have a satellite observation. This is an example of that blending procedure.

The field I am looking at now is a scalar. It is the wind stress curls. This is the derivative of the two components of the winds, the east-west component and the north-south component. It is going to be noisy, but it is scalar. So, I can show the effect of our blending technique, and it also points out some important meteorological features, that are a benefit of this particular procedure. Wind stress curl extrema are positive in the Northern Hemisphere, in the region of atmospheric cyclones. These are storms in the storm track regions of the Pacific and Atlantic Oceans.

Associated with these storms are frontal systems, and these are almost washed out in the weather center products. This is the wind stress curl from the National Centers for Environmental Prediction on a given day in 1996. This particular blending is for the N-scat system. We do this on a regular basis for QuickSCAT, and that data is available from the data archive system from NCAR. Overlay the N-scat swaths in the middle panel. That really doesn't show up well. Actually, I think this is in the abstract as well.

The swaths from the satellite, all you can see is that the wave number properties in the satellite data are much richer at high wave numbers than they are—due to blow up in the north Atlantic. So, here is that cyclone in the North Atlantic. You are looking at it from the bottom of the ocean. Here it is from the top. The frontal system is here in yellow. Here are the overlying scatterometer plots. You can see that the scatterometer detects the very sharp spatial features of the front, and the high-amplitude wind stress curl that occurs there when it crosses. The blending procedure, because it is a spatial model, can't keep track of the propagation of this system. The space-time model will,

and we are working on that. Because it is a spatial model, within the eight degree squares in the gap region, high-amplitude wind stress curls, commensurate with the wind stress curl that occurs in the front, is distributed. That is important for driving a global ocean model. In fact, in 1999, we wrote a paper that showed that the ocean models that we used for climate forecasting, for climate stimulations, were very sensitive to this change in the surface wind fields.

So, you drove a model with the weather center winds, and we are not getting this high wave number forcing. This high wave number forcing turns out to be important. This is the annual mean response after a spin of three degrees global ocean model. Here is the difference with respect to a calculation that was done with the weather center model. So, it is the blended winds minus the weather center winds.

You can see, up to 7 meters per second, 3 1/2 knot differences in the currents of the upper oceans. More important, there is a big divergence pattern in the Tropics. This is the region of the El Niño Southern Oscillation signal. So, the divergences there have big implications for cold water at the surface, and the propagation or not of an El Niño signal into the eastern Pacific.

So, there is a reason to do this for climate. I am going to shift now to regional applications, and perhaps a little more deep statistical modeling. This comes from my learning at the feet of Berliner and Wikle.

This is an AVHR, a radiometer image. It is basically a temperature in the infrared and visible regions of the spectrum, of the Labrador Sea. This is the coast of Labrador, here is the southwestern coast of Greenland. This is one of a few regions in a world ocean where the so-called ocean deep convection process occurs, and this is critical to climate. This is a place where the properties of the surface atmosphere are subducted to great depth and for very long times into the ocean. This is the so-called thermal haline circulation, or the global conveyor belt, that you might have heard of.

The Labrador Sea is one place. The eastern Mediterranean is another, and a few points around Antarctica are the other places where the atmosphere and the deep ocean are in contact, very briefly, during these very brief ocean convection events, and they drive the redistribution of heat on the planet. The ocean part of that redistribution happens here. So, those convective triggers are associated with weather patterns of the kind we see here. This is called a polar low. The low-pressure system is centered around the middle of the basin.

The winds that are associated with this, the surface winds that are associated with this signal, drag dry, cold continental air across the relatively warm sea surface and exchange those properties that I talked about in the beginning of the talk—heat, moisture and momentum—and superdensify the surface ocean and provide this plunging physical mechanism. Within an hour of this AVHRR image, the NASA scatterometer, or N-scat, a fragment of that swath occurred, and you can see the signature of the polar low here. So, it is understanding these convective triggers, and certainly the surface wind field associated with them, is sort of the target science problem that we dealt with.

This is another polar low signal in the Labrador Sea. This sets up a Bayesian hierarchical model that we use to retrieve a uniform surface wind field with estimates of uncertainty at each grid point, from the scatterometer data.

So, we are going to build a posterior distribution for the wind at the diamond grid points—that is our target grid—given the scatterometer data and a prior, based on a

physical balance. This is the work of Andy Royal, who was another post-doc at GSP, and Mark Berliner and myself. So, I think, as a physicist, about Bayesian models in stages. The first is the data model stage, or what you would call the likelihood, and I think this is a very natural entity for satellite data. It is wrong to think of satellite data only as moorings for balloon traces that happen to be right next to each other in space. Instead, they inform probability distributions. So, probabilistic models are an essential new technique, I think, that we need a great deal of help with in the geophysical field. I think our sort of preliminary pilot studies show that there is a great deal of play here. The likelihood model gives us a distribution for the data that naturally arises from measurement error models that come from every satellite mission that is ever launched.

We do what are called calibration validation studies. Calibration validation studies will inform the likelihood distribution to excellent precision and allow the satellite data—the volume of it—to actually speak to the posterior very clearly. In the prior process model, we used heritage and geophysical fluid dynamics that go back for generations. We developed essential dynamical descriptions of processes of interest, and there is a whole branch of atmospheric and oceanographic science to do just this. We can blend these two heritages, the observational side of our field and the process model side of our field, to develop very useful posterior distributions. Of course, the analytic derivation of the posterior is out of range because the normalizer is intractable. So, we rely on the advances in your field and GIB sampling and mark up chain Monte Carlo algorithms.

The data statement for this particular problem—Bayesian formalism, as I hope I will get to at the end of the talk—is very amenable to multi-platform observation. In fact, we have a prototype model here that uses altimeter and scatterometer for the same problem. The altimeter measures the sea-surface height, from which we can infer ocean-surface currents. So, what we use as a data statement involves an incidence matrix, and this is another problem that satellite data introduced, and that the statistical community can readily address, changes of support.

We have a single point observation, perhaps, within the swath of a satellite data. We want to infer something about a process in the grid cell for a model. You people know how to do this, and we are beginning to deal with that. That is the kind of information that goes into this incidence matrix, K . On the other hand, when we are given the abundance of data that comes with the satellite overpass, the incidence matrix need not be very sophisticated, we have found, and we have simple nearest-neighbor algorithms at present that will yield the results that I am about to show. Then, as I said before, the measurement error models are the calibration validation studies.

For the process model, we use what we call stochastic geostrophy. This is a fundamental balance between the gradient of a pressure field and the implied velocity. Because we are on a rotating planet, any gradient in pressure or potential will initiate a flow from high potential to low potential but, because we are rotating, the resultant flow, which accounts for this rotation vector, will be along the lines parallel to the gradient. This is called the geostrophic relation, and we can translate this differential expression for the geostrophic relation into a probabilistic statement.

So, for our priors, we say that the zonal wind, given some pressure, some hidden process pressure and variance, is distributed normally, and the mean of that normal distribution is proportional to the gradient of the pressure, and the variance is expressed

in terms of the covariance of the wind field that we might know about from our observations. The second level, we prescribed a pressure field. The pressure, you know, is a good news/bad news sort of field. It is the reason why we can't do massively parallel calculations very efficiently in climate and weather forecasting, because it is quasielliptic. So, the perturbations in the pressure from remote places have a very important impact on the pressure at the grid point of interest. So, it is bad in that sense. It is good in the following sense: Since it is quasi-elliptic, it is relatively smooth, and it is well approximated by harmonic operators.

It turns out it is a good thing to hide in a Bayesian hierarchical models, because there are analytic expressions for surface pressure that fit well with meteorological processes, and we have done some regional studies with drifters in the region that give us space and time scales of variability for the pressure field there. So, we can prescribe a pressure process solely in terms of its covariant structure for models of that kind. Building that prior distribution, building the data distribution, using a Gibb's sampler, we generate the following posterior mean distribution for the surface winds.

There is already an important result here. Had the prior dominated the flow, as I told you, should be parallel to the isobar. In fact, the wind, the posterior mean wind here, is crossing isobars, and that means that the satellite data has spoken. The Bayesian formalism requires that we get a distribution for not just the dependent variable of interest in the deterministic sense, but also all the parameters and the hidden processes. So, in addition to the surface wind, we have a posterior distribution for surface pressure as well.

The right-hand panel shows what the weather center forecasts for this particular time also, and came up with in a deterministic model. This was a single realization from a forward model. All they came up with is the following pressure distribution. When we overlay the original satellite data it shows, in fact, they misplaced the low-pressure center. So, their region of ocean deep convection triggering would have been in the wrong place and, in fact, the intensity was considerably weaker than it is in the posterior mean distribution from the Bayesian hierarchical model. We have done a similar and more sophisticated Bayesian hierarchical model to retrieve surface winds in the Tropics.

In fact, thanks to Tim Hoar we are providing 50 realizations of the most specifically reasonable surface winds from the Indian Ocean to the dateline, 40 degrees North and 40 degrees South, four times a day. That is going to be available, and it will be interesting to see what the geophysical community does with it. I know what I am going to do with it, but there is a great deal that can be done with 50 realizations in terms of putting error bars on the deduction of weather and climate processes that we study.

Typically, for example, John Bates mentioned the Madden-Julian Oscillation. Well, what we typically have to do to study the process of the Madden-Julian Oscillation, which takes about 10 days to propagate across the Indian Ocean and into the western Pacific, is average several events. These events happen every 40 to 50 days when they happen, and then they don't happen for a while. So, the background flow system is completely different in the situations that you have to composite, in some sense, to get an idea of what the generic Madden-Julian Oscillation looks like. Now, with 50 realizations of the surface winds for a single Madden-Julian Oscillation, we have a different concept of what the error bars are going to be on in relationships between, for example, surface convergence and propagation of this wave. That is an aside. The Bayesian hierarchical model that describes that wind blending in the Tropics has been published in JASA.

There is a Wikle et al. (2001) paper that I would refer you to, and also Chris Wikle's Web page. That is one of the 30 or 40 recent publications on his page. What I would like to talk about now is the prototype atmosphere ocean model that is also set in the Bayesian hierarchical context.

This is an analog, a probabilistic analog, for the centerpiece tools for climate analysis and climate forecast. People who analyze climate run massive—we really do mean massive now—atmosphere-ocean coupled simulations on the largest supercomputers that they can find, and they provide a single deterministic realization at the end of the day. I think that this community can guide well those kinds of calculations, which are very expensive, by building the essential PDF, that whatever formal models of simulation they choose to run have to go through in some mode or some parameter sense. I can talk more about that in the breakout session.

What we did was combine the atmosphere model that I have just described for the Labrador Sea and an ocean model with slightly more sophisticated physics for the prior. We separated in the data stage or the likelihood the errors with respect to the atmospheric process, and the scatterometer data from the errors in the altimeter data from the ocean process. So, those were independent. In the process model stage, we simply factored the joint distribution between the atmosphere and ocean processes.

In the atmospheric process times an ocean process, we come up with the posterior interests, which are a process for atmosphere and oceans, all the parameters, given scatterometer and altimeter data. Then the horrible normalizer, of course, is the simulation method, which is here. The simulation method is particularly clever, and this was Mark Berliner's design, and I will come to that, I hope, in the end. So, the process model was built on now a dynamic differential equation that has proved itself. It is the original ocean model, actually. It is called quasi-geostrophy.

We have terms for the evolution of the ocean stream function, ψ , non-linear effects, convection, planetary vorticity, forcing by the surface wind, bottom friction and internal friction. The first step that a deterministic modeler would take would be to discretize these on a grid of interest and form matrix operators, and that is done here. This is changing a differential equation into a difference equation, a very standard technique.

You will notice that it is also very Markovian. We have matrix expressions operating on the previous time-level stream functions to give us an estimate of the next time-level stream function.

We also separate out the boundary conditions which, when we jump to probabilistic state, will become a boundary process, and that is a very big issue in geophysical modeling of limited area domains. So, here is the leap to probabilistic ocean stream function, and these operators are modeled directly after their finite-difference counterparts in the deterministic world. We have the linear operators on the previous time level, the non-linear operator, surface wind stress, boundary process, and we have added a model misfit term. This model misfit term is the only account for model error that forward model data simulation systems can make. In contrast, we have distributional parameters, which make these random variables in front of every term here.

So, we have a term-by-term management of uncertainty, and the uncertainty can interact, in the way thatvection uncertainty should interact with the diffusion uncertainty in the dynamic. Along with this come several economies. Because the

deterministic system is stiff, elliptic, difference equation, they are constrained to take very small time steps that are not relevant to the physics of interest.

It is not important to a polar low, what is happening on 15 second time intervals but, because it is an elliptic system, the way it is written in deterministic space, they are constrained to take 15-second time steps, and this drives a huge expense. They are also constrained to take very small spatial steps in their models. In a probabilistic model, we are not so constrained.

There must be a constraint that makes physical sense of some kind in the probabilistic world, and this is the sort of theoretical problem that I think this community could pursue and make large contributions. Nonetheless, we can take six hour time steps and three times the grid space in our air-sea hierarchical Bayesian model. The algorithm—I can't go through it given time—is a clever combination of Markov chain Monte Carlo for the atmosphere, and important sampling Monte Carlo for the ocean. That is atmosphere-ocean physics in probabilistic space. The importance weights are the likelihood distributions for the ocean data.

So, I build a catalog of forcing from the atmosphere. I go ahead and generate the ocean stream functions that come from every member of that catalogue. Then, I say the important ones have to do with how they—the data distributions from the ocean sensor, the altimeter oriented.

We tested this model in what is called an observing system simulation experiment. We had a truth simulation from primitive equations, a more sophisticated physical set, very high resolution, and compared the Bayesian hierarchical model, posterior distribution, with that truth simulation over 10 days. First, we had to spin up the truth. So, for a year we forced the box that looked like the Labrador Sea with this kind of wind, and generated this kind of ocean stream function equivalent in primitive equations.

There is a cyclonic eddy in the southwestern corner, and a rim current that is similar to the Labrador Sea, and then closed eddies on the interior of that rim current. Then, we idealized the surface forcing of a polar low, and sampled it as though we had a scatterometer, and sampled the ocean as though we had an altimeter, corrupted those data with proper measurement noise, fed those to our data stages in the Bayesian hierarchical models.

So, these are days one, three, five and seven of the simulated data. You can see the ocean stream function evolving underneath the altimeter. The altimeter tracks are here. This is representative of the TOPEC system. The simulated scatterometer is representative of the sea wind system, and you can see the polar low, which is perfectly circular and propagating perfectly zonally across this box, sampled, and then depart from the domain.

This is a comparison of truth simulation on the left and the Bayesian hierarchical model posterior mean distribution on the right, for the same four days, one, three, five and seven of a 10-day simulation. I will show you a difference map in a minute, but the main difference is that the Bayesian hierarchical model is actually more responsive, in a posterior mean sense, to this polar low, than was the primitive equation model.

As a physicist, my conception of statistical models always used to be, yes, they were generally right, but man, they were really sluggish and smooth, and the real detail that I needed wasn't available. Well, that seems to be quite the opposite in a posterior mean, let alone the realizations from the posterior distributions.

Here are the difference plots, and I draw two columns here. One is the difference for the full Bayesian hierarchical model. The other is the difference when I exclude that importance weighting. So, here, on the right-hand column, is a Bayesian hierarchical model for which we did not supply altimeter data in a separate data stage. You can see that this allows us to quantify, in a distributional sense, the value added of the altimeter data to the atmosphere ocean problem.

Interestingly, all of the differences isolate with features of interest. So, the cyclone in the southwestern corner is a place where differences exist and, because we have a posterior distribution, it is a place where uncertainty exists. This is a map of the standard deviation as a function of space for day seven, day five, day three, day one. Notice also that the boundary process is emerging as a source of uncertainty, and that is very consistent with the experience in forward modeling in limited area domains in the atmosphere and ocean. I am going to skip my last slide, which is a summary slide for this model that is in the abstract, and get to my conclusions.

The regional and global surface wind data sets from space are important. With two sea wind systems, there will be eight times 10^5 surface vector wind retrievals every 24 hours. That is a Level 2 product. A Level 1 product is an order of magnitude bigger than that. Those are the backscatter observations. I would never use a Level 3 product—I hate to say this—because Level 3 depends very much on what you want to do, and I will build my own Level 3 products, as I have seen. I have blended winds from the weather center and the satellite.

[Question off microphone from audience.]

MR. MILLIFF: Level 3 is for eye candy. It makes for the prettiest slides and animation and things, but you can't do science. The problem that polar-orbiting and even equatorial-orbiting satellites pose for geophysicists is that they don't appear on regular grids, they don't have uniform spatial—they don't leave a global field with uniform spatial and temporal resolution. So, that is a key issue that Amy brought up.

We have used physical constraints to drive a process to build those uniformly distributed spatially and temporally varying grids. Multi-resolution wavelets to impose this spectral constraints. Bayesian hierarchical models exploit the massive remote sensing data sets. I think what I expect to hear, in parts of this meeting, is that we have a problem of trying to find a needle in a haystack.

I think what geophysicists need to say is that, wow, there is a haystack that we can use. Never mind the needles. We used to just put the needles out there as a few moorings in a few places. Now, in fact, there is a whole probability distribution that needs to be exploited that comes from these satellite data.

Bayesian hierarchical models are amenable and readily adaptable to multiplatform data. The modern ocean observing system will involve remote sensors from space and in situ drifting, autonomous systems. The changes of support and distributional interactions of the uncertainty in the signals from those data, I think, are readily handled by the Bayesian hierarchical model approach. There has been a demonstration of air sea interaction through a Markov chain Monte Carlo-important sampling Monte Carlo linkage. Thanks.

Global and Regional Surface Wind Field Inferences Given Spaceborne Scatterometer Data

Ralph F. Milliff

Colorado Research Associates (CoRA) Division,

NorthWest Research Associates (NWRA)

collaborators:

L. Mark Berliner

Ohio State University

Christopher K. Wikle

University of Missouri

Doug Nychka

Tim Hoar

National Center for Atmospheric Research

Jan Morzel

CoRA/NWRA

research support:

NASA ESE Ocean Vector Winds Science Team

presentation to:

NRC Committee for Applied and Theoretical Statistics

Workshop on Massive Data Streams

13–14 December 2002

Washington, DC

**GLOBAL AND REGIONAL SURFACE WIND FIELD INFERENCES FROM SPACE-BORNE
SCATTEROMETER DATA**

Ralph F. Milliff

Colorado Research Associates (CoRA) Division,
NorthWest Research Associates (NWRA)

Collaborators:

L. Mark Berliner (Ohio State University)

Christopher K. Wikle (University of Missouri)

Doug Nychka (National Center for Atmospheric Research)

Tim Hoar (National Center for Atmospheric Research)

Jan Morzel (CoRA)

The global ocean surface wind field transfers momentum, and modulates the transfers of heat and material properties (e.g. fresh water, CO_2 , etc.), between atmosphere and ocean. Momentum inputs create and sustain the wind-driven general circulation of the ocean, and heat and fresh water exchanges drive the thermohaline general circulation; both of which have important implications for Earth climate. On regional scales, the surface wind field is an indicator of synoptic variability affecting weather forecasts.

The surface wind vector field over the ocean has been observed by active scatterometer systems in space with increasing precision, coverage, and resolution since 1978. Table 1 indicates characteristics of past, existing, and planned scatterometer missions since sustained earth observing missions began in 1991. In scatterometry, radar pulses of known frequency and polarizations are directed at the ocean surface where they are scattered by capillary waves. The space-borne sensor detects the backscatter signal, from several different geometries (e.g. look, azimuth, and incidence angles) and across two polarizations, to return a normalized radar backscatter cross section, or σ_0 . The σ_0 are spatially averaged within so-called wind vector cells (WVC) that form an array spanning the satellite ground track along its orbit. For each WVC, a geophysical model function is fit to relate averaged σ_0 to wind speed and direction.

The NASA QuikSCAT (QSCAT) satellite bears the first SeaWinds scatterometer instrument. The WVC are ordered at 25 km resolution across an 1800 km swath, along a polar orbit that is declined 8°. Roughly 1,000,000 surface wind vectors are retrieved over the global ocean in about 14 orbits every 24 hr by QSCAT. A second SeaWinds instrument is planned for launch aboard the ADEOS-2 satellite of the Japanese Space Agency (NASDA) on

14 December 2002 (i.e. the same day this talk is scheduled for delivery at NRC/CATS!). For the first time, tandem scatterometer missions will return true synoptic resolution of the global ocean surface wind field.

Global and regional applications of surface wind data from scatterometer systems often require regularly gridded surface vector wind fields with physically consistent treatments for missing data (e.g. due to rain contamination and/or attenuation of the radar signals). Computer models for the simulation of the ocean general circulation have been shown to be sensitive to surface wind forcing on diurnal time scales. The implied space-time requirements do not match well with the native organization of surface vector winds from scatterometer systems occurring in swaths from westward precessing orbits. A variety of statistical models have been developed to infer global and regional surface vector wind fields from scatterometer observations, on regular grids, and at diurnal temporal resolution.

Blending QSCAT and Weather-Center Analysis Winds

A statistical model has been developed to blend scatterometer surface vector winds with surface wind fields from weather-center analyses to create global ocean datasets, 4-times per day, at 0.5° resolution (<http://dss.ucar.edu/ds744.4>). The blending methodology is constrained by an approximate power-law relation that is observed, with regional and monthly variability, in wavenumber spectra for surface winds from scatterometer systems. The weather-center winds are used only in swath gaps and missing data regions with respect to the QSCAT orbits. They must be augmented at high wavenumbers to retain the regional and seasonal power-law relation that is observed. The augmentation is implemented in a multi-resolution wavelet procedure designed by Chin et al (1998).

Figure 1 depicts three panels of the global wind stress curl field for 24 January 2000 at 1800 UTC. The wind stress curl is a scalar summary of the surface vector wind field that is useful to illustrate the blending method. The top panel shows the wind stress curl from the weather-center analyses field. Analysis fields combine the latest forecast field with relevant surface observations that accrue over the forecast interval (e.g. between initialization and verification times). In the middle panel, the wind stress curl derived in the QSCAT swaths for this time period are superposed on the analysis field. Note the higher wavenumber content in the surface wind stress curl from the satellite observations. The blending method operates to augment the higher wavenumber content of the weather-center analyses such that the

blended field is consistent with power-law spectral properties observed by the QSCAT. The third panel shows the wind stress curl for the blended field.

The blended winds have been used to drive regional and global ocean model simulations. Milliff et al. (1999) demonstrated realistic enhancements to the response of a relatively coarse-resolution ocean general circulation model (OGCM) to the higher-wavenumber winds in the blended product. Higher resolution OGCM experiments are in progress now.

Bayesian Inference for Surface Winds in the Labrador Sea

The Labrador Sea is one of a very few locations in the world ocean where surface exchanges of heat, momentum and fresh water can drive the process of ocean deep convection. Ocean deep convection can be envisioned as the energetic downward branch of the so-called global ocean conveyor belt cartoon for the thermohaline general circulation that is important in the dynamics of the Earth climate. The energetic exchanges at the surface are often associated with polar low synoptic events in the Labrador Sea.

A Bayesian statistical model has been designed to exploit the areal coverage of scatterometer observations, and provide estimates of uncertainty in the surface vector wind inferences for the Labrador Sea. Here, the scatterometer system is the NASA Scatterometer or NSCAT system that preceded QS-CAT. It has proved convenient to organize the Bayesian model components in stages. Data Model Stage distributions are specified almost directly from precise information that naturally arises in the calibration and validation of satellite observing systems. The Prior Model Stage (stochastic geostrophy) invokes a simple autonomous balance between surface pressure (a hidden process in our model) and the surface winds. The posterior distribution for the surface vector winds is obtained from the output of a Gibbs sampler.

An application of the Labrador Sea model for surface winds will be described at the end of this presentation. The first documentation of this model appears in Royle et al. (1998).

Bayesian Hierarchical Model for Surface Winds in the Tropics

The Bayesian Hierarchical Model (BHM) methodology is extended in a model for tropical surface winds in the Indian and western Pacific Ocean that derives from Chris Wikle's postdoctoral work (Wikle et al., 2001). Here, the Data Model Stage reflects measurement error distributions for QSCAT in the tropics as well as for the surface winds from the NCEP analysis. The Prior Model Stage is prescribed in two parts. For large scales, the length scales

and propagation of the leading modes of the equatorial β -plane are used. At smaller scales, once again, we invoke a wavelet decomposition constrained by the power-law behavior for wavenumber spectra in the tropics.

A recent implementation of this model generates 50 realizations of the surface wind field, 4-times per day, at 50 km resolution, for the domain 20° N to 20° S, 40° E to 180° E, for the QSCAT data record for the calendar year 2000. Figure 2 depicts snapshots of five randomly selected realizations for zonal wind and divergence fields for 25 December 1999 at 0000 UTC. Differences are smallest in regions recently sampled by QSCAT. This implies that the uncertainty in the observations is smaller than the uncertainty in the approximate physics assigned in the prior model stage.

Surface convergence in the tropics is a critical field in the analysis of the atmospheric deep convection process. However, single realizations of this field are rarely useful because divergence is an inherently noisy field. The production of 50 physically sensible realizations can begin to quantify the space-time properties of the signal vs. noise. The first use of this dataset will be to diagnose surface convergence patterns associated with the Madden-Julian Oscillation (MJO) in the regions where the MJO is connected to the surface by atmospheric deep convection.

A Bayesian Hierarchical Air-Sea Interaction Model

The Bayesian Hierarchical Model methods extend naturally to multi-platform observations and complex physical models of air-sea interactions. Berliner et al (2002) demonstrate a prototype air-sea interaction BHM for a test case that mimics polar low propagation in the Labrador Sea, given both simulated altimeter and scatterometer observations. Hierarchical thinking leads to the development of a Prior Model distribution for the surface ocean streamfunction that is the product of an ocean given atmosphere model, and a model for the atmosphere. The Prior Model stage for the atmosphere is a model similar to the Labrador Sea wind model introduced above.

Figure 3 compares the evolution of the ocean kinetic energy distribution in the air-sea BHM with a case from which all altimeter data have been excluded. The BHM resolutions are 3 times coarser in space and $O(1000)$ times coarser in temporal resolution with respect to a high-resolution “truth” experiment also shown in the comparison. Also, the “truth” fields are computed in a physical model that incorporates more sophisticated physics than those that form the basis of the Prior Model Stage in the air-sea BHM. Implications of this methodology to data assimilation issues in coupled general

circulation models will be discussed.

References

- Berliner, L.M., R.F.Milliff and C.K.Wikle, 2002: "Bayesian hierarchical modelling of air-sea interaction", *J. Geophys. Res., Oceans*, in press.
- Chin, T.M., R.F.Milliff, and W.G.Large, 1998: "Basin-scale, high-wavenumber sea surface wind fields from multi-resolution analysis of scatterometer data", *J. Atmos. Ocean. Tech.*, **15**, 741–763.
- Milliff, R.F., M.H.Freilich, W.T.Liu, R.Adas and W.G.Large, 2001: "Global ocean surface vector wind observations from space", in *Observing the Oceans in the 21st Century*, C.J.Koblinsky and N.R.Smith (Eds.), GODAE Project Office, Bureau of Meteorology, Melbourne, 102–119.
- Milliff, R.F., W.G.Large, J.Morzell, G.Danabasoglu and T.M.Chin, 1999: "Ocean general circulation model sensitivity to forcing from scatterometer winds", *J. Geophys. Res., Oceans*, **104**, 11337–11358.
- Royle, J.A., L.M.Berliner, C.K.Wikle and R.F.Milliff, 1998: "A hierarchical spatial model for constructing wind fields from scatterometer data in the Labrador Sea." in *Case Studies in Bayesian Statistics IV*, C.Gatsonis, R.E.Kass, B.Carlin, A.Cariquiry, A.Gelman, I.Verdinelli, and M.West (Eds.), Springer-Verlag, 367–381.
- Wikle, C.K., R.F.Milliff, D.Nychka and L.M.Berliner 2001: "Spatiotemporal hierarchical Bayesian modeling: Tropical ocean surface winds", *J. Amer. Stat. Assoc.*, **96**(454), 382–397.

Figure Captions

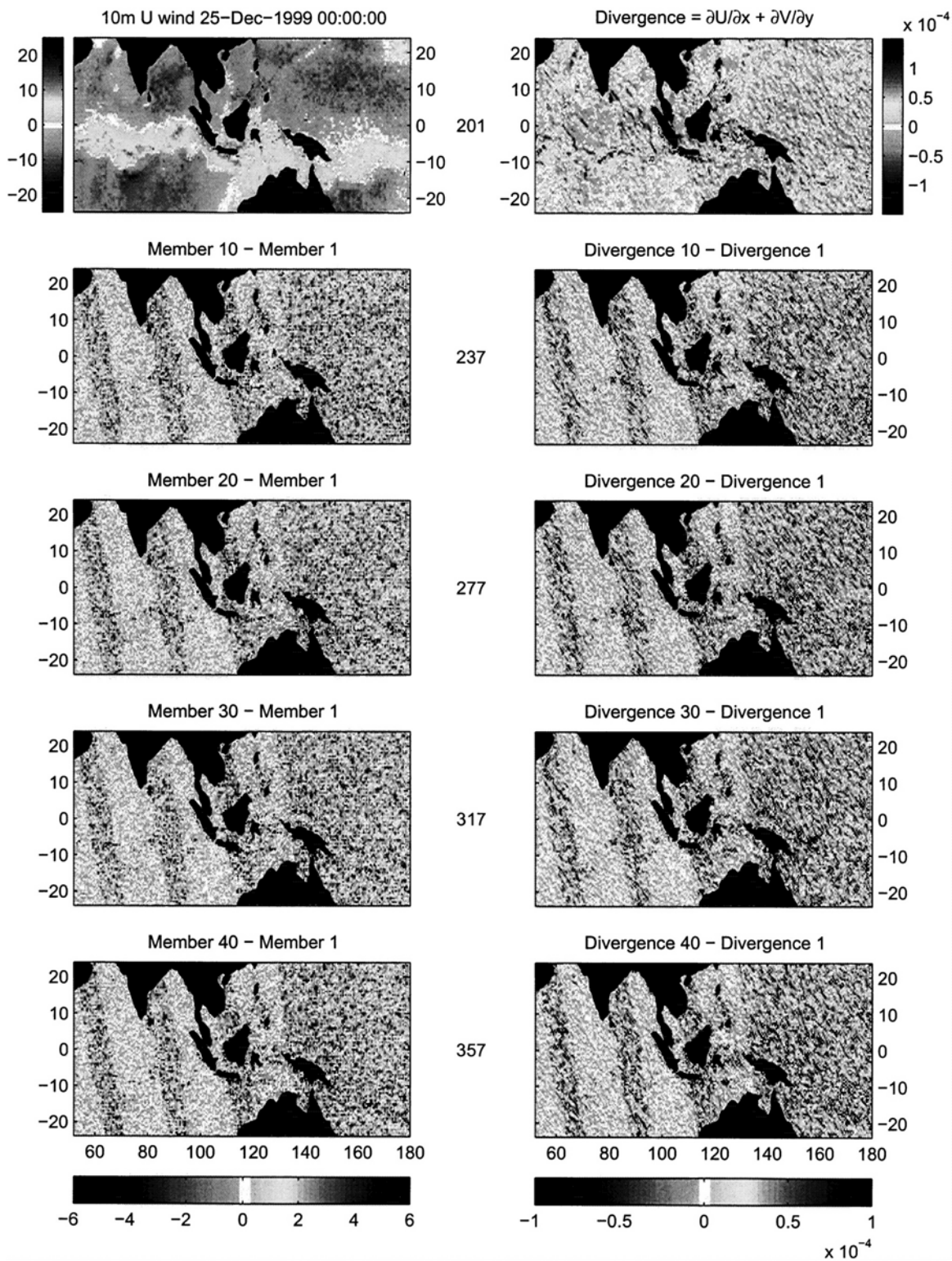
Table 1. Past, present, and planned missions to retrieve global surface vector wind fields from space (from Milliff et al., 2001). The table compares surface vector wind accuracies with respect to in-situ buoy observations. Launch dates for SeaWinds on ADEOS-2 and Windsat on Coriolis have slipped to 14 and 15 December 2002, respectively.

Figure 1. Three panel depiction of the statistical blending method for surface winds from scatterometer and weather-center analyses. Panel (a) depicts the wind stress curl for the weather-center analyses on 24 January 2000 at 1800 UTC. Wind stress curl from QSCAT swaths within a 12-hour window

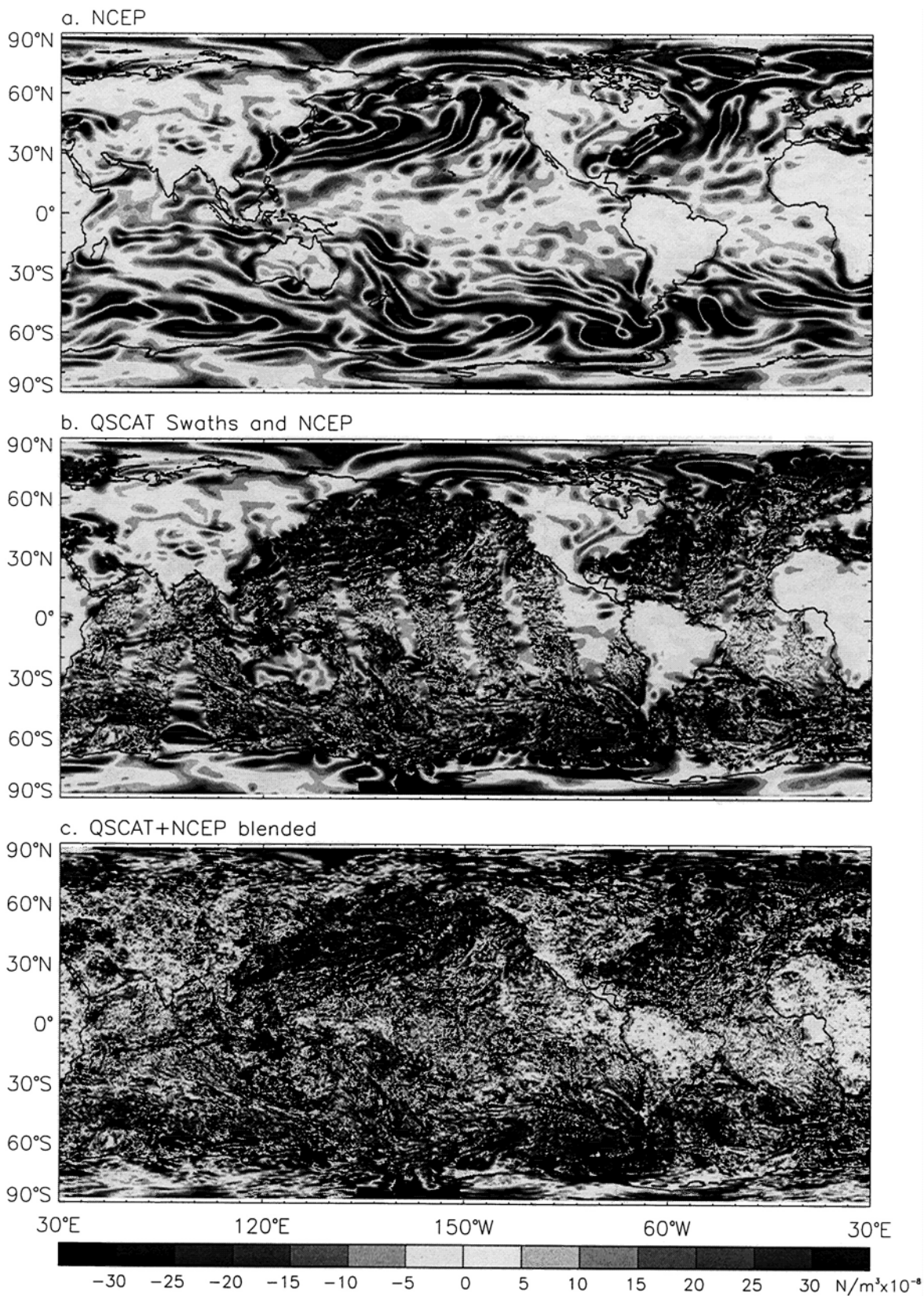
centered on this time are superposed on the weather-center field in panel (b). Panel (c) depicts the wind stress curl for the blended field. Derivative fields such as wind stress curl are particularly sensitive to unrealistic boundaries in the blended winds.

Figure 2. A Bayesian Hierarchical Model is used to infer surface vector wind fields in the tropical Indian and western Pacific Oceans, given surface winds from QSCAT and the NCEP forecast model. Five realizations from the posterior distribution for (left) zonal wind and (right) surface divergence are shown for the entire domain on 30 January 2001 at 1800 UTC. The two panels in the first row are zonal wind and divergence from the first realization. Subsequent rows are zonal wind differences and divergence differences with respect to the first realization. The differences are for realizations 10, 20, 30, and 40 from a 50 member ensemble of realizations saved from the Gibbs sampler.

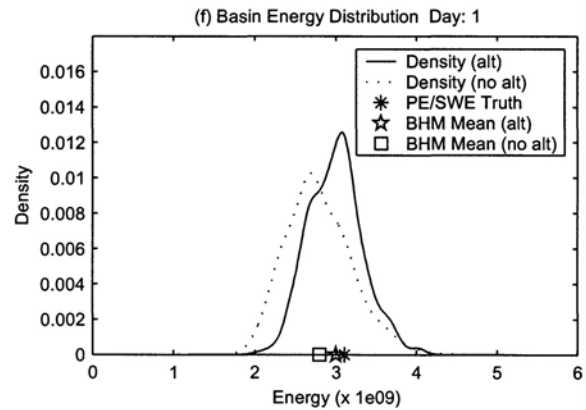
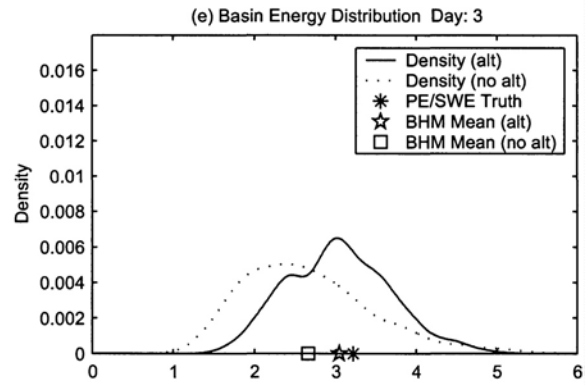
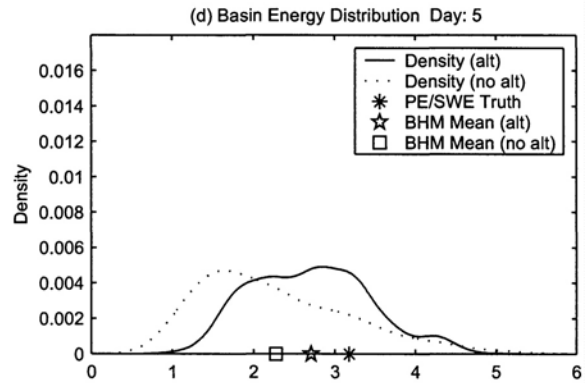
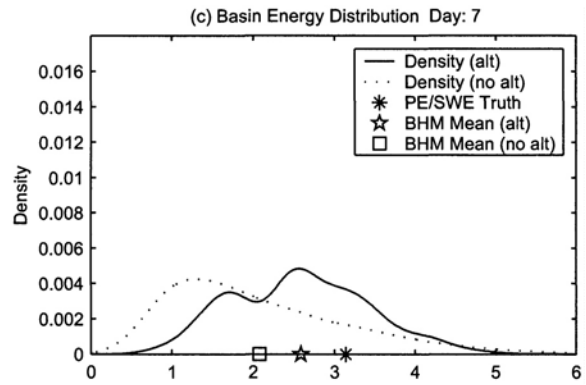
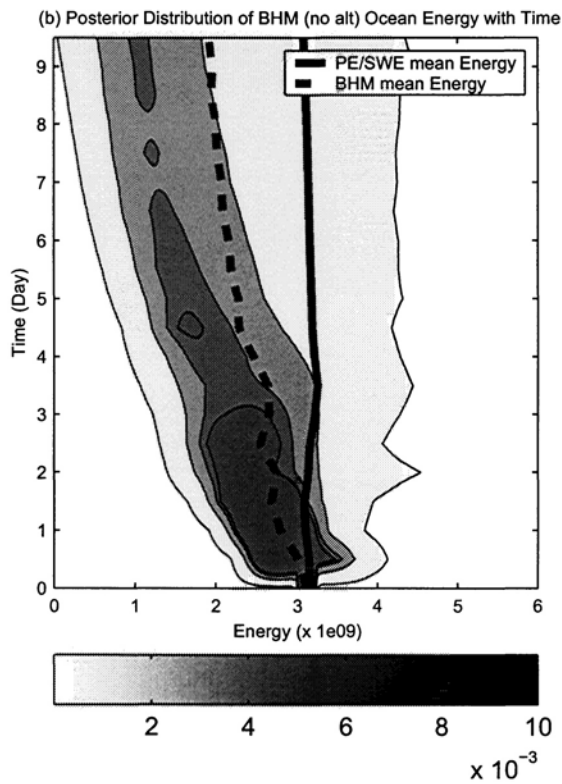
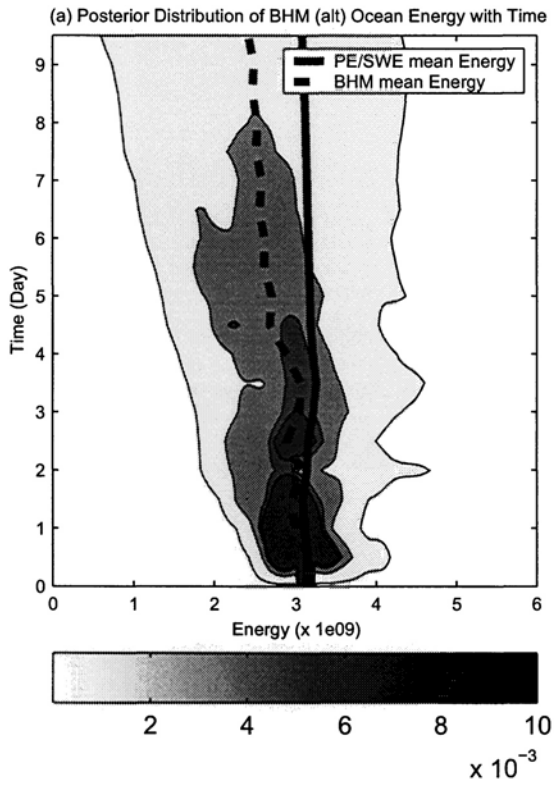
Figure 3. Summary plots for the Air-Sea interaction Bayesian hierarchical model (from Berliner et al., 2002). The basin average ocean kinetic energy distributions as functions of time are compared with a single trace (solid) from a “truth” simulation described in the text. The posterior mean vs. time (dashed) is indicated in panel (a) for the full air-sea BHM, and in panel (b) for an air-sea BHM from which all pseudo-altimeter data have been excluded. Panels (c-f) compare BHM probability density function estimates at days 1, 3, 5, and 7.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

| Mission | Measurement approach | Swath (km) daily cov. | Resolution (km) | Accuracy(wrt buoys) | URL(http://) |
|--|---|------------------------|-----------------------|--|--|
| ERS-1/2 AMI 4/91–1/01 | C-BAND SCATT. | 500/41% | 50 (~70) | 1.4–1.7 m/s rms spd 20° rms dir ~2 m/s random comp. | earth.esa.int |
| ASCAT/ METOP NSCAT 9/96– 6/97 | C-BAND SCATT. Ku-BAND SCATT. (fan beam) | 2×550/68% 2×600/75% | 25 50 (12.5) 25 50 | Better than ERS 1.3 m/s (1–22 m/s) spd 17° (dir) 1.3 random comp. | esa.int/esa/progs/ www.METOP.html winds.jpl.nasa.gov/ missions/nscat |
| SeaWinds/ QuickSCAT 7/99–present | Ku-BAND SCATT. (dual conical scan) | 1600/92% (1400) | 12.5 25 | 1.0 m/s (3–20 m/s) spd 25° (dir) 0.7 random comp. | winds.jpl.nasa.gov/ missions/quickscat |
| SeaWinds/ ADEOS-2 2/02 | Ku-BAND SCATT. (w/u- wave Rad.) | 1600/92% (1400) | (12.5) 25 | Better than QuickSCAT | winds.jpl.nasa.gov/ missions/seawinds |
| WINDSAT/ CORIOLIS 3/02 | DUAL-LOOK POL. RAD. | 1100/~70% | 25 | ±2 m/s or 20% spd ±20°?? | www.ipo.noaa.gov/ windsat.html |
| CMIS/ NPOESS 2010? | SINGLE-LOOK PO. RAD. | 1700/>92% | 20 | ±2 m/s or 20% spd ±20°?? (5–25 m/s) | www.ipo.noaa.gov/ cmis.html |

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

Summary

1. Global and regional surface wind datasets from spaceborne scatterometers are "massive" and important for *climate* and *weather*. Applications require:
 - regular grids
 - uniform spatial $O(10\text{ km})$ and temporal $O(\text{diurnal})$ resolution
2. Blended scatterometer and weather-center analyses provide global, realistic high-wavenumber surface winds
 - impose spectral constraints via multi-resolution wavelets
3. Bayesian Hierarchical Models to exploit massive remote sensing datasets
 - measurement error models from cal/val studies (likelihoods)
 - process models from GFD (priors)
 - advances in MCMC
4. Tropical Winds Example (Wikle et al. 2001)
5. Bayesian Hierarchical Model for Air-Sea Interaction (Berliner et al 2002)
 - multi-platform data from scatterometer and altimeter
 - stochastic geostrophy (atmos) and quasi-geostrophy (ocean) priors
 - MCMC to ISMC linkage for posteriors
 - term-by-term uncertainty
 - realistic covariance structures

Report from Breakout Group

Instructions for Breakout Groups

MS. KELLER-MC NULTY: There are three basic questions, issues, that we would like the subgroups to come back and report on.

First of all, what sort of outstanding challenges do you see relative to the collection of material that was in the session? In particular there, we heard in all these cases that there are real specific constraints on these problems that have to be taken into consideration. We can't just assume we get the process infinitely fast, whatever we want.

The second thing is, what are the needed collaborations? It is really wonderful today. So far, we are hearing from a whole range of scientists. So, what are the needed collaborations to really make progress on these problems?

Finally, what are the mechanisms for collaboration? You know, Amy, for example, had a whole list of suggestions with her talk.

So, the three things are the challenges, what are the scientific challenges, what are the needed collaborations, and what are some ideas on mechanisms for realizing those collaborations?

Report from Atmospheric and Meteorological Data Breakout Group

MR. NYCHKA: The first thing that the reporter has to report is that we could not find another reporter except for me. I am sorry, I was hoping to give someone the opportunity, but everybody shrank from it.

So, we tried to keep on track on the three questions. I am sure that the other groups realized how difficult that was.

Let me first talk about some technical challenges. The basic product you get out of this is a field. It is maybe a variable collected over space and time. There are some just basic statistical problems of how you summarize those in terms of probability density functions, if you have multiple samples of those, how you manipulate them, and also deal with them. Also, if you wanted to study, say, like a particular variable under an El Niño period versus a La Niña period, all those kinds of conditioning issues. So, that is basically, sort of very mainstream space-time statistics.

Another important component that came out of this is the whole issue of uncertainty. This is true in general, and there was quite a bit of discussion about aligning these efforts with the climate change research initiative, which is a very high level kind of organized effort by the U.S. government to study climate. Uncertainty measures are an important part of that, and no surprise that the typical deterministic geophysical community tends to sort of ignore these, but it is something that needs to be addressed.

There was also sort of the sentiment that one limitation is partly people's backgrounds. People use what they are familiar with. What they tend to do is limited by the tools that they know. They are sort of reticent to take on new tools. So, you have this sort of vicious circle that you only do things that you know how to do. I think an interesting thing that came out of this—and let me highlight this as a very interesting technical challenge, and it is one of these curious things where, all of a sudden, a massive

data set no longer becomes very massive. What John was bringing up is that these large satellite records typically have substantial non-zero biases, even when you average them. These biases are actually a major component of using these. So, a typical bias would be simply change a satellite platform that is measuring a particular remotely sensed variable, and you can see a level shift or some other artifact. In terms of combining different satellites, you need to address this. These biases need to be addressed empirically as an important problem.

The other technical challenge is reducing data. This is another interesting thing about massive data sets, that part of the challenge here is to make them useful. In order to make them useful, you have to have some idea of what the clientele is. We have had some discussion about being careful about that, that you don't want to sort of create some kind of summary of the data and have that not be appropriate for part of the user community. The other thing is, whatever summary is done, the assumptions used to make it should be overt, and also there should be measures of uncertainty along with it.

Collaborations, I think for this we didn't talk about this much, because I think they were so obvious. Obviously, the collaborators should be people in the geophysical community that actually work and compile this data with the statisticians.

Some obvious centers are JPL, NCAR, NOAA—Ralph, do you volunteer CORA as well?

AUDIENCE: Sure.

MR. NYCHKA: John, NCDC, I am assuming you will accept visitors if they show up.

AUDIENCE: Sure will. It is a great place to be in the summer, between the Blue Ridge and the Great Smokeys.

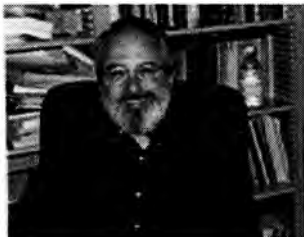
MR. NYCHKA: Okay, so one thing statisticians should realize is that there are these centers of concentrations of geophysical scientists, and they are great places to visit. The other collaboration that was brought up is that there needs to be some training of computer science in this. The other point, coming back to the climate change research initiative, is that this is another integrator, in terms of identifying collaborations. In terms of how to facilitate these collaborations, one suggestion was—this is post docs in particular—post docs at JPL.

I tried to steer the discussion a little bit, just to test the waters. What I suggested is some kind of regular process where there are meetings that people can anticipate. I am thinking sort of along the interface model or research conference model. It seems like the knee jerk reaction in this is simply, people identify an interesting area that they declare, let's have a workshop. We have the workshop, people get together, and then that is it. It is sort of the final point in time. I think John agreed with me, in particular, that a single workshop isn't the way to address this. So, I am curious about pursuing a sort of more regular kind of series of meetings. Okay, and that is it.

David Scott, Chair of Session on High-Energy Physics

Introduction by Session Chair

Transcript of Presentation



BIOSKETCH: David Scott is the Noah Harding Professor of Statistics at Rice University. He earned his BA in electrical engineering and mathematics in 1972 and his PhD in mathematical sciences in 1976, both from Rice University.

Dr. Scott's research interests focus on the analysis and understanding of data with many variables and cases. The research program encompasses basic theoretical studies of multivariate probability density estimation, computationally intensive algorithms in statistical computing, and data exploration using advanced techniques in computer visualization. Working with researchers at Rice, Baylor College of Medicine, and elsewhere, he has published practical applications in the fields of heart disease, remote sensing, signal processing, clustering, discrimination, and time series. With other members of the department, Dr. Scott worked with the former Texas Air Control Board on ozone forecasting, and currently collaborates with Rice Environmental Engineers on understanding and visualization of massive data.

In the field of nonparametric density estimation, Dr. Scott has provided a fundamental understanding of many estimators, including the histogram, frequency polygon, averaged shifted histogram, discrete penalized-likelihood estimators, adaptive estimators, oversmoothed estimators, and modal and robust regression estimators. In the area of smoothing parameter selection, he has provided basic algorithms, including biased cross-validation and multivariate cross-validation. Exciting problems in very high dimensional data and specialized topics remain open for investigation.

Dr. Scott is a fellow of the American Statistical Association (ASA), the Institute of Mathematical Statistics, the American Association for the Advancement of Science, and a member of the International Statistics Institute. He received the ASA Don Owen Award in 1993. He is the author of *Multivariate Density Estimation: Theory, Practice, and Visualization*. He is currently editor of the *Journal of Computational and Graphical Statistics*. He is past editor of *Computational Statistics* and was recently on the editorial board of *Statistical Sciences*. He has served as associate editor of the *Journal of the American Statistical Association* and the *Annals of Statistics*. He has also held several

offices in the Statistical Graphics Section of the American Statistical Association, including program chair, chair-elect, chair, and currently past chair.

TRANSCRIPT OF PRESENTATION

MR. SCOTT: This is a very statistically oriented committee, but we were very much interested in bringing in research scientists to help us understand the data opportunities out there, and to bring a good description of problems that might be available for research.

Certainly, our second topic today falls into this category in a nice way. It deals with high-energy physics. We have three outstanding speakers, two physicists and a computer scientist, to lead us in the discussion.

We want to remind everybody that we intend, in some sense, to have a question or two during the talks, if possible, as long as it is for clarification and hopefully in the discussion at the end, when we come back together, you will have a chance to sort of express your ideas as well. We would like to capture those.

I am editor of JCGS and again, I would like to extend an invitation to the speakers to consider talking with me about putting together a small research article for the journal, a special issue of the journal, later this year—next year.

With that, I would like to turn it over to our first speaker, who tells me that, as all physicists, he has traveled all around the world. He is now at Berkeley.

Robert Jacobsen

Statistical Analysis of High Energy Physics Data

Transcript of Presentation and PowerPoint Slides

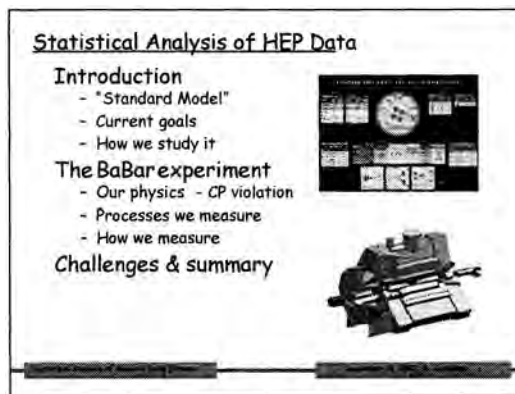
BIOSKETCH: Robert Jacobsen obtained a BSEE from Massachusetts Institute of Technology in 1978. He spent 1976 through 1986 working in the computer and data communications industry for a small company that was successively bought out by larger companies. He left in 1986 to return to graduate school in physics, obtaining his PhD in experimental high energy physics from Stanford in 1991.

From 1991 through 1994, he was a scientific associate and scientific staff member at CERN, the European Laboratory for Nuclear Physics, in Geneva, Switzerland. While there, he was a member of the ALEPH collaboration concentrating on B physics and on the energy calibration of the LEP collider. He joined the faculty at Berkeley in 1995.

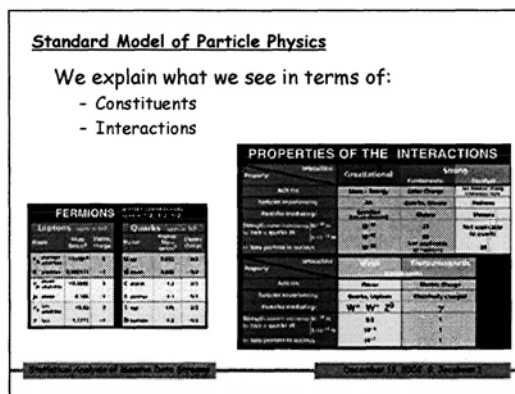
TRANSCRIPT OF PRESENTATION

MR. JACOBSEN: My name is Bob Jacobsen, and I am starting these three talks. So, I am going to lay some of the groundwork for what some of my colleagues will follow with.

My science is different from what you have just been hearing about because we actually have a chart which explains it all. Because we have that chart, which is really what we call a standard model, it affects how we do our science. It affects what questions are interesting and what we look at in an important way.



So, I am going to start with an introduction that actually talks about this science, the goals of it and how we study it. Then I am going to talk about a very specific case, which is something that is happening today, an experiment that has been running for two years and will probably run for three or four more years—that is a little bit uncertain— what we do and how we do it in the context of the statistical analysis of high-energy physics data. I am going to leave to you the question at the end, as to whether we are actually analyzing a massive data stream or not. Then, I am going to end with a few challenges, because I think it is important to point out that, although we get this done, we are by no means doing it in an intelligent way.



So, the standard model of particle physics explains what we see at the smallest and most precise level in terms of two things, a bunch of constituents, which are things like electrons and neutrinos and quarks, and a bunch of interactions between them, which

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

you can think of as forces, but we tend to actually think of as particles that communicate with each other by the exchange of other particles. Depending on how you count, there are basically three types of stuff, fermions, and there are basically four forces and we omit the fact that we don't actually understand gravity. We just sweep that under the rug and ignore it, and we do our experiments on electricity, magnetism, the strong force and the weak force.

These combine to what we see experimentally:

| Baryons and Antibaryons | | | | | |
|---|-------------------------|--------|------|------------|----------|
| Baryons and Antibaryons | | | | | |
| These are about 10 ⁻²⁶ cm in diameter. | | | | | |
| Symbol | Quark Content | Charge | Spin | Mass (MeV) | Life (s) |
| p | uud | 1 | 1/2 | 938 | ∞ |
| \bar{p} | $\bar{u}\bar{u}\bar{d}$ | -1 | 1/2 | 938 | ∞ |
| n | udd | 0 | 1/2 | 939 | ∞ |
| \bar{n} | $\bar{u}\bar{d}\bar{d}$ | 0 | 1/2 | 939 | ∞ |
| Λ | uds | 0 | 1/2 | 1115 | ∞ |
| $\bar{\Lambda}$ | $\bar{u}\bar{d}\bar{s}$ | 0 | 1/2 | 1115 | ∞ |

| Mesons | | | | | |
|---|---|--------|------|------------|-------------------------|
| Mesons and Antimesons | | | | | |
| These are about 10 ⁻¹⁵ cm in diameter. | | | | | |
| Symbol | Quark Content | Charge | Spin | Mass (MeV) | Life (s) |
| π^+ | u \bar{d} | 1 | 0 | 137 | 2.6 × 10 ⁻⁸ |
| π^- | d \bar{u} | -1 | 0 | 137 | 2.6 × 10 ⁻⁸ |
| ρ^+ | u \bar{d} | 1 | 1 | 770 | 1.7 × 10 ⁻⁸ |
| ρ^- | d \bar{u} | -1 | 1 | 770 | 1.7 × 10 ⁻⁸ |
| π^0 | $\frac{1}{\sqrt{2}}(u\bar{u} - d\bar{d})$ | 0 | 0 | 135 | 8.4 × 10 ⁻¹⁷ |
| η | $\frac{1}{\sqrt{6}}(u\bar{u} + d\bar{d} - 2s\bar{s})$ | 0 | 0 | 548 | 7.1 × 10 ⁻¹⁷ |

$e^+e^- \rightarrow \gamma^* \rightarrow B^0 \bar{B}^0$

An electron and positron (antiparticle) colliding at high energy can annihilate to produce B⁰ and B^{0-bar} mesons. This is what we study in our experiments.

December 12, 2001 8:24am

It is important to point out that, just like you can't see—in the talks that we have just been hearing—you can't actually see the underlying processes that are causing weather. You can only see the resulting phenomenon. We can't see any of the things in that fundamental theory. What we actually see are built-up composite structures like a proton, which is made up of smaller things, and protons you can actually manipulate and play with.

More complicated ones are the mesons, which are easier for us to experiment with, so we generally do, but even then, we have to use these techniques to do our experiments. We have no other instruments except the forces we have just described for looking at these particles. So, what we do is, we interact them through things that look like this big blob on, for you, it would be the right-hand side.

For example, you bring an electron and anti-electron together. They interact through one of these forces, something happens and you study that outgoing product. This particular one is the annihilation of an electron and its anti-particle to produce two particles called B's, which is what we study. The actual interactions of those particles is what my experiments study. In general, when you put together approximately a dozen constituents, four different forces, many different possibilities for how these can happen, there is a very large bestiary of things that can happen.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

OK, but is it science?

Ideas backed by formal mathematical structure
 Precise prediction in many cases
 Backed up by many measurements

10.4. W and Z decays
 The partial decay width for gauge bosons to decay into fermions is given by (10.4.1)

$$\Gamma(W \rightarrow f_1 \bar{f}_2) = \frac{G_F^2 M_W^4}{4\pi} \sum_{\text{spins}} |\mathcal{M}|^2 \approx 2.083 \times 0.6 \text{ MeV} \quad (10.4.1)$$

$$\Gamma(W \rightarrow \nu \bar{\nu}) = \frac{G_F^2 M_W^4}{4\pi} \sum_{\text{spins}} |\mathcal{M}|^2 \approx 2.083 \times 1.5 \text{ MeV} \quad (10.4.2)$$

$$\Gamma(Z \rightarrow f \bar{f}) = \frac{G_F^2 M_Z^4}{4\pi} \sum_{\text{spins}} |\mathcal{M}|^2 \approx 0.2 \text{ MeV} \quad (10.4.3)$$

10.5. Introduction
 The standard electroweak model is based on the gauge group SU(3) x SU(2) x U(1), with gauge bosons $G_{8,3}$, $W_{3,2}$ and $B_{1,1}$. The SU(3) and SU(2) fermions, leptons, and their corresponding group coupling constants g_3 and g_2 . The SU(2) fermion doublets $\psi_L = \begin{pmatrix} u \\ d \end{pmatrix}$ and $\begin{pmatrix} \nu \\ e \end{pmatrix}$ of the SU(2) fermion doublets are doublets under SU(2), where $\mathbf{2} = \mathbf{2} \oplus \mathbf{2}$, and $\mathbf{1}$ is the SU(2) singlet. The SU(3) fermion triplets $\psi_L = \begin{pmatrix} u \\ d \\ s \end{pmatrix}$ are triplets under SU(3), where $\mathbf{3} = \mathbf{3} \oplus \mathbf{3}$, and $\mathbf{1}$ is the SU(3) singlet. The SU(3) fermion singlets $\psi_R = \begin{pmatrix} u \\ d \\ s \end{pmatrix}$ are singlets under SU(3), where $\mathbf{1} = \mathbf{1} \oplus \mathbf{1}$, and $\mathbf{1}$ is the SU(3) singlet. The SU(2) fermion doublets $\psi_L = \begin{pmatrix} u \\ d \end{pmatrix}$ and $\begin{pmatrix} \nu \\ e \end{pmatrix}$ are doublets under SU(2), where $\mathbf{2} = \mathbf{2} \oplus \mathbf{2}$, and $\mathbf{1}$ is the SU(2) singlet. The SU(2) fermion singlets $\psi_R = \begin{pmatrix} u \\ d \end{pmatrix}$ and $\begin{pmatrix} \nu \\ e \end{pmatrix}$ are singlets under SU(2), where $\mathbf{1} = \mathbf{1} \oplus \mathbf{1}$, and $\mathbf{1}$ is the SU(2) singlet. The U(1) fermion singlets $\psi_L = \begin{pmatrix} u \\ d \end{pmatrix}$ and $\begin{pmatrix} \nu \\ e \end{pmatrix}$ are singlets under U(1), where $\mathbf{1} = \mathbf{1} \oplus \mathbf{1}$, and $\mathbf{1}$ is the U(1) singlet. The U(1) fermion doublets $\psi_L = \begin{pmatrix} u \\ d \end{pmatrix}$ and $\begin{pmatrix} \nu \\ e \end{pmatrix}$ are doublets under U(1), where $\mathbf{2} = \mathbf{2} \oplus \mathbf{2}$, and $\mathbf{1}$ is the U(1) singlet. The U(1) fermion triplets $\psi_L = \begin{pmatrix} u \\ d \\ s \end{pmatrix}$ are triplets under U(1), where $\mathbf{3} = \mathbf{3} \oplus \mathbf{3}$, and $\mathbf{1}$ is the U(1) singlet. The U(1) fermion singlets $\psi_R = \begin{pmatrix} u \\ d \\ s \end{pmatrix}$ are singlets under U(1), where $\mathbf{1} = \mathbf{1} \oplus \mathbf{1}$, and $\mathbf{1}$ is the U(1) singlet.

Now, my wife likes to say that this is not actually science. This is a cartoon. I get a little bit defensive about that. It actually is backed up by a formal mathematical structure that lives on a little card that is about this big, which looks like that, and I am not going to go through it.

The advantage that we have with this actual physical theory is that we can make from it predictions. We can—there are a few little asterisks here, and we won't go over the details very much, but in many cases, we can actually make predictions to the level of four significant digits of what will happen in a particular reaction.

Open questions remain

Insoluble mathematics of QCD
 Restricts our domain of calculation
 Analogy to turbulence
 Use modeling & simulation to cope

The 18+ parameters are incalculable within the SM
 Too many?
 1000's of compounds => 4, then eventually 100 elements
 => 3 (proton/neutron/electron) to hundreds of "particles"
 => 5 (electron, neutrino, quarks) to 88 "particles"
 => ?

Now, these predictions have been checked many different ways, but there are some open questions that remain. The two basic categories of this are that we have the equivalence of turbulence. We have things that we cannot calculate from first principles. They happen for two reasons.

One reason is that the mathematics of this theory are, like the mathematics of turbulence, beyond us to solve for first principles. We do not yet know how to do that. We know how to model it, we know how to approximate it, we know how, in many cases, to pick regimes in which the thing simplifies to the point where we can make precise calculations.

If you walked up to us with a physical situation, we may not be able to calculate the answer to that. Further, because of the way these processes work—and this is a cartoon here very similar to the one that I just talked about, the interaction between an electron and an anti-electron, which is very simple but, as time gets on, it gets more and more complicated. It starts as a very high-energy pure reaction but, as that energy is spread among more and more particles and their reaction products, the situation becomes

more and more highly dimensional, becomes more and more complicated, just like the cascade of energy and turbulence. By the time you get down to the smallest level, you are dealing with something that is inherently probabilistic.

In our case, it starts probabilistic at the very topmost level. These interactions could go any one of a number of ways when they happen.

AUDIENCE: Could you tell us how much energy you are talking about?

MR. JACOBSEN: It depends on the experiment. This particular picture was made for one that was 100 giga-electron volts in the center mass. My experiment is actually 10. So, it is many times the RAS mass of the particles that we see in the final thing, and you are making a lot of stuff out of energy.

The other problem here is that, like every physical theory, there are some underlying parameters. Depending on how you count, there are approximately 18 numbers that are in this theory that are not predictable by the theory. They are just numbers that are added in. Now, we have measured them many different ways. We are a little sensitive about the fact that 18 is very large.

You know, is that too many? Wouldn't you like a theory of everything that has the zero or one numbers in it. Is 18 too many? I just say, thousands of chemical compounds were originally explained by Earth, wind and fire but, when people worked on it for a long time, they discovered they really needed 100 elements. That was good, because we could explain 300 elements in terms of protons, neutron and electrons but, when they really looked into it, those protons, neutrons and electrons but when they really looked into it, those protons, neutrons and electrons of the 1930s became hundreds of mesons and particles in the 1950s.

Those were eventually explained in terms of five things, the electron, the neutrino and three quarks but, when you look really into it, it turns out there are 88 of those. The modern version of the standard model particle physics has 88 distinguishable particles in it. So, no matter what we do, we seem to keep coming back to approximately 100.

This gives us a question as to whether or not there is another level of structure. Right now, we don't know the answer to that. The standard model doesn't need another level of structure, but it is certainly something we are looking for.

So, with these 18 parameters, we measure them. That is what we do. Is this complete? Is there anything more there? We don't know the answer to that. We know that some measurements that were just announced over the last couple of years and just updated about a week ago require us to add at least a couple more numbers.

Is it complete? Aside from numerology?

Recent neutrino measurements require extension to add neutrino mass
"trivial" vs "extensive" depends on point of view, and eventually data
But at least SM gets more parameters!

One part remains unconfirmed
"Higgs sector", origin of aspects of mass
Many possible detailed theories exist...

Statistical Analysis of Massive Data Streams December 15, 2009 8:30am-11:00am

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

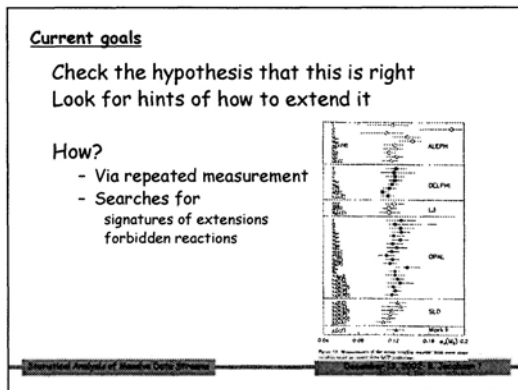
Well, that is bad. The theory is getting more and more complicated, but we are still explaining all of the data, and there is one part where we have experimental evidence completely lacking, for whether the theory is really working the way it is supposed to. Including people who will talk to you later on in this session, we are working very hard on that issue.

Current goals

Check the hypothesis that this is right
Look for hints of how to extend it

How?

- Via repeated measurement
- Searches for signatures of extensions
forbidden reactions

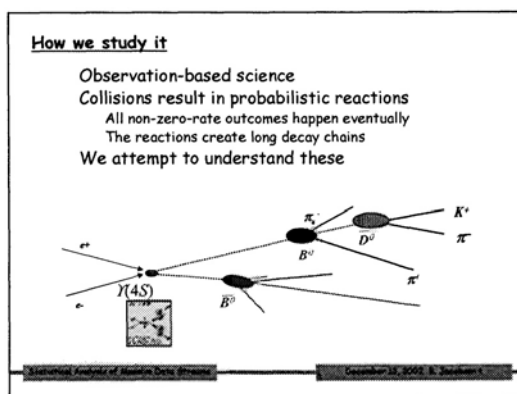
The slide contains text and a plot. The text is as follows: 'Current goals', 'Check the hypothesis that this is right', 'Look for hints of how to extend it', 'How?', '- Via repeated measurement', '- Searches for signatures of extensions', and 'forbidden reactions'. The plot is a scatter plot with several data points and error bars, labeled with 'ALEPH', 'DELPHI', 'L3', 'OPAL', and 'SLD'. The plot shows a distribution of points with error bars, likely representing experimental data from different experiments.

My current goal is to check a very specific hypothesis, which is that this theory, in its structure, as it stands today, is correct in describing. It may not be the entire story— but is it right? While we are at it, we are going to look for hints of how to extend it, but mostly, I am interested in how it is right.

The usual method of doing this in science, which was invented by Galileo, was that you measure things 50 different ways. If they are all explained by one theory with one underlying parameter, the first one tells you nothing, because all you have done is measure the parameter. The second one, though, should be consistent, and the third one should be consistent and the fourth one should be consistent.

The theory of gravity predicts that all bodies attract to the Earth at the same acceleration. The first object you drop tells you nothing, because you have just learned what that acceleration is. It is the second one that tells you something. So, this is an example of a whole bunch of measurements with their error bars made by different experiments of different quantities, all of which are related by this underlying theory, to a singular parameter.

We express these are measurements of the underlying parameter, but that is not why we are doing it. Nobody actually cares what that number is. What we are trying to do is determine the theory which is saying, all those things are related. They are all different except for the fact that some of them are replicated from different experiments. They are described by one single underlying phenomenon, and in this case, they are disgustingly well distributed. The chi squared of all those things is about .05 per degree of freedom. That is a separate problem.

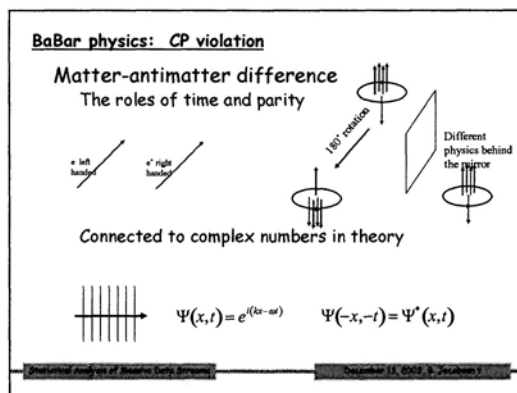


Okay, so, how do you study it? The problem is that we can't actually do an experiment in the classic sense. Like astronomers or people observing weather, we can't go out and say, I want a storm here, right now. That is not going to happen. So, it is an observational science. We collide particles in our machines to put a lot of energy in a small volume, and then we let these reactions take over and everything that is permitted by nature will eventually happen by nature. Some might be more common than others.

The typical reaction, though, is a little bit hidden from us. If I want to study the box that I showed you earlier, I can arrange for it to happen, but I cannot see the direct result. I can't see the interaction happen. That happens over a 10^{-29} of a second. I can't even see the things that it produces, because their lifetimes are just picoseconds. They will be produced. They will go some tiny little distance—that is a couple of microns.

They will decay into things. Some of those things will decay further into other things and, if I am lucky, I may be able to see all of these things and try to infer backwards what actually happened.

I have no control over this downstream process. I have only limited control over what I can measure from the things that come out of it. There are practical difficulties. What I want to see is what actually happens inside these little blocks.



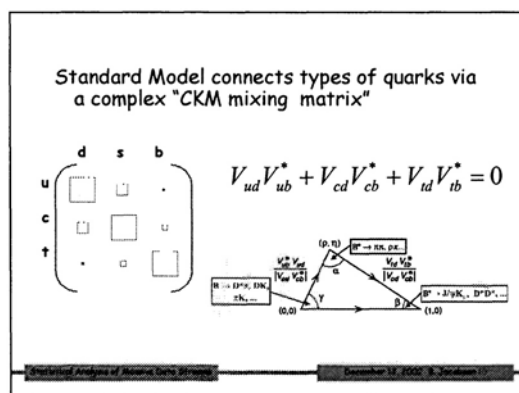
That is the introduction to high-energy physics. Now, I want to specialize in what we do. We are trying to understand why matter and antimatter are different. The world has a lot more of one than the other.

In this mathematical structure of the theory, this is intimately associated with the idea of time. Particle versus antiparticle, time and parity, left-handedness versus right-

handedness are all tied together in the mathematical structures in ways that have only really been elucidated in the recent past. I consider 15 years to be the recent past.

For those of you who are statisticians and, therefore, very familiar with numbers, there is a deep connection between the appearance of complex numbers in this theory that represent a wave, a propagating wave, as e^i times something. This connection between changing parity, which changes one direction to another, like looking in a mirror, that just changes the sign of your X coordinates. Time reversal, time going forward and backwards changes the sine of your time coordinates. These waves become their own complex conjugates, under these reversals, under these symmetries.

The bottom line is that we are trying to measure parameters, not just in terms of amplitudes, but also in terms of the complex nature of them, in a theory. We are interested in measuring complex numbers that nature seems to have. There are very few constants of nature that are complex. The gravitational constant, mass of the electron, tax rate—well, maybe not the tax rate—none of these are intrinsically complex.



The standard model has a 3-by-3 matrix of complex numbers that just appears in it as a mathematical construct. Because it is a matrix that is relating how things transform into others, it is inherently unitary. Everything you start with becomes a one of three possibilities. That means that these three amplitudes for doing this—for example, one of these rows, this thing could become one of those three things that is the sum of these.

When you actually put the mathematics of unitary together, what you get is—basically dotting these matrices into each other—you get an equation that looks like that. These rows are orthogonal to each other. You can represent that as a triangle. These are three complex numbers that sum back to zero, so they are a triangle.

The important point that you need to carry away from this is that these constants are controlling physical reactions. So, by measuring how fast some particular physical reaction happens, I can measure this piece or I can measure this piece or I can measure this piece. I am measuring things that go into this construct, the triangle. By measuring a very large number of reactions, I can—

AUDIENCE: What does CKM stand for?

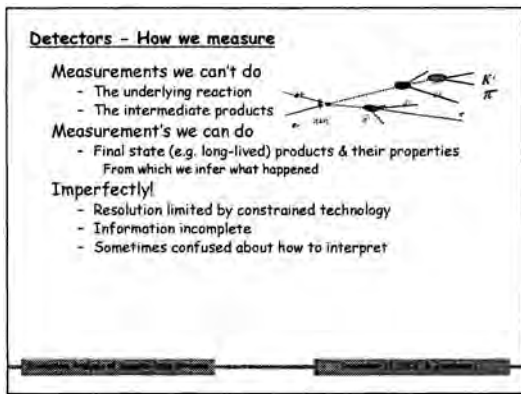
MR. JACOBSEN: Cabibbo-Kobayashi—Charlie will now give you the names of the three physicists.

CHARLIE: Cabibbo-Kobayashi-Maskawa.

MR. JACOBSEN: The three people who built this part of the theory. That was

before my time.

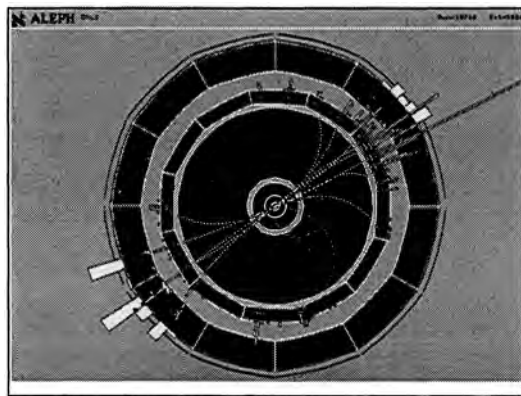
You build a large number of measurements and you try to over-constrain this. You try to see whether the hypothesis of this theory is correct, which says that, all those measurements, when combined properly, will result in a triangle that sums to zero. That is what we want to know.



Now, other experiments will have similar things they are trying to do. This is the cartoon of what actually happens. Let me remind you that we can't measure the underlying reaction or the intermediate products. We can only measure the things in the final state and, from that, we have to infer it.

Now, even worse, our experiment does not tell us everything that we want to do. The properties of these outgoing particles, their momenta or even their type, are only measured with finite resolution and it is not as good as we need it to be.

The information is incomplete in two senses. One is that sometimes things are not seen by the detector. There are pipes that carry particles in and cooling water in and information out. Particles that go in there are not measured, just like the satellites don't always tell you what you want at the time you want to see it. Our sampling of these events is imperfect.



It is often confused as to how to interpret. This is not from my experiment, but it is a great example. This is what happened in the detector during one of these collisions. There is no actual picture here, where the computer readout sensors inside the thing. The yellow green lines are where the computer said, ah, a particle went there because I can

line up all these little dots that correspond to measurements, and there is some energy deposited out here.

You will notice that there is one sort of going down toward 5:00 o'clock here that is not color. The computer says that that is not a real particle. No real particle made that line. The reason is, it has got a little tiny kink in it, part way along it. You can't really see it, but there is a missing dot and a kink right there.

What it probably did was bounce off an atom and deflect. Instead of making a nice smooth trajectory, it hit an atom and deflected. That doesn't happen very often, but it doesn't fit my hypothesis that something just propagated through the deflection. We will come back to this kind of problem. In this case, what the machine knows about what happened here is missing an important piece.

Acquiring & using data

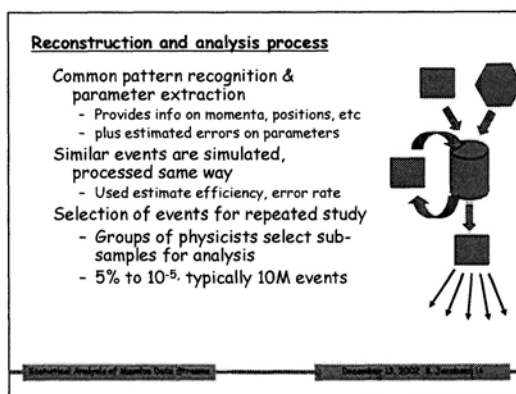
- 250M beam crossings per second (22TB/sec)
 - Mostly Coulomb scattering, rejected by hardware trigger
- 1K/s crossings result in end-products seen by detector (60MB/sec)
 - Most are "hard-scatters" and cosmic rays
 - Rejected by software trigger
- 100/s can't be distinguished from interesting events
 - Recorded to disk (600GB/day)
 - Forms basis for physics analysis
- We want as much (useful) data as we can get:
 - Typically run 2/3 of 24x7x365 for years
 - Currently looking at a few billion recorded events

I didn't expect to do a mine is bigger than yours kind of slide, but let me start. Twenty-two terabytes a second. We don't keep it all, because most of it is—we don't keep it all. Most of it is stuff that people understood 200 years ago. So, we aren't interested in that. The fact that like charges repel and opposite charges attract, or maybe it is the other way around, we have mastered that. We don't need to study that any more.

So, most of the 250 million collisions a second are just that, and there is nothing more to them than that. About a thousand result in things being thrown into the detector in ways that indicate something may have happened. So, the hardware detects that, drops this data stream down, both in space and time, fewer bytes per crossing, because you don't read out the parts of the detector that have nothing in them, and fewer per second.

Then, there is software that takes that 60 megabytes per second down to about 100 per second that can't be distinguished from interesting events. It turns out, we will talk later about how many events, but we record about 600 gigabytes a day of these events that are interesting. This is what we analyze.

Because there are so many things that can happen in this decay chain, and because only certain reactions are interesting, we need a lot of events, and I will talk about those statistics later on. So, we run as much as we can and we typically keep everything together for about two-thirds of always, and we are now looking at a few billion of these events, 500 terabytes, but I will come back to that number later on.



The process of dealing with it is that you take the data that starts in this little hexagon, which is the detector itself, you shove it onto disks. Then you have a file of processors that looks at it and tries to convert it into usable data.

I don't quite understand the terminology of Earth sensing, but I think this is Level 2. It is where you have taken it from raw bits, as to wire seven had so many erts, and you converted it to something went that-a-way, with a certain amount of precision. You try to get not just this precise measurement of that particular particle went that-a-way, but also an estimate of how precise that is. It is strictly MISR statistics estimates. The RMS of the scatter on this is expected to be [word missing].

We also do a tremendous amount of simulation which is very detailed. The simulation knows about all the material in the detection, it knows about all the standard model, it knows how to throw dice to generate random numbers. It is generating events where we know what actually happened because the computer writes that down, very similar to your simulations.

We try to have at least three times as much simulated data as real data but, when you are talking hundreds of terabytes, what can you do? We feed these all through farms and what comes out the bottom of that is basically in real time. We keep up with that. Then it goes through a process where I am not sure where it fits into the models that we have described before, but it is very reminiscent of what we heard the NSA does. We look for what is interesting.

We have had the trigger where we have thrown stuff away. Which phone you tap is sort of the analog of that. But now we take all the stuff that we have written down and individual people doing an analysis, looking for some particular thing they want to measure, will scan through this data to pick out the subset they look at.

The ones who are smart will accept something like one in 10^{-5} . The real things they are looking for are 10^{-6} , 10^{-7} , but they accept a larger set. People who don't put a good algorithm in there will accept 5 percent, and will have a huge data sample to do something with.

AUDIENCE: [Question off microphone.]

MR. JACOBSEN: The skins, the actual running through the thing, is supposed to be done three times a year. It is really done sort of every six months. It is a production activity.

AUDIENCE: [Question off microphone.]

MR. JACOBSEN: They will pick 10 million events out of this store, and then they will go away and study them for an extended period of time and write a paper. The

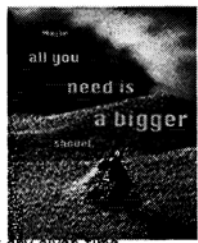

process of actually picking from billions of events is a large-scale computer project. So, they will—I will talk more about how they look over and over again at these things, but the actual project running to do the selection is scheduled and done as a collaborative effort several times a year.

AUDIENCE: [Question off microphone.]

MR. JACOBSEN That is about right. I will show you some plots.

Our development and computational task:

- Thousands of processors
- 100's of TB of disk
- Millions of lines of code
- Hundreds of collaborators



Limits what we can achieve at any given time

International Journal of High Energy Physics, December 20, 2006, © Jacobson, D.


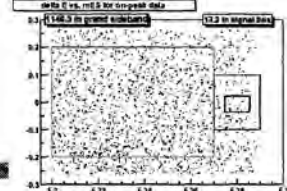
So, our task involves thousands of processors, hundreds of terabytes a disk, millions of lines of code, hundreds of collaborators, and I remind you that we still screw up a lot. We are doing the best we can but, for example, fixing the algorithm that didn't find that track involves interactions on all these levels, lots of code, more computer time, lots of people signing it off as the right thing to do.

I have to show this. I love this. We are in the state of this poor guy. The task has grown and our tools have not, and this limits what we can do in science. This goes directly to our bottom line.

Typical Analysis 1: Cut & Count for Specific Rate

Correct events will have lots of properties:

- Integers: Number of decay products, charge, etc.
Reject events with the wrong value
- Total energy, invariant mass
Define signal and background "boxes"



with 0.5% of the original data
0.2 to 0.3 in signal box
0.2 to 0.3 in signal box

So, once you have this sample of some number of events, what do you do with it? You are looking for a particular reaction. You are looking for this decayed to that, because you want to measure how often that happens, to see the underlying mechanism. You want to find out how many storms produce how much something or other.

So, you remove events that don't have the right properties. The events you are looking for will have a finite number of final state particles. The electric charge has to add up to the right thing, due to charge conservation, etc. You cut everything away that

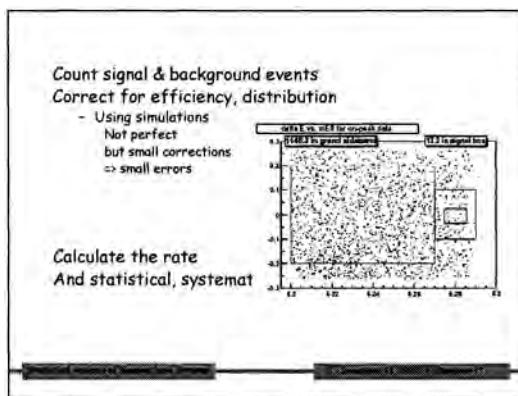
doesn't match that. Since the detector makes mistakes, you will have thrown away events that really were what you were looking for.

A typical efficiency in terms of that kind of mistake for us is 20 percent. We can discuss whether it is 10 percent, 25 percent, but it is routine to throw away the majority of events that actually contain what you are looking for because of measurement errors and uncertainty. We will come back to that.

In the end, you usually pick a couple of continuous variables like the total energy and the invariant mass. Invariant mass is just the idea that if you have something that blows up from the products, you can calculate how big the thing was that blew up. Everything is right, but you are still not certain they are the right ones, and you plot those two properties on these two axes for each of those events. You will get something that looks like this.

This is actually a simulated run. In this bigger box are 1,100 events, approximately, and in this little blue-purple box, are 13.2 events. You get fractional events, because this is simulated data and you have to adjust for the variable number, and we don't have exactly as many real data as simulated data.

So, all the ones that are in the blue box are most likely to be the right thing, because that is centered on the right values of these quantities. This big box is not centered on the right quantities. It is the wrong ones. It is our background. Those are accidentals that get through all of your cuts, that something else happened.



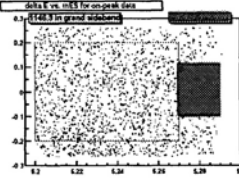
It is easy to see what you have to actually do. You just count the events in those two boxes, take the stuff in the big box, and project how much of the stuff in the little box is background, and you have to worry about the shape and all the other stuff. Do the subtraction, and that is your answer, and then you write the paper, Nobel Prizes are awarded, etc.

The way you do the extrapolation and the correction for efficiency is with these stimulations. They are not perfect but, to the extent that they are only small corrections, how big an error can you make on a small correction? So, you have to add systematic errors for that. Historically, though, there has been a tremendous concern, because everybody wants to be the first person to discover a particular reaction.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

Blind analysis

Historical concern about "tuning the cuts"
BaBar has adopted "blind analysis"



- Cover signal region until all details final
- OK to tune on simulat

Statistical Analysis of Massive Data Streams December 17, 2008 B. Berger

So, they tend to sort of put the lines in the place that gets that last event in. The BaBar has adopted a policy, which is followed about 85 percent of the time, called blind analysis. Blind but not dumb, is the phrase we actually use.

The analysts basically promise to make all their plots on their screen with this little box over the signal, while they are doing all their tuning. There is nothing that we can do to enforce that. These tools are on the desktop, but they do it. Then, once they have got all the details of their analysis final and they promise not to change anything, you pop that off and you look at it. This seems to be very effective, because nobody worries about it any more.

Typical Analysis 2: Fitting for the answer

Final answer extracted by max-likelihood fit

- Linear combination of PDFs to find rate(s)
- Plus underlying SM parameters
- Plus resolution, efficiency parameters
- Plus the kitchen sink...

From 1 to 10 (typical) to 215 parameters

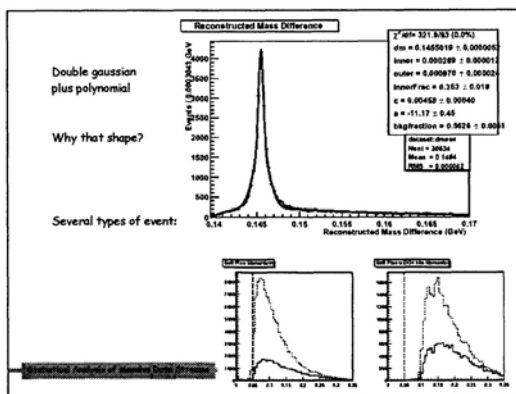
Basic problem: We really don't understand the distributions from first principles

Statistical Analysis of Massive Data Streams December 17, 2008 B. Berger

As time goes on, we will get more complicated analyses. We will do a maximum-likelihood fit that has some linear combination of probability density functions. To the statisticians these probably aren't probability density functions, but that is what we call them, to find the rates of many things that are going on.

Maybe that fit will have directly in it the standard model parameters that we are looking for. Maybe there will be parameters that describe a resolution. How much something is smeared out on the plot may actually be fit for, as it goes along.

A typical fit has 10 parameters in it, but I have seen ones that have up to 215. The basic problem here is that, if you are fitting in a maximum-likelihood fit, you want to understand the distribution. We do not understand most of these distributions on first principles.



We can extract them from simulations that we don't trust, or we can look at the data to try and figure out what they are. Now, we are not smart about it. This is a fit probability density function to some particular thing. You can't read this, because it is one of our tools, but it is the best fit by the sum of two Gaussians.

So, we parameterize it as the sum of two Gaussians and feed it into one of these huge fits. The real reason that it is the sum of two Gaussians is there are two distributions there. There are actually two populations.

In this plot down here, the high-energy physicist will motivate the fact that they are actually two types of events, but we do not separate them. It is very unusual, it was considered a major breakthrough for somebody to actually do a fit by doubling the number of parameters separating these two populations. That is how you get up to 215, a bunch of simultaneous fits.

Usually, we just parameterize these. The reason is subtle. The reason is that, even though we know it is two populations, we are still going to have to fit for two Gaussians, because we don't have a first order understanding of what those things should actually look like. So, even though we calculate the errors on many quantities, we calculate them in a very simplistic Gaussian way, and we don't have any way of representing the more complicated things that are going on.

Features of typical analysis

- Typically 10k to 100k events in final selection
 - Generally fits on the desktop
- Analysis done with small scale tools
 - PAW, ROOT plus user code
 - Analysis turnaround time comparable to attention span

Are we still doing "Statistical Analysis of Massive Data Streams" at this point?

To come back to your question, there are typically 10K to 100K events in that final selection that was done in production, and that fits in a desktop computer, and that is really what sets how many of those are.

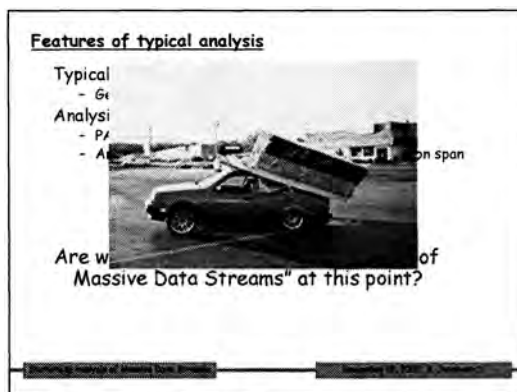
The analyses are done with very small scale tools. They are physicist-written,

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

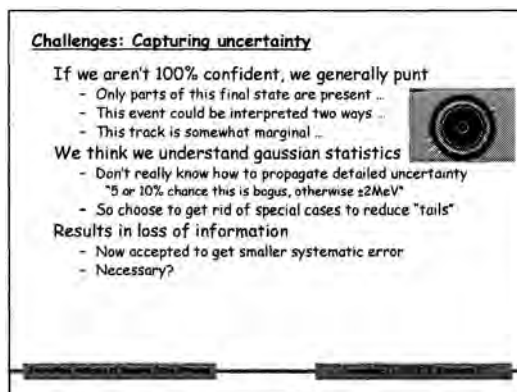
general-purpose tool boxes, but users write their own algorithms for selecting events. The analysis turn around time is analogous to the attention span. That means that they will write code that is just complicated enough that it will complete before they come back with a cup of coffee.

Now, my question to you—we don't have to answer this now—is I am not sure that we are doing statistical analysis of massive data streams at this point. The only way we know how to do it right now is to get it onto somebody's desktop, somehow make it fit, and in the next talk we may hear about better ways to do this.

This is the situation we are in. We have bigger jobs, and our tools are just not up to it. Every time we attempt it, we get embarrassed.



Look at this picture, actually taken at a Home Depot. You will notice the engine is actually running. This guy tried to drive home.



In my last couple of slides, I want to turn this around the other way. I want to say something to the statisticians in the office, what I perceive the real challenges in your area to be. All of them are this idea of capturing uncertainty. We do a tremendously bad job of dealing with the fact that our knowledge is poor, in many different ways.

What we do basically now is, if we suspect that we have something funny going on, we punt. We throw the event away, we run the detector for an extra week, month, year, whatever. So, if only parts of the final state are present because of a measurement error, if, when we look at the event, we could interpret it multiple ways, which can happen, there was this event—this cartoon is to remind you of the marginal track.

If something is marginal because maybe something a little bit more complicated than the first order thing I am looking for has happened, we throw it away. You can easily do back of the envelope estimations that we are talking about factors of three in running time for experiments that cost \$60 million by doing this. Physicists, particularly high-energy physicists, think that we have completely mastered Gaussian statistics, and refuse to admit that there are any others, although secretly we will mention Poisson.

We don't know how to propagate detailed uncertainty. We don't know how to deal with the fact that the real distribution of error on one of these tracks is, 98 percent of the time it is an MISR of this width and 2 percent of the time it is a Gaussian of that width. We don't know how to combine all that throughout all the stages of this analysis to get out something that makes sense in the end.

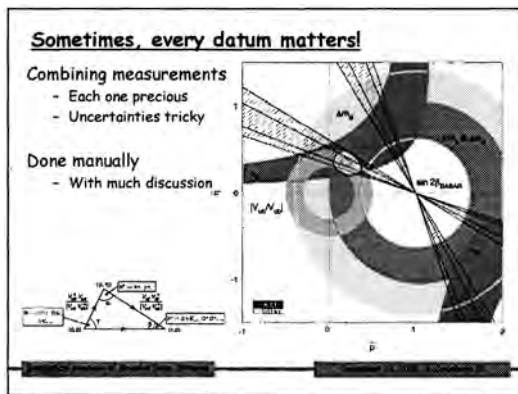
So, we choose to get rid of those special cases out of a fear of tails in our distribution. When we can't do that, we model it with these PDFs, which we don't really understand, and which are throwing away all the correlation information between those two, because we can't see that at all.

This is accepted now, because it causes a lot of statistical power. In the absence of a way of dealing with it—I don't want to say formally, I want to say principled way of dealing with it, we do not know how to assign a systematic error if we allow these things to come in.

It is not even so much that the systematic error from these will be large. We don't know what it should be. So, we don't really know how to balance off the complexity of our true information and the statistical power of our true information.

As you go through the literature in high-energy physics, you will find people who will address this in one particular place in an analysis where it is useful.

There has been no systematic approach, which is where the real gain is. The real gain will come from ten 5 percent gains, not from one 5 percent gain. Now, sometimes, every data matter. When what you are doing is the real final result, taking that triangle which is shown in cartoon form here to remind you, and combining measurements that took hundreds of people years to make, because different experiments are measuring different things, and trying to see whether this is right, you can't throw out the work of 500 people because there is a tail on the distribution. You will get lynched.



So, the way that we do that now is that we do it manually. People called theorists get together at conferences and they argue.

This particular plot is the result of six theorists for three weeks, trying to deal with

the statistics of combining data with non-Gaussian error. The way that you interpret this is that each little color band or hatched band is a different measurement from a different experiment, and this little egg-shaped thing here, the yolk, is the best estimate of where the corner of the triangle is in this plane. The triangle actually lies along the horizontal midplane and sort of extends up to there. As these measurements get smaller and smaller error bars on them, this will tighten up until eventually, hopefully, some of them will become inconsistent with each other.

AUDIENCE: [Question off microphone.]

MR. JACOBSEN: No, actually. What we do, remember, I said that this combination of all possible processes has to obey unitarity. Everything has to become only one possible thing. So, unitarity also implies that certain things don't happen. Some things sum to zero. So, we have this triangle which each side of it is related to certain rates. To make it easier to deal with, since we don't know any of them, we divide all the sides by the length of this and put it in an arbitrary zero to one space.

So, we call these things an η and ρ , but they really are physical measurements. Everything is put in terms of one high-precision measurement. So, that is actually one of the things that is really spurious. That is not a good word to use.

One of the things that is very hard about this is that everything, with the possible exception of some of the bands that appear in the theory, and one of the plots—but not all of the plots—is subject to experimental verification. We have no scale in this plot that is not subject to experimental verification.

From here to here—I am not sure where the center is—there is the center, and there is the side. That is actually an experimental number. The width of that is actually secretly an experimental number.

As fundamental physicists, we worry about that. When you learned electricity and magnetism, you probably learned it in terms of the charge of the electron. That did not tell us the charge of the electron. Somebody had to go out and measure it and, every time that measurement changes, it changes everything. It is now known to nine significant differences, so it doesn't change your electric bill, but at this level, it has to be taken into account that everything is subject to experimental verification.

It is an unfortunate fact that there is nothing that is really well known in that plot. In 15 years, we will be doing this in high school, but right now, it is a very hard thing because there is a lot of uncertainty in that plot.

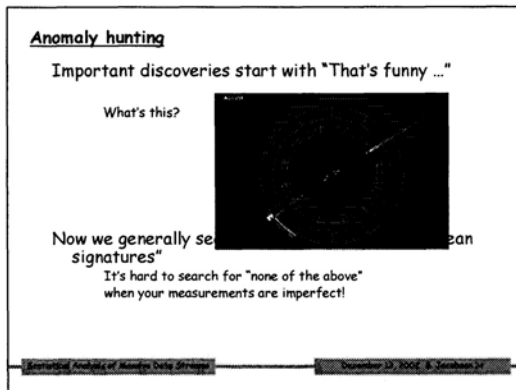
MR. SCOTT: Can you just briefly, when the theorists are sitting around in this room deciding how to draw it, what exactly are they using for this?

MR. JACOBSEN: Leaving the names out, there is one guy who says, the only way you can possibly do this is take all the likelihood curves from all of these measurements, pour them into a computer and have it throw numbers, and then make a scatterplot of the possible outcomes.

Then, I always discuss whether we draw contour lines or not. Someone else will say, but that is completely bogus because there are large correlations in the systematic errors that these experiments have made. They have made correlated assumptions, and then how do you deal with that?

His response was, well, we vary the underlying assumptions in one of these computer models. Someone else says, no, they told us it is plus or minus something in their abstract, that is all I know, that is what I am going to draw.

In fact, the one that actually won was the first one that I described. They put all this stuff in, they ran a computer model that put dots on a screen. The question is, if you have got a whole bunch of dots on the screen, where do you draw the 90 percent? One has to enclose 90 percent of those. Which 90 percent? I am not a statistician. My job is to make that little angled line in there. I don't understand, really, this, but I know that they don't either, because I can show you six other plots, and they are not the same.



One last thing. I promised I would get done close to on time. Everybody—I teach freshman physics a lot. Everybody believes that science starts with “Eureka” and Archimedes running down the street naked. Not my science. My science starts with, “Hmm, I wonder what went wrong there.” Let me give you an example of this.

This is a famous event. It has actually been redrawn with a different depictographic package, so it is actually a simulated event. The real event that looked like this led to the discovery of the τ , Nobel Prizes were awarded, etc., etc., etc. What this event was, was this can't happen. In the standard model, as it was established at that time, the rate for this process was identically zero. Zero does not fluctuate to one.

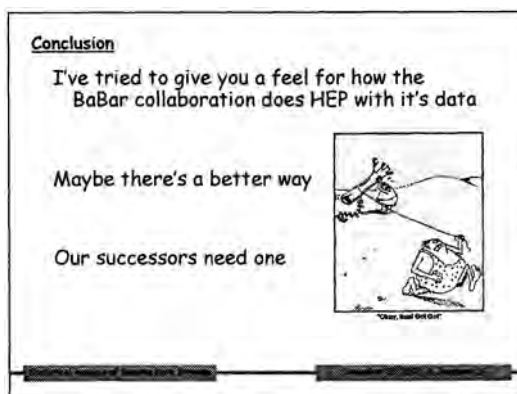
This was not noticed by a computer program that was looking for it. Why do you look for something that can't happen? That is not a great way to get measurements. It was noticed by someone flipping through events and saying, “Oops, what is that?” The process that I have outlined, which is the modern process of the last five years, is not the process of flipping through a significant fraction of events. You can't look through a significant fraction of a billion events.

So, we are, in fact, quite weak on the fact of finding things that don't belong. This is particularly difficult because nowadays, although we have a theory that predicts this, one of my students checked. Approximately 15 percent of the events that look like this are due to mistakes in the detection.

Anomalies can be due to mistakes. The standard models predict that an electric charge is 100 percent conserved all the time. If you could prove that an electric charge was not conserved, automatic Nobel Prize. A lot of our events don't conserve charge, because we will offset the charged particle. Something disappeared and we didn't see it. So, if you naively add up the charges, it doesn't sum to zero. It sums to something else. So, how do you look for anomalies in the presence of mistakes is something that we don't know how to do.

The way we do it now is say, “Oh, this weird thing that can't happen, could be happening.” Let's go look for it as a clean signature. We search positively. Let's see if

electric charge is conserved. We do a long, complicated analysis, and the bottom line is that, with such and such confidence level, an electric charge is conserved. It is a search paper, and there are a lot of those. They are all searches for specific things. It is very hard today to notice an anomaly.



Okay, here are my conclusions. I tried to give you a feeling for how we do high-energy physics with these data. In the discussions, I really hope to hear about better ways, because our successors definitely need one.

I am in the position of these guys, who are back at the invention of the kite, and they are working with the technology they know and understand, and they are just going to use it as much as they can, but they are not going to succeed. So, thank you very much.

MR. SCOTT: While our second speaker is setting up, I want to thank Bob very much, and we do have a few minutes for questions.

AUDIENCE: Why do you do this parametrically, if you don't know what your distribution is?

MR. JACOBSEN: What other choice have I got?

AUDIENCE: Non-parametrics.

MR. JACOBSEN: I am embarrassed to say, I don't know how to do that, but let's talk. How do I put this? A lot of what we do is because we learned to do it that way. So, new ideas, in fact, distribute themselves in a very non-linear fashion in high-energy physics. Neural networks are a classic example. Neural networks for a long time, the basic reaction of the high-energy community was something like this: keep this away from me, I will get out the garlic.

Then a few people showed that they actually could be understood. Once that had been shown, you don't even mention it any more. So, methods that we don't understand we resist until suddenly they are shown to be worthwhile.

AUDIENCE: Where are they used?

MR. JACOBSEN: Neural networks? They are used everywhere now, but they are predominantly used as categorizations of large groups, at least at the rejection steps before you get to the precision measurement.

For example, the skins that are sorting out samples for people? I guess about a third of them probably use some sort of neural network recognition. It takes many variables to try to make a guess about these things. We do apply them on a large basis. We have simulated ones. We don't have neural network chips. We have neural network

algorithms.

AUDIENCE: What are the number of— [off microphone]?

MR. JACOBSEN: Ten is the—plus or minus two.

AUDIENCE: And the categorization is— [off microphone.]

MR. JACOBSEN: Yes no.

Paul Padley

Some Challenges in Experimental Particle Physics Data Streams

[Abstract of Presentation](#)

[Transcript of Presentation and PowerPoint Slides](#)

BIOSKETCH: Paul Padley is a professor of physics at Rice University. His research interests lie in experimental elementary particle physics. He conducts much of his research at Hadron collider facilities such as the Tevatron at FermiLab and the Large Hadron Collider at CERN, the world's largest particle physics laboratory. Dr. Padley earned his BSc from York University in 1981 and his PhD from the University of Toronto in 1987.

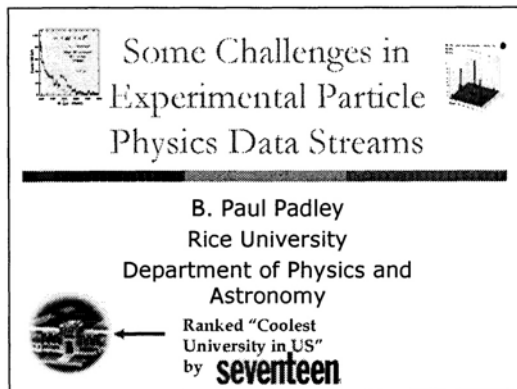
ABSTRACT OF PRESENTATION

Some Challenges in Particle Physics Data Paul Padley, Rice University

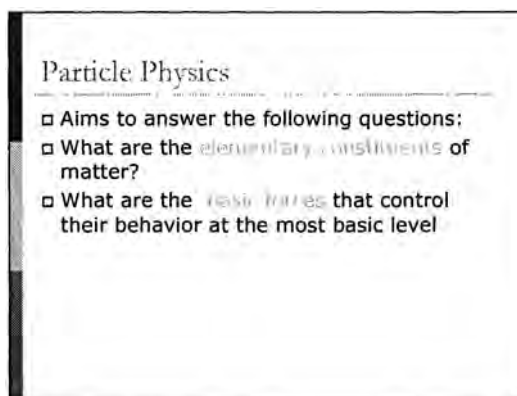
Experimental particle physics attempts to understand the most basic constituents of matter and the forces that act upon them. The research is carried out at national and multinational laboratories, such as Fermilab and CERN, by large international collaborations. The next generation of experiments will produce data at a rate of 40 TB/s, which will be reduced with real-time hardware. The resulting 800 GB/s data stream will then be processed in real time with a 10*6 MIPs computer cluster and reduced to terabytes of data per day which will subsequently be analyzed offline. Each step in the process involves the statistical analysis of the data to search for the signatures of interesting (but possibly unanticipated) physics.

TRANSCRIPT OF PRESENTATION

MR. SCOTT: A decade ago, Steve Sain, who is sitting in the back corner here, was trying to write his thesis. These physicists kept bugging him to do these nonparametric methods, and finally found the top quark. Paul Padley has very graciously agreed to sort of give us a talk that really illustrates statistical challenges explicitly that are out there and, without any further ado, I will turn it over to Paul.



MR. PADLEY: I am going to do my introduction, too. The most important thing I want you to leave with today is, I come from Rice, and it was ranked the coolest university in the United States by *Seventeen* magazine. While my daughters will challenge with me on that issue, who am I to argue with *Seventeen* magazine?



So, I am doing particle physics which you just heard about, and my brief summary of what we tried to do is answer the following questions. What are the elementary constituents of matter, and what are the basic forces that control their behavior at the most basic level?

Weighty Questions!

- Why does a point particle - the top quark - have the mass of 175 protons?
 - protons are extended objects with things inside them
- We don't know why things have mass
- (nor do we have a quantum theory of gravity - which is some sort of clue)

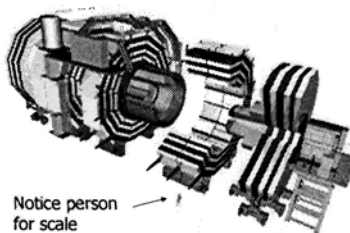
In the process of doing that, we asked some very weighty questions. I mean that literally. So, one of the outstanding problems—and Bob may reference this—is this bit of the theory that is marked, not been checked, or not been confirmed.

The bit of the theory that has not been confirmed is the thing that tells us why we have mass. I assume we have mass, you have mass. We don't know where that comes from. A point particle, the top quark, has the mass of 175 protons. A proton is a big, extended object. It has quarks and things in it. So, why is this point particle, that is infinitely small, have a mass? So, there are some pretty big, weighty questions that haven't been answered by the standard model. So, we don't know why things have mass.

We don't have a quantum theory of gravity, which is some sort of clue, since gravity is the thing that interacts with mass. So, to do that, we do weighty experiments.

Weighty Experiments

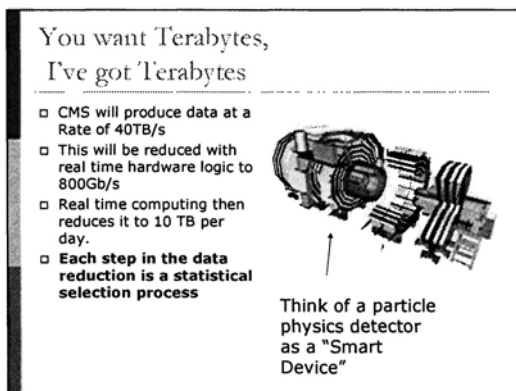
- Exploded view of CMS experiment at CERN
- It will weigh 12,500 t



Here is the next-generation experiment that we are building at CERN. It will weigh 12,500 tons. Notice the person, for scale. You can think of this as a big instrumented device. Those 12,500 tons of detector are instrumented.



To do this, we need big international collaborations. There is an experiment I currently work on at DO. There is a small subset of the people who worked on building that experiment, which is running and taking data now. It is one of the two experiments that discovered the top quark. A free beer if you can find me in there. I am there. Another free beer if you can figure out which client was upside down.



You can think of a big particle physics detector as a smart device. It is a realtime smart device doing real-time analysis. The next-generation experiment will, in principle, produce data at about 40 terabytes of data a second. What we do is use this big smart device to self-select what data is interesting.

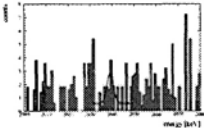
We will knock it down to something like 800 gigabits per second in real-time on the detector itself. We never try to get the 40 terabytes of data off. I, for example, work on the bit out in the edge that tries to self identify whether what happened in there was interesting or not. I, for example, work on this bit of the detector which has artificial intelligence on it to say, "Hey, wow, something interesting happened here; let's think about it some more."

We then shift the data out in real-time computing, and we will reduce the data to something like 10 terabytes of data a day. Each step in that process is a statistical process.

A Question Posed

- Given this heavy need to use statistics one can ask the question
- How do particle physicists use statistics?
- Often **BADLY!**

Supposed observation of neutrinoless double beta decay



You can ask, given this heavy need to use statistics, one can ask the question, how do particle physicists use statistics? Bob made reference to the fact that we all use Gaussian statistics. In fact, if you went and polled a group of particle physicists and asked them, if two hypotheses have a good chi-squared, which hypothesis do you take, the vast majority would say, the one with the lower chi-squared, and not answer that both are valid. Your typical particle physicist, when you say, “Well, actually, both hypotheses are valid, what are you going to do with that?,” they will look at you dumbfounded.



It was announced last year the observation of neutrinoless beta decay and they published it. They got it published.

There is an effort to rectify this

Physicists and statisticians get technical in Durham

Particle physicists and statisticians got together in Durham, UK, last March to discuss statistical techniques of relevance to particle and astroparticle physics analysis.

“Almost 100 physicists and two professional statisticians gathered at Durham’s IPPP in March to discuss statistical techniques in particle physics.”




Okay, there is an effort to rectify this. Here is a headline from what I will call a trade magazine, the *CERN Courier*, a standard popular magazine about a business: “Physicists and Statisticians Get Technical in Durham.” There was a big conference in Durham last March on applying statistical techniques to particle and astrophysics data.

How many statisticians even attended this conference? None. They then go on proudly to say, “this teaming of physicists and statisticians.” There they all are. Look for your colleagues in there. Almost 100 physicists and 2 professional statisticians gathered. That is a direct quote.

Some Credits

- I am here representing a number of people who have done the real work and who should get the credit:
 - Andrew Askew
 - Bruce Knuteson
 - Hannu Miettinen
 - Sherry Towers
 - Plus others
- I personally would like to thank David Scott for all his support. (He is always there to answer my naive questions)

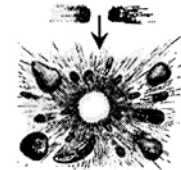


Okay, we have a problem. Before I go on, some of us have actually tried to rectify this situation. I am going to talk about some work that has gone on, and I am representing other people who have done the real work. The results I will show come from my graduate student, Andrew Askew, and a former student at the Rice, Bruce Knuteson, a colleague, Hannu Miettinen, another colleague, Sherry Towers at Fermilab.

Then, I owe a lot of thanks to David Scott, because he is always there and I can come and ask him my naive questions like, “You mean you don’t take the lowest chisquared?” He will patiently answer them for me.

But what do we do?

- Collide particles (protons for example)
- Watch what comes flying out in our detector, event by event
- Statistically analyze the events looking for known or new physics



It's a bit like smashing two strawberries together and getting a bowl of fruit.

First, a little bit of what we do. We talked about how we collide protons, for example, and look at what comes pouring out into our detector event by event, and then we statistically analyze that looking for new physics or known physics.

It is a little bit like taking two strawberries, smashing them together. You get this bowl of energy that then turns into a bowl of fruit, and we look at the fruit coming out.

A Simple Example

- Select all the events with correct topology
- Make the invariant mass of the photon pairs
- Plot and look for bumps

Higgs to 2 photons ($M_H < 140$ GeV)

$H^0 \rightarrow \gamma\gamma$ is the most promising channel of M_H in the range 90–140 GeV. The high performance PbWO₄ crystal electromagnetic calorimeter in CMS has been optimized for this search. The $\gamma\gamma$ mass resolution at $M_H = 100$ GeV is better than 1%, resulting in a S/B of $\sim 1/20$.

$M_{\gamma\gamma} = 100$ GeV

Here is a specific example. This is something where you get a trip to Sweden. This would be a proton antiproton—proton proton collision, say the two photons. What we do, this is what we see in the detector, and we have to statistically analyze that to figure out what went on.

Then we will make an invariant mass distribution of these two photon signals over a large statistical event to look for a bump. That would be the discovery of the Higgs signal.

Looks Simple!

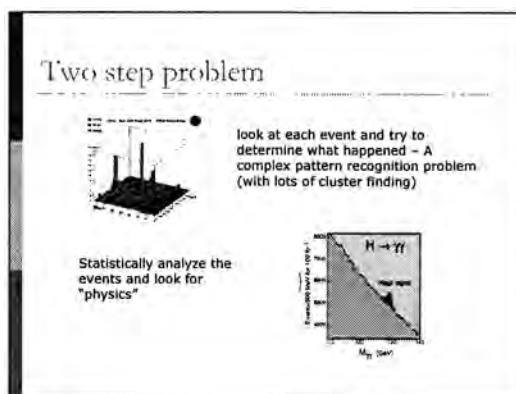
- The phrase “Select all the events with correct topology” represents a very complex step!

Of course, this phrase, “select all the events,” with the correct apologies, that is a very complex step. So, here is an event. What I have done is altered the detector outlook and energy deposited in the detector.

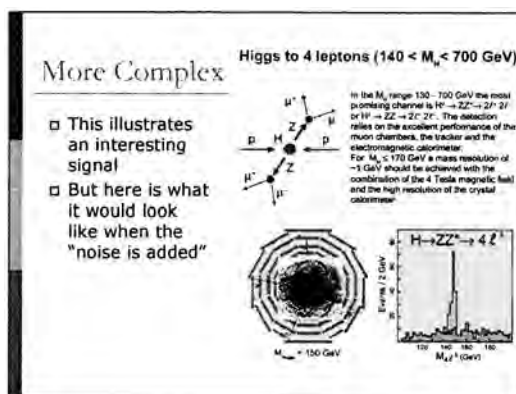
An example of what you see in the detector

To recognize the clusters of energy and decide what they are is a difficult challenge

There are still obvious big things there with missing energy and an electron. Then, as I look at the event I find, well, there is a jet from the fundamental quark, and another one and another one and another one. Those things get pretty hard to fish out in the data. So, we need pretty good tools for doing that.

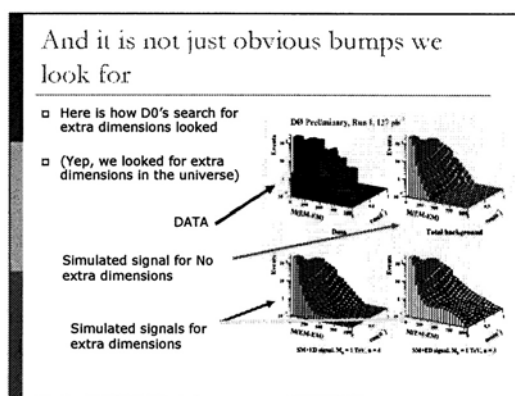


So, we have sort of a two-step problem. We look at each event and try to figure out, event by event, what has happened, and that is a complex pattern recognition problem with lots of cluster finding and other things that we need to do, to track fitting and identifying things. Then we have to take a cohort of events and statistically analyze them and look for physics. It gets pretty complex.



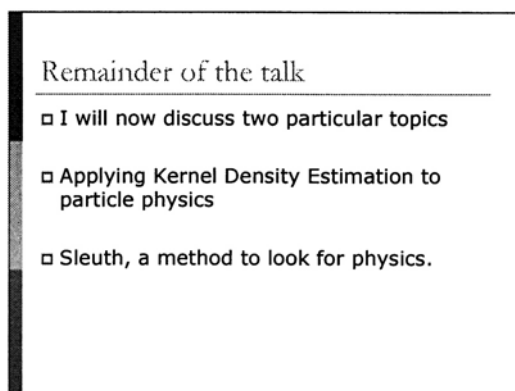
In the next-generation experiment, what you will actually see, here is an event with four muons coming out. If you were looking at it, all you would see is that, and you have to find the tracks in that mess. I have made it worse than it really is, because I have compressed it all into two dimensions, a three-dimensional thing.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



It is not just those bumps we look for. Sometimes, what we are looking for are differences from the standard model that are a little bit more subtle. For example, this is a simulation distribution of two parameters—it doesn't matter what—for our standard model particle physics.

Here is simulated distribution, assuming there were extra dimensions to the universe. The fact that we could use particle physics experiments to look for extra dimensions was mind-boggling to me, not something I had expected. Here is what we saw in the data. Then we have to ask the question, well, are we seeing extra dimensions in that data or not. Clearly, we are not in this case, but sometimes it is a little bit more subtle.



So, I want to talk about two particular attempts to try and go a little bit beyond the sort of analysis that Bob was talking about. He made reference there about using neural networks, and we have been trying to use kernel density estimation. Then, a method for looking for new unexpected things.

PDE

- In the last decade or so, Neural Networks have been more commonly (but not universally) used in particle physics analysis.
- The "Black Box" nature of Neural Networks worries many in the field
- A group at Rice developed PDE (Probability Density Estimation) as an understandable alternative.

Neural networks have become commonly used as a pattern recognition tool, but the black box nature of them worries many in the field. I mean, physicists don't really like being able to visualize what they have seen in that.

PDE

- Originally formulated to look for top quarks
 - L. Holmstrom, S. Sain (statisticians at Rice)
 - H. Miettinen
 - B. Knuteson (when he was an undergraduate)
- An understandable multivariate approach
- It has equations
- You can plot things and understand what it is doing.

So, a group at Rice, named here—is Sain here? There he is. I have never met him, this happened before I got at Rice. They developed a method called PDE that was formulated to look for top quarks. It is a multivariate approach where you can plot things to understand what is going on.

Discriminant Function

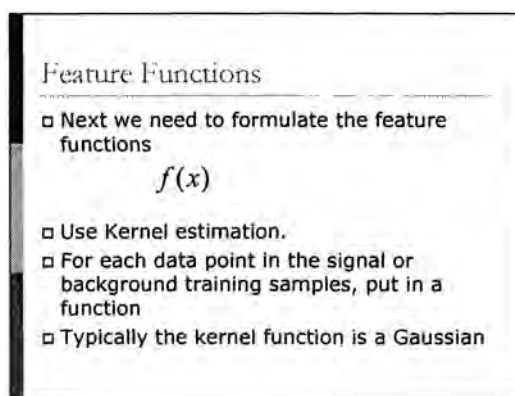
- In any method, want a Discriminant function that can separate signal from background
- A general form could be:

$$D(x) = \frac{f_s(x)}{f_s(x) + f_b(x)}$$

Where x is a vector of the parameters used in the analysis

So you want to form a discriminate function that can discriminate signal from background. So, you can have a general form like this, where x is some vector of parameters that you have measured. So, you have a function describing the signal and

then the background, and you try to make it discriminate.



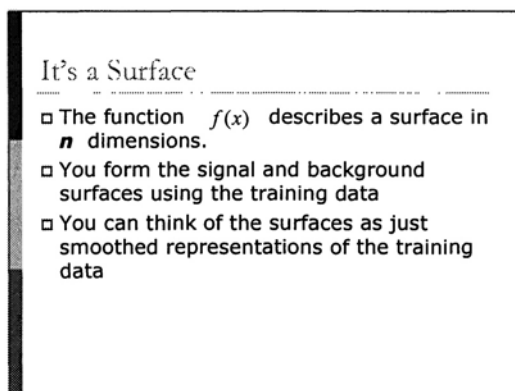
Feature Functions

- Next we need to formulate the feature functions

$$f(x)$$

- Use Kernel estimation.
- For each data point in the signal or background training samples, put in a function
- Typically the kernel function is a Gaussian

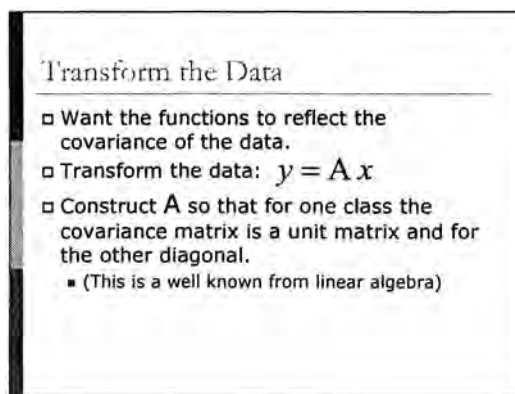
So, we need to formulate these feature functions. What they did was they used kernel estimation. So, for each data point in the signal or background, they put in a function, typically a Gaussian function—actually, we have only ever used Gaussian functions.



It's a Surface

- The function $f(x)$ describes a surface in n dimensions.
- You form the signal and background surfaces using the training data
- You can think of the surfaces as just smoothed representations of the training data

This function describes a surface and end dimensions, and you form the signal and background surfaces using this Monte Carlo simulated data of the signal you are looking for in the background. So, you can just think of these as smooth surfaces representing the data. It is much more straightforward to think of than thinking about what the neural network is doing.



Transform the Data

- Want the functions to reflect the covariance of the data.
- Transform the data: $y = A x$
- Construct A so that for one class the covariance matrix is a unit matrix and for the other diagonal.
 - (This is a well known from linear algebra)

It reflects the covariance of the data. You construct a transformation, so that one class of the data has a covariance matrix that is a unit matrix, and the other is diagonal. That is something that you can do, and then you write it in mathematical form.

Mathematical Form

$$f(x) = \frac{1}{N h_1 \dots h_d} \sum_{i=1}^N \prod_{j=1}^d K\left(\frac{x_j - x_{ij}}{h_j}\right)$$
$$h_j = h^0 \times \sigma_j$$

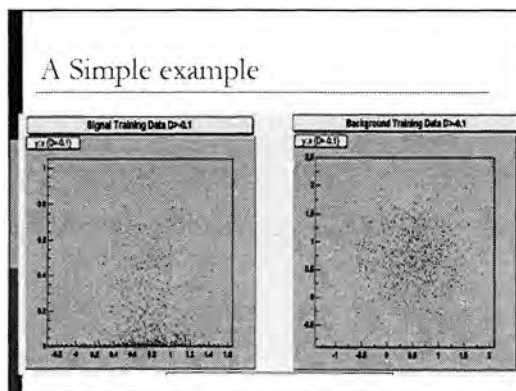
Note that there is one arbitrary parameter in the method, which should be found empirically h^0

There is a free parameter that enters.

A Method for Discriminating Signal from Background

- By following the recipe given, one obtains a function $D(x)$ which can be used to separate signal from background.
- Typically one will require $D(x) > 0.8$ or something similar
- This is a graphical cut! You can visualize what you are doing

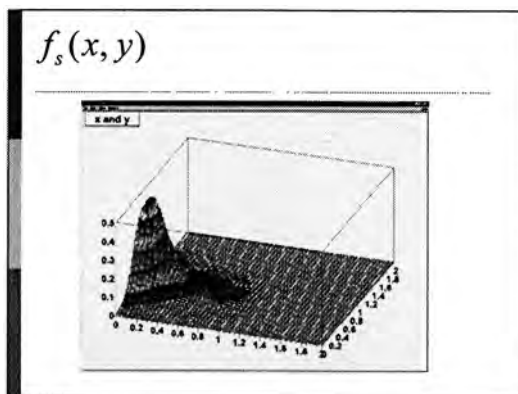
So, by following a recipe where we make these kernel functions, it can make a discriminate function and make a graphical cut, something where you can visualize what you are doing.



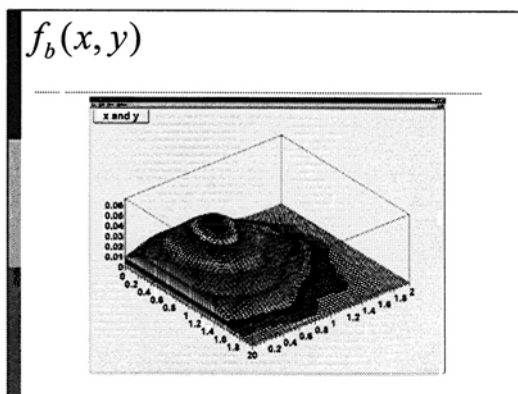
Here is just a Monte Carlo, arbitrary parameter signal, that we wanted to look for

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

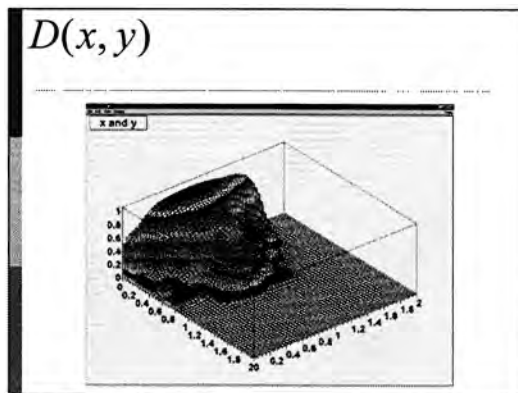
distribution, and here is a background.



Then, you apply this technique and you get a model of the signal,



and a model of the background,



and then here is this discriminate function, and you would make a cut somewhere here, and then be able to pick the signal out.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

Adaptive Kernel

- We have implemented in RootPDE a variant on the method using adaptive Kernels.
- The idea is to make the width of the Gaussians wider in regions with few events
- Modify the width by

$$h_i = h^0 \sigma_i \left[\frac{f(\bar{x})}{f(x)} \right]^\alpha$$

- Where \bar{x} is a reference point (we are currently using the mean)

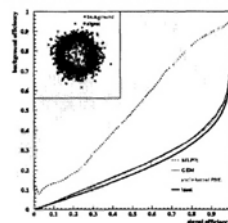
We have modified that original method a little bit. One of the problems with that method is if you have outliers. The outliers sort of get a little Gaussian put around them. So, we have made a way to adapt that and smooth it out. So, we have introduced another parameter.

GEM

- A colleague (Sherry Towers) independently came up with a similar method: GEM
- Difference is, it uses a local covariance matrix

Our colleague at Fermilab, Sherry Towers, independently came up with a similar method that uses a local covariance method.

Density Estimation vs Neural Networks



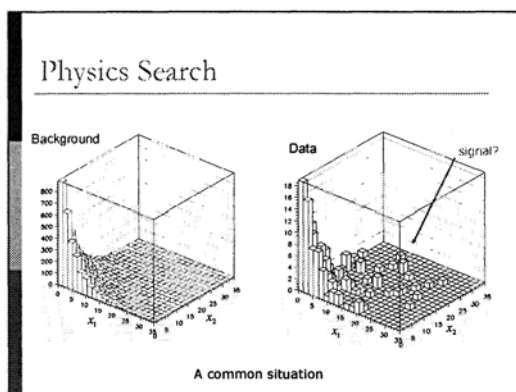
Here is an example where density estimation does much better than a neural network.

In general though, the methods give similar results

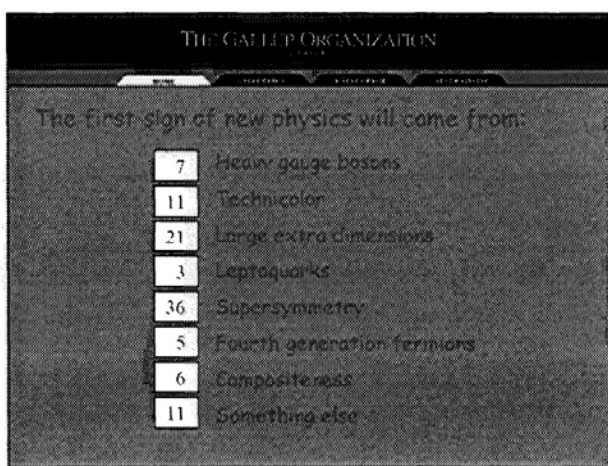
Here is a comparison of these methods for particularly hard signal and background samples, where a neural network just does a terrible job and our methods do a good job.

So, that is one thing. So, one thing that we have tried to do is, in applying these

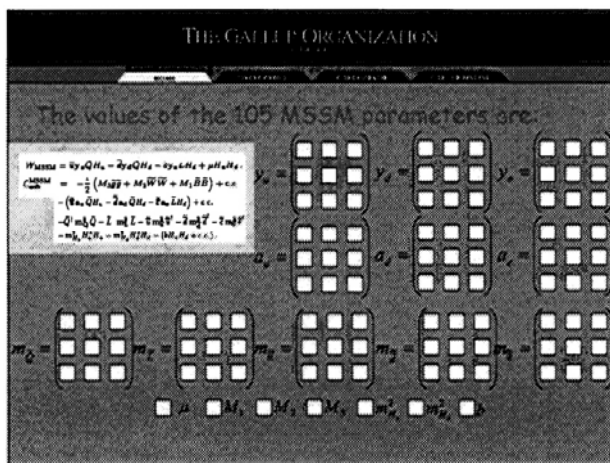
cuts and sort of making cuts and boxes, as was described before, try and use—a lot of people are using neural networks, but there is a group of people who are trying other techniques, like support vector machines, internal density estimations and that.



Another problem we come up with continuously is, you have background, and here, background means the standard model. What you see in the data is this. So, have we found new physics in these points out here or have we not? That is a common problem.



In fact, if you were to take a poll, as a particle physics community, as to what the new physics would be, you would get a whole pile of answers. So, we have this standard model that works beautifully and explains all the known data. I think if you made a poll of the particle physics group, nobody would believe that the standard model is right. There are all these free parameters, you don't really understand mass in this context, and there are a lot of ideas out there for what the new physics would be, new bosons, all these things that are sort of meaningless. A big contingent would say something else. So, this was just a straw poll then.



Even if you pick the most popular model, supersymmetry, of what you are looking for, there are 105 pre-parameters in the theory. So, how do you possibly know — and changing those parameters changes what you will see in the detector.

The problem in a nutshell

- We have a well defined standard model.
- We need to look for interesting physics events in this data. (lots of data)
- We need to statistically analyze these events to determine if we have found something of physics interest
- But we possibly don't know what will be the interesting physics

So, the problem, in a nutshell, that we face is that we have a well-defined standard model. We need to look for interesting physics in the data—there is lots of data. We need to statistically analyze these events to determine if we have found some physics of interest, but we probably don't know what the physics of interest is that we are looking for. So, we are looking for something unknown in this vast mass of data.

Motivation

Most searches follow a well-defined set of steps:

- Select a model to be tested
- Find a measurable prediction of the model differing as much as possible from the prediction of the Standard Model
- Check those predictions against the data

This approach becomes problematic if the number of competing candidate theories is large . . . and it is!

Is it possible to perform some kind of "generic" search?

Now, the method that was described to you before is, typically, what you do is select a model to be detected, you find a measurable prediction and a couple of

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

parameters of that model, and you go check those predictions against the data. That is fine, when you have a finite set of models to test.

In this huge plethora of possible models and every variation of the parameters, and saying supersymmetry is a different model with different consequences for the experiment, this becomes a real problem. So, at the DO experiment, a generic search was tried. This is something that was called Sleuth.

Motivation

The word "model" can connote varying degrees of generality

- A special case of a class of models with definite parameters
mSUGRA with $M_{1/2}=200$, $M_0=220$, $\tan\beta=2$, $\mu<0$
- A special case of a class of models with unspecified parameters
mSUGRA
- A class of models
SUGRA
- A more general class of models
gravity-mediated supersymmetry
- An even more general class of models
supersymmetry
- A set of even more general classes of models
theories of electroweak symmetry breaking

generality

Most new physics searches have generality $\approx 1\frac{1}{2}$ on this scale
We are shooting for a search strategy with a generality of ≈ 6 . . .

So, typically, what is done in a physics analysis is that you have some class of model, minimal supergravity, supersymmetric supergravity, with some particular set of parameters, and you can go and try and look for that case. You can consider looking for some larger set of the parameters. What we really want to do is try to make our searches where we are looking for something new in general.

So, the typical physics analysis done in a particle physics experiment is done at 1.5 on this scale, and we would like to be up here at 6.0, searching through the data, looking for new things.

More Motivation

Another related issue:

How do we quantify the "interestingness" of a few strange events *a posteriori*?

After all, the probability of seeing exactly those events is zero!

How excited should we be?

How can we possibly perform an unbiased analysis after seeing the data?

CDF $e e \gamma \gamma$ Candidate Event

$E_T = 30$ GeV

$E_T = 55$ GeV

$E_T = 60$ GeV

$E_T = 55$ GeV

The other problem that comes up all the time is you get one unique event. Well, Bob showed you a unique event before. Well, how do you find those unique events and decide that they are unique. So, we would like our method to be able to do that in an unbiased way.

Sleuth **Step 1: Exclusive final states**

We consider exclusive final states

We assume the existence of standard object definitions
 These define $e, \mu, \tau, \gamma, j, b, E_T, W,$ and Z

All events that contain the same numbers of each of
 these objects belong to the same final state

So, what we did is basically tried to boil the data down into a finite set of fundamental things that you can measure. We looked at different combinations of the particles, what they are, and then we tried to see what it is about those particles that we are measuring that is interesting.

Sleuth **Step 2: Variables**

What is it we're looking for?
 The physics responsible for EWSB

What do we know about it?
 Its natural scale is a few hundred GeV

What characteristics will such events have?
 Final state objects with large transverse momentum

What variables do we want to look at?

p_T 's

At this point in time, it is highly likely that exciting new physics can be found at large values of p_T .

The method can be extended to other cases, but we have not done so.

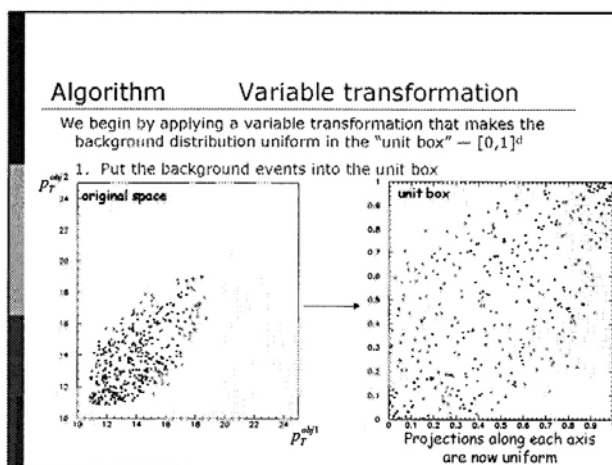
In our case, we have the advantage that the known physics happens at low energy. So, at low transverse energy in the experiments. So, if we look for things that happen at high energy, they are likely to be interesting.

Sleuth **Step 2: Variables**

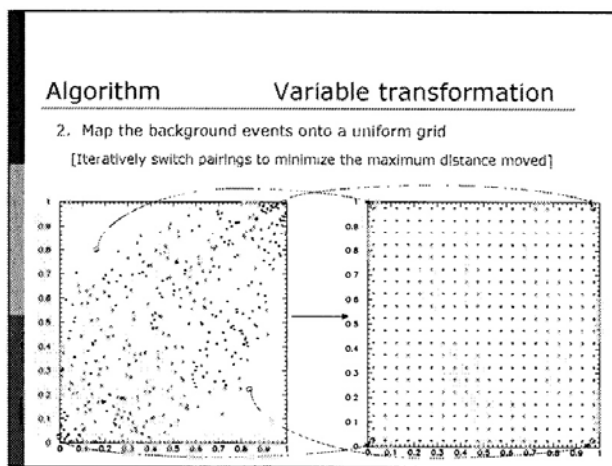
| | |
|---|--|
| <p>If the final state contains</p> <ul style="list-style-type: none"> 1 or more lepton 1 or more $\gamma/W/Z$ 1 or more jet missing E_T | <p>Then consider the variable</p> <ul style="list-style-type: none"> $\sum p_T^l$ $\sum p_T^{\gamma/W/Z}$ $\sum p_T^j$ E_T |
|---|--|

So, we picked a set of parameters which is just basically the momentum in the perpendicular direction of the things that we were looking for.

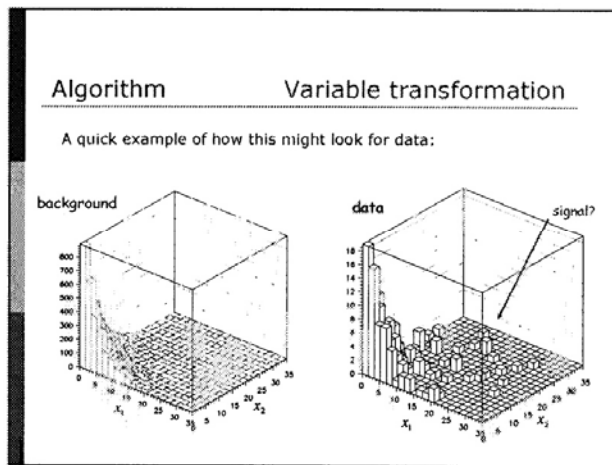
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



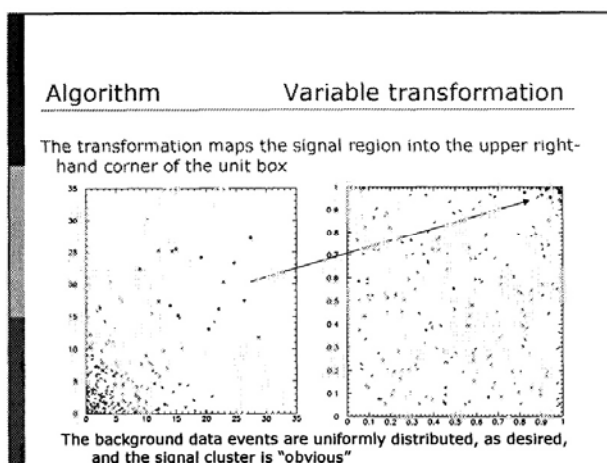
Then we go one more step. We do a variable transform.



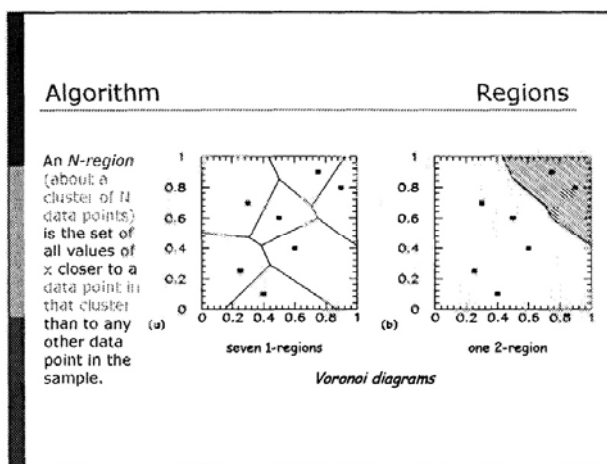
So, say we have our data gathered on a space like this. We basically push that data out into a unit box, and then map the data onto a uniform grid. We are taking a simulated model background data, and we map it out onto a uniform grid in a unit box, and dimensions.



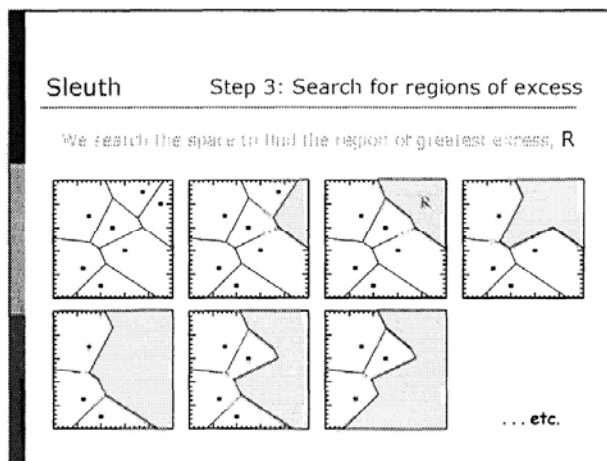
We can go back to that example that I gave before and look at what happens to the signal or not.



The way we set up our variables, it would tend to push those events up into a cluster in the corner.



Then we have to ask the question, is that an interesting cluster that we are looking at? So, what we do is, we create what are called Voronoi regions, which is just a cluster of data points as the set that has—the region is the set of all values of x closer to a data point in that cluster than to any other data point in the sample.



So, you break the data up into regions and then you look through those regions and try to look for an x set in that space.

Sleuth Step 3: Search for regions of excess

- define the "interestingness" of an arbitrary region
 - the probability that the background within that region fluctuates up to or beyond the observed number of events
- search the data to find the most interesting region, R
- determine **P**, the fraction of *hypothetical similar experiments* (hse's) in which you would see something more interesting than R

Basically, you can assign a probability as, what is the probability that you will see something that is more interesting than what you saw.

Sleuth Sensitivity

If the data contain no new physics, Sleuth will find P to be random in (0,1)

If we find P small, we have something interesting

If the data contain new physics, Sleuth will *hopefully* find P to be small

The method has been tested on known physics – for example the search for the top quark. It finds the top quark but with less sensitivity than a dedicated search. That is the price one pays for a generalized search

So, the data contain no new physics. These are just fine P's that are random between zero and one. If the data does contain new physics, then you hopefully find P to be small. This method was tested on known physics. For example, the search for the top quark, which was a big discovery, was reproduced using this. It did find—this message did find the top quark, but the price you pay for this general search is that you have less sensitivity to the new thing that you are looking for.

Results DØ data

Sleuth was used to scan the DØ data for evidence of "New Physics".

In a standard analysis, each mode on the right would be a single model dependant analysis.

Sleuth allowed a "quasi" model independent search of many channels at once.

No evidence of new physics is observed

| Data set | P |
|---------------|---------------|
| topX | |
| topX | 0.14 (+1.00e) |
| topX | 0.45 (+0.10e) |
| topX | 0.33 (+0.00e) |
| topX | 0.71 (-0.00e) |
| W+jetlike | |
| W 2j | 0.29 (+0.00e) |
| W 3j | 0.19 (+0.00e) |
| W 4j | 0.33 (-0.00e) |
| W 5j | 0.81 (-0.00e) |
| W 6j | 0.22 (+0.71e) |
| W 7j | 0.76 (-0.71e) |
| W 8j | 0.17 (+0.00e) |
| W 9j | 0.19 (+1.10e) |
| Z+jetlike | |
| Z 2j | 0.32 (-0.00e) |
| Z 3j | 0.73 (-0.00e) |
| Z 4j | 0.82 (-0.00e) |
| Z 5j | 0.72 (-0.00e) |
| Z 6j | 0.83 (-0.00e) |
| Z 7j | 0.04 (+1.10e) |
| Z 8j | 0.68 (-0.47e) |
| Z 9j | 0.36 (+0.00e) |
| Z 10j | 0.06 (+1.50e) |
| Z 11j | 0.08 (+1.41e) |
| (47)(17)(10)X | |
| ee | 0.89 (-1.50e) |
| 2γ | 0.84 (-0.00e) |
| Zγ | 0.62 (-0.00e) |
| γγ | 0.88 (-1.17e) |
| γγγγ | 0.23 (+0.00e) |
| γγZ | 0.66 (-0.41e) |
| γγW | 0.21 (+0.00e) |
| γγZγ | 0.80 (+0.00e) |
| Wγγ | 0.18 (+0.00e) |
| Zγγ | 0.11 (+0.00e) |
| P | 0.89 (-1.50e) |

What was amazing with this is, here is a list of all of the channels and the limits we could set, looking for new physics in all these channels. In the traditional sort of

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

analysis that Bob was describing, and normally done, is a graduate student would come and pick a channel and take a model, and test that channel against some model. With this method here, we were able to search down through a whole list of channels in one fell swoop, going through our masses of data.

Now, one of the things that made this possible is, at the time of this analysis, it was mature data. So, we had very good simulations of the detector in that. So, we really could model the standard model in it very well. It was very mature, well-understood data. So, that made it easier to go searching through. We think in here, there is this hint where we have this general problem of looking for unknown things.

Conclusions

- Particle Physics presents a number challenges of interest – I have only scratched the surface
- We have large data streams that we must search in real time and offline for signals of (possibly unanticipated) physics.
- We do have a well defined “Standard Model” against which to compare – more on how we generate that comparison data set in the next talk
- We have begun to apply techniques from statistics but this is in its infant stages.
- I have shown a couple of initial steps, Kernel Density Estimation and Sleuth.

So, just to conclude, I think particle physics presents a number of challenges of interest. You have just seen a little taste, a little scratch of the many places, every step along the way, we face statistical challenges. We certainly have large streams of data that we must search in real-time, and offline, for signals of interest, that are possibly—in fact, the most interesting ones would be unanticipated. We have the advantage of a well-defined standard model to test against, and actually, techniques to generate the data that we use for that will be talked about in the next talk.

There are people in the community who have actually talked to a statistician now. That is a step in the right direction. Of course, we always have this hurdle, we know we are smarter than everybody else, so we try to reinvent things for ourselves from scratch. There is a small group of people who have actually spoken to at least two statisticians. So, we know it happens. So, we are in the very early infant stages.

I have shown some of what I call baby steps that have gone on at our experiment at DO which are unique. I mean, the number of people even within the experiment who would understand the phrase kernel density estimation would be very small, or even know what a kernel is. So, there is a lot of progress that needs to be made.

The statistics conference that occurred is going to be a continuing series, it looks like. There is another one scheduled for next year. So, there is at least a recognition in the community that we should be talking to the statistics community, and that dialogue should start. Hopefully, we will get smarter about how we do analysis. I will finish there.

AUDIENCE: [Question off microphone.]

MR. PADLEY: Yes, I think—that has to be done individually for each combination of particles. In fact, really, you don't need to do it. I mean, really, if you are smart, you could bypass that whole step. I think next time around methods will be used to bypass that. It is the idea of trying to find—the problem you always get is you have

some distribution that is fine with exponential pay off.

You have two events way out here, far away from the standard model. So, is that interesting? I mean, Carlo Rubbia made a career out of discovering things out there that, as they took more data, the distribution— [off microphone.]

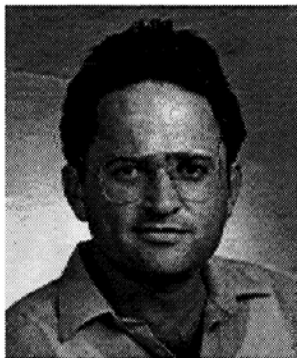
MR. SCOTT: I have a question, Paul. You have 6, 7, 10 dimensions of data. Are these real measurements or derived measurements, typically?

MR. PADLEY: That is almost level three in this case, which is one of the problems. So, you know, part of what made that analysis possible at the end is, what we started off with is about a million measurements per event. You then have to distill that down into, say, vectors, and that will represent—there will be hundreds of those in an event. We have tried to knock it down to five or six or ten parameters that characterize the events. That is like level three. That was really only possible because of the maturity of the data and, by the time that analysis was done, there was a lot of confidence in the other steps that went into informing that data set.

Miron Livny

Data Grids (or, A Distributed Computing View of High Energy Physics)

Transcript of Presentation and PowerPoint Slides



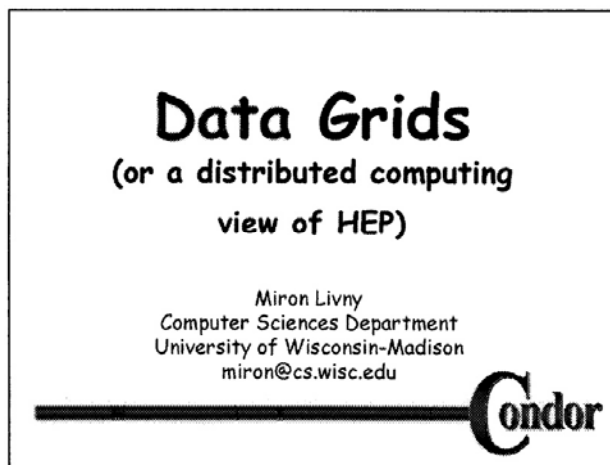
BIOSKETCH: Miron Livny is a professor of computer science at the University of Wisconsin at Madison. His interests are in high throughput computing, visual data exploration, experiment management environments, and performance evaluation. He received his PhD in computer science from the Weizmann Institute of Science, in Rehovot, Israel, in 1984.

High-throughput computing is a challenging research area in which a wide range of techniques is employed to harness the power of very large collections of computing resources over long time intervals. His group is engaged in research efforts to develop management and scheduling techniques that empower high throughput computing on local and wide area clusters of distributively owned resources. The results of these efforts are translated into production code and are incorporated into Condor, a widely used, high-throughput computing system. The worldwide user community of Condor plays an important and active role in Dr. Livny's research, and researchers from a wide spectrum of scientific disciplines collaborate with his group in the development and evaluation of Condor.

In the area of visual exploration of information, Dr. Livny's group works on developing a framework and tools for intuitive graphical interaction with collections of multimedia data. His framework is based on a declarative approach to the creation of active visual presentations of tabular data. He implements his tools in Java and works closely with domain scientists on testing and evaluating them with real data. Some of these are stored in scientific databases that are connected to real-time and off-line data sources.

TRANSCRIPT OF PRESENTATION

MR. SCOTT: The next speaker is very appropriate. If you are going to have massive data sets, you need massive computing and “grid” is the buzzword. We are very fortunate to have one of the leaders in the field to talk to us about the relationship between grid computing and high-energy physics.



MR. LIVNY: Thank you. Since I really didn't know what was expected from me in this presentation, I will try to communicate sort of three—address three areas. One is sort of our view from the computer science perspective of what high-energy physics is. The other one is to touch upon what I believe is a huge challenge, is how to do real interdisciplinary work. So, what does it mean to change a title from a computer science professional to a computer scientist. That was not easy, and we are still working on it. The third one is sort of to give you some update on technology and what can be done and how it works because, at the end of the day, a lot of what was described earlier depends on computing resources and stuff like that. So, we will see how far we can go with all of that.



I will give you a little bit of background because I think it is important to understand (a) where we are coming from, and (b) what we have experienced so far.

The Condor Project (Established '85)

Distributed Computing research performed by a team of ~35 faculty, full time staff and students who

- face software/middleware engineering challenges in a UNIX/Linux/NT environment,
- are involved in national and international collaborations,
- actively interact with users,
- maintain and support a distributed production environment (more than 2000 CPUs at UW),
- and educate and train students.

Funding - DoD, DoE, NASA, NIH, NSF, AT&T, INTEL, Micron, Microsoft and the UW Graduate School

Condor

www.cs.wisc.edu/condor

So, at Wisconsin we have been running the Condor project now for over 15 years, and we probably are the proud holders of the title of the oldest computer science project that is still doing the same thing.

This is the good news and the bad news. The good news is that there is a lot of work to be done there. The bad news is that it is really hard to do it right. When I say to do it right, I think it is important to understand that, if you want to do it, then you have to do it for real, which means that you have to develop new software, and you have to face all the challenges of software engineering, middleware engineering, whatever you want to call it, because it has to run on all the platforms and it has to do it right.

You have to become part of these collaborations, and it is not that easy to be this dot in the picture that you saw earlier, and to survive there. We definitely, in computer science, think that a collaboration of three scientists is a huge effort. Suddenly realizing that we have to work in these larger collaborations—and I will talk about politics later — it is an important part of it, and has a huge implication. So, we have to learn how to do it, and it is not simple.

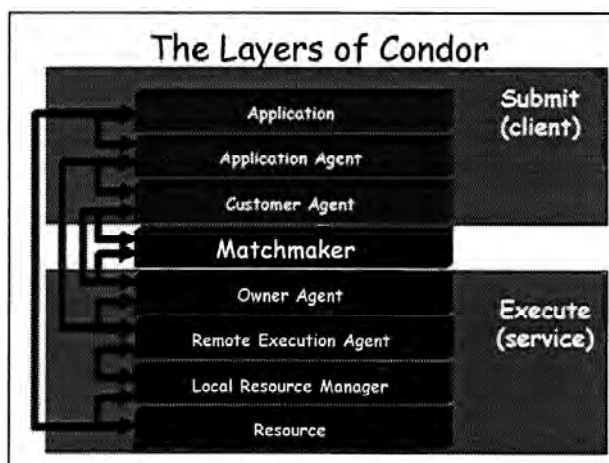
The other part of it is that we have to work with real users. So, we cannot come and say, “Yes, we would like to help you but can you change your distribution a little bit? I mean, you know—if the distribution would have been a little bit different, it would have been much easier for us.” The same thing is true for us as computer scientists.

The other part of it is really this business of practicing what you preach. If you develop a method or you develop a tool, if you are not actually using it and figuring out how it works and when it works and when it doesn't work, then I think there is very little hope that it will be accepted by an end user.

To remind you, the high-energy physics community, as an example, was very self-contained until recently. I think what you have been hearing here regarding statistics, and what we have experienced on the computer science side, is that they realized that they need some help or can use help.

This has not been an easy process, again, and we have to develop things that actually work. If we come and provide something and say, this is the solution, and it falls on its face, the next time doing it becomes very, very difficult.

Now, the good news is that today what we are doing—distributive computing grids—all this is very fashionable and, therefore, we are getting a lot of support. It is good news and bad news because, when something is very fashionable, everyone is stepping to the plate, and people who have been doing a lot of very different things suddenly are experts in distributive computing, and that can be very dangerous.



So, sort of one message I want to leave with you, something we have learned over the last 15 years, is that when we build a distributive computing environment, we have to sort of separate it into three main components, one that represents the clients, the consumer. This is not the trivial part of the equation. Then, there is the other part that represents the resource. Again, this is a very intelligent component, especially when we get into all the political issues of distributed ownership.

What we have realized has worked extremely well is that we interface these two with a framework that we call match-making, that allows providers of resources and requesters of resources to sort of come together in the way that we, as humans, come together.

I think one of the big messages that we have as a result of all of our work is that we should look at these computing environments more as communities rather than a computing system, where nanoseconds and picoseconds are what matter. The actual activity of why are you part of this community and what do you contribute and what do you get out of it becomes much more important than the latency of the network.

The Grid: Blueprint for a New Computing Infrastructure
Edited by Ian Foster and Carl Kesselman
July 1998, 701 pages.

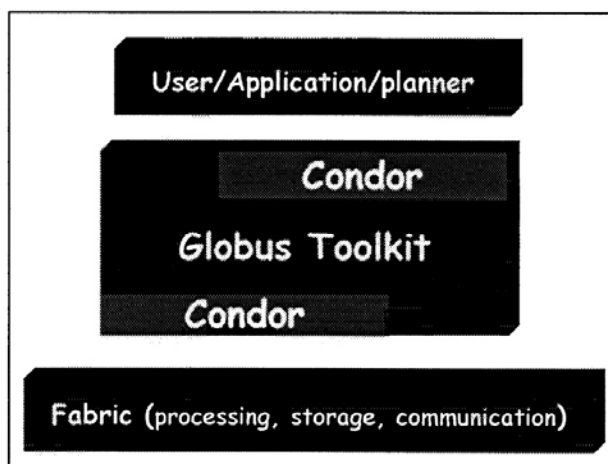
"This is a source book for the history of the future."
Vint Cerf, Senior Vice President, Internet Architecture and Engineering, MCI Communications

The grid promises to fundamentally change the way we think about and use computing. This infrastructure will connect multiple regional and national computational grids, creating a universal source of **pervasive and dependable** computing power that supports dramatically new classes of applications. The Grid provides a clear vision of what computational grids are, why we need them, who will use them, and how they will be programmed.

In the mid-1990s, the grid concept came to the front of our activities, and this is the Bible, and we are working on the New Testament now, the second version of the grid book.

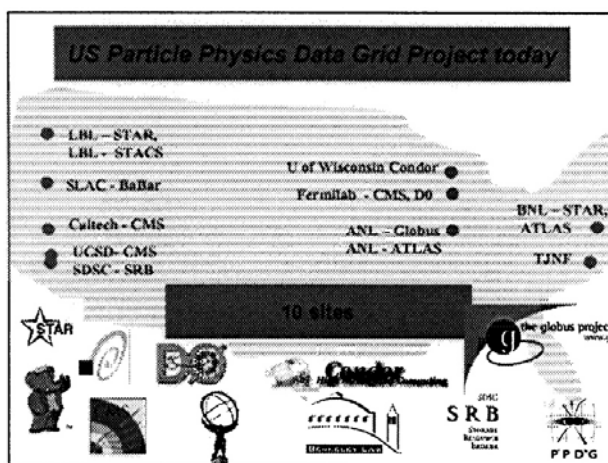
The main message there, we are trying to create this pervasive and dependable computing facility, and as I said, there is a lot of activity on this. I can give you another talk to two hours to show it is related to distributive computing, and there are a lot of concepts that are sort of resurfacing, and there is a lot of stuff that goes back to what we

have been doing 20 and 30 years ago.



So, if you look at this notion of the grid, there is the user who wants to do some analysis, who wants to create some kind of a histogram or scatterplot or what have you, or ask a question, or generate a million events, and I will show you a little bit about that.

That leaves the fabric that has to do the work, and in between there is this thing that we call the grid of the middleware, that has to bring them together.

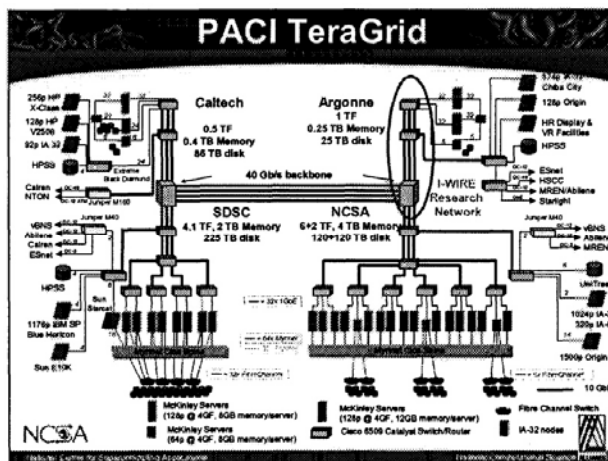


Actually, following the division of labor that I showed you earlier, we have been doing, on our side, and have been sort of contributing and moving it into the high-energy physics community, is to comment and say there is what is called the Globus tool kit, which is the inter-domain capability that allows you to cross domains and create them into a single domain, and then we have taken the technology that we have developed and attached it on one side to the fabric and on the other side to the computing environment, and this is what we have to add to it in order to make the whole thing work.

Now, what I will do today is focus more on the application, user side, because I think that this is much more applicable to what we are talking about here. One of the questions that we have to ask at the end of it is, how do we write applications, how do we develop interfaces that can actually take advantage of these capabilities. That, I think, is where some of the algorithmic and software development issues are coming into play.

Now, we are even in a much worse situation than what you have seen earlier, because you can see only the handles on that side, but when we are getting into a grid effort—in this case, the particle physics data grid which is a DOE activity, we have to deal with several of these experiments, and we have to deal with several software

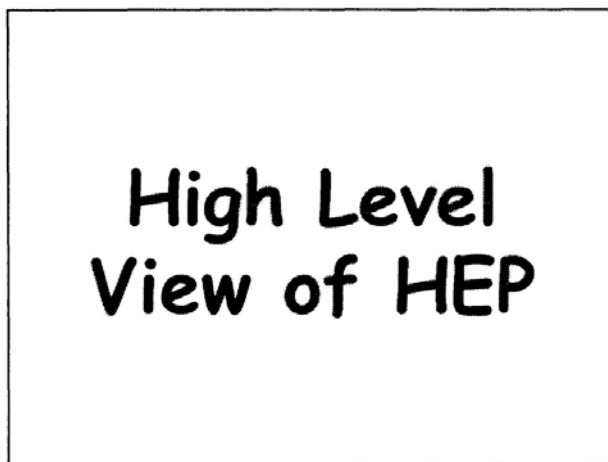
contributors, in order to make the whole thing work and in order to understand what is in common and what is different.



This is, as I said, one example and it is chopped on the left. By the way, if you are looking for the use of Compus, logo generation is a huge industry these days. There is a continuous stream of new logos, and I am sure there is a lot of money in that also.

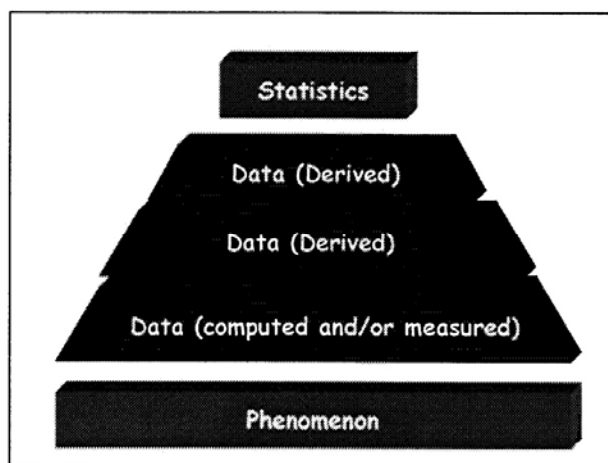
Now, the hardware is also pretty challenging. This is one grid, in terms of infrastructure. This is the thoroughgrid. I think it is a \$45 million or \$50 million effort of NSF.

Part of it is, okay, if you want to do high-energy physics calculation, here are a few flops that you can use and a few pedabytes to store the data. The question is, how do you get to that, and how do you get your results back?



Let me try to generalize or abstract and say, this is the way we look at it, and that is the way we are trying to develop tools to help high-energy physics. So, the two examples that I have, one is from high-energy physics and the other one is from the Sloan Digital Sky Survey, which is more in the astronomy side.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.




So, this is a simplified view of what is going on. These are the phenomena down there that can either be described by the detector or described by a set of models. That is sort of the sort of information about the real world. Then it goes through this pyramid of data refinement and, eventually, at the end, what they wanted is statistics.

We have to deal with the flow of data up, and we have to deal with the issue of a query going down. When a scientist comes in with a question about statistics, which is basically, in most cases, “Give me a histogram of these values on these events, under these conditions,” then it is going down.

One of the challenges here is how far you have to go down depends on the question. So, the projection question is challenging because you might get it from the top or you may have to go down. As you saw earlier, the amount of data involved here are huge and, in many cases, the computation needs is huge as well.

Distributed Computing Environment

- > **Physical Distribution - Processing and storage resources are interconnected by local and wide area networks**
- > **Distributed Ownership - Resources are autonomous and locally managed**


www.cs.wisc.edu/condor

What makes this more interesting is that all this is happening in a distributed environment. The distribution is in two dimensions. It is what we are used to, which is the physical distribution—namely, we have networks, we have machines that are interconnected by these networks, and these networks can be local area and these networks can be wide area, or can be wireless, or whatever they are.

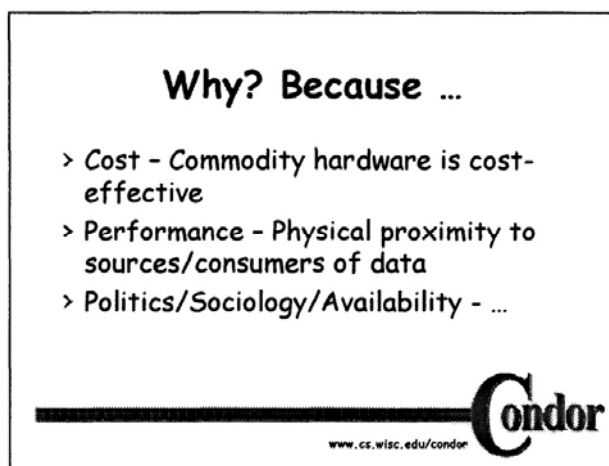
So, the machines are in different places. They are heterogeneous and all these kinds of wonderful things which, by the way, brings up one of the biggest challenges of this environment, is if you don't have a single reboot button. It is so wonderful when one of these wonderful machines misbehaves. You reboot it and you bring it back to a stable state and you can keep going.

When you have distributed distribution, you can really never have the system in a

stable state. The other principle that we learned from the physicists, and that is the importance of time. What you know is what happened in the past and, by the time you react on it, it is even later. So, never assume that what you are doing has anything to do with reality.

The second part, which is even more challenging, is the distributed ownership. Since computing is so cheap, and so powerful, we get more and more, a smaller and smaller organization owning more and more computing power. These computing resources are autonomous, are managed by their own local software, but are also managed by their own local administrators that reflect local policy.

So, if you want to operate in this environment, you have to be prepared to deal with all that, which really means that you ought to take an opportunistic view of the world. You ought to go and use whatever is available for as long as it is, and then be ready to back off when this is gone, and be able to make sure that you behave according to the rules.



So, what is driving the fact that it is distributed? Cost, obviously. Commodity computing is here to stay with us, I think, for a long time, at least, and if we have to compute on desktops or we have to compute on Play Station or whatever, the computational needs, as you saw them earlier, are so huge.

I heard earlier that they would like to have threefold Monte Carlo. I heard earlier that you would like to get 10-fold Monte Carlo. Some people tell me that you can barely get today one-fold Monte Carlo, in many cases.

So, if we can make it available to the HEP community and the others—by the way, the biologists are coming in with even more than that, I just talked earlier this week with one biologist who wanted to do pairwise comparison of 120,000 proteins that exist today, and another 300,000 that are coming down the pike later this year. So, we have to go after commodity, whatever it is, and it is distributive in nature.

The other part of it is performance, because we need to be close to where we collect the data. We want to be at CERN or we want to be at SLAC where the detector is, we want to be close to where the scientists are, in order to do the analysis.

It is not only the performance, but it also brings in the availability issue. Why? Because if I have the data here and I have the computing here, I am in charge. I touch the mouse, all this is mine, even if it is not as big as it could have been if everyone would have gone to the single place.

That is also the politics. So, you have these international efforts and, for example,

the United Kingdom now has invested a huge amount of money in e science, and they want the resources to be in the United Kingdom. All of a sudden, BaBar has more of a presence in the United Kingdom, not because that is what makes sense, but that is where suddenly the money is. So, that is the politics.


The other one is the sociology. I want to have control, I want to show that I am a contributor, I am doing it. If we don't understand it from the outset, and we really build everything around it—this also goes back to my previous comments—we have to understand how to operate in a large collaboration.

It is getting even more difficult when it is interdisciplinary, when we come in with different cultures and a different way of thinking and we have, eventually, to solve the same problems.

While we are doing computer science and they are doing physics, at the end of the day, they have to find one of these particles, which I have been trying to give them particles along and here it is, and let's forget about all these other worlds, but they don't want my particles. They are looking for another one.

Key planning issues

- > Predict processing and storage requirements of each data generation/derivation/aggregation step.
- > Decide when to derive and when to retrieve the data needed to answer a query. (Virtual Data, Data Provenance, Data Equivalence)
- > Control movement of data and/or function.


www.cs.wisc.edu/condor

Now, what can we bring to the table? There are a lot of principles we have learned over the years in computer science. I think one of them, which is actually coming from the database world, is the value of planning. One of the nice things about databases is that you come in with a logical request. You don't know anything about the physical implementation of the system or the data. Then, you let somebody do the planning for you. There are a lot of challenges in doing the planning, and there are a lot of challenges in convincing the user that what you did is right.

For example, a typical physicist will not trust any piece of software. Coming to them to say, "Trust me, give me a high-level request and here is the histogram," they will say, "No, no, where was the byte, when was it moved, by whom was it generated, by which operating system, which library?" —all these kinds of things, because they have learned over the years that they are very sensitive to it.

Now, whether it has to be this way or not is an interesting question which I think has also statistical implications because, on the one hand, everything is random. On the other hand, you want to know exactly what the voltage distribution of the machine was when you ran it.

So, that is getting into—the second item here is data provenance. There is a lot of work today in understanding, if this is the data, where did it come from? How much do we have to record in order to convince the young scientist that this data is valid, or that

these two data sets are the same? In the database world, whether it has materialized or not is left to the database, and that is connected to the other concept that I have here, which is virtual data.

We have the whole IT output, which is called Griffin, which is dealing with virtual data, which is again coming to the end user and saying, "Tell me what you want and I will do it for you. Write it in sequel, and I will decide what join method I should use and where I should use it, and whether I materialized it earlier or not is up to me." There is a huge trust issue here, when you allow somebody else to do the planning for you.

Now, the main issue in the planning is to figure out what is the research requirement. As I point out later, a huge question is how much is the space requirement of an operation, because this is sort of a bottleneck and a source of deadlock, if you cannot write your data or bring your data in when you are trying to really run a large operation.


There is this question of when to derive and when to retrieve. If you want a million events that were simulated, what is cheaper, to go and retrieve them from CERN, or to rerun it on your local farm. Now, if you can guarantee that the two are statistically equivalent, then maybe I should reproduce them on the local farm, rather than wait for the data to be shipped from CERN.

We have not solved, I think, even in databases the question of when to move the data and when to move the function. Given the amount of data involved and the amount of resources, we are facing this problem on a larger scale. Where do we do it, how long do we wait for the data to come, and where do we move the data? Can we push selection down to where the data is?

For the issues that are coming from the database world, from computer science, we have to apply them not in the standard way, which is the way we have been doing it all along.

Data/Work Flow

- > Phenomenon -> data: real-time constraints of instruments, processing capacity, storage capabilities and communication bandwidth for Monte Carlo applications
- > Data -> data: multi stage feature and meta-data extraction, indexing and compression.
- > Data -> statistics: select, project and aggregate.


www.cs.wisc.edu/condor

The other part of it is that we really have a huge data workflow problem here that we have to manage and control, because if we screw up, it can be very bad.

If you really want to live in this brave new world where we have grids and we have computing and we can do things everywhere, then we have this continuous movement of data, of computing. We are talking about tens or hundreds of thousands of things that we want to do in this environment. If we don't keep track of what is happening, or somebody really misbehaves or loses control, we can grind the whole

system to a halt.

So, when we move the data from the phenomenon to the data, when we measure it, so we have a real time constraint if the data is coming from an instrument. Whether it is a telescope or a detector, we have to make sure that this data goes in and we cannot lose anything, because this is real data.

We also have to deal with the production of data coming from the Monte Carlo production, which is, again, a stream of data coming in.

I think many of us have focused on the problem of how to optimize read. The problem of how to maintain a pipeline where there are rights and data has to go in is still an open question.

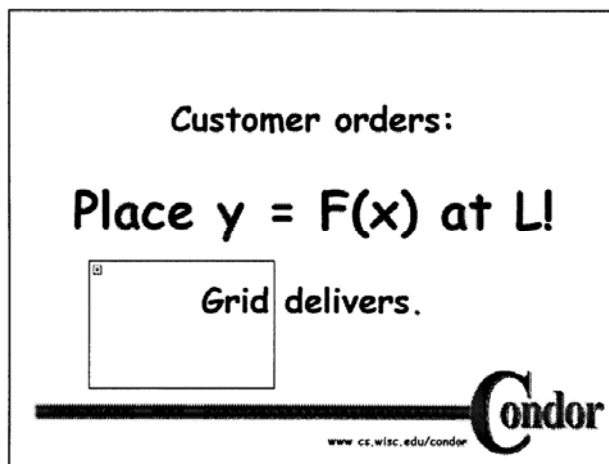
On the Web, we are all focusing, again, on how can I get the document quickly. We are not dealing with how do I inject a lot of data into a common space.

Now, when we are doing the data to data, then we have this multistage, where what we are doing is, we are doing feature extraction, we are doing indexing, we are extracting metadata, we are doing compression. It is basically the same, depending on how you want to look at the output of it.

We have to deal with all these stages. We have to keep track of them. We have to know how the data was produced, the data provenance, and we have also to record how it is done, so that if we want to redo it on the fly, we can do it automatically. In the end, what we have to do is, again, we have to select, we have to project and we have to aggregate.

Now, the selection may involve a lot of distributed patches. So, I have to figure out where the data is. The index can tell me where it is, but it is distributed all over. Maybe some of it has to be reproduced, but eventually, I get the data. At what level I project, again, it depends. Sometimes I want an attribute which is in the metadata, and sometimes I have to go deeper, even into the raw data, in order to get the attribute that I want.

So, the typical approach that we have, that we have sort of the whole thing and we do the selection and then the selection. Again, we need something that is more of the semi-joined structure, that we look at the attributes and what is going on, and then we go to the real couples, and the real couples may be very deep in the hierarchy and may require quite a lot of computing to get the data out.



So, let me try to give you sort of a simple example of what is involved in doing what I would view the most basic operation on the grid, and try to make it abstract.

Let's assume I have an x . I want to apply on it an F . I get the result, which is a y , and I want to store it somewhere, in location L , which can be my screen or can be a CERN archive. I don't care, but eventually, I have to park it somewhere. What I think we have to keep in mind here, which traditional computing, definitely high performance computing, has ignored is that moving this data in and out is an integral part of the problem, not just computing.

So, getting a very fast computation on a high-performance machine, but then waiting several weeks to get the data on and off the machine is not the solution here.

Data Placement (DaP)
is an integral part of the
end-to-end
performance problem

- Space Management
- Data Transfer Capabilities

Condor
www.cs.wisc.edu/condor

So, we have to bring in the data placement activity as part of the end to end solution. Here are sort of the six basic steps that we have to carry out in order to do this y equal F effects or $2F$.

A simple plan for $y=F(x) \rightarrow L$

1. Allocate $\text{size}(x)+\text{size}(y)$ at $SE(i)$
2. Move x from $SE(j)$ to $SE(i)$
3. Place F on $CE(k)$
4. Compute $F(x)$ at $CE(k)$
5. Move y to L
6. Release allocated space

Storage Element (SE); Compute Element (CE)

Condor
www.cs.wisc.edu/condor

So, first of all, we have to find some parking space for x and y . As I pointed out earlier, how do we know how big x is? That is relatively easy. How big is y is getting even trickier, because it can depend on a lot of internal knowledge. Then we have to move it from some kind of a storage element to where we want to actually move it to move x . Then we may have to place the computation itself, because the computation itself may not be a trivial piece of software that resides anywhere in this distributed environment. Then, we have the computation to be done. Then, we have to move the results to wherever the customer orders us and, in the end, we have to clean up the space.

Just doing this right is tough. I can assure you that you don't do it right, even today, on a single machine. How many of you, when you open a file or you write to a

file, check the return codes of the write, whether it succeeded or not?

I am sure that most of your applications will die if the disk is full, and it will take root intervention, in many cases, just to recover it. We cannot afford it in a distributed environment because it has to work in an autopilot with a lot of applications.

**What we have here is
a simple six-nodes
Directed Acyclic Graph
(DAG)**

**Execution of DAG must be
Controlled by the client**

Condor
www.cs.wisc.edu/condor

So, what we really have here, if you think about it, it is really a DAG, a simple DAG in this case, although a shishkabob. Do this, do this, do this, do this.

Keep in mind that we have to free the space, even if things have failed in between, which creates some interesting challenges here.

This has to be controlled by the client, because you are responsible for it. Somebody has to be in charge of doing all these steps and, again, you can look at it, if you really want to, as a transaction that has to go end to end.

Challenges


- > "Heavy Lifting"
 - Scheduling and management of "bulk" data transfers
 - Flow control, routing and buffering of very large "packets"
- > Space (lot) Management
 - Who gets "parking space" where, when and for how long
 - Co-allocation of compute storage and data transfer resources
- > Just in time delivery of input data
- > Timely removal of output data
- > "Local" storage capabilities
 - Easy to manage
 - Cost effective

Condor
www.cs.wisc.edu/condor

I think I am sort of running out of time here. Here is a list of challenges. How do we move the data? The closest we can get to it is a quota system on some machines, but even that doesn't guarantee you that you can actually write the data when you need it.

Approach


- > Treat data placement as a "first class citizen(job)"
- > Robust and flexible storage management services
- > Distributed storage capabilities (storage appliances)
- > Uniform framework for defining and managing processing and data placement jobs


www.cs.wisc.edu/condor

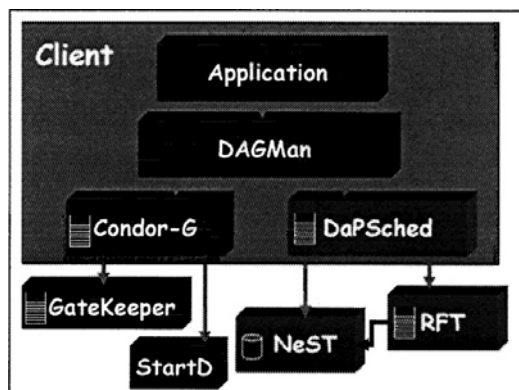
So, I already talked a little bit about it, because I want to move faster to the examples to show you that we can actually do something with all that, but the approach that we have been taking is, first of all, to make data placement first-class citizens. That means that when you write an application, when you design a system, make sure that getting space, moving the data, releasing it, is a clear action that is visible from the outside, rather than buried in a script that nobody knows about it and, if it fails, it really doesn't help us much. We have to develop appliances that allow us to use a managed storage space in this environment in a reasonable way, and then create a uniform framework for doing it.

It Works!!!!

High Energy Physics Simulations


www.cs.wisc.edu/condor


So, let me show you what we have been able to do so far. The first one is, how we can generate the simulated event, with millions and millions of simulated events.



This is sort of the high-level architecture of what we have deployed out there. So, the application is generating a high-level description of what has to be done.

DAGMan

- > **D**irected **A**cyclic **G**raph **M**anager
- > DAGMan allows you to specify the *dependencies* between your jobs, so it can *manage* them automatically for you.
- > (e.g., "Don't run job "B" until job "A" has completed successfully.")



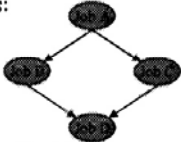
www.cs.wisc.edu/condor

This is getting into what we call—this is the Directed Acyclic Graph Manager that is responsible for controlling it.


Defining a DAG

- > A DAG is defined by a *.dag file*, listing each of its nodes and their dependencies:

```
# diamond.dag
Job A a.sub
Job B b.sub
Job C c.sub
Job D d.sub
Parent A Child B C
Parent B C Child D
```



- > each node will run the job specified by its accompanying Condor submit file


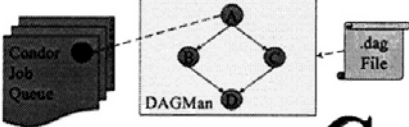


www.cs.wisc.edu/condor

Now, for some of you, if it reminds you of the old days of JCL, yes, it is like JCL, at the higher-level, but then it goes to what we call Condor-G, which is the computational part, and we have a data placement schedule that uses the other tools to do that.

Running a DAG

- > DAGMan acts as a "meta-scheduler", managing the submission of your jobs to Condor based on the DAG dependencies.



www.cs.wisc.edu/condor

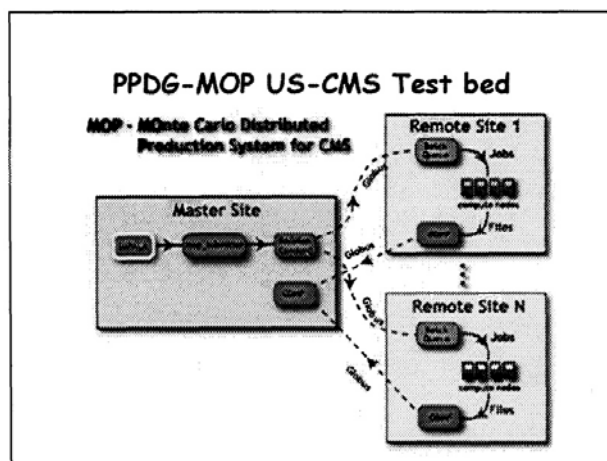
So, I am not going to talk about that since I have four minutes, and just—the

physicists have way too much free time on their hands. So, they can generate this wonderful animation.

Running a DAG (cont'd)

- > In case of a job failure, DAGMan continues until it can no longer make progress, and then creates a "rescue" file with the current state of the DAG.

The diagram illustrates the Condor workflow. On the left, a stack of papers represents the 'Condor Job Queue'. An arrow points from this queue to a central box labeled 'DAGMan', which contains a diamond-shaped Directed Acyclic Graph (DAG) with four nodes. An arrow points from the DAGMan box to a document icon labeled 'Rescue File'. Below the diagram is the Condor logo and the URL 'www.cs.wisc.edu/condor'.

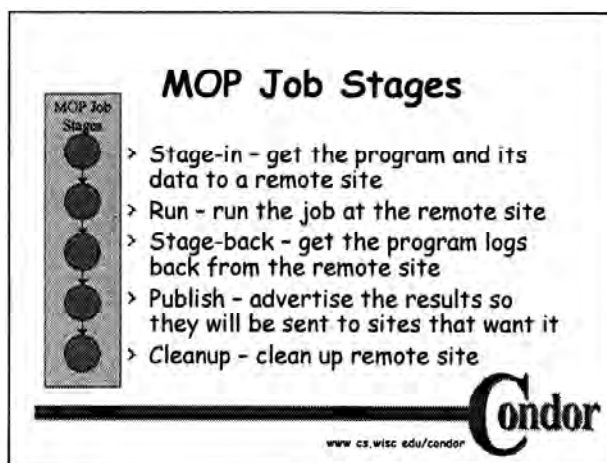


So, here is the way it works. We have a master side. IMPALA is the CMS. They have an even bigger detector than the BaBar detector, that is generating the events themselves. This is the master side. Then we have all these other sides where we send out the computations, get the data back, publish it, move the data in and we keep going.

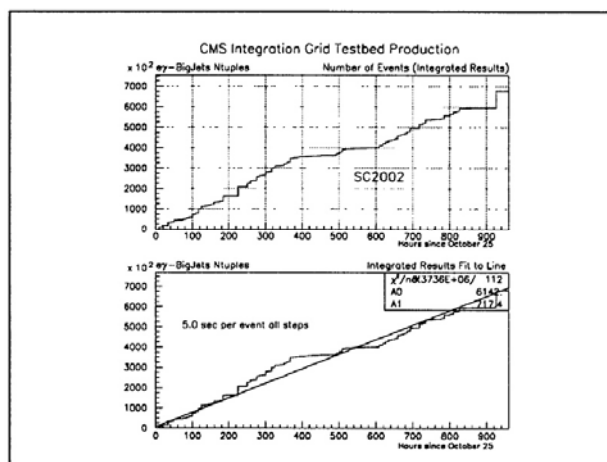
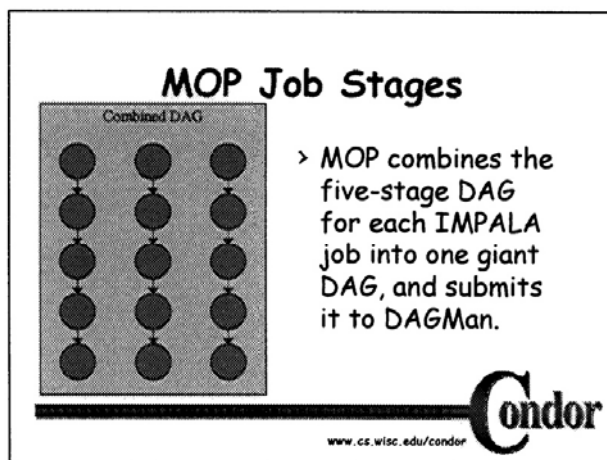
How Does MOP Work?

- > From the perspective of the CMS production system (IMPALA), MOP is almost like a local batch system. Instead of submitting jobs to PBS or Condor, the system can submit them to MOP.
- > For each physics job that IMPALA submits to MOP, MOP creates a DAG containing sub-jobs necessary to run that job on the Grid...

The Condor logo and the URL 'www.cs.wisc.edu/condor' are located at the bottom of the slide.

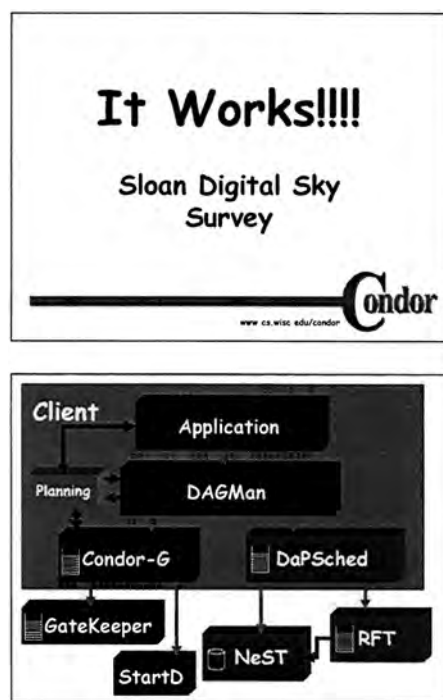


Basically, each of these jobs is a DAG like this, and then we move them all to larger DAGs that include some controls before and after, and that is the way it works.

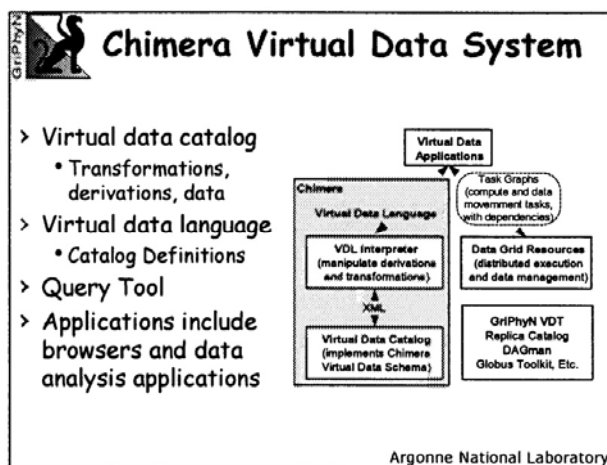


So, here is an application that this graph is, after 900 hours, this is hours since it started. So, this is one of the CMS data challenges, and this is the number of events that we have to generate. So, a job is two months, and it has to keep going, and we have to keep generating the event. This is what we have been generating using that infrastructure.

Let me show you another example of what happens when you have to do it in a more complex environment.

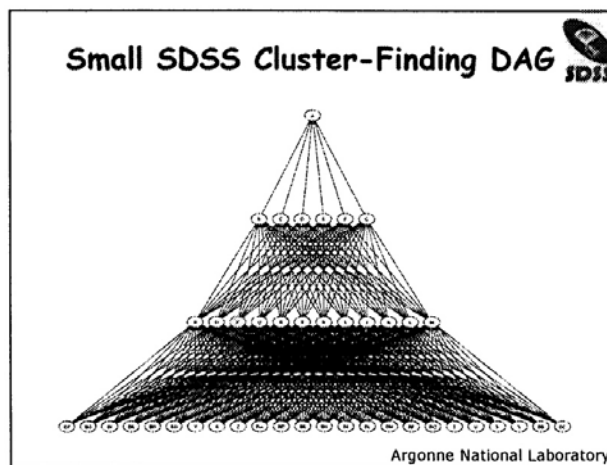
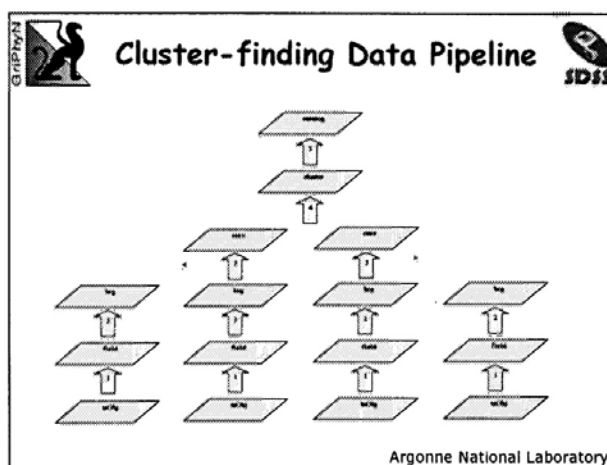


That is where we are putting in the planning. So, it is the same architecture as I showed you earlier, but there is a planning box there that is trying to make a decision on when to do it, how to do it, and what are the resources we should use for this.

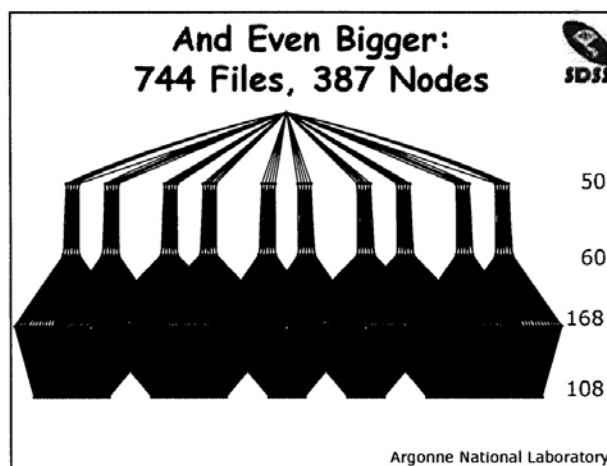


This is based on work that is actually done at Argonne National Labs and the University of Chicago as part of the GriPhyN Project, and that was because of the data system that includes higher-level information about the derivations that are formally defined and, from the derivation, we create transformations, which are the more specific acts that have to be done.

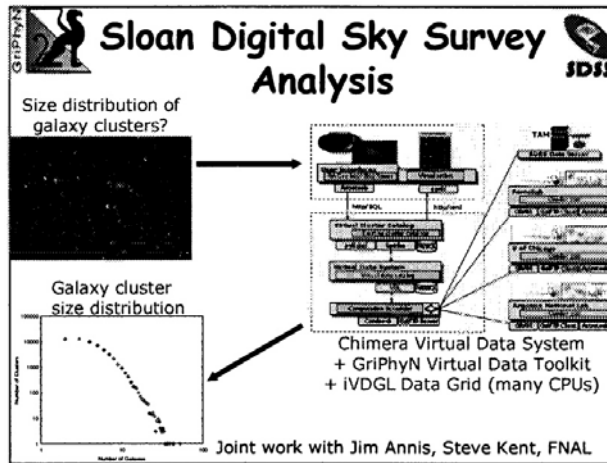
This is creating the DAGs, but they are not being executed by the architecture. As we go through, we go back to the virtual system where we come and say, tell me, now, what to do.



So, this is what we have to do there. We have to aggregate information. We have all these images. We have to pull them together. We have to create distributions, as I showed you, of galaxy sizes or whatever it is.



So, this is sort of the DAG that is going up rather than going out. This is an example of one job. This is an example of a collection of these jobs that we are actually executing. Each of the nodes in this DAG is a job that can be executed anywhere on the grid, and this is where we start.



This is the computing environment that we use to process the data, and these are the statistics.



I will leave you with that, that if you want to write applications that work well in this environment, (a) be logical. The other one, you have to be in control, because if you don't get the right service from one server, you should be prepared to move on to somebody else, if you want to use it effectively. Everyone wants lunch.

Report from Breakout Group

Instructions for Breakout Groups

MS. KELLER-MC NULTY: There are three basic questions, issues, that we would like the subgroups to come back and report on.

First of all, what sort of outstanding challenges do you see relative to the collection of material that was in the session? In particular there, we heard in all these cases that there are real specific constraints on these problems that have to be taken into consideration. We can't just assume we get the process infinitely fast, whatever we want.

The second thing is, what are the needed collaborations? It is really wonderful today. So far, we are hearing from a whole range of scientists. So, what are the needed collaborations to really make progress on these problems?

Finally, what are the mechanisms for collaboration? You know, Amy, for example, had a whole list of suggestions with her talk.

So, the three things are the challenges, what are the scientific challenges, what are the needed collaborations, and what are some ideas on mechanisms for realizing those collaborations?

Report from High-Energy Physics Breakout Group

GROUP TWO PRESENTER: I only took a few notes, so I am trying to stall, but I am glad to see Mark Hansen has arrived.

So, we talked about experimental physics. What is interesting is that there is sort of a matrix in my mind of what we discussed. I think Paul had mentioned there was a conference in Durham earlier this year in March, in which there were 100 physicists and two statisticians starting to scratch the surface of issues. There is a follow-up meeting in Stanford in September. Somebody named Brad Efron is the keynote speaker. So, presumably, there will be at least one statistician.

I think what was clear is that, sort of in the current context of what experimental physics is doing, there is a list of very specific questions that they think they would like answered. What we had discussed went beyond that. We were really looking, gee, if we had some real statisticians involved, what deeper issues could we get into.

I think that, after a good round of discussion for an hour, we decided there were probably a lot of really neat, cool things that could be done by somebody who would like to have a career changing event in their lives. Alan Wilkes is feeling a little old, but he thinks he might be willing to do this. I think on the good note is what you have, which is often—on another good note—collaborations are clearly in their infancy. There are only a few statisticians in the world, is sort of my observation. So, there is a reason why there are not a lot more collaborations than there should be, perhaps. If you look at Doug's efforts in climatology, there are really some very established efforts. If you look at astronomy, you have had some efforts in the last four years that have really escalated to the next level, and I think physics is high on the list of making it to the next step. I think there are probably a lot of agencies here in this town that would help make that happen.

The thing that gets more to sort of the issue at hand here is that there are a whole

lot of statistical things involved in what are called triggering. So, things are going on in this detector and the thing is when to record data, since they don't record all 22 terabytes a second, although they would like to, I guess, if they could.

The interesting statistic that I heard was, with what they do now, they think they get 99.1 percent of the interesting events among all the billions of ones that turn out not to be interesting. So, 99.1 is perhaps not a bad collection ratio. So, much of the really interesting statistics that we have talked about is sort of the off-line type. In other words, once you have stored away these gigabytes of data, there are lots of interesting pattern-recognition problems and stuff. Sort of on the real-time data mining sort of issue, we didn't sort of pursue that particular issue very deeply. What struck everybody was how time-sensitive the science is here, and that the way statisticians do science is sort of at the dinosaur pace and the way physicists do it is, if they only sleep three hours a night, the science would get done quicker, and it is a shame they can't stay up 24 hours a day. There is lots of discussion about magic tricks to make the science work quicker.

All in all, I think the conversation really grew in intensity and excitement for collaborations, and almost everybody seemed to have ideas about how they could contribute to the discussion. I think I would like to leave it there and ask anybody else in the group if they wanted to add something.

Daryl Pregibon

Keynote Address: Graph Mining—Discovery in Large Networks

[Abstract of Presentation](#)

[Transcript of Presentation and PowerPoint Slides](#)



BIOSKETCH: Daryl Pregibon is head of the Statistics Research Department at AT&T Shannon Research Labs. His department is responsible for developing a theoretical and computational foundation of statistics for very large data sets. He has been with the Labs for over 20 years. He has interest in and has made contributions to the three main areas of statistics: modeling, data analysis, and computing. His specific contributions include data analytic methods for generalized linear and tree-based models, incorporating statistical expertise in data analysis software, and designing and building application-specific data structures in statistical computing. He is very active in data mining, which he defines as an interdisciplinary field combining statistics, artificial intelligence, and database research.

Dr. Pregibon received a PhD in statistics from the University of Toronto in 1979 and an MS in statistics from the University of Waterloo in 1976. He is a fellow of the American Statistical Association and has published over 50 articles in his field. He was coauthor of the best applications paper, “Empirical Bayes Screening for Multi-item Association in Large Databases,” at Knowledge Discovery and Data Mining 2001 (KDD2001) and the best research paper, “Hancock: A Language for Extracting Signatures from Data Streams,” at KDD2000. He is the past chair of CATS (Committee on Applied and Theoretical Statistics, National Academy of Sciences). He was co-chair of KDD97 and has been either a special advisor or member of the KDD program committees for the past 3 years. He is co-founder of SAIAS (Society for Artificial Intelligence and Statistics). Currently he is a member of CNSTAT (Committee on National Statistics, National Academy of Sciences), a member of the SIGKDD Executive Committee, a member of the Steering Committee of IDA (Intelligent Data Analysis), and a member of the Editorial Board of Data Mining and Knowledge Discovery.

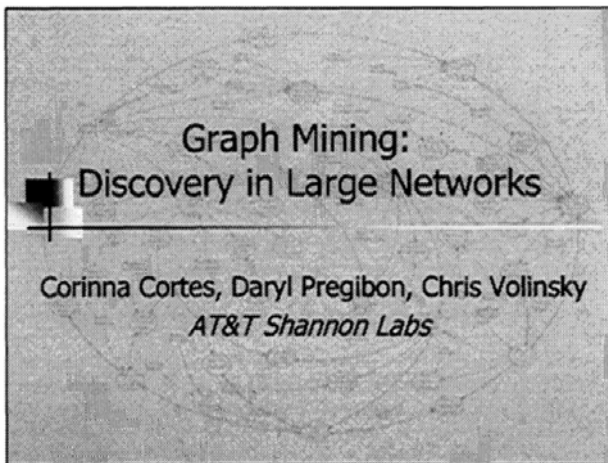
ABSTRACT OF PRESENTATION

Graph Mining: Discovery in Large Networks Daryl Pregibon (with Corinna Cortes and Chris Volinsky), AT&T
Shannon Labs

Large financial and telecommunication networks provide a rich source of problems for the data mining community. The problems are inherently quite distinct from traditional data mining in that the data records, representing transactions between pairs of entities, are not independent. Indeed, it is often the linkages between entities that are of primary interest. A second factor, network dynamics, induces further challenges as new nodes and edges are introduced through time while old edges and nodes disappear.

We discuss our approach to representing and mining large sparse graphs. Several applications in telecommunications fraud detection are used to illustrate the benefits of our approach.

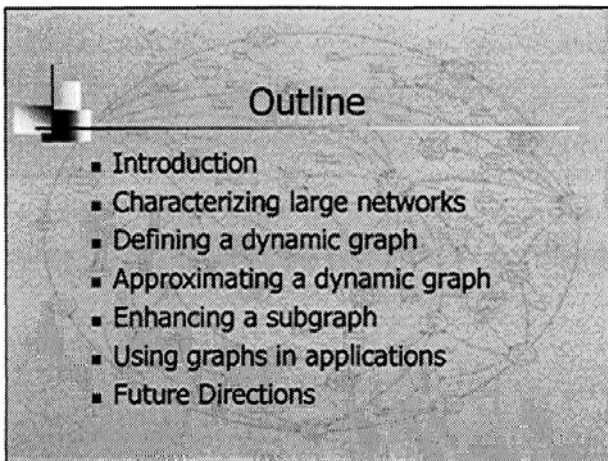
TRANSCRIPT OF PRESENTATION



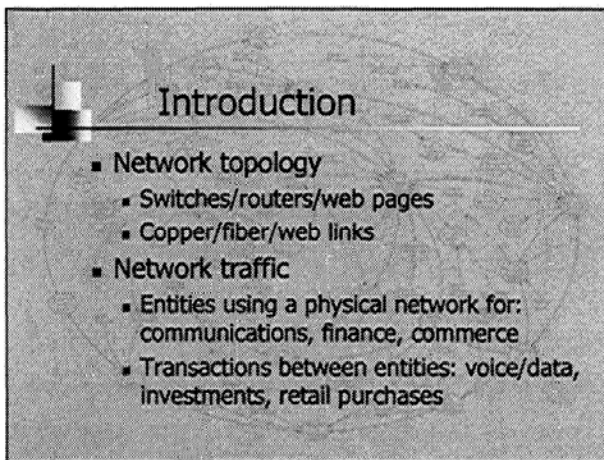
MR. PREGIBON: I guess I want to think of the words to apologize. I am listed as a keynote speaker, but I think any of the presentations we heard this morning, and probably this afternoon and tomorrow probably would qualify for this lunchtime keynote speech. The only thing I can think that it is marked as keynote is maybe it has caught the funder's eyes the most. So, with NSA being the major funding agency, this is probably an area that they are quite interested in. That is my excuse for why someone has named this a special talk. Otherwise, I think what we will see is this morning, this afternoon and tomorrow, there are many different shapes and forms for this large data problem and large data streams.

So, what you are going to hear is something completely different than what you heard this morning. This isn't big science. This isn't science at all. In fact, I am a bit jealous. It is nice to have science that you can at least hope to explain what is going on, in some phenomenon in your large data.

I am not sure that what I am going to talk about today is going to make any contribution to big science. The only thing I could think of is, I might donate my laptop to the scientific community because I think it holds some new physics in it. It is very sensitive to the characteristics of electrons and, if I just touch it the wrong way, it is going to die and reboot. It only weighs seven pounds. So, it is not a couple ton accelerator, but I really think there are some secrets in here that the physics community might be able to unlock.



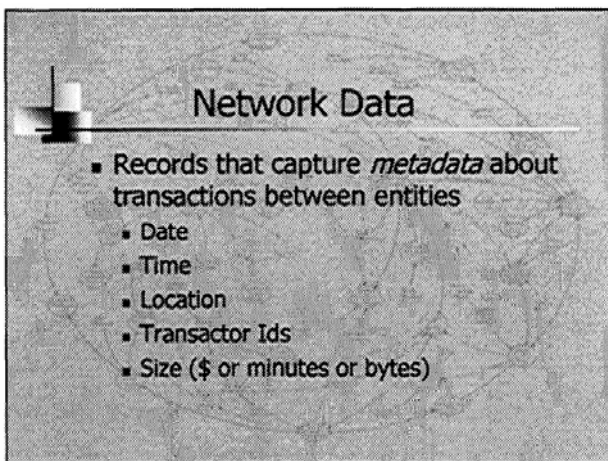
Let me get on to the topic. We are going to talk about network data. I will give you a brief introduction of what sort of network data I want to talk about. Then I want to go into a little bit of a spiel where I want to show you some network data. The reason I am doing that is to basically guide how we think about such data, and you are going to find out that, throughout the talk, there aren't a lot of statistics per se in the talk, but I think you will be able to see how statistical thinking guided a lot of sort of the ideas and the things we migrated to. By virtue of describing this network data, ideas of a dynamic graph pop up. So, we will talk a little bit about defining a dynamic graph, necessarily how we approximate that, and sometimes how we post-process that and then use it in some applications.



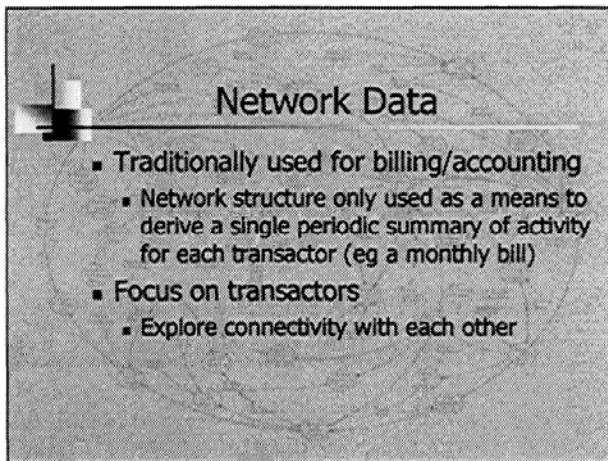
So, let me jump right in to say what the topic of my talk is versus other sorts of network data that might exist. So, generally speaking, with regard to networks, you can sort of think of static versus dynamic. I will think of network typology as a somewhat static beast. Even though the number of switches in a telephone network and the number of pieces of copper or fiber connected to them, they do change through time, they are relatively static.

They are, in many cases, physical objects. In the cases of Web pages, maybe they don't have a physical entity or identity, and Web links come and go, but they don't come and go as fast as other sorts of things. By that I mean, the thing that I am going to be talking about is actual traffic on a physical network. That is the sort of data that makes up the topic of my talk. So, the things of interest here are the entities that are using the physical network. They are using this physical network to conduct something — communication, finance, commerce, it could be transportation.

These transactions involve other entities. They could be voice data transactions, investment transactions, and retail purchases. These things are totally out of the sort of control of the person collecting the data.



Now, the data that are associated with these transactions isn't the content of the transaction. Largely speaking, they are just a few bites of information that describe some salient characteristics of the transaction, such as the date and time stamp that it took place, possibly encoded in the metadata is the location for the transaction, maybe something—definitely in my case, and I think many of these cases—some identifiers for the transactors, and often some element of size. It is the number of bytes involved in the transaction, the number of minutes the transaction lasted. Maybe in a financial or a commerce application it could be the size in dollars of the transaction. So, it may not say what was purchased, but you would know the dollar amount associated with it.



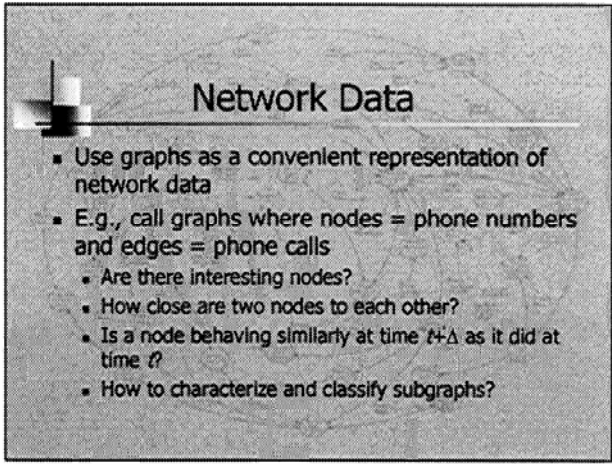
Now, traditionally, these data aren't new. They have been around forever, and they have been around for a darned good reason, because people make money, or this captures records that are used for billing. This is the thing that actually paid, and continues to pay, for the collection of such data. What we are going to talk about is trying to use these data—this morning we talked about Level 0, Level 1, Level 2 data, and for this talk, you might think of these as Level 0 data.

In fact, as the fields that I have described them, they are probably Level 1. The Level 0 data that we get off the network are pretty ugly. It is a raw, unstructured form called AMA format that we process and put into a different format called Level 1. We have our own name for it.

As we go up the food chain, I think what we try to do is add value as we go up the levels. So, this talk is almost on, as you go up, and with large data sets, you do have to filter your data. If you can think of adding value as you go up this level stack, that is kind

of what we are talking about, or one of the messages I want to get across to you in my talk.

Maybe the lesson here is that each record has just a few bytes of information, who talked to who, when and how long. So, they are pretty trivial. If you put together a couple hundred million of those a day, and you do that day in and day out, you can get a very good picture of what is going on in your network. So, that is kind of the message here.



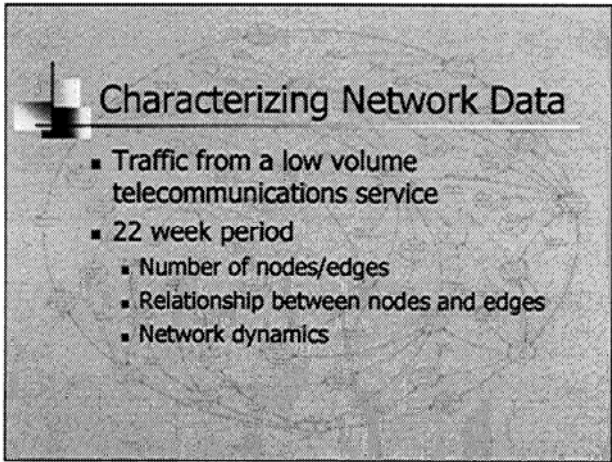
Network Data

- Use graphs as a convenient representation of network data
- E.g., call graphs where nodes = phone numbers and edges = phone calls
 - Are there interesting nodes?
 - How close are two nodes to each other?
 - Is a node behaving similarly at time $t+\Delta$ as it did at time t ?
 - How to characterize and classify subgraphs?

I think I used graphs in the title of my talk, and anyone who is associated with network data, this is a no-brainer. Graphs are a convenient representation of the data and, in the applications that I am going to be talking about, we are talking about call graphs.

So, every node in a network, the network identifier is a phone number, and the edges we are talking about are phone calls or aggregate summaries of phone calls between telephone numbers.

Some of the questions you might ask of such data, are there interesting nodes in your network? How close are two nodes to each other, not in geography, but in sort of the network topology, where the network is not the physical network, but it is the communication network. Other sorts of nodes or other questions that are interesting to me concern the temporal nature of the network. Is the node that I am looking at today behaving similarly to the way it behaved in the past. Then you might actually talk about subgraphs of the large graphs and how you might capture them and classify them.



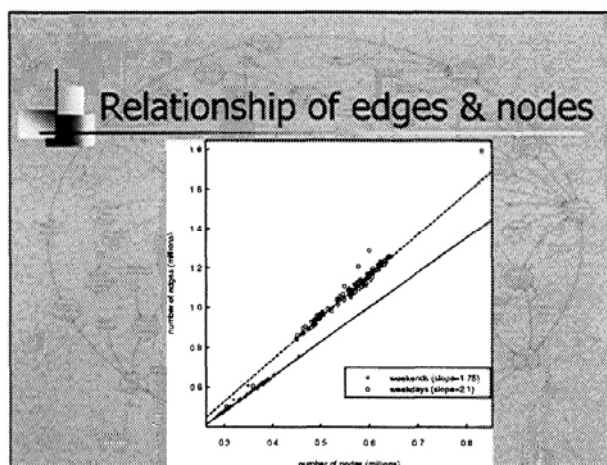
Characterizing Network Data

- Traffic from a low volume telecommunications service
- 22 week period
 - Number of nodes/edges
 - Relationship between nodes and edges
 - Network dynamics

So, to try to motivate some of the ways we think about the data, let me show you

some actual data. It is going to be from what we will call a low-volume service. We get data all the time. So, it is not quite an accelerator application, but people are making phone calls and we bill them for their phone calls. So, we have a continuous stream of data.

This is going to be a stream of data from one of our services. I picked it up for 22 weeks. What I am going to do is just give you an idea of the size, shape and scope of network data. Let me just go through a series of graphs, and hopefully you will be able to see some of these features.

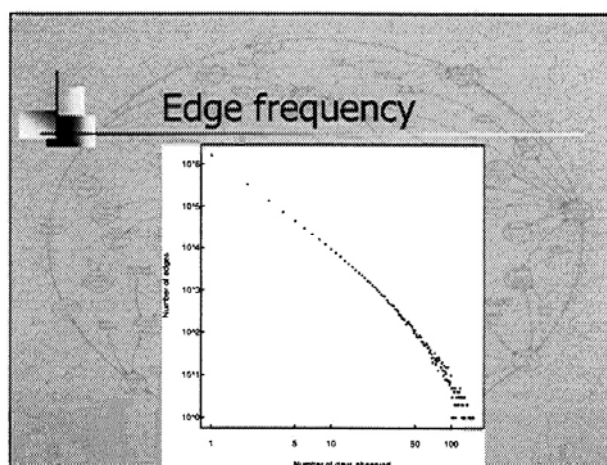


So, this is a plot. It is a scatter plot and, for each of the 154 days in the study period, I have a point. Here is a point, there is a point, and there are a bunch of points. That point contains the information on the number of nodes that were active in this service on that day, and the associated number of edges for that day.

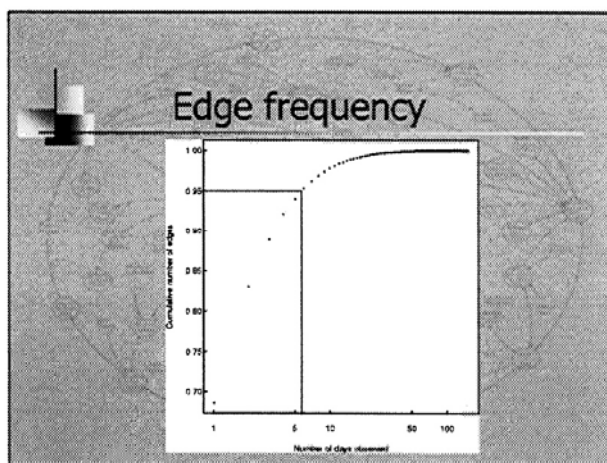
So, generally, on a weekday, we have roughly half a million transactors on our network, and they are transacting with two others. So, there are about a million edges a day, and about a half million nodes a day. On the weekends, we see it quite different, maybe only a third of a million nodes, and half a million edges.

There are some outliers that correspond to special events, holidays, etc. So, this gives you an idea, a little bit, of how many transactors we are seeing on the network, how many transactions, and also gives you a little hint of the sparsity of the graph.

If you have a graph that has n nodes, roughly, there are n^2 edges. This is hardly n^2 . I mean, it is a factor of two. So, the graphs that we are talking about, and are going to talk about, are very, very sparse, and that is a good thing, or can be a good thing.

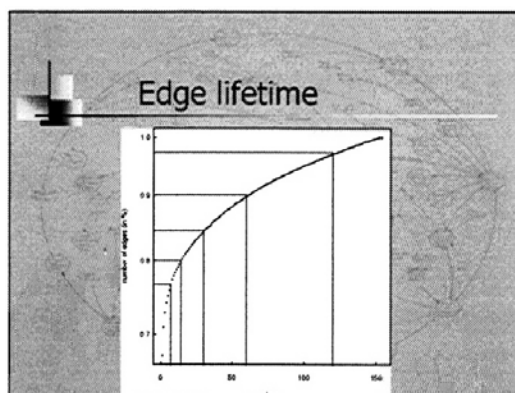


This gets at the sparsity in a slightly different way. This is a plot of the number of edges versus the number of days on which they were observed. This is a typical plot you might see in an analysis of text and other things. Most edges occur only once. So, over a million of these edges, you only see once, over about half a million of the edges you only see twice over the study period. Way down here, you see some of the edges every day.



So, this gives you a little sense of the temporal nature and how frequently you observe things. These are the same data presented slightly differently, and it just shows you the cumulative frequency.

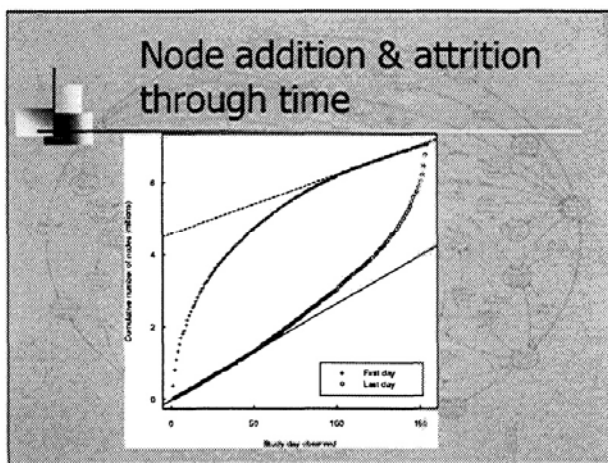
So, 95 percent of the edges have occurred on six or fewer days, of a 154-day study period. So, that gives you an idea of the episodic nature with which these edges come and go.



This is a different way to view the data. This talk is about the lifetime of the edges. So, we have an edge and, if we have only seen it once, its lifetime is a day.

Some edges we have seen twice. So, we say, have we seen them today and tomorrow, or did we see one the first day of the study period, and the second time we saw it was the last day of the study period.

What this display is meant to convey is that roughly, say, 75 percent of the edges came and went within a week. Eighty percent of the edges came and went in about two weeks. Eighty-five percent came and went within a month, and 90 percent of them, the first time we saw them and the last time we saw them was about two months, which is less than half the study period. So, these things come and go and you may never see them again, and their half lives are generally very short.



This is one of my favorite plots, and someone criticized me for saying it is my favorite one because it takes so long to explain, and a good plot, you shouldn't have to explain it, but let me make a crack at it. What I want to get at in this plot is just the fact that nodes come in and go out through time, and the numbers are pretty staggering.

What I did was take the entire study period, and I got the complete list of network identifiers. There were about 6.5 million of them. Then I said, okay, let's start at day one, or how many are unique on day one. Well, they are all unique on day one, because that is where you are starting. On the second day, how many new ones did you see? On the third day, how many new ones that you hadn't seen the first couple of days and so forth.

After a while you sort of get over this initial starting period and you get to a sort of steady state. I have fitted a trend line to this steady state up here. The slope of that corresponds to the fact that every day we see 16,000 new nodes in our graph. Then, I did the same going in reverse. I said, okay, we have all these nodes and now the steady state is going to be the reverse process.

When was the first day of the study period? For how many nodes was that the last day I saw that node? The second day of the study period, for how many nodes was that the last day that I saw it, and fitted a trend line to the beginning of that, because the truncation for that will occur over here. Generally speaking, the slope of that trend line is that we are losing 27,000 new nodes a day. So, this is not a good business to be in.

Again, what we are trying to capture is, nodes come in and go out and there are lots of them. It is not trivial numbers. So, you have to take this into account.

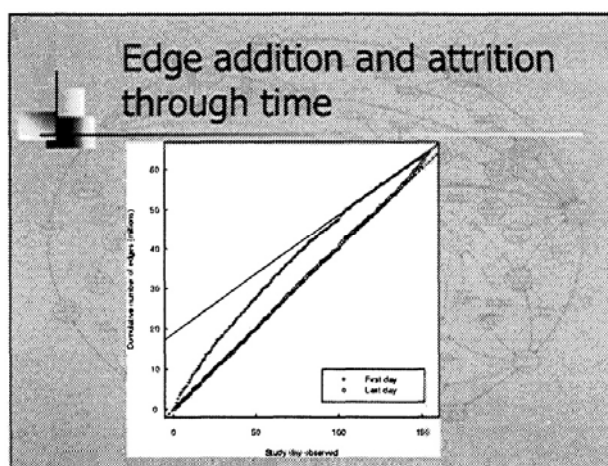
AUDIENCE: [Question off microphone.]

MR. PREGIBON: They haven't converged yet. The question was, do these lines converge? My response was they haven't converged yet.

AUDIENCE: [Question off microphone.]

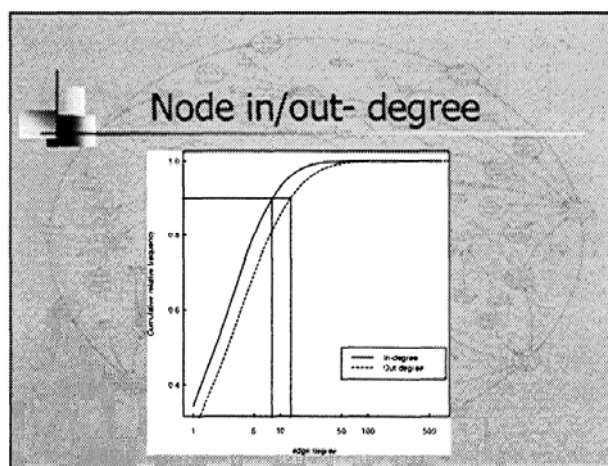
MR. PREGIBON: The second question was, what would happen if I had run this for more weeks. I believe what would happen, I would see the same slopes, and I think the lines would be closer to intersection than they are right now.

This business, by the way, I don't think it is that proprietary. It is our calling card business. Calling cards are buying out by virtue of cell phones and prepaid cards. So, this is a business that we are essentially harvesting. We are not investing in it, people aren't using it, and this is the rate at which that business is, you know, going in the toilet.



So, the final thing is a similar plot—I guess it is not the final one, there is another plot after this—the same sort of business on the edges. What is a little interesting to me is that the attrition of edges is nearly linear. I can't explain that. This is the addition of edges.

The numbers here are also big. The slopes of these lines, we are seeing 300,000 new edges a day that we have never seen before in the steady state, and we are losing 400,000 edges a day that we will never see a again. That gives you sort of an idea of both the node set and the edge set.



This is the final graph in this set. This is speaking toward, again, the connectivity or the sparseness of these data. I have done it in a cumulative plot and I think I have tried

to capture the 90th percentile. So, 90 percent of the nodes on this graph have an in degree of eight or fewer edges, and I think 90 percent of the nodes on this graph had an out degree of 13 or fewer edges, and this is over the entire study period.

So, that is, again, a bit of a background for, when you are in this business of studying graphs or traffic or transactions on a network, that is the type of volatility that you have to deal with.

Defining the network graph

What does the graph at time t mean?

Some notation:

\oplus is the graph addition operator

$$G_t = \alpha G_1 \oplus \beta G_2$$

such that G_t contains the union of the nodes and edges in G_1 and G_2 , and the weights on the edges of G_t are:

$$w(G_t) = \alpha w(G_1) + \beta w(G_2)$$

Now, let's get into how that data motivated sort of the structures we used to capture this graph. I keep saying this graph as if there is a graph. It is changing all the time. So, therein lies the problem. We have to come up with a definition of what we need.

So, I will introduce a Microsoft operator. I found that in the symbol table. It looks like a little plus sign. That is a graph operator.

Basically, all we are going to say is that we can take two graphs—graph one and graph two—and combine them or take a linear combination of them, such that the resulting graph will have the union of all the nodes and edges of graphs one and two, but that we are just going to take a linear combination of the weights along those edges to find a new graph, or graph three.

Defining the network graph

- today's nodes and edges:
 $G_t = g_t$
Too narrow!
- union of all time periods:
 $G_t = g_1 \oplus g_2 \oplus \dots \oplus g_t = \bigoplus_{i=1}^t g_i$
Too broad!
- moving average of the most recent time periods:
 $G_t = g_{t-k} \oplus g_{t-k+1} \oplus \dots \oplus g_t = \bigoplus_{i=t-k}^t g_i$
Too many!

I will show you how that is used on this viewgraph. Again, there are many different ways to define what we mean by the network graph. I am putting most of these up here as a straw man to say, none of these are the ones we chose, but these are all

legitimate choices for legitimate purposes.

Again, you can just define the graph to be the network traffic you see on some period, say, today. That could be the graph you are interested in. Given the volatility that we see, 300,000 new edges today that we didn't see, and 400,000 are going away, that graph may be viewed as being too narrow for some applications, of too high variance.

The sort of complement of that is to say, don't throw anything away, and let your graph just grow over time and don't sort of delete anything, or just keep adding stuff to it. That possibly, for some applications, is too broad. If you are using this graph for inferential purposes, you may not want traffic from a year ago to affect inferences on your graph today. For some applications, you may exactly want that. For some of the things we do, we don't want that.

Defining the network graph

We adopt: $G_i = \theta G_{i-1} \oplus (1-\theta) g_i$

i.e., today's graph is defined as a convex combination of yesterday's graph and today's data

Alternatively: $G_i = \omega_1 g_1 \oplus \omega_2 g_2 \oplus \dots \oplus \omega_i g_i = \bigoplus_{j=1}^i \omega_j g_j$

where $\omega_i = \theta^{i-1} (1-\theta)$

i.e., an exponentially weighted moving average

Advantages:

- only one graph need be stored
- recent data has most influence

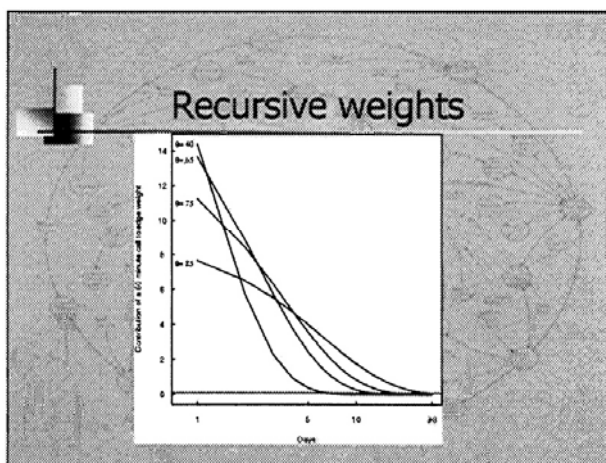
Something closer to what we want is this. We want the graph to be defined as what is happening recently with the network. You can think of this as simply a moving window. So, we have traffic coming over the network. We are going to window this.

The only problem we have with this is that, to maintain this, we have to maintain too many little graphs. So, in order to update this graph tomorrow, we have to throw this guy away, and then add another one on the other end. Then, depending on how wide this window is—if it is a month—we are going to manage 30 graphs all the time.

So, it is not a surprise to think of where I am going with this. I sort of said, well, we like this windowing operating but we don't like to pay for the storage and maintenance of all these graphs.

Well, one way to get a benefit of that is to simply do an exponential average where we take a convex combination of the graph like it was through yesterday, and then just add in the graph for today, using sort of a factor of θ , which is going to sort of guide how wide of a window you are looking at for your application. Again, just doing the simple algebra, you can expand the recursion and basically—this isn't anything new — most people who study this type of data, I think, this would be a natural definition for them.

The benefits of this, from the recursion you see that recent data would have the most influence. From a computational point of view, it is great, really because only one graph needs to be stored. You have yesterday's graph, you put today's data in, and you immediately have the new graph for today.



That parameter, θ , is, again, adjustable. I just show you some values of θ that we use for different applications.

If we said θ equals about .85 and you follow that curve down, for our applications, if we do daily processing, that means an hour phone call will last, in our analysis, for roughly about a month. So, it won't wash out for a month.

For other values of θ , I think that is a θ of .4, if we set the value of .4, that one hour call will wash out in about a week. Again, it is completely adjustable. It really depends on your application. In telecom, phone numbers, when they go out of existence, they are reassigned. Typically, they can be reassigned within a month. So, we typically use a 30-day value and a θ of .85. So, we don't blend activity on the new user of a phone number with activity on the previous owner.

Constructive representation of the graph as the union of all subgraphs

$\forall v \in G$

$I_v = \{v_i \rightarrow v; w_i\}$ List of edges that terminate at v

$O_v = \{v \rightarrow v_i; w_i\}$ List of edges that originate at v

$S_v = \{v : I_v, O_v\}$ Diameter 1 subgraph centered on v

$G = \bigcup_{all v} S_v$

- Redundant since each edge stored twice
- Allows fast expansion of larger sub-graphs centered on each node

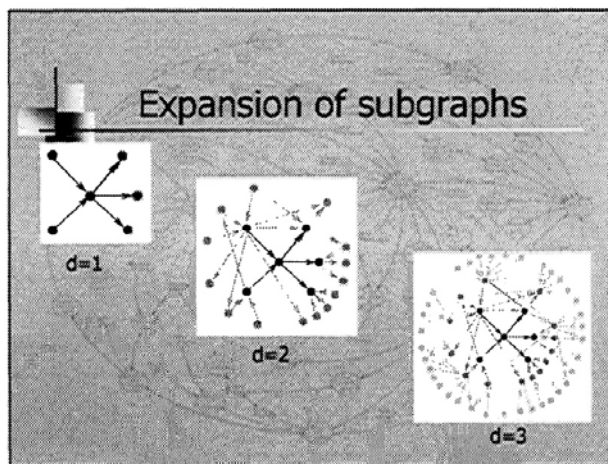
So, let's talk about the graph again. So, that captures the dynamic nature of our graph. Now, how are we going to represent it?

One way that we like to represent the graph—and you will sort of see the applications that motivate this—is in a constructive sense, and we are going to think of it as a union of all subgraphs, where the subgraphs are, for every node in the network—so, this is important here, and we will get to this again later—for every node in the graph, we keep the set of edges going out of it or coming into it, the set of edges going out of it, and that defines a diameter one subgraph centered on that node, and we are going to do that for every node. Then the graph, by definition, is just a union of all those subgraphs.

Now, there is a big penalty here, because I have stored every edge twice. If I call

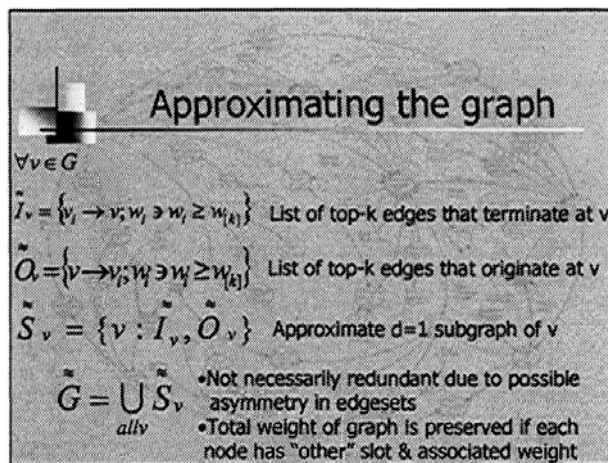
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

Amy up, that edge is going to be in my subgraph, and it is also going to be in Amy's subgraph. So, this is pretty redundant.



The main reason we do this is, it allows for fast expansion of subgraphs centered on every node. By that I mean, in a lot of the applications we are interested in not just who called me and who I called, but we want to take that out.

For every one of the people I called, who called them and who did they call. By building this redundant structure, it is very easy to go from this down to this. So, it is very easy to traverse this data structure to build subgraphs of arbitrary depth, and literally within a second. So, that is the main reason why we like this constructive representation.



We don't use that, though, in practice. What we actually do is approximate the graph. The nature of the approximation kind of builds off this constructive definition.

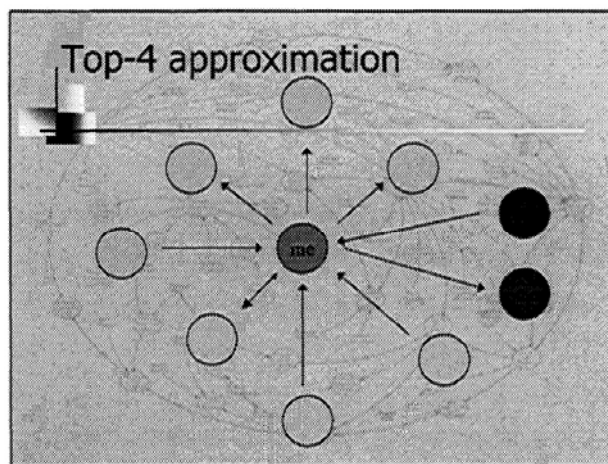
So, we are going to approximate our large graph by sacrificing something. Now, I am not going to sacrifice information on the node set. I want information on every node in my graph. What I am going to sacrifice is some of the edges. So, I am only going to retain edges in my graph that are sort of big enough or important enough.

That can be defined by the user, but I am going to truncate my edge set, say I am only going to keep the k edges with the largest weights on them going on. The k edges with the largest weights going out, and now I can define my subgraph now, centered on every node, and it is going to be approximate. Then I can define the big graph to be the union of all these subgraphs.

Now, through sleight of hand, this graph isn't necessarily redundant. The reason

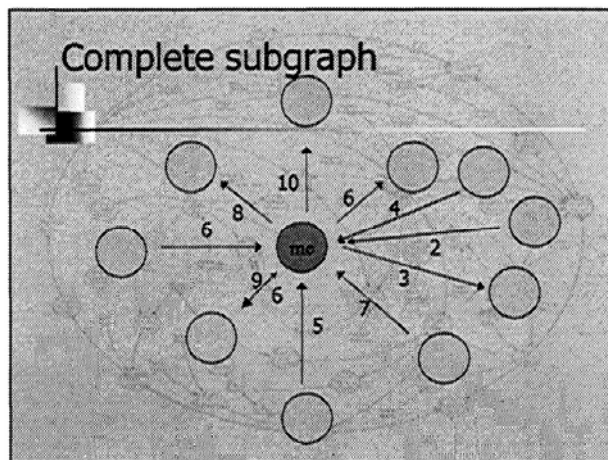
for that is sort of asymmetry. Amy might be very popular. When she calls me, I might fall out of her subgraph because she only spoke to me for a minute, where she talks to her family members and friends for quite long. I might be unpopular, so her edge may stick around in my subgraph for a long time. So, even though there is a possibility of redundancy in this representation, it may not occur.

The other thing we try to do in our approximation is the following. We threw away edges, but we are going to try to maintain the total weight of the subgraph by maintaining the weight in a sort of funny node for each node in the graph, and we call it slot other. Every node will have two slot others, that sort of catch the carry over, and let me show you how this works.



This is, let's say, my subgraph. I guess I labeled it me in the middle. Bill Gates is responsible for rotating some of these names up here. I just typed them in and I must have twisted something when I put in my circles, but when I went to type in the names, that is what it wanted to do and I couldn't undo it. So, if Bill Gates is responsible for doing it, it is my responsibility for not knowing how to undo it. So, anyway, this is my subgraph.

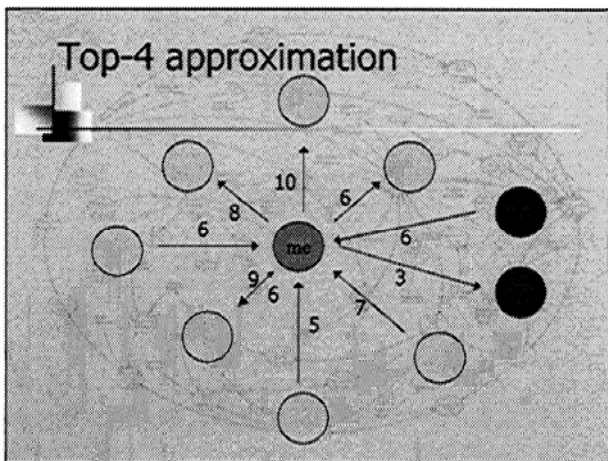
Now, suppose I wanted to do a top four approximation for this. What this means is that I want to maintain only four of the nodes going in and four of the nodes going out.



I am going to do that by saying, I am going to drop off the nodes with the smallest weight. If we look around here, it looks like these guys have the smallest weight. So, when my boss calls, or when Paul calls me, these guys are going to be left, and when I

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

call Bill, I don't call him enough, so I am going to whack that node.



What I have done is, I have replaced those nodes here with two other nodes, just labeled “other.” So, they are not going to carry the names associated with who made those calls, or the number of calls that were in there. So, if I tried to mimic Amy's animation, I can do that. So, you can see how I am collapsing several nodes down to one, and then replacing them with node “other.”

AUDIENCE: [Question off microphone.]

MR. PREGIBON: So, the question is, what are the weights. So, the weights typically would be associated with some characteristic of the transaction. So, the weight might be the dollar value of a transaction, length of a phone call, number of bytes of the connection between this IP address and that IP address. So, sorry for the confusion.

Anyway, the top four approximation for my subgraph, again, only maintains the top four in and out directions, plus overflow nodes that I have thrown away the labels on, but I have maintained the weight. So, my approximate subgraph has every node in the network in it. It has the total weight as the original subgraph. All I have done is pruned some of the edges. So, that is the beast we have chosen to work with.

AUDIENCE: [Question off microphone.]

MR. PREGIBON: So, the question was, it is not really the top four because of the temporal nature of the data. I could have had something in my top four that slid off the top four but now is coming back in and would get in my top four.

$$\tilde{G}_t = \text{topk} \left\{ \theta \tilde{G}_{t-1} + (1-\theta)g_t \right\}$$

$$= \bigcup_{v \in V} \tilde{S}_v(t)$$

e.g., Update the graph by updating all of the atomic subgraphs

| Old top-4 edges | (1 0) | Today's edges | = | New top-4 edges | |
|-----------------|-------|---------------|------|-----------------|-----|
| node-labels | wt | node-labels | wt | node-labels | wt |
| XXA6305407 | 3.9 | XXA6302467 | 2.0 | XXA7502656 | 5.2 |
| XXA7502656 | 3.0 | XXA7302856 | 6.2 | XXA6329187 | 4.8 |
| XXA6329187 | 4.4 | XXA6329187 | 0.8 | XXA6329187 | 3.8 |
| XXA634231 | 2.3 | XXA634231 | 10.0 | XXA634231 | 2.0 |
| XXA6343142 | 1.9 | | | XXA6343142 | 1.6 |
| XXA7354212 | 1.8 | | | XXA7354212 | 1.5 |
| XXA4231423 | 0.8 | | | XXA4231423 | 1.5 |
| XXA4231423 | 0.5 | | | XXA4231423 | 6.7 |
| XXA506432 | 0.2 | | | XXA506432 | 6.4 |
| Other | 0.1 | Other | 0.0 | Other | 6.3 |

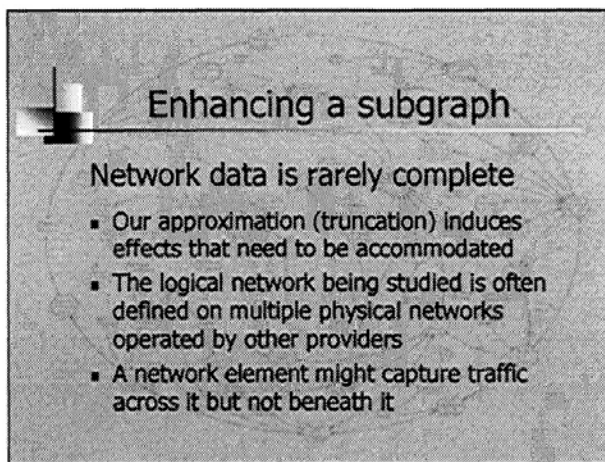
That is the segue into this view graph, is how do we update this thing. You are

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

absolutely right. If something falls off, it will go into other, and it has to fight its way back into the top four.

The way it works, sort of at the atomic level, is that we are going to update this graph by updating all of the subgraphs at the atomic level. So, we may have the node set from yesterday, we have today's calls, and we take the convex combination of the weights, and then, when we do that, we then do the sort and only the top k nodes in that sort are retained.

If you were in my top four and you then fall into other, I have got to keep calling you in order for you to fight your way back into my top four. So, then the approximate graph, then, through time, is then defined by that operation.



So, the final thing I will talk about is something that we are going to do day in and day out. As the data streams in, we are going to construct a network topology of the data streaming in and maintain that. You can think of this as kind of a database. Basically, we are building what you might call a materialized view of the raw transactions, and this view is the network view of the data that can be queried very, very fast.

Again, you put in a seed node and, within a second, you can get a subgraph out surrounding this node. When you abstract that subgraph, you may want to enhance it. This is just a fact of life, and reflects the fact that you data really aren't everything that you would like.

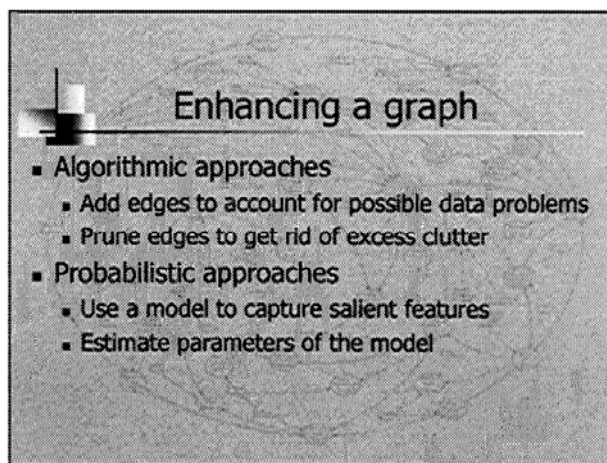
For instance, this is something that we brought on ourselves. We threw away some stuff. So, we might want to think about ways that, at least when we analyze data, that we account for the fact that we threw away some stuff. Then, there are these two cases where we would have liked other stuff, but we didn't get it. So, you know, in the good old days, AT&T was a monopoly, so we had all the network transactions. So, our graph was complete. Today, we are not a monopoly. So, there are edges that we don't observe because those calls aren't carried on our network.

Now, in building these graphs, for security's sake, you may want to have all these edges in, even though your data may not have observations on these edges. So, that is a potential problem. Another one is that, you know, you may be capturing intercepting data at one level in the network stack, and there are communications going on beneath that, that you won't see.

So, in my application, AT&T is primarily a long-distance company. So, I will see long-distance phone calls. If I call my parents in Ohio and my brother, who also lives in my same town calls them, the AT&T network data will see calls going to my parents

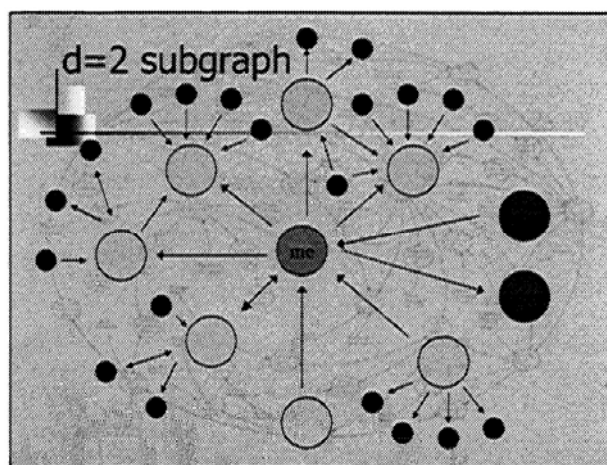
from each of our numbers. They won't see me calling my brother, because we live in the same town and that is a local call.

You can think of it the same in IP space. There are routers, in different parts of the Internet. You may see traffic crossing some of the routers, but you are not going to see stuff below. If you are really trying to build up network connectivity for all of these network elements, that may be important to you.

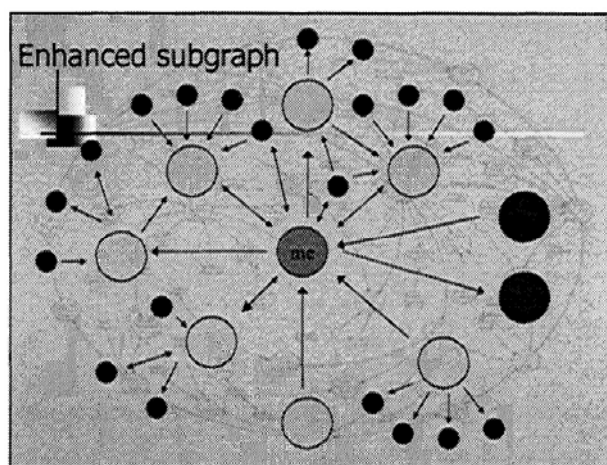


We are exploring several approaches to enhancing the graphs. I will show you examples of both of them. One is strictly algorithmic, the other probabilistic.

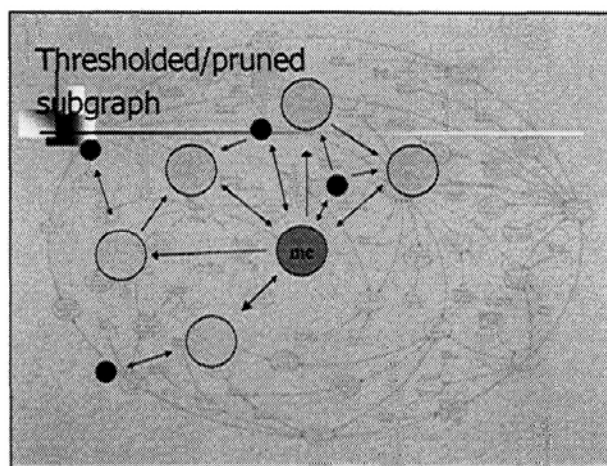
So, the algorithmic ones first. What we try to do here is capture some of the richness that our data doesn't bring to us that we would like to bring back in. So, in a deterministic fashion, we will add some edges and then we will do some pruning to get rid of them.



Again, through a cartoon, how we do that, this might be my diameter two subgraph, again, showing the category other, and then for the nodes I called, or that called me, who they conversed with.



I may add some edges in. So, for some reason or other, these edges weren't in my data, but I wanted to add them in.



Then, after I added them in, I probably have too much junk in there and I will get rid of them. One algorithm we use to get rid of things is just running strongly connected components or some other fast graph algorithm, to basically prune out stuff.

Maybe I should say something here that is obvious to anyone who has looked at these graphs. Graphs are like classification trees. People think of them, oh, they are simple, they are easy to interpret. Classification trees, like graphs, are easy to interpret if they are small. Classification trees, regression trees and graphs are really hard to interpret if they are big. So, pruning is almost necessary in every application, because these things get big. We seldom go out more than a radius of two in our graphs. You just bring in too much junk. Even at diameter two, you have to prune away junk.

Probability distributions on subgraphs

Consider a subgraph $S=\{v,e\}$ such that

$$p(e) \propto \exp \left[\sum_{i \leq j} \rho_{ij} e_{ij} e_{ji} + \sum_{i \neq j} \theta_{ij} e_{ij} \right]$$

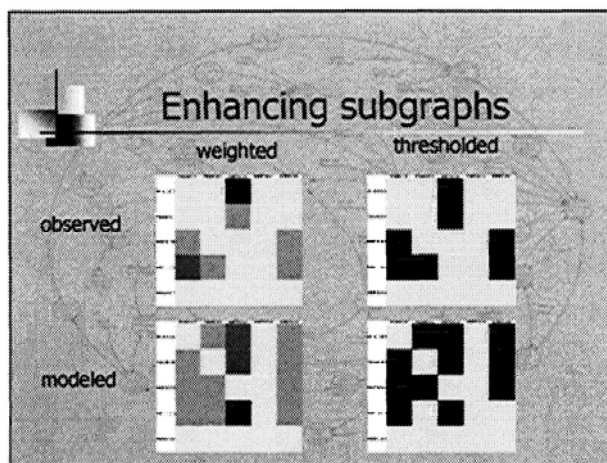
$\rho_{ij} = \rho$ ρ =graph reciprocity
 $\theta_{ij} = \theta + \alpha_i + \beta_j$ θ =graph density
 α =node expansiveness
 β =node attractiveness

The second way we add stuff into our graphs is using probability models. The class of models we are using here are log-linear models, which are quite nice, because they give nice interpretable parameters, both for the graph overall, and also parameters at the node level. So, if you wanted to think about computing these models for every node, you can then cluster your parameters, to cluster your nodes. It is a very flexible type of model.

The way we account for the fact that we are missing data is to put some priors on some of these parameters, and you can decide which ones you want priors on. Then, once you have those, you can use an N type algorithm to estimate the parameters, and this is one of the things we are experimenting with.

Generally, we are at the point now for about a 200 node diameter two subgraph in the R language, we can compute these models in about two minutes. That is an interpretive language. So, we can't do it for a large number of subgraphs. If we rewrote that in C, we think we could get it down to about two seconds per subgraph, but we are still not satisfied with it at the level of modeling that we are doing.

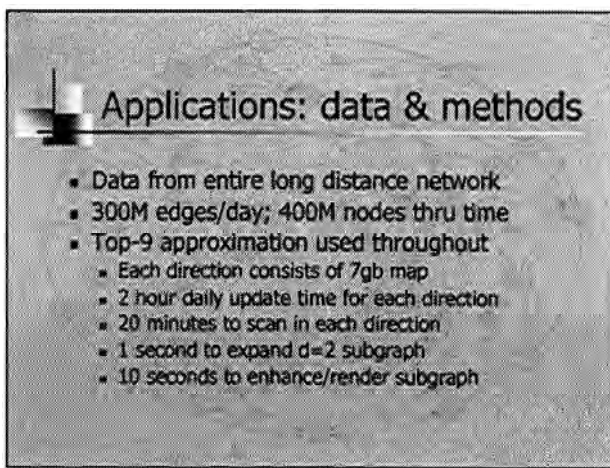
The other nice thing about these models, as a statistician, we would like to think about the parameters of the model. There are also the ultimate fitted probabilities. So, in the model that I put up before, we are actually modeling sort of the probability of an edge. You can think of then using the output of the statistical model as a way to prune and enhance your graph.



For instance, this is actually, I think, a very small subgraph around me. I think there are only six nodes in this subgraph. So, these are the six nodes, and these might be

my observed data. If I was going to draw a graph of that data, this might be the threshold version of data that I use for making that graph.

If I fit a model to that data, this might be the probabilities associated with the edges on that model and, if I threshold this thing, this might be the graph I plot. So, this is a way that we are going to be using to observe data, add in edges probabilistically, and then prune them back to something that we hope we can understand, and that captures the salient features of the data.

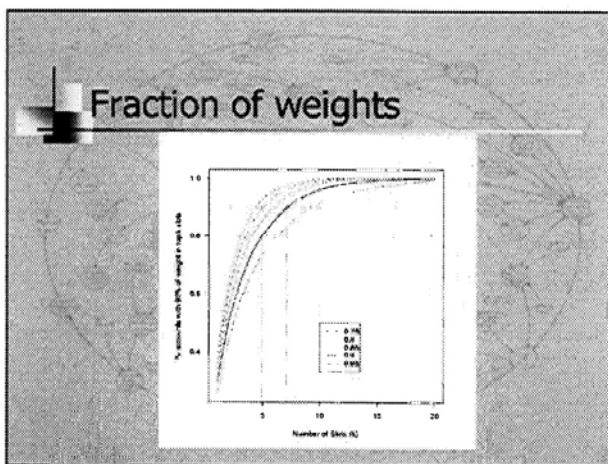
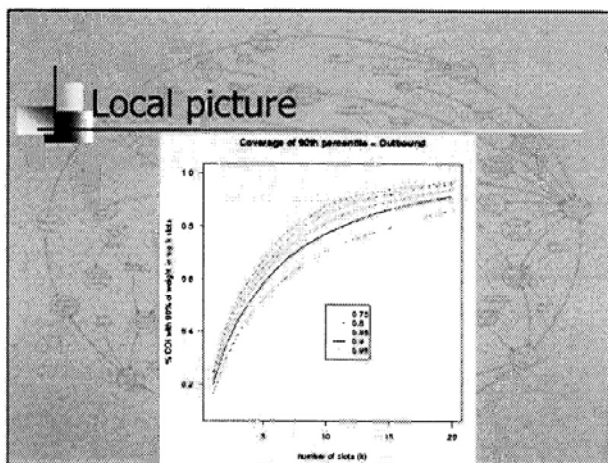
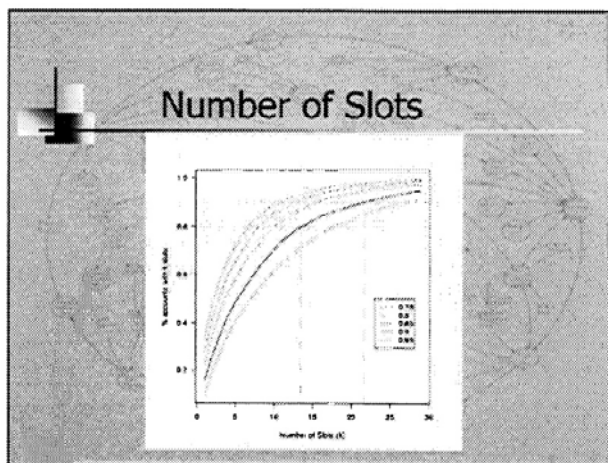


Let me, in the last five minutes or so, go over the applications and give you an idea for the volumes we are talking about. I mentioned before, we are a long-distance company. So, this is the data from the long-distance network.

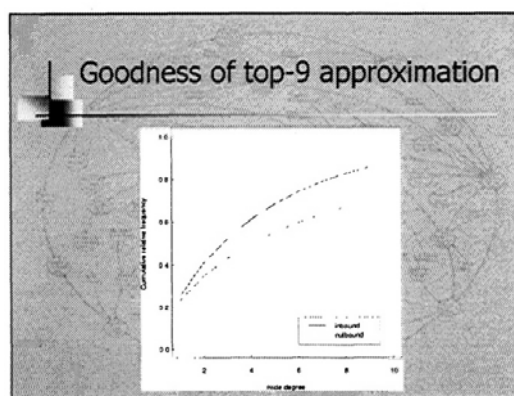
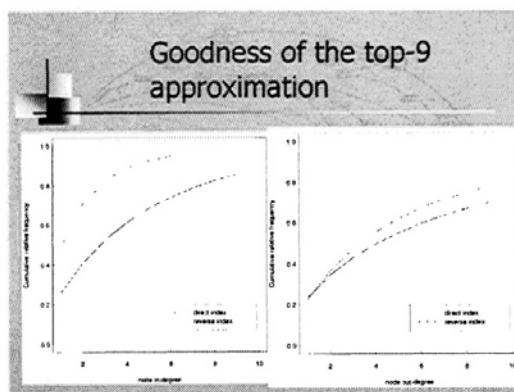
We see about 350 million edges a day. That is a reasonably big number by this morning discussion. The bigger number is the number of nodes that we see through time, nearly half a billion nodes through time. So, it is a pretty big graph when you look at it through time. In our work, we tend to use the top nine approximation. That is for historical reasons. We have been changing that over the past year.

I will give you an idea about the sizes of these materialized views. Each direction, inbound and outbound, are about 7 gig. It takes us about two hours to process the data daily. If we wanted to scan each of these maps and compute something for each, it takes about 20 minutes. It is easily under a second to expand a diameter two subgraph for any node.

When we want to do the enhancement, flavor it and then render it on someone's Web site, it is basically about 10 seconds from "give me the information" to seeing the subgraphs. Those are some of the operating characteristics that we are dealing with.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



Subscription fraud

The problem.

- Customers subscribe for service with no intent to pay
- Identity theft and/or flawed processes let them in the door
- If left undetected, 60-90 days pass before service is disconnected

Okay, so without further ado, let's talk about the applications. The primary use of these tools is in fraud detection. I will outline one type of fraud that we deal with, and those in the audience with similar problems, they probably have similar versions of this. They aren't necessarily for fraud.

Customers subscribe for service. We like that. They don't pay us, we don't like that, and there are a lot of reasons why this happens. Some of it is our own doing. It is a competitive market. We work hard to get customers. We don't screen them, maybe, as well as we should. Other times, people out there who want to be bad are pretty good at being bad. They will steal identities. No matter how good we check on their credentials, it comes up good because this is a perfectly good credit record that this individual has.

The only problem is that it is not the person.

Then, the problem gets exacerbated, because not only did we give them service, we basically gave it to them for a couple of months. They use the service for 30 days, we send them a bill. Thirty days later, we find out they didn't pay the bill. So, we start reminding them, gently at first and more vigorously later. Eventually, we lose our patience, but literally, 60 to 90 days. So, it is not just getting service, but it is service over an extended period. So, this is something we don't like.

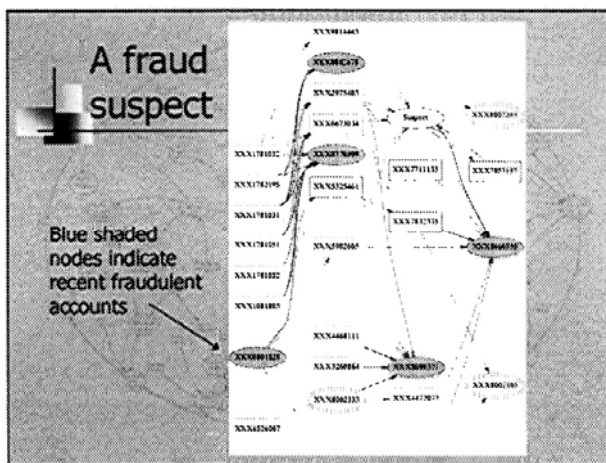
In a security operation, you may not want bad guys floating around for 60 or 90 days without you knowing about them. You would like to know about them within minutes, days, weeks, rather than 60 or 90 days.

Subscription fraud

The plan.

- Fraudsters seldom work in isolation
- This implies a clustering of fraudulent network activity
- Exploit this affinity group to rank suspects by 'guilt by association'

So, part of the plan we have is, you know, bad guys don't work in isolation, or you count on the fact that they don't. So, there is going to be some clustering of bad behavior in your network, and we will exploit this to come up with ways to describe this affinity group, and just to rank suspects by who they are associated with.



So, for instance, I think I have outlined a node up here. This is a suspect. We can compute their diameter two subgraph, add some value to it and then—I won't go into detail on what the nodes of different shapes and sizes mean.

Think of some as cellular handsets and others as land lines, thickness and color of the lines depicting the weight or the strength of that association. Color is the main thing here. Any node that is colored was associated with fraud very recently. So, this

particular phone number that was a suspect, in their community of interest, has five other bad guys associated with it. This is what we want to bring to the attention of the security staff, that this would be someone good to investigate.

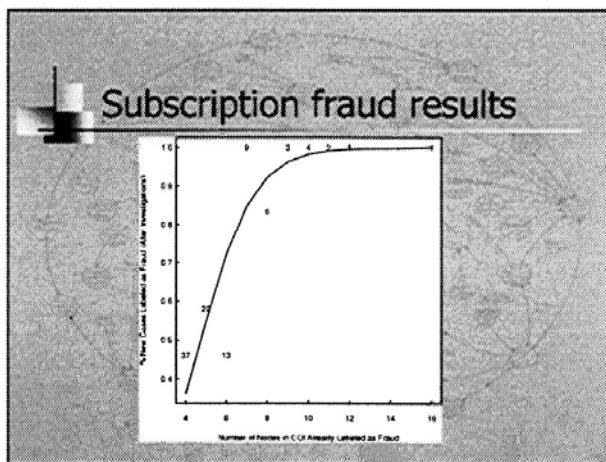
Subscription fraud

The process.

- Compute $d=2$ subgraph for each new account 7 days after 1st sighting
- Enhance and threshold subgraphs
- Color nodes in subgraphs according to recent reports from fraud center
- Rank new accounts according to number of fraudulent nodes in their subgraph
- Render subgraph through web-app

So, the way we do this is, for every new account we see in our network for a particular type of service, we will compute their subgraph seven days after we sight it. Seven days is, again, just some made up number, a waiting period. You want the calling circle to mature, depending on the level of activity. It could be one day, it could be six days, but seven is what we settled on.

We will do some enhancement of those subgraphs and then threshold them. We will go to a database and look for recent fraud cases. So, we will color those subgraphs. We will rank them and put them on a Web site and the security associates then just go through and look at this, and do their investigations.



This is a result of what we saw from a trial when we first developed this. On this axis is the number of nodes in a diameter subgraph that are associated with previous fraud cases. This was the outcome of new cases post investigation.

The number plotted at each plotting position is the number of cases. So, there were 13 cases that we presented to security that had six bad guys in the community of interest, and about 45 percent of those turned out to be fraud.

Out here, there were six cases that we presented that had about nine bad guys in it. Over 80 percent of those turned out to be bad. As soon as you get beyond that, it is

almost 100 percent. If you are surrounded by badness, by golly, you are bad, no doubt about it.

Where the action, though, is down here where you don't see. This is the mess we were cleaning up when we started this. Now, when you see one or two bad guys around a new person, that is a big indicator that things are bad. So, by rolling this out and having active investigation, we are able to really clean up the mess.

Account linkage

The problem.

- Fraudsters (or their customers) are seldom prosecuted and they remain 'loyal' until it becomes tedious
- While the node identifier has changed, the entity behind the node is the same
- If left undetected, 60-90 days pass before service is disconnected (once again)

The second thing I will talk about, and just close on this, is tracking bad guys. You can think about this as account linkage. People who are bad, as I said, they are good at it. Just like customers, they exhibit loyalty. If you have got a good product, they are going to be loyal. Bad guys are the same way. If you have got a bad product, they will be loyal to you.

If you let them abuse your network for a year without catching them, they will come back tomorrow and abuse it for another year because you are not hassling them. Fraud detection versus some of the security things, we don't have to be perfect. We just have to be better than the competition. The fraud is not going to go away. It is going to go to some network. We just don't want it to be ours.

So, the nature of the game is to really hassle these guys. So, we want to hassle them, but again, they have their own tricks. So, identity theft is not quite science yet, but there are some people that are darned good at it.

Account linkage

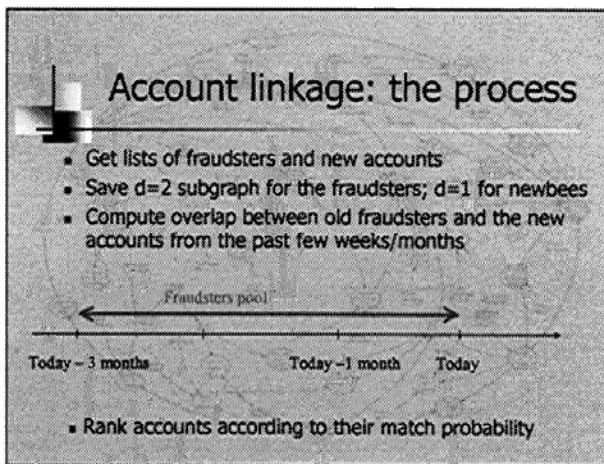
The plan.

- While the node identifier has changed, the calling circle of the entity should be similar (you are who you call!)
- Keep library of subgraphs of bad accounts and compare to those of newbees
- Rank new accounts according to the quality of overlap with subgraphs in the library

So, you know, you can knock them off your network and because they don't cost

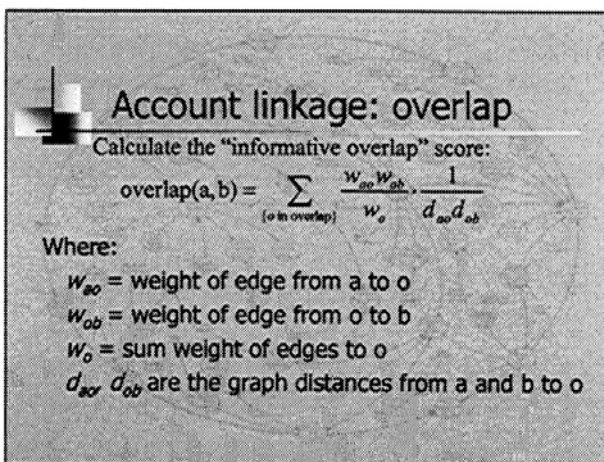
you as much as you would like to, they will come back on your network. If you are dumb enough not to know they are back, they are going to abuse you again. So, it is the same old game except, you know, burned once, you don't want to get burned again.

So, even though the node identifier—so, the phone identifier on your network is new, is it possible that the person behind that identity is the same. So, we are after them. Basically, we want to exploit the fact that you are who you call. You may be burning your cell phone or a prepaid card at a fast rate, but maybe your friends and family, your girlfriends or whatever, they are not doing the same thing.



So, their lifetime in your network has a different half-life than yours, and we want to exploit that fact. So, what we do is, we keep a library of subgraphs for these baddies and then new guys come along and we want to do a graph matching.

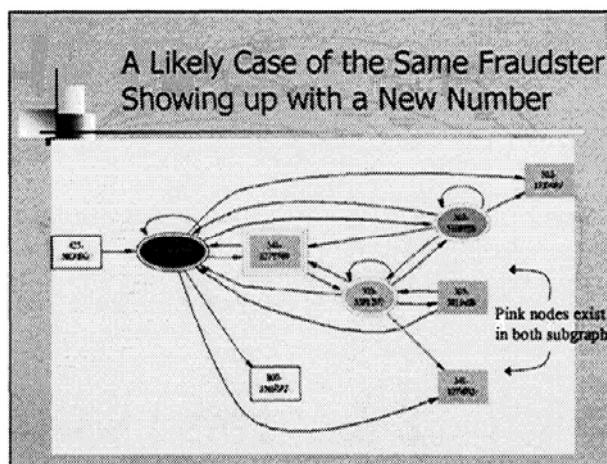
We want to say, are any of these newbies likely to be any of these ones we have caught before. So, it is a big problem. Again, it is a graph matching problem, and they don't necessarily come back on your network the same day that you bumped them off. So, you are collecting a pool of baddies with a pool of new customers, trying to do graph matching and then ranking these pairs to present to security.



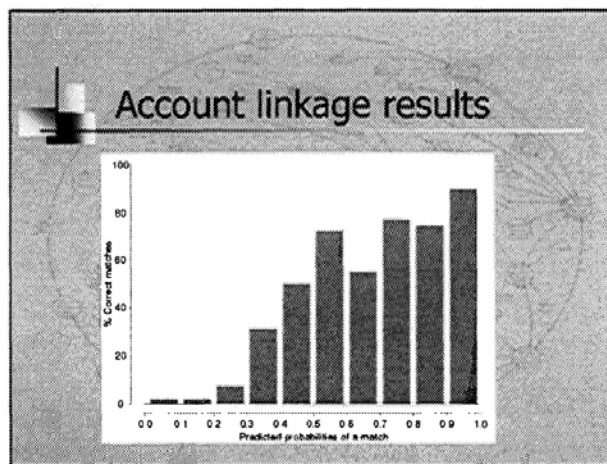
Again, there is a lot of engineering involved, and what do you mean by graph matching? So, basically, you are taking a graph and a subgraph, overlapping them and saying, you know, how good is the overlap. You know, this is just something we have come up with. It mimics, I think, a little bit of what goes on in text analysis, some of the scoring methods that are used there. You want to account for the fact that big weights

associated with edges are good, but if a node is very common, it is not very discriminatory, you want to down weight it.

The fact that both the newbie and the old guy call Lands' End isn't very informative. If they both called a number in Yemen, maybe that is informative, if no one else called that number. So, that is what this is getting at.



I am not going to go into details. I guess this is a cartoon—it is actually a real case of this. Anything that is in pink or red are the overlaps on two subgraphs. The network IDs for the subgraphs we are overlaying are both in here. Anything that is green is associated with only the top number here, anything in blue the bottom one. This just shows that these two identities that are appearing on your network at different times are quite likely the same person behind it.

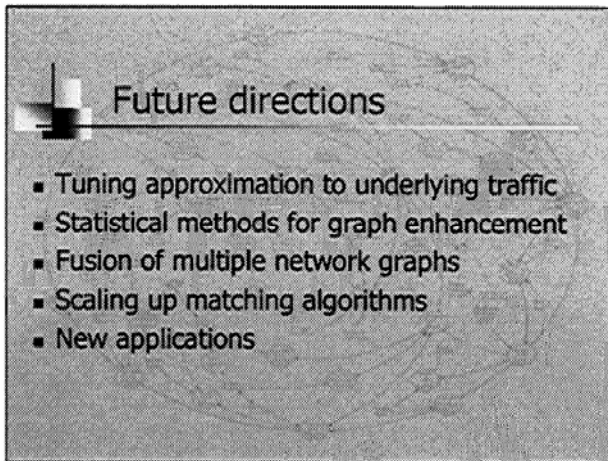


Quickly, just to show you the results of implementing this in the investigative sense, giving a bunch of leads to our spot associates, based on the deciles of predicted match, and then looking through time to say how many of those cases that were presented turned out to be true matches, and that the pairs that we presented were actually matched pairs.

It just shows we are reasonably well calibrated. This thing would be perfectly on a 45 degree, on a perfect calibration. It is showing that, when we think it is a match, generally, after investigation, it is a match. The thing that, to us, makes this so powerful, again, speaks to a number of things that we are seeing today.

Generally speaking, we are getting tens of thousands of new customers on our network a day, tens of thousands of baddies of different sorts, not paying their bills or whatever, a day. So, the fact that we are able to distill this down to less than a thousand things for our fraud team to investigate, it is a big crunching of this down.

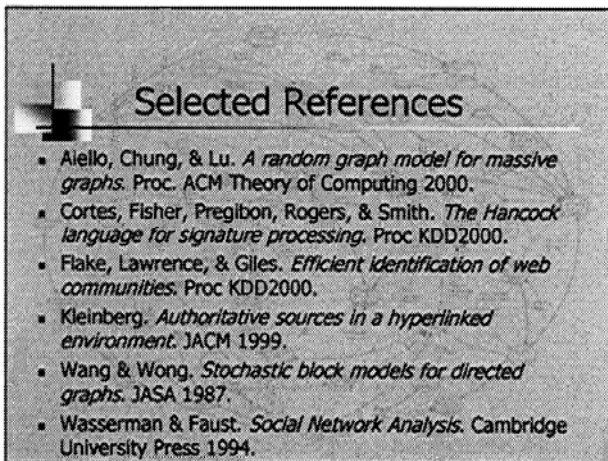
In a physical sense, as a physicist, you have got all of this data. Where do you look? So, these tools are to guide our physicists to show, this is where you have got to look.



Future directions

- Tuning approximation to underlying traffic
- Statistical methods for graph enhancement
- Fusion of multiple network graphs
- Scaling up matching algorithms
- New applications

I am just going to close things by saying this is ongoing work, a lot going on, and thank you for your patience and your time. Thanks.



Selected References

- Aiello, Chung, & Lu. *A random graph model for massive graphs*. Proc. ACM Theory of Computing 2000.
- Cortes, Fisher, Pregibon, Rogers, & Smith. *The Hancock language for signature processing*. Proc KDD2000.
- Flake, Lawrence, & Giles. *Efficient identification of web communities*. Proc KDD2000.
- Kleinberg. *Authoritative sources in a hyperlinked environment*. JACM 1999.
- Wang & Wong. *Stochastic block models for directed graphs*. JASA 1987.
- Wasserman & Faust. *Social Network Analysis*. Cambridge University Press 1994.

MS. KELLER-MC NULTY: While we are getting John hooked up for the next session, if there are a couple of questions?

[Question off microphone.]

MR. PREGIBON: The question is, do I ever use spectral techniques for these graphs. I think possibly what you mean there is some of the Hudson Authority type computations?

[Comments off microphone.]

MR. PREGIBON: No, we have not. We should probably talk off line. There is some mathematics associated with analysis of graphs in the literature, with argon analysis. Those sorts of spectral methods are used to characterize and process findings of special nodes on the graph. For people who are familiar with the work of John Kleinberg from Cornell, his Hudson Authority work is of that ilk.

Sallie Keller-McNulty, Chair of Session on Integrated Data Systems

Introduction by Session Chair

Transcript of Presentation

BIOSKETCH: Sallie Keller-McNulty is group leader for the Statistical Sciences Group at Los Alamos National Laboratory. Before she moved to Los Alamos, Dr. Keller-McNulty was professor and director of graduate studies at the Department of Statistics, Kansas State University, where she has been on the faculty since 1985. She spent 2 years between 1994 and 1996 as program director, Statistics and Probability, Division of Mathematical Sciences, National Science Foundation. Her on-going areas of research focus on computational and graphical statistics applied to statistical databases, including complex data/model integration and related software and modeling techniques, and she is an expert in the area of data access and confidentiality. Dr. Keller-McNulty currently serves on two National Research Council committees, the CSTB Committee on Computing and Communications Research to Enable Better Use of Information Technology in Government and the Committee on National Statistics' Panel on the Research on Future Census Methods (for Census 2010), and chairs the National Academy of Sciences' Committee on Applied and Theoretical Statistics. She received her PhD in statistics from Iowa State University of Science and Technology. She is a fellow of the American Statistical Association (ASA) and has held several positions within the ASA, including currently serving on its board of directors. She is an associate editor of *Statistical Science* and has served as associate editor of the *Journal of Computational and Graphical Statistics* and the *Journal of the American Statistical Association*. She serves on the executive committee of the National Institute of Statistical Sciences, on the executive committee of the American Association for the Advancement of Science's Section U, and chairs the Committee of Presidents of Statistical Societies. Her Web page can be found at <http://www.stat.lanl.gov/people/skeller.shtml>

TRANSCRIPT OF PRESENTATION

MS. KELLER-MCNULTY: Our next session has to do with integrated data streams. Actually, it has been alluded to in the sessions prior to this as well, the multiplatforms, how do you integrate the data?

We are going to start off with a talk that is sort of overview in nature, that is going to present some pretty broad problems that we need to start being prepared—we need to start to prepare ourselves how to address. That is going to be by Doug Season, who is one of the deputy lab directors in the threat reduction directorate at Los Alamos. He has been involved with different presidential advisors at OSTP throughout his career, for both Clinton and Bush, has a long history of looking into and being interested and doing, himself, science in this whole area.

That is going to be followed by a talk by Kevin Vixie, who will look at some hyperspectral analyses, kind of focus in on a piece of this problem. He is a mathematician at Los Alamos.

Finally, our last speaker will be John Elder, who has been looking hard at integrating models and integrating data, and both hardware and software methods to do that.

J. Douglas Beason

Global Situational Awareness

[Abstract of Presentation](#)

[Transcript of Presentation](#)



BIOSKETCH: Douglas Beason is the director of the International, Space and Response (ISR) Division at the Los Alamos National Laboratory, responsible for over 400 professionals conducting research and development in intelligence, space, sensor, and directed energy programs. He has over 26 years of R&D experience that spans conducting basic research to directing applied science national security programs and formulating national policy.

Dr. Beason previously served on the White House staff, working for the President's Science Advisor in both the Bush and Clinton administrations. He has performed research at the Lawrence Livermore National Laboratory; directed a plasma physics laboratory; taught as an associate professor of physics and director of faculty research; was deputy director for directed energy, USAF Research Laboratory; and is a member of numerous national review boards and committees, including the USAF Science Advisory Board and a Vice Presidential commission on space exploration. He retired as a colonel from the Air Force after 24 years, with his last assignment as commander of the Phillips Research Site, Kirtland AFB, New Mexico.

Dr. Beason holds PhD and MS degrees in physics from the University of New Mexico, an MS in national resource strategy from the National Defense University, and is a graduate of the Air Force Academy with bachelor's degrees in physics and mathematics. The author of 12 books and more than 100 other publications, he is a fellow of the American Physical Society, a distinguished graduate of the Industrial College of the Armed Forces, a recipient of the NDU President's Strategic Vision Award, and a Nebula Award finalist.

ABSTRACT OF PRESENTATION

Global Situational Awareness

Douglas Beason, Los Alamos National Laboratory

Battlefield awareness can sway the outcome of a war. For example, General Schwarzkopf's "Hail Mary" feint in the Gulf War would not have been possible if the Iraqis had had access to the same overhead imagery that was available to the Alliance forces. Achieving and maintaining complete battlefield awareness made it possible for the United States to dominate both tactically and strategically.

Global situational awareness can extend this advantage to global proportions. It can lift the fog of war by providing war fighters and decision makers capabilities for assessing the state anywhere, at any time—locating, identifying, characterizing, and tracking every combatant (terrorist), facility, and piece of equipment, from engagement to theater ranges, and spanning terrestrial (land/sea/air) through space domains. In the world of asymmetric warfare that counterterrorism so thoroughly stresses, the real-time sensitivity to effects (as opposed to threats from specific, preidentified adversaries) that is offered by global situational awareness will be the deciding factor in achieving a dominating, persistent victory.

The national need for global situational awareness is recognized throughout the highest levels of our government. In the words of Undersecretary of the Air Force and Director of the National Reconnaissance Office Peter Teets, "While the intelligence collection capabilities have been excellent, we need to add persistence to the equation. You'd like to know all the time what's going on around the face of the globe."

Global situational awareness is achieved by acquiring, integrating, processing, analyzing, assessing, and exploiting data from a diverse and globally dispersed array of ground, sea, air, and space-based, distributed sensors and human intelligence. This entails intelligently collecting huge (terabyte) volumes of multidimensional and hyperspectral data and text through the use of distributed sensors; processing and fusing the data via sophisticated algorithms running on adaptable computing systems; mining the data through the use of rapid feature-recognition and subtle-change-detection techniques; intelligently exploiting the resulting information to make projections in multiple dimensions; and disseminating the resulting knowledge to decision makers, all in as near a real-time manner as possible.

TRANSCRIPT OF PRESENTATION

MR. BEASON: Thanks, Sallie. This will be from the perspective of a physicist. While I was flying out here, I sat by a particle physicist. I, myself, am a plasma physicist. I found out that, even though we were both from the same field, we really couldn't understand each other. So, good luck on this.

Global situational awareness is a thrust that the government is undertaking, largely due to the events surrounding September 11 of 2001. Basically, it is a thrust to try to give decision makers the ability to be able to assess the socioeconomic or tactical battlefield situation in near-real time. The vision in this is to be able to do it anywhere any time, and this is versus everywhere all the time. Now, everywhere all the time may never be achieved, and we may never want to achieve that, especially because of the legal ramifications. The vision to be able to monitor nearly everywhere any time, what I am going to do is to walk you through some of the logic involved.

First of all, what does it mean by that? What does it mean by some of the sensors? Then, really get to the core of the matter, which is how do we handle the data. That really is the big problem, not only the assimilation of it, understanding it, trying to fuse it together. We will mine it, fuse it, and then try to predict what is going to happen.

Here is an outline of the talk. What are the types of capabilities that I am talking about in the concept of operations? Then, I will spend a little bit of time on the signatures. That is kind of the gravy on here. Again, I am a physicist, and this is where the fun part is. What do we collect and why, a little bit of the physics behind it, and how do we handle the data, how do we mine it, and then how do we fuse it? What scale of problem am I talking about? If we just purely consider—if we try to decouple the problem from the law enforcement to the space situational awareness, and just look, for example, at the battlefield awareness, what people are talking about in the Defense Department is some kind of grid on the order of 100 by 100 kilometers. So, that is 10^4 kilometers. Then, up to 50 kilometers high, and knowing the resolution down to a meter. That is like 10^{14} points. This is just the battlefield itself. So, it just staggers your mind, the problem. So, let me give some examples, and what do we mean about the global capabilities.

First of all, the argument is being made that it is more than visible. That is, it is more than looking at photographs. It is more than looking at imagery. It includes all types of sensors, and I am going to walk you through this in a minute. It also includes cyberspace, Web sites, e-mail traffic, especially if there is a flurry of activity. What you would like to do is, you would like to have the ability to be able to look at what we call known sites, and to visit these in a time where, first of all, things don't change very much. That is, it could be a building going up and you may only have to revisit this site perhaps weekly, daily or even hourly, if you would like. These are sites where something may be going on, or even Web sites, but you know that the delta time change is not very much, so you don't have to really revisit it too much.

The second thing is that you really want to have the capability for monitoring for specific events. If there is a nuclear explosion, if missiles are being moved around, if terrorists are meeting somewhere, you want to have those specific events. You want to be able to telescope down to them, to be able to tap into them. You want to be able to do it on a global scale. Second of all, for those kinds of activities, you may have to have

some kind of a tip-off. You don't know, either through any kind of intercepts, like telephone conversations or visual intelligence. You may have to have human intelligence direct you to what is going to be happening. So, that is what you would like on a global scale. On a local scale, you would want very specific things to occur.

For example, perhaps when equipment is being turned on or off, if this is a terrorist that you have located that you are communicating, you want to be able to not only geolocate the terrorist, but also to determine some of the equipment that they may be using. This thing of dismounts, right now, this is one of DARPA's largest problems. A dismount is, if you can imagine a caravan going through the desert and you are tracking this caravan and all of a sudden somebody jumps off the caravan, you don't want to divert your observation asset away from that caravan, but yet, this person who jumped off may be the key person. So, you would want to have the capability of not only following that caravan, but to follow this person across the desert, as they jump into a car, perhaps, drive to an airport, jump in a plane and then fly somewhere to go to a meeting.

So, how do you do something like that? Again, it is not just visual. If you can imagine an integrated system of sensors that combine, say, acoustic sensors that are embedded in the ground that can follow the individual, and then hand off to some kind of RF—a radio-frequency sensor—that could follow the car, that could, again, follow the plane that the person may go into.

So, what type of sensors are needed, and how do you integrate this in a way so that you don't have a bunch of scientists sitting in a room, each person looking at an oscilloscope saying, okay, this is happening and that is happening and then you are going to hand it off to the next person. What type of virtual space do you need to build to be able to assimilate all this information, integrate it and then hand it off. So, these are some of the problems I will be talking about.

The traditional way of looking at this problem is to build a layered system of systems. That is, you have to balance everything from sensitivity resolution, coverage and data volume. I will give you a quick example. Back in the Bosnian War, the communications channels of the military were nearly brought to their knees. The reason was not because of all the high information density that was going back and forth on the communications channel. It was because, when people would send out, say, air tasking orders or orders to go after a certain site, they would send them on PowerPoint slides with 50 or 60 emblems, each bit mapped all around the PowerPoint.

So, you had maybe 20 or 30 megabytes of a file that had maybe 20 bits of information on it. So, you have to be smart in this. So, the point there is that when you are making these studies, the answer is not just to build bigger pipes and to make bigger iron to calculate what is going on. You have to do it in a smart way. Again, this is part of the problem, and now what I am going to do is walk you through, first of all, some of the sensors and some of the ways that people think we may be able to attack this problem.

First of all, there is more to the sensing than visual imagery. Let me walk you through some examples of these. The case I am trying to build up here is that, for this global situational awareness, the problem is not really inventing new widgets. It is the information, and the information is really the key. It is where the bottleneck is.

So, I am going to walk you through just some examples of some sensors that already exist. Some of them are already being used. It is not, again, a case of building new technology all the time. On the lower left-hand side, what you are looking at is a

defense threat reduction agency. So, this is why I am here, is to translate that, our project. It is a hard, deeply buried target project.

We are basically looking at an underground cavern and trying to determine where assets are in this underground cavern. Of course, that is a timely question today. You can do this by using acoustic sensors. That is, you know the resonances that are built up in these three-dimensional cavities. Just like you can calculate the surface of a drumhead when it is struck, in a three-dimensional cavity, if you know where the resonances are located, what you can do is back out of that where the assets are, if you know that trucks are in there, for example, or that people are walking around.

It is kind of like a three-dimensional pipe organ. This just shows some unique characteristics that arise from the power spectrogram of that. On the upper right-hand side it is just showing that, believe it or not, there is a difference in the acoustic signatures of solid state and liquid fuel—not solid state, but solid propellant liquid fuel rockets.

You can back out what the differences are, and you can identify whether or not somebody has shot off, not only what type of rocket, if it solid or liquid fueled, but also the unique rocket itself from that. So, there are other types of sensors, using sonics—and I will talk a little bit more here when I talk about distributed networks. If you can have the ability to be able to geolocate your sensors in a very precise manner—say, by using differential GPS—then what you can do is correlate the acoustic signatures that you get. You can, for example, geolocate the position of snipers. You can imagine, then, that those distributed sensors don't even have to be stationary, but they could also be moving, if you have a time-resolution that is high enough.

What are the types of sensors that we are talking about? Well, radio-frequency sensors. For example, on the lower left-hand side, it shows a missile launcher that is erecting. Most of the examples I am using for global situational awareness are military in nature, but that is because of the audience that this was pitched at.

What occurs in a physics sense, any time you have a detonation that happens in an engine, you have a very low temperature plasma that is created. Any time you have a plasma, plasmas are not perfect. That is, they are not ideal NHD plasmids. You have charge separation, which means that you have radio-frequency emissions. You are able to pick that up. In fact, the missions are dependent upon the cavity that they are created in. So, there is a unique signature that you can tag not only to each class of vehicle, but also the vehicle itself that you can pick out.

Up on the right-hand side, it shows the same type of phenomenology that is being used to detect high explosives when they go off. I am not talking about mega-ton high explosives. I am talking about the pound class, 5- to 10-pound classes of explosives. Again, it creates a very low temperature plasma. An RF field is generated, and you can not only detect that RF field, but also, what you can do is, you can geo-located those. These are, again, examples of sensors that can be added to this global sensor array. We have two examples here of spectral type data. On the right-hand side is data from a satellite known as multi-thermal imager. It is a national laboratory satellite, joint effort between us and Sandia National Laboratory. It uses 15 bands in the infrared. It is the first time that something like this has been up and has been calibrated.

What you are looking at is the actual dust distribution the day after the World Trade Center went down. This is from the hot dust that had diffused over lower Manhattan. I can't tell you exactly what the calibration is on this, but it is extremely low.

On the lower left-hand side, what you are looking at is an example of another sensor, which is a one photon counter. What this sensor does, it works in either a passive or an active mode. This is in an active mode where we have used a source to illuminate a dish. This is a satellite dish. It was actually illuminated from above from an altitude of about 1,000 meters. What we are looking at are the returns, the statistical returns from the photon system that came back.

The technical community now has the ability to count photons one photon at a time and to do so in a time-resolved way with a resolution of less than 100 picoseconds. What that means is that we can now get not only a two-dimensional representation of what a view is, but also, with that time-resolution, you can build up a three-dimensional representation as well. What you are looking at, the reason you can get the pictures from the bottom side, it is through a technique called ballistic photons. That is, you know when the source was illuminated, and you can calculate, then, on the return of the photons the path of each of those individual photons. So, basically, what this is saying is that you can build up three-dimensional images now. You can, in a sense, look behind objectives. It is not always true, because you need a backlight for the reflection. Again, there is more to sensors than visual imagery. That is kind of the fun part of this, as far as the toys and being able to look at the different things we are collecting.

The question then arises, how do we handle all this data. Finally, how do we go ahead and fuse it together. I would like to make the case of—I talked about a paradigm earlier about one way to collect data or to build bigger pipes and to make bigger computers to try to run through with different kinds of algorithms to assess what is going on. Another way to do this is to let the power of the computer actually help us itself by fusing the sensor with the computer at the source.

It is possible now, because of a technique that was developed about 10 years ago in field programmable gate arrays—that is, being able to hardwire instructions into the registers itself, to achieve speeds that are anywhere from a hundred to a thousand times greater than what you can achieve using software, that is because you are actually working with the hardware instead of the software, to execute the code. Since these things are reprogrammable—that is, they are reconfigurable computers—then you can do this not only on the fly, but also, what this means is that you can make the sensors themselves part of the computation at the spot, and take away the need for such a high bandwidth for getting the data back to some kind of unique facility that can help process the information.

Plus, what this gives you the ability to do is to be able to change these sensors on the fly. What I mean by this is, consider a technology such as the software radio. All you know is that radio basically is a receiver, and then there is a bunch of electronics on the radio to change the capacitance, the induction. All this does, the electronics, really, is to change the bandwidth of the signal, to sample different bits in the data stream, and that type of thing. It is now possible to go ahead and make—because computers are fast enough—and especially reconfigurable computers—to go ahead and make a reconfigurable computer that can do all the stuff that the wires and years ago the tubes and, now, the transistors do.

What this means is that, if you have a sensor, say, like a synthetic aperture array, and you want to change the nature of the sensor because it is detecting thing in an RF, to say an infraredometer, you can do it on the fly. What this provides people the power with

is that—or if you have platforms that you are going to put out in the field, be they ground-based, sea-, air- or space-based, you don't have to figure out, 5 to 10 years ahead of time, what these sensors are going to be. That is, if it is going to be an RF sensor, then all that is important is the reception of this, and the bandwidth that you have for the reconfigurable computer. You can change the nature, the very nature, of the sensor on the fly.

That is a long explanation of what this chart is, but what this shows is that, by putting the power of the computation next to the sensor, then what you do is greatly reduce the complexity of the problem and the data streams that you need. You are still going to need the ability to handle huge amounts of data, because remember, I was talking about 10^{14} different nodes. What this does is help solve that problem. I don't want to go too much longer on that, but you all know about the distributed arrays. I talked a little bit about the power of that earlier. Basically, having a non-centralized access to each of these, once you have the positions of these things nailed down in a way—say, by using GPS or, even better, differential GPS—then they don't even have to be fixed, if you can keep track of them.

What is nice about distributed networks is that every node on this should automatically know what every other node knows, because that information is transmitted throughout. So, it degrades very gracefully. What this also gives you the power to do is not only to take information in a distributed sense, but also, if you know the position of these sensors well enough, then you will have the ability to phase them together, and to be able to transmit.

What this means is, if you have very low transmitters in here at each of these nodes, say, even at the watt level, by phasing them together, you get the beauty of phasing, if you can manage to pull this off. It is harder to do at the shorter wavelengths, but at the longer wavelengths, it is easier to do. Once you have all this data, how are you going to move it around in a fashion where, if it is intercepted, then you know that it is still secure? When using new technologies that are starting to arise, such as quantum key distribution, this is really possible. For example, two and only two keys are created, but the keys are only created when one of the wave functions is collapsed, of the two keys that exist. This is something that arises from the EPR paradox—Einstein, Polinski, Rosen — and I would be happy to talk to anybody after this about it. It involves quantum mechanics, and it is a beautiful subject, but we don't have really too much time to get into it. Anyway, keys have been transmitted now 10 kilometers here in the United States, and the Brits have, through a collaboration, I think, with the Germans, have transmitted keys up to, I think, 26 kilometers through the air. We also have the ability to use different technologies to transmit the data that aren't laser technologies. Why laser technologies? Well, laser energy is very opaque to certain atmospheric conditions.

We know that RF can transmit, especially where there are holes in the spectrum, through the atmosphere. So, to be able to tap into regions of the electromagnetic spectrum that have not been touched before, in the so-called terahertz regime—this is normally about 500 gigahertz up to about 10 terahertz—is possible now, with advances in technology. What I have shown here is something that was initially developed at SLAC. IT is called the clistrino, which is based on their clistron RF amplifier.

The key thing here is that the electron beam is not a pencil beam but, rather, a sheet beam which spreads out the energy density. So, you don't have a lot of the

problems that you used to have for the older-type tubes. So, we talked a little bit about handling and we talked about the sensors. Let me talk about the data mining.

What is envisioned here is having some kind of what we call distributed data hypercube. That is, it is a multidimensional cube with all sorts of data on it. On one axis, you have probably heard in the news, this is Poindexter's push at DARPA, tapping everything into credit cards on one axis to airlines transactions on another axis, another axis being perhaps telephone intercepts, another axis being RF emissions, another axis perhaps visual information or human intelligence. Tapping into that, and being able to do so in a legal way—because there are large legal implications in this, as well as things that are prohibited by statute, as it turns out, especially when you talk about intelligence databases—to be able to render that using different types of algorithms and then be able to compute that and then feed that back in does two things.

First of all, it is to give you a state of where you are at today and, second of all, it is to try to predict what is going to happen. I will give you a very short example of about five minutes here of something that is going on, that is taking disparate types of databases to try to do something like this. So, that is kind of the mining problem, and there are various ways to mine large types of data. Let me talk to you about two examples here where, again, you are not relying just on the algorithm itself, but you are relying on the computer to do this for you.

This is an example of a program called GENIE that, in fact, was developed by a couple of the individuals that are here in this room. It is a genetic algorithm that basically uses an array of kernels to optimize the algorithm that you are going to render, and it does so in a sense where the algorithm evolves to find you the optimal algorithm. It evolves because, what you do is, you tell the algorithm or the onset, or tell the computer, what are the features, or some of the features, that you are looking for.

On the left-hand side, this is an example of—this is an aerial overhead here of San Francisco Bay. What you want to do is look for golf courses on this. Let's say that we paint each of the known golf courses green and then we tell the algorithm, okay, go off and find those salient characteristics which define what a golf course is. Now, as it turns out, it is very tough to do because there is not a lot of difference between water and golf courses. A lot of golfers, I guess, think that is funny.

The reason is because of the reflectivity, the edge of the surface, which has no straight lines in it. So, it is kind of tough. Especially when you are talking about something like global situational awareness, if you find a golf course, you have got to make absolutely sure that it is a golf course. You can imagine that this could be something else that you are going after. What you do is, you let the computer combine those aspects, especially if you have hyperspectral data. That could be information in the infrared that may show the reflectivity, for example, of chlorophyll. It could be information about the edges. The computer assembles this data, using again, this basis of kernels that you have, and comes up with and evolves a unique optimized algorithm to search for these things.

Now, this is different from neural nets because you can actually go back through and you can, through a deconvolution process, find out what the algorithms are, and it does make sense, when you look at it. Basically, it is by using the power of computation to help you itself, what you do is, you are reducing the complexity of the problem. Now, if you have taken that, you can go to the next step, and you can accelerate that algorithm,

not just to run on a computer, but if you hardwire it into a reconfigural computer, with that floating point gate array I told you about, what I will do is, I will show you an example of how you can do things on a—I don't know if I am going to be able to do this.

Say that you have streaming data coming in as video arrays. What is occurring here is that we have asked it to locate the car that you will see—that is the one in the green on the right-hand side—at rates that are approaching 30 frames per second. The point is that, by marrying different types of technology, we will be able to help out and help you determine to do things in a near-real-time manner. Other techniques for pulling very small or high—very low, I should say, signal-to-noise data out. What I have shown here is pulling some data out by using a template on the lower left-hand side. On the right-hand side, it is looking at some spectrographic data, and being able to pull out some chemical species. So, these are all examples of data fusing techniques.

Let me really wrap this up and leave time for some questions here. The whole goal of this is to be able to synthesize what we call a complete view of a situation from disparate databases. Just trying to pull things together to give people a wide range of ability, be it from the national scene to the person, it could be a law enforcement officer, who may only want to know things that are happening 30 or 40 feet around him. What I have done is, I have put double-headed arrows on there, to show that there should be a capability for those sensors to be tasked themselves, which kind of makes the people who run the sensors kind of scared.

On the other hand, if you don't have that ability, then you don't have the ability to allow feedback into the system. There is an example of something going on like this, that is not nearly as complex, where some Forest Service data is being combined on wild fires, where they start, how they originate, combine that with climatology data, looking at soil, wetness, wind data—what else, soil information. Then, combine that with Department of Justice data on known arsonists, where they started forest fires before and where they are located now.

What this is attempting to do, with this very small database, is to combine these disparate databases to predict—first of all, to give you the situation where we are now, and then perhaps to be able to predict if an arsonist were to strike at a particular place. So, this is a real, no kidding, national security problem, forest fires, because you can imagine—well, we, at Los Alamos ourselves, had devastating fires two years ago, that nearly wiped out a national lab with that. So, by using small test problems like this, we will show not only the larger problems that will arise, but also hopefully that doing something like this is not merely a pipe dream.

In conclusion, I think it has been determined that a need for a global situational awareness really exists. Again, this is a synthesis and an integration of space situational awareness, battlefield situational awareness, law enforcement situational awareness, to be able to be used in anti-terrorist activities. A lot of work needs to be done. This is not a one-lab or a one-university project. It is something that I think will really tap the S&T base of the nation. The key here is seamless integration. It is the data and it is not really the sensors, but it is integrating it in a way to be able to show it in a way that makes sense in a seamless sense. So, that is the talk. It is accelerated about 10 minutes faster than I normally give it. Might I answer any questions you might have?

AUDIENCE: In that golf course example, you actually had trained data, spectral data from real golf courses, and trained the model on that and predicted other golf

courses?

MR. BEASON: Actually, what we did on that was, we had that picture and we located—we knew where the golf courses were, and we painted those with whatever technique we used, and then let the computer itself use that as its training aid. So, we allowed it to pick out—we didn't give it any information a priori as to what might be a course. We let it decide itself. You did see some errors. So, there is an effort to push down the number of errors involved.

Also, we were able to find—for example, we went back and looked at some forest fire data that had occurred around Los Alamos, and what we were able to find was that there were three instances where the Forest Service had started fires before and they had not told us about it, and we were able to pick those out. That really ticked us off, when we found that out.

MS. KELLER-MC NULTY: I would like to point out that Nancy David and James Theiler over here are the GENIE genies, if some people have some questions about that.

AUDIENCE: I was going to ask a question about this term, data fusion. Is that the same as data assimilation?

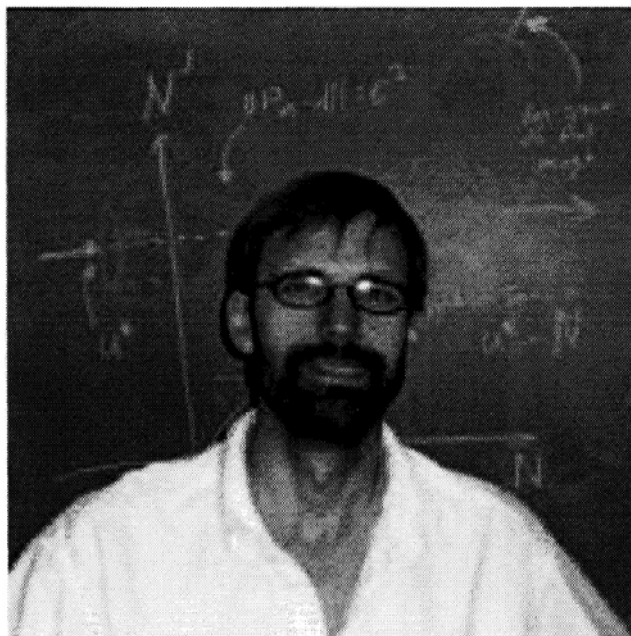
MR. BEASON: I am not sure what you mean by data assimilation. I think I know. Fusion, that is all in a smart way, because you can't just bring things together. You have to know what the context is.

Kevin Vixie

Incorporating Invariants in Mahalanobis Distance-Based Classifiers: Applications to Face Recognition

[Transcript of Presentation](#)

[Technical Paper](#)



BIOSKETCH: Kevin Vixie is a mathematician in the computational science methods group at Los Alamos National Laboratory. His research interests are in inverse problems; image analysis and data-driven approximation; computation; and modeling. More specifically, he is interested in the following main areas: data analysis techniques inspired by ideas from partial differential equations, functional analysis, and dynamical systems; nonlinear functional analysis and its applications to real world problems; geometric measure theory and image analysis; high dimensional approximation and data analysis; and inverse problems, especially sparse tomography. The problems he is interested in tend to have a strong geometrical flavor and a focus not too far from the mathematical/real data interface.

TRANSCRIPT OF PRESENTATION

MR. VIXIE: Thanks for your patience. So, the problems we are interested in range over a lot of different kinds of data, including hyperspectral data.

The problem we chose to look at, for the methods I am going to tell you about, is space data, because not only has it been worked on a lot and it is hard, but there are nice sets of data out there, and there is also some way to make very nice comparisons among a bunch of competing algorithms.

The people involved in this are myself, Andy Frazier, Nick Hengartner, who is here, Brendt Wohlberg. A little more widely, there is a team of us that is formulating that includes also other people who are here, like Peter Swartz, James and, of course, Nick.

The big picture is that we have these very large data sets. To do the kind of computations that we need to do, we need to do dimension reduction. The classical, of course, is PCA, and we like to look at non-linear variance of that approach. The challenge we chose to address was how to build metrics which are invariant to shifts along surfaces in the image space which represent changes of, like scaling, rotation, etc., that don't change the identity, but they do change the image.

This is preliminary work, like a couple of months of work, but the five-dimensional space we are working on is represented by the shift in X , shift in Y , scale in X , scale in Y and rotation. The next steps are to look at three-dimensional rotation and change elimination.

So, the prior work that we believe is relevant is on the order of 160 papers that we looked through. This morning, out of curiosity, I did a site search, also an INSPEC search, just to see what would come up when I typed in "face recognition." Isis gave me about 900 hits and INSPEC gave me about 3,200. That is not too far from believable that 5 percent may be kind of poor.

So, the couple of papers that we found that we thought were of help to us was this eigenfaces versus Fisher faces, and then a paper on using tangent approximations to increase the classification rates for character recognition. The data that we used was the FERET data from the FERET database.

The test bed we chose was the Colorado State University test bed. We chose this because, in this test bed, they have a very uniform way of testing many different algorithms. So, including ours, there are 13 algorithms there, so with ours, it was 14. There is a standard pre-processing that is done, and everybody trains on the same data with the same pre-processing and there is a standard test. That allowed us to compare, in a very fair and unbiased manner different algorithms.

Here is a sampling of, this is the performance. I will explain this a little bit. The idea is that these algorithms train on the training data, and then they end up with an algorithm that is a trained algorithm that you can hand data to, and it will hand you back a distance matrix.

So, if we hand you 100 faces, you end up with about 100 matrices, and this is the distances between the faces. Then, based on that, you can rank results and say, if I have two images of the same person, what I would like to happen is that, in the column corresponding to image one of this person, the image two is ranked number one. It is the closest.

This is sort of a Monte Carlo experiment, a histogram of how often, if you hand

an algorithm 100 faces, it is going to get the right person in rank one. So, it ranges. The bottom is the PCA base, putting in distance, which I will explain a little bit, and the top is this linear discriminate analysis based on a correlation angle.

AUDIENCE: So, if you train on 100 faces, why doesn't it cover those 100 faces?

MR. VIXIE: First of all, it is a different training and testing, and I will get to the ethos of how many it is trained on.

Now, the pre-processing is done to all the training data and all the test data; you shift, scale and rotate based on eye coordinates. So, these slides aren't actually the very latest. I made some changes last night after I came and they are not showing.

So, this is two pictures of the same guy. There is a different facial expression. You can also see at the bottom, these are the pre-process. These are the raw images.

So, you scale, shift and rotate based on eye coordinates and location. Then you mask. You use an elliptical mask. Then you equalize the histogram. This is standard in image processing, and then you shift and scale until you get a mean zero, standard deviation one.

For training, 591 images were used, and that was 197 individuals, three pictures each. Just a little flavor of what these different methods do, for PCA, of course, what you do is simply take the images, form the empirical covariance matrix, do the Haganie composition, and then project down to some specified number, and here it was 60 percent. So, you end up with a 354-dimensional space.

Now, that is actually a very big reduction in dimension, because the images are on the order of maybe a quarter-million pixels. So, it is a big reduction in dimension.

The LDA is just—probably everybody here knows about this—the Fisher basis. In this case, you have a knowledge about what is the within-class variance and what is the between-class variance. So, you know what differences correspond to differences between the same individual and what differences correspond to difference between different individuals. You try to pick a basis that can differentiate between-individual differences and within-individual differences, optimally.

For the linear discriminate analysis, we simply take the within-class. So, there are 591 pictures. So, you have the differences. For each set of three images, you take all those differences, and you throw them together, build a covariance matrix, and that gives you the within-class covariance matrix.

For the Euclidian distance, this is very simple. You just project into the PCA basis, and then take the coefficients. This is image A and image B. The distance between those images is just simply the difference of the coefficient squared sum dot square root. That corresponded to actually the worst performance in that set of algorithms.

The correlation that was the best is an angle-based classification. Here, after you project it on an LDA basis, you take the mean and subtract them out and simply take an inner product. Subtract that from one. So, if the end product is maximized, then they are close. You want that to be a distance. You subtract that from one and it turns into something like a distance.

So, what did we do? Well, what we did was dictated not only by some interest in faces, but more in an interest in many kinds of data, including hyperspectral, voice, etc.

So, we said, well, we would like to know, in a principal way, how to include in the distance metric the notion that there are differences that don't really matter.

So, the blue curve represents the low dimensional manifold. In our case, it is going to be five-dimensional because there are five parameters we are modifying the image by.

That represents sort of an orbit of the face in the image space that represents all the same individual. So, what I would like to do is build a metric that says, if there are two points—a point here and a point there—there is no distance between them, if they are the same individual. So, these manifolds are non-linear and it is not necessarily easy to compute them. So, as sort of a first thing, we do a linear approximation.

So, this just represents the tangent manifold approximation to that surface. So, what we are going to now is, we are going to try to modify this covariance matrix, which uses a kernel to build the distance, or the inverse of that kernel, modify it with the knowledge that differences along this direction don't matter.

Now, of course, that is not quite true, because if this is highly curved, and if I go too far out, that linear approximation isn't very good. So, we want to use the second-order information, which we can also compute, to limit how widely we let ourselves move along this direction with no penalty. We built this new covariance matrix, and it enabled us to use the classification method.

Now, notice that the key feature here is that, in fact, when you do this, you end up with different modifications at each point. Even though we start with the same within-class covariance, we end up with different modifications of that. So, it is a localized modification of something that seems to be non-linear.

I will come back to some of the details and show you the results. So, this is the same graphic as before with some faces removed. So, this was the worst, this is the best, this is what we got. What we got was an improvement—the next-to-the-last curve was what we did without the tangent modification. So, we got this big improvement—again, our performance is untuned, and the thing you have to understand about the face stuff is that often tuning makes a very big difference.

So, we were encouraged because untuned performance, and then we got this big jump. Sort of the next step is to add that to the angle base metric, in which you get an improvement there. Again, really, the real goal for this isn't to do faces the best of anybody. It is really to have a tool that is flexible, fast, and generally applicable.

Here are some details. There is this nice picture, again. I like this picture. So, you imagine there are a bunch of individuals and you have a space of images. So, this plane I am drawing represents the space of images.

If you have 100 by 100 pixelated images, you get 10,000 in actual space. Then, I imagine that I have a space of parameters that controls—that this individual controls where I am on this manifold.

This is the transformation that maps you from individual and parameter to the image space. What you can do is—to make a sensible little explanation—assume the image equals this $\tau(F, \theta)$ but I have got some noise. I am going to assume that is distributed normally, and that the θ is also distributed normally, which is a helpful fiction that is not too far off.

You can sort of convince yourself that it might be reasonable because all of this face recognition is done where you are first trying to locate them the same but you make at least sub-pixel errors. So, you might expect that the errors you make are going to have something that might be something like a Gaussian distribution.

So, if we expand τ in a Taylor series, then we get this. In our case, θ is a five-dimensional vector, and G is going to be the number of dimensions you are using by five, and then we have the second-order term here.

This is just a note that, because of linearity, if θ distributes according to this distribution, then $GF\theta$ is just that.

So, let me give an example of the shifts. Now, when I show this to people first they say, what is the big deal? You are just shifting it. The deal is that in images this is non-linear. The shift in image is up and the pixel space is non-linear. I will give it to you a little later. You want to convolve these images with the kernels to smooth them before you take the derivative.

This is the image you start with and then you try to shift it up nine pixels, and then this is the second-order of correction.

Now, notice, you can see artifacts here, not that well, but there are some artifacts. It gets a little too light here and dark there. So, there are some artifacts with this second-order of correction, because we have shifted it a little bit past the validity of this combination of kernels and image. So, we have to tune the sort of combination of how big a kernel we use to smooth it and how far we shift it.

AUDIENCE: [Comment off microphone.]

MR. VIXIE: Yes, so let's take an example. If you have a very simple image and you have just one pixel that is black and everything else is white, and you shift it once, we are going to do it with quiet differences.

You shift it one pixel. Now, take the difference between those pixels. One is the new spot and minus one is the old spot. Add that to the old image to get the new image. It is shifted one pixel. Now, I would like a five-pixel shift. So, how about if you multiply that little difference vector by five? Are you getting a five-pixel change? No. Does that make sense now?

AUDIENCE: Yes, but that is not usually what— [off microphone.]

MR. VIXIE: What I mean is I want to shift things— [off microphone.]

AUDIENCE: When you say linear, do you mean linear operating on the image, not on the location?

MR. VIXIE: Yes. Yes. Okay, some details. So, it is a second-order error, if everything after zero and the first-order term is small, then the distribution for S and F is going to be normal, according to this distribution. It is going to have a mean at τ after zero, and this covariant.

So, the maximum likelihood of classification is simply this minimizing over I , this distance, where the kernel is now modified with the derivative.

Okay, so, the question is, how do you pick the θ . We have conflicting goals here. Do you want the $\sigma(\theta)$ large, do you want to move out on that approximating manifold, but you want the second-order to be small?

So, the solution is to maximize the determinant of $\sigma(\theta)$, while constraining the second-order error.

If you do that, it is fairly straightforward. You end up that $\sigma(\theta)$ equals α times the inverse of this modified second-order term.

What we have done here is, we have modified it so that it looks strictly positive. You might think that, okay, this weights all pixels evenly. So, this is the sum of the pixels.

The way to view this second-order term is as a third-order tangent because at each pixel you have, in our case, a five-by-five matrix.

So, what I would like to do is actually make it so that the second-order errors are discounted, or it is not paid as much attention to if those second-order errors are in directions which the a within, or the within-class covariance is ignoring anyway.

So, then, what we do is, we decompose the within-class variances and we get the components of this tangent in each of those directions, and then simply weight those, because this final within-class covariance by the inverse square root of the I — [off microphone.]

Now we have the sense that, if there is a second-order error but it is in a direction that the within-class covariance is totally ignoring, then we can go much farther out. Again, we used the switch and we the constrained optimization before.

Okay, now just a couple of notes. The trick that makes this thing work quickly is the fact that, when you do the smoothing and just taking derivatives, so you can transfer the derivatives to the kernel, and then you can use the fact that even when you get—in this case, with the shift, you don't get a spatially varying kernel, but in other cases, especially when you are doing the second-order derivatives, you get a lot of spatially varying kernels, and that is hard to do with the FFT.

So, we simply pre-process the image by multiplying by that spatially ordering term, and then we can do the FFT. That is what makes this work very quickly.

I guess that is all the slides I have here. I had more in here. The main points that I wanted to close with are that this method of looking at modeling data is generally applicable. We are not interested, for a couple of reasons—legal reasons— [off microphone].

I might feel bad about—I don't think I am going to do this, but if I created something that was perfect at spatial recognition, maybe I would have a conscience issue.

At any rate, this is something that is actually applicable to a very wide range of data. It is fast to compute and it is headed toward the use of knowledge that we have about what sorts of transformation of identity or classifications— [off microphone.]

The things we want to do next would be, first of all, add the change and modifications to the angle base. Also, look at more detailed, or more difficult, transformations, like three-dimensional locations would be much more difficult. Lighting is also an issue. Thank you.

MS. KELLER-MC NULTY: Are there some questions while John is getting set up?

AUDIENCE: How do you find the—[off microphone.]

MR. VIXIE: We let the CSU test bed do that, simply because we wanted our comparison to be absolutely unbiased with the other test cases. So, we used their algorithm. I didn't take care of that part. My job was the code to compute all the derivatives and second-order transforms. I didn't do that piece of it or recode that piece of it.

MS. KELLER-MC NULTY: Other questions? Do you want to make a comment on the data fusion, data assimilation question earlier?

MR. VIXIE: Yes, somebody asked about data fusion. Well, classically, when I think of data assimilation, I think of common filtering, something where you have a dynamic process and you want to get back to a state space, but that state space is not fully

measured.

I have a ten-dimensional space and I am taking one-dimensional measurements, and the idea is that over time I build up those one-dimensional measurements, and I know about the dynamic connections. Then I can get back to the state. I preserve the state. When I think of assimilation, I think of that. In fact, I think that is what is commonly meant when you say data assimilation. You can correct me if you know better.

Data fusion could include something like that. In essence, to do fusion, you might approach it this way, where you have an idea that there is some big invariant state space that is like hidden, and fusion is enabling you to get back to that hidden space.

AUDIENCE: [Off microphone.]

MR. VIXIE: Fundamentally, there is no difference in the way I stated it. I think the way people often think about it, and what you find written about it is quite different. I mean, I have a way of thinking about it that makes sense to me.

INCORPORATING INVARIANTS IN MAHALANOBIS DISTANCE BASED CLASSIFIERS: APPLICATION TO FACE RECOGNITION

Andrew M.Fraser
Portland State University and Los Alamos National Laboratory
Nicolas W.Hengartner, Kevin R.Vixie, and Brendt E.Wohlberg
Los Alamos National Laboratory
Los Alamos, NM 87545
USA

Abstract—We present a technique for combining prior knowledge about transformations that should be ignored with a covariance matrix estimated from training data to make an improved Mahalanobis distance classifier. Modern classification problems often involve objects represented by high-dimensional vectors or images (for example, sampled speech or human faces). The complex statistical structure of these representations is often difficult to infer from the relatively limited training data sets that are available in practice. Thus, we wish to efficiently utilize any available *a priori* information, such as transformations of the representations with respect to which the associated objects are known to retain the same classification (for example, spatial shifts of an image of a handwritten digit do not alter the identity of the digit). These transformations, which are often relatively simple in the space of the underlying objects, are usually non-linear in the space of the object representation, making their inclusion within the framework of a standard statistical classifier difficult. Motivated by prior work of Simard et al., we have constructed a new classifier which combines statistical information from training data and linear approximations to known invariance transformations. When tested on a face recognition task, performance was found to exceed by a significant margin that of the best algorithm in a reference software distribution.

I. INTRODUCTION

The task of identifying objects and features from image data is central in many active research fields. In this paper we address the inherent problem that a single object may give rise to many possible images, depending on factors such as the lighting conditions, the pose of the object, and its location and orientation relative to the camera. Classification should be invariant with respect to changes in such parameters, but recent empirical studies [1] have shown that the variation in the images produced from these sources for a single object are often of the same order of magnitude as the variation between different objects.

Inspired by the work of Simard et al. [2] [3], we think of each object as generating a low dimensional manifold in image space by a group of transformations corresponding to changes in position, orientation, lighting, etc. If the functional form the transformation group is known, we could in principle calculate the entire manifold associated with a given object from a single image of it. Classification based on the entire manifold, instead of a single point leads to procedures that will be invariant to changes in instances from that group of transformations. The procedures we describe here approximate such a classification of equivalence classes of images. They are quite general and we expect them to be useful in the many contexts outside of face recognition and image processing where the problem of transformations to which classification should be invariant occur. For example, they provide a framework for classifying near field sonar signals by incorporating Doppler effects in an invariant manner. Although the procedures are general, in the remainder of the paper, we will use the terms *faces* or *objects* and *image classification* for concreteness.

Of course, there are difficulties. Since the manifolds are highly nonlinear, finding the manifold to which a new point belongs is computationally expensive. For noisy data, the computational problem is further compounded with the uncertainty in the assigned manifold.

To address these problems, we use tangents to the manifolds at selected points in image space. Using first and second derivatives of the transformations, our procedures provide substantial improvements to current image classification methods.

II. COMBINING WITHIN CLASS COVARIANCES AND LINEAR APPROXIMATIONS TO INVARIANCES

Here we outline our approach. For a more detailed development, see [4]. We start with the standard Mahalanobis distance classifier

$$\hat{k}(Y) = \underset{k}{\operatorname{argmin}} (Y - \mu_k)^T C_w^{-1} (Y - \mu_k),$$

where C_w is the within class covariance for all of the classes, μ_k is the mean for class k , and Y is the image to be classified. We incorporate the known invariances while retaining this classifier structure by augmenting the within class covariance C_w to obtain class specific covariances, C_k for each class k . We design the augmentations to allow excursions in directions tangent to the manifold generated by the transformations to which the classifier should be invariant. We have sketched a geometrical view of our approach in Fig. 1.

Denote the transformations with respect to which invariance is desired by $\tau(Y, \theta)$, where $Y \in \mathcal{Y}$ and $\theta \in \Theta$ are the image and transform parameters respectively. The second order Taylor series for the transformation is

$$\tau(Y, \theta) = \tau(Y, 0) + V\theta + \theta^T H\theta + R,$$

where R is the remainder,

$$(V_k)_i = \left. \frac{\partial \tau(Y_k, \theta)}{\partial \theta_i} \right|_{\theta=0}, \text{ and } (H_k)_{i,j} = \left. \frac{\partial^2 \tau(Y, \theta)}{\partial \theta_i \partial \theta_j} \right|_{\theta=0}.$$

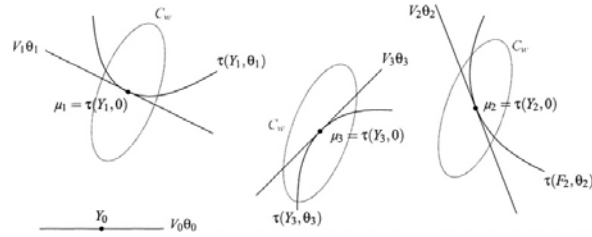


Fig. 1. A geometrical view of classification with augmented covariance matrices: The dots represent the centers μ_k about which approximations are made, the curves represent the true invariant manifolds, the straight lines represent tangents to the manifolds, and the ellipses represent the pooled within class covariance C_w estimated from the data. A new observation Y is assigned to a class $k \in \{1, 2, 3\}$ using $\hat{k}(Y) = \operatorname{argmin}_k (Y - \mu_k)^T C_k^{-1} (Y - \mu_k)$. The novel aspect is our calculation of $C_k = C_w + \alpha \tilde{C}_k$ where α is a parameter corresponding to a Lagrange multiplier, and \tilde{C}_k is a function of the tangent and curvature of the manifold (from the first and second derivatives respectively) with weighting of directions according to relevance estimated by diagonalizing C_w .

We define

$$C_k = C_w + \alpha V_k C_{\theta,k} V_k^T, \quad (1)$$

Where $C_{\theta,k}$ is a $\dim(\Theta) \times \dim(\Theta)$ matrix. We require that $C_{\theta,k}$ be non-negative definite. Consequently $V_k C_{\theta,k} V_k^T$ is also non-negative definite. When C_k^{-1} is used as a metric, the effect of the term $V_k C_{\theta,k} V_k^T$ is to discount displacement components in the subspace spanned by V_k , and the degree of the discount is controlled by $C_{\theta,k}$. We developed [4] our treatment of $C_{\theta,k}$ by thinking of θ as having a Gaussian distribution and calculating expected values with respect to its distribution. Here we present some of that treatment, minimizing the probabilistic interpretation. Roughly, $C_{\theta,k}$ characterizes the costs of excursions of θ . We choose $C_{\theta,k}$ to balance the conflicting goals

Big: We want to allow θ to be large so that we can classify images with large displacements in the invariant directions.

Small: We want $\theta^T H \theta \in \mathcal{Y}$ to be small so that the truncated Taylor series will be a good approximation.

We search for a resolution of these conflicting goals in terms of a norm on θ and the covariance $C_{\theta,k}$. For the remainder of this section let us consider a single individual k and drop the extra subscript, i.e., we will denote the covariance of θ for this individual by C_θ .

If, for a particular image component d , the Hessian H_d has both a positive eigenvalue λ_1 and a negative eigenvalue λ_2 , then the quadratic term $\theta^T H \theta$ is zero along a direction e_0 which is a linear combination of the corresponding eigenvectors, i.e. $(\gamma e_0)^T H_d (\gamma e_0) = 0 \quad \forall \gamma$. We suspect that higher order terms will contribute to significant errors when $\gamma \geq \min(|\lambda_1|^{1/2}, |\lambda_2|^{1/2})$ so we eliminate the canceling effect by replacing H_d with its *positive square root*, i.e. if an eigenvalue λ of H_d is negative, replace it with $-\lambda$. This suggests the following *mean root square* norm

$$|\theta|_{H_{\text{mrs}}} \equiv \sqrt{\sum_{d=1}^N \theta^T \sqrt{H_d} H_d \theta}. \quad (2)$$

Consider the following objection to the norm in Eqn. (2). If there is an image component d which is unimportant for recognition and for which H_d is large, e.g. a sharp boundary in the background, then requiring $|\theta|_{H_{\text{mrs}}}$ to be small might prevent parameter excursions that would only disrupt the background. To address this objection, we use the eigenvalues of the pooled within class covariance matrix C_w to quantify the importance of the components. If there is a large within class variance in the direction of component d , we will not curtail particular parameter excursions just because they cause errors in component d .

We develop our formula for C_θ in terms of the eigendecomposition

$$C_w = \sum_d e_d \lambda_d e_d^T$$

as follows. Break the $\dim(\Theta) \times \dim(\gamma) \times \dim(\Theta)$ tensor H into components

$$H_d \equiv e_d^T H \quad (3)$$

Then for each component, define the $\dim(\Theta) \times \dim(\Theta)$ matrix

$$H_d^+ \equiv \sqrt{(H_d)^T H_d}, \quad (4)$$

and take the average to get

$$\bar{H} \equiv \sum_d H_d^+ |\lambda_d|^{-1/2} \quad (5)$$

Define the norm

$$|\theta|_{\bar{H}} \equiv \sqrt{\theta^T \bar{H} \theta}.$$

Given H and C_w , one can calculate \bar{H} using Equations (3), (4), and (5). Then by using the determinant $|C_\theta|$ to quantify goal **Big**: (allow θ to be large) and using $\mathbb{E}|\theta|_{\bar{H}}^2$ to quantify goal **Small**: (keep $\theta^T H \theta \in \mathcal{Y}$ small), we get the constrained optimization problem:

Maximize the determinant $|C_\theta|$

Subject to

$$\mathbb{E}|\theta|_{\bar{H}}^2 \leq \gamma, \quad (6)$$

where γ is a constant.

The solution to the problem is

$$C_\theta = \alpha (\bar{H})^{-1} \quad (7)$$

where α , which is a function of γ , is a constant that balances the competing goals.

To verify that Eqn. (7) indeed solves the optimization problem, note:

$$\begin{aligned} \mathbb{E}|\theta|_{\bar{H}}^2 &= \mathbb{E} \left(\sum_{k,l} \theta_k \bar{H}_{k,l} \theta_l \right) \\ &= \sum_{k,l} \bar{H}_{k,l} \mathbb{E}(\theta_k \theta_l) \\ &= \text{Tr}(\bar{H} C_\theta). \end{aligned}$$

In the coordinates that diagonalize \bar{H} Eqn. (6) only constrains the diagonal entries of C_θ . Of the symmetric positive definite matrices with specific diagonal entries, the matrix that has the largest determinant is simply diagonal. So C_θ and \bar{H} must be simultaneously diagonalizable, and the problem reduces to

$$\begin{aligned} \text{Maximize: } & \prod_{l=1}^{\dim(\Theta)} \sigma_l \\ \text{Subject to: } & \sum_{l=1}^{\dim(\Theta)} \sigma_l h_l = \gamma. \end{aligned}$$

The Lagrange multipliers method yields Eqn. (7).

Summary: Given a new image Y , we estimate its class with $\hat{k}(Y) = \underset{k}{\text{argmin}} (Y - \mu_k)^T C_k^{-1} (Y - \mu_k)$

where $C_k = C_w + \alpha V_k C_{\theta,k} V_k^T$. We have derived the parameters of this classifier by synthesizing statistics from training data with analytic knowledge about transformations we wish to ignore.

III. FACE RECOGNITION RESULTS

We tested our techniques by applying them to a face recognition task and found that they reduce the error rate by more than 20% (from an error rate of 26.7% to an error rate of 20.6%). We used an analytic expression for transformations in image space and developed procedures for evaluating first and second derivatives of the transformations. The transformations have the following five degrees of freedom:

- Horizontal translation
- Vertical translation
- Horizontal scaling
- Vertical scaling
- Rotation

To implement the test, we relied on the FERET data set [5] and a source code package from Beveridge et al. [6], [7] at CSU for evaluating face recognition algorithms.

Version 4.0 (October 2002) of the CSU package contains source code that implements 13 different face recognition algorithms, scripts for applying those algorithms to images from the FERET data set, and source code for Monte Carlo studies of the distribution of the performance of the recognition algorithms. Following Turk and Pentland [8], all of the CSU algorithms use principal component analysis as a first step. Those with the best recognition rates also follow Zhao *et al.* [9] and use a discriminant analysis. For each algorithm tested, the CSU evaluation procedure reports a distribution of performance levels. The specific task is defined in terms of a single *probe* image and a *gallery* of NG images. The images in the gallery are photographs of NG distinct individuals. The gallery contains a single *target* image, which is another photograph of the individual represented in the probe image. Using distances reported by the algorithm under test, the evaluation procedure sorts the gallery into a list, placing the target image as close to the top as it can. The algorithm scores a success at rank n if the target is in the first n entries of the sorted list. The CSU evaluation procedure randomly selects $NG \times 10,000$ gallery-probe pairs and reports the distribution of successful recognition rates as a function of rank.

Restricting the test data set to those images in the FERET data that satisfy the following criteria:

- Coordinates of the eyes have been measured and are part of the FERET data.
- There are at least four images of each individual.
- The photographs of each individual were taken on at least two separate occasions.

yields a set of 640 images consisting of 160 individuals with 4 images of each individual. Thus we use $NG=160$. Of the remaining images for which eye coordinates are given, we used a training set of 591 images consisting of 3 images per individual for 197 individuals. The testing and training images were uniformly preprocessed by code from the CSU package. In [6] the authors describe the preprocessing as,

“All our FERET imagery has been preprocessed using code originally developed at NIST and used in the FERET evaluations. We have taken this code and converted it...

Spatial normalization rotates, translates and scales the images so that the eyes are placed at fixed points in the imagery based on a ground truth file of eye coordinates supplied with the FERET data. The images are cropped to a standard size, 150 by 130 pixels. The NIST code also masks out pixels not lying within an oval shaped face region and scales the pixel data range of each image within the face region. In the source imagery, grey level values are integers in the range 0 to 255. These pixel values

are first histogram equalized and then shifted and scaled such that the mean value of all pixels in the face region is zero and the standard deviation is one.”

Each recognition algorithm calculates subspaces and fits parameters using the preprocessed training images and knowledge of the identity of the individuals in the images. Then, using those parameters, each algorithm constructs a matrix consisting of the distances between each pair of images in the testing set of 640 images. Thus, in the training phase, one can calculate the mean image, μ_k , of an individual, but in the testing phase, the algorithm has no information about the identity of the individuals in the images.

We developed three recognition algorithms: the first consists of the general techniques of Section II combined with minor modifications to fit the test task. We developed the second two algorithms after observing that the CSU algorithms based on angular distance perform best (see Fig. 2). In Section II we supposed that we would have several examples of each class, making an estimate of each class mean μ_k plausible, but for the task defined by the CSU evaluation procedure, we must simply provide 640×640 interimage distances.

The most obvious method for fitting our classification approach within this distance-based framework is to define the distance between image Y_k and Y_l as the Mahalanobis distance

$$d_0(Y_k, Y_l) = (Y_k - Y_l)^T C_k^{-1} (Y_k - Y_l).$$

Note, however, that this distance is not symmetric, since the augmented covariance is only relevant to one of the two images. Consequently, the symmetrized distance

$$d'_0(Y_k, Y_l) = \frac{d_0(Y_k, Y_l) + d_0(Y_l, Y_k)}{2}$$

is used for the distance matrix. After observing that of the CSU algorithms, those based on angular distance perform best (see Fig. 2), we developed two additional algorithms. The “Mahalanobis Angle” distance is

$$d_1(Y_k, Y_l) = \frac{Y_k^T C_k^{-1} Y_l}{\sqrt{Y_k^T C_k^{-1} Y_k} \sqrt{Y_l^T C_k^{-1} Y_l}},$$

with symmetrized version

$$d'_1(Y_k, Y_l) = \frac{d_1(Y_k, Y_l) + d_1(Y_l, Y_k)}{2}.$$

Instead of symmetrizing $d_1(Y_k, Y_l)$, we also define the symmetric distance

$$d'_2(Y_k, Y_l) = \frac{Y_k^T A_{kl}^{-1} Y_l}{\sqrt{Y_k^T A_{kl}^{-1} Y_k} \sqrt{Y_l^T A_{kl}^{-1} Y_l}}$$

where

$$A_{kl} = (C_k + C_l)^{-1}.$$

Evaluating each of the first two distances on the test set of 640 images takes about 30 minutes on a 2.2 GHz Pentium III. We found that the second distance performed better than the first. Because we estimated that evaluating the third distance would take about 160 hours, we instead implemented a hybrid, constructed by computing $d'_1(Y_k, Y_l)$ and then computing $d'_2(Y_k, Y_l)$ only for those distance below some threshold (further detail may be found in [4]).

Each of our algorithms operates in a subspace learned from the training data and uses an estimated covariance,

$$C_k = C_w + \alpha V_k \bar{H}_k^{-1} V_k^T,$$

associated with each image Y_k . We list the key ideas here:

- Use the training data (which includes image identities) to calculate raw within-class sample covariances, C'_w . Regularize the raw covariances as follows: (1) Do an eigenvalue-eigenvector decomposition to find $C'_w = Q \Lambda' Q^T$ (2) Sum the eigenvalues, $S = \sum_i \lambda'_i$. (3) Set $C_w = C'_w + \delta S \mathbb{I}$, which has no eigenvalues less than δS .
- Conceptually convolve the test image with a Gaussian kernel that has mean zero and variance

$$\begin{bmatrix} (\frac{h-1}{8})^2 & 0 \\ 0 & (\frac{h-1}{8})^2 \end{bmatrix},$$

where h is an adjustable parameter in the code that must be an odd integer. Change variables to transfer differentiation from the image to the kernel. Evaluate the matrices V_k and \bar{H}_k by convolving (using FFT methods) differentiated kernels with the image.

Thus α , δ , and h are three adjustable parameters in the estimate of C_k . We investigated the dependence of the performance on these parameters [4], and chose the values $\alpha=100$, $h=11$, and $\delta=0.0003$. Our experiments indicated that the classification performance was not sensitive to small changes in these choices.

Results are displayed in Fig. 2 and Fig. 3. Each of our algorithms performs better than all of the algorithms in the CSU package.

IV. CONCLUSIONS

We have presented techniques for constructing classifiers that combine statistical information from training data with tangent approximations to known transformations, and we demonstrated the techniques by applying them to a face recognition task. The techniques we created are a significant step forward from the work of Simard et al. due to the careful use of the curvature term for the control of the approximation errors implicit in the procedure. For the face recognition task we used a five parameter group of invariant transformations consisting of rotation, shifts, and scalings. On the face test case, a classifier based on our techniques has an error rate more than 20% lower than that of the best algorithm in a reference software distribution.

The improvement we obtained is surprising because our techniques handle rotation, shifts, and scalings, but we also preprocessed the FERET data with a program from CSU that centers, rotates, and scales each image based on measured eye coordinates. While our techniques may compensate for errors in the measured eye coordinates or weaknesses in

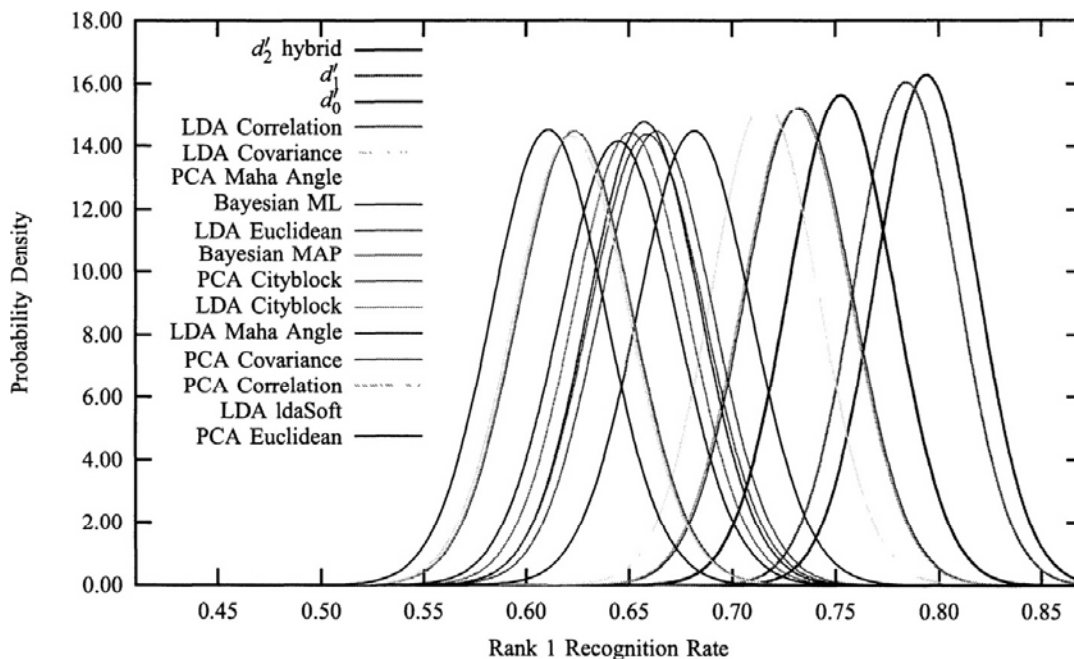


Fig. 2. Approximate distributions for the rank one recognition performance of the algorithms. For each algorithm, a Gaussian is plotted with a mean and variance estimated by a Monte-Carlo study. Note that the key lists the algorithms in order of decreasing mean of the distributions; the first three are the algorithms described in Section III, and the remainder are those implemented in the CSU software distribution.

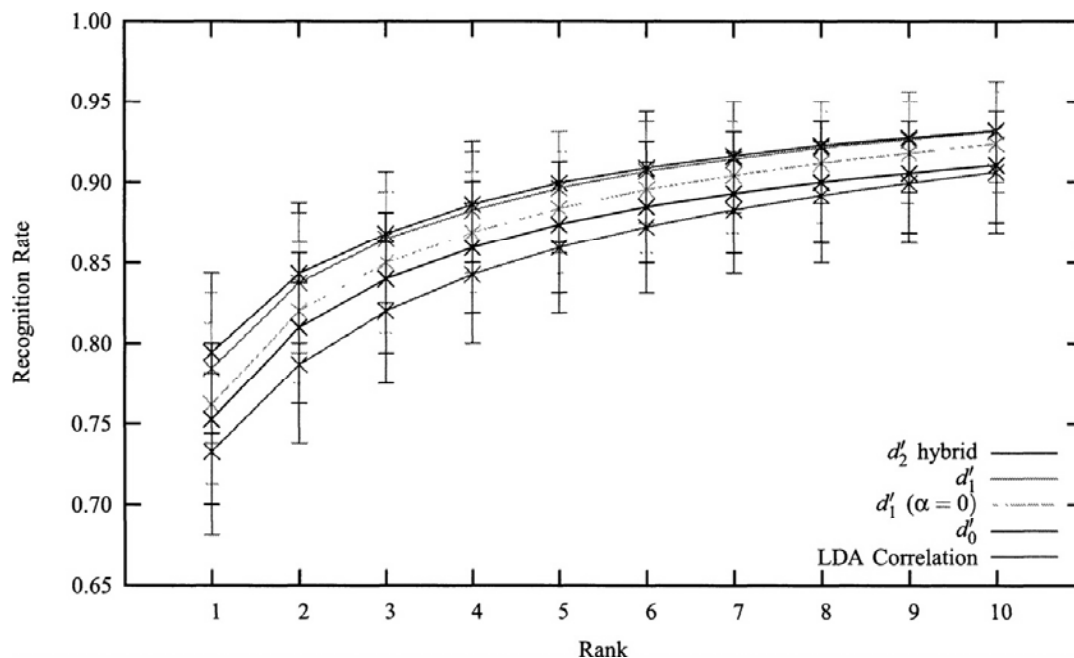


Fig. 3. The mean recognition rate and 95% confidence intervals as a function of rank for the following algorithms: d'_1 hybrid (the hybrid of d'_1 and d'_2), d'_1 (the symmetrized Mahalanobis Angle with tangent augmentation), $d'_1 (\alpha = 0)$ (the symmetrized Mahalanobis Angle with no tangent augmentation, illustrating the benefit obtained from the regularization of C'_w), d'_1 (the symmetrized Mahalanobis distance), and LDA Correlation (the best performing algorithm in the CSU distribution).

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

the preprocessing algorithms, we suspect that much of the improvement is due to similarities between the transformations we handle and differences between images. For example, a smile is probably something like a dilation in the horizontal direction.

V. ACKNOWLEDGMENT

This work was supported by a LANL 2002 Homeland defense LDRD-ER (PI K. Vixie) and a LANL 2003 LDRD-DR (PI J. Kamm).

REFERENCES

- [1] A.S.Georghiadis, P.N.Belhumeur, and D.J.Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, June 2001.
- [2] P.Y.Simard, Y.A. L.Cun, J.S.Denker, and B.Victorri, "Transformation invariance in pattern recognition—tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade*, G.B.Orr and K.-R.Muller, Eds. Springer, 1998, ch. 12.
- [3] P.Y.Simard, Y.A.Cun, J.S.Denker, and B.Victorri, "Transformation invariance in pattern recognition: Tangent distance and propagation," *International Journal of Imaging Systems and Technology*, vol. 11, no. 3, pp. 181–197, 2000.
- [4] A.Fraser, N.Hengartner, K.Vixie, and B.Wohlberg, "Classification modulo invariance, with application to face recognition," *Journal of Computational and Graphical Statistics*, 2003, invited paper, in preparation.
- [5] P.J.Phillips, H.Moon, P.J.Rauss, and S.Rizvi, "The feret evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, Oct. 2000, available as report NISTR 6264.
- [6] J.R.Beveridge, K.She, B.Draper, and G.H.Givens, "A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001. [Online]. Available: <http://www.cs.colostate.edu/evalfacerec/index.html>
- [7] R.Beveridge, "Evaluation of face recognition algorithms web site." <http://www.cs.colostate.edu/evalfacerec/>, Oct. 2002.
- [8] M.Turk and A.Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, 1991.
- [9] W.Zhao, R.Chellappa, and A.Krishnaswamy, "Discriminant analysis of principal components for face recognition," in *Face Recognition: From Theory to Applications*, Wechsler, Phillips, Bruce, Fogelman-Soulie, and Huang, Eds., 1998, pp. 73–85.

John Elder

Ensembles of Models: Simplicity (of Function) Through Complexity (of Form)

Transcript of Presentation and PDF Slides

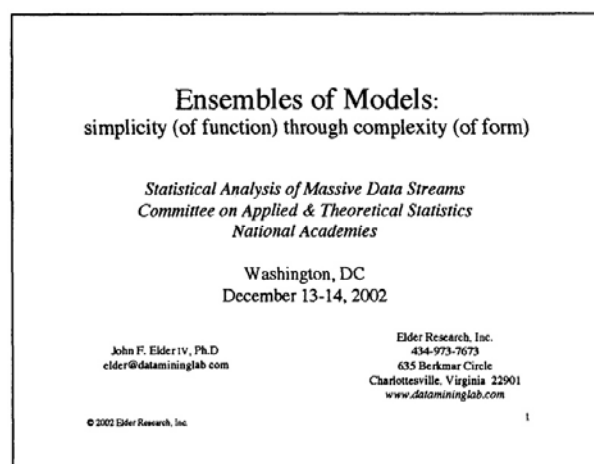
BIOSKETCH: John Elder is chief scientist of Elder Research, Inc. (ERI), a consulting firm with offices in Charlottesville, Virginia, and Washington, D.C. (www.datamininglab.com).

Dr. Elder earned electrical engineering degrees from Rice University and a PhD in systems engineering from the University of Virginia, where he is currently an adjunct professor, teaching optimization. He spent 5 years in high-tech defense consulting, 4 years heading research at an investment management firm, 2 years in Rice University's Computational and Applied Mathematics Department, and has led ERI since 1995.

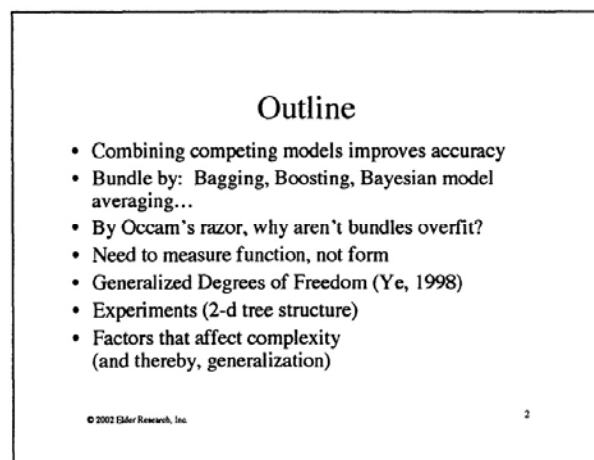
Since 1995, Dr. Elder has led ERI's projects in credit scoring, direct marketing, sales forecasting, stock selection, image pattern recognition, drug efficacy estimation, volatility forecasting, fraud detection, biometrics, and market timing. He writes and speaks widely on pattern discovery techniques and is active on statistical and engineering journals and boards.

Dr. Elder is active in statistics and engineering conferences and boards and is a program co-chair of the 2004 Knowledge Discovery and Data Mining Conference. Since the fall of 2001, he has served on a congressionally appointed panel guiding technology at a division of the National Security Agency.

TRANSCRIPT OF PRESENTATION



MR. ELDER: I am going to talk today on two topics, ensembles of models—that is, combining competing models together to improve accuracy, and then the complexity of doing that. It tends to help you to combine models on new data, which is sort of counter to the intuition or religion of simplification being important for generalization. So, I want to explore that.



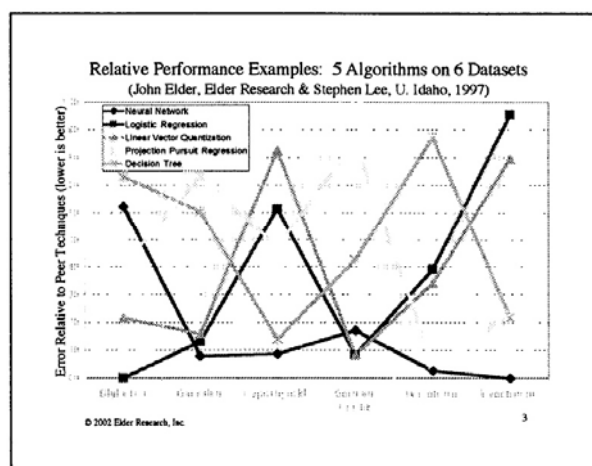
Combining models almost always improves generalization accuracy. The mechanisms for this aren't completely known, but there are a lot of intuitive reasons for that. I will mention a couple of methods of combining models, which I call bundling. You will notice that all good methods have to start with the letter "B".

The question it raises, though, is why, if you take a model that is relatively complex already, and perhaps even slightly overfit, based on your experiments, and combine it with other models that are of a similar ilk, you don't get even further overfit. It certainly goes against the minimum description length principle for model selection, which I find very elegant and beautiful, that a model—that is related to the communications world by saying, I have to transmit an output variable to you.

We all have the same input variables, and I can send that as a signal and then send the noise. Since the error distribution of the noise, which has to be sort of completely described, is tighter, given a good model, if I am able to describe the gist of the data very well, then I will only have to send a little bit of correction for you to reproduce the data. However, if my model is not very good, the error component will be much larger. So,

this is very appealing and it gets people into counting bits in terms of how to describe trees and so forth. The model ensembles really mess with this whole thing because they are very difficult to describe, and yet, generalize well. So, that is a puzzle.

I think an answer is, to measure complexity differently than the form of the model. If it looks complex, it isn't necessarily, to look at the function of the model, to look at its behavior rather than its appearance. One of the metrics that does this was recently introduced by Jianming Ye and I want to describe that briefly, if I can. Then, a very simple experimental example to look at how it works out in practice, and then conclude with just some factors that affect complexity and, ergo, generalization. So, you can see I am still wedded to the Occam's razor viewpoint, which has been under attack recently, and for good reason. So, let's see if this is a partial answer to that.



Here is an experiment that Steven Lee at the University of Idaho and I performed, using five different algorithms, on six different well-studied problems. Most of these data sets have fewer points in the data set than they have papers that have been written with them, so these are, in some sense, not necessarily representative. In particular, the investment problem is completely made up there on the right. They are ordered on the x-axis in terms of increasing difference on absolute terms between the estimates.

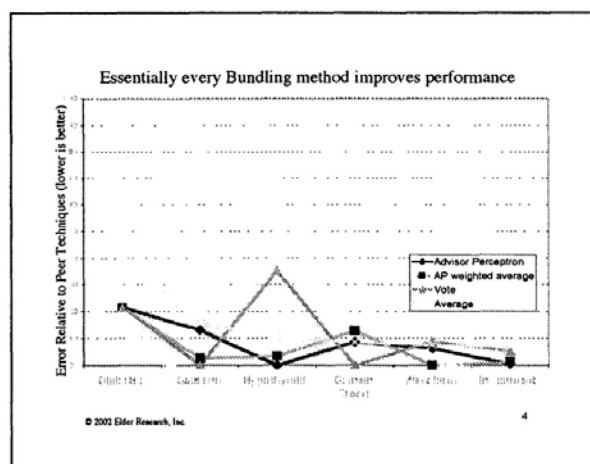
So, there is a higher variance of differences on the right-most data set than there is on the left-most. What I have plotted are the relative errors. So, the very worst would be near one at the top, and the very best would be at the bottom. So, we can see that this selection of five algorithms, which is not my five favorite algorithms—this is the five we could get a hold of in S at the time, yet they are very different.

We see that the age-old question of which algorithm is best really has just the kind of answer that we want. It depends on the problem.

From this small subset of data, probably the neural nets would be the one. Again, this is on out-of-sample data, so this was after the fact. I don't have shown here what we knew about the training data beforehand. After the fact, it looks like neural nets, the red line, is closest to the zero line in terms of relative error. You can see that most techniques are competitive in at least one of the problems.

The reason we wanted to do all these was to introduce a way of combining models. So, we looked at that, but then had to swallow our own medicine and look at simpler versions of it. We were introducing a technique called Advisor Perceptron, and that is the red one here, and it does very well. Simple model averaging, although it outperforms on each of the problems—that is the yellow line—it still does just fine and

it is a whole lot easier.



So, our point was to try to introduce a new technique, and it looks like splitting hairs compared to the big issue, which is virtually any reasonable method of model combinations. So, this is on the same scale as the previous slide, and you can see that the error variance, if you will, of choosing any one particular favorite technique is much greater than building five models and averaging the outputs together, and you get a much more robust answer. In fact, there was no pre-filtering on the individual models. They didn't have to reach any kind of performance threshold to be included in the average. They were just all averaged together.

So, this is exciting. It is not five times more work to build five models. It is maybe 10 percent more work, because all the work is getting the data ready, and actually the fun part is running models. So, you get to do more fun things, and you get five times as much output, which is good, when you have a client or boss, and yet, your answer is better and you avoid the hard choices of which one.

Now, I think this isn't done much, because we tend to specialize, and people tend to become experts in a particular technique, and they don't really feel confident in other techniques. Perhaps this problem has been alleviated somewhat by the commercial tools which keep becoming more and more improved, keeping it a little easier, the learning curve. It is certainly easier than it used to be in terms of using something.

I remember when MARS was freeware, it took about a month to get it up and running. Now it is a commercial product, and I would rather pay a few thousands than spend the month. Only graduate students would rather spend the month than actually pay real money. They don't realize their time is money, even for graduate students.

The good news is I think it is going to be possible for people to run more algorithms on the same problem, with 5 percent, 10 percent more effort. That is the fun part and the payoff is really good. So, that is a plug for modeling.

Bundling estimators consists of two steps:

- 1) Constructing varied models, and
- 2) Combining their estimates

Generate component models by varying:

- Case Weights
- Data Values
- Guiding Parameters
- Variable Subsets

Combine estimates using:

- Estimator Weights
- Voting
- Advisor Perceptrons
- Partitions of Input Space, X

© 2002 Eder Research, Inc. 5

Just stepping back, bundling consists of two basic steps. First, build a variety of models and, secondly, combine them in some way. I have just shown here that there are a lot of different options for how you can do those two steps. You can use different case weights, which is a generalization of bootstrapping, or you can modify the data values. You can just run with the same data and change your guidance parameters, or you can use different subsets of the variables.

So, you can think of data coming from different sources and building models that are specialized from that source and later combining, without having to merge the entire data sets and start whole. That is a possible way.

Then, combining them, you can vote or weigh or do other fancy things, or partition the space and have models that are experts in one area gradually hand over their authority to models that are experts in overlapping areas.

Bundling Techniques

- **Bayesian Model Averaging:** sum estimates of possible models, weighted by posterior evidence
- **Bagging** (Breiman 96) (*bootstrap aggregating*) -- bootstrap data (to build trees mostly); take majority vote or average
- **Boosting** (Freund & Shapire 96) -- weight error cases by $\beta_t = (1 - \epsilon(t)) / \epsilon(t)$, iteratively re-model; average, weighing model t by $\ln(\beta_t)$
- **GMDH** (Ivakhenko 68) -- multiple layers of quadratic polynomials, using two inputs each, fit by Linear Regression
- **Stacking** (Wolpert 92) -- train a 2nd-level (LR) model using leave-1-out estimates of 1st-level (neural net) models
- **ARCing** (Breiman 96) (Adaptive Resampling and Combining) -- Bagging with reweighting of error cases; superset of boosting
- **Bumping** (Tibshirani 97) -- bootstrap, select single best
- **Born-Again** (Breiman 98) -- invent new X data ..

© 2002 Eder Research, Inc. 6

Now, all of these have been attempted. I am just going to list here a handful of the major ways of combining models, and we have some experts here in a number of these areas. I also want to mention, Greg Ridgeway and I did a tutorial on this, and the notes from that are available on our web site, Greg from RAND.

Perhaps the one that has captured the most attention, bagging, is perhaps the easiest one. It is just bootstrap aggregating. You take the same set of data and you build bootstrap replicates of the data, which I kind of—which are certainly easy to do, and build models on those replicates and then aggregate them together, or vote.

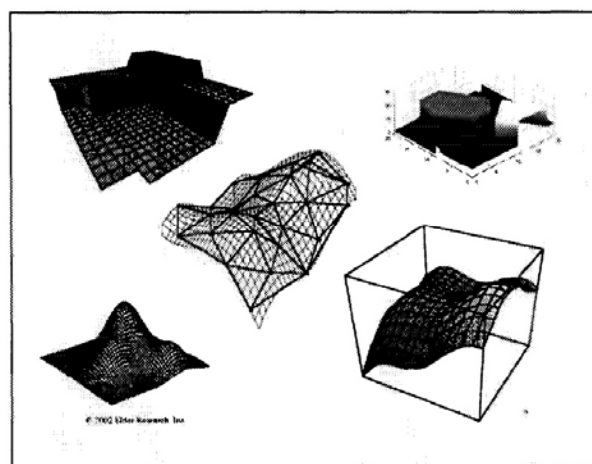
One that is a much more sophisticated, complex and so forth is boosting, where the accuracy of the first—you build a model, and the accuracy of that model is evaluated

for all the particular data points. Initially, everyone starts out with the same weight, every observation. The observations that you are able to nail get very low weight, and the ones that you have errors on get more weight, and you do a second pass.

Then, the things that you still have trouble with, the weight continues to go up and the weight gets less and less, and you get these obsessed models, as you build these stages, that are looking more and more focused on fewer and fewer things. Sort of the things that you do right are taken for granted and—it is really like marriage, really. The socks on the floor really become important. These models are a little crazy. You don't use that final obsessed model. You use a weighted sum of the whole sequence of models, and this, astonishingly, has very good performance.

So, a number of good statisticians—it wasn't invented by statisticians. They would never come up with anything quite that crazy, but it was looked at after the fact as to why it is working, and Jerry Friedman and Dick Shraney and Hasty have done some good analysis of that, perhaps why—once you sort of analyze why something works, you can start to twiddle with it and change parameters, and get papers out of it. So, that is a good thing.

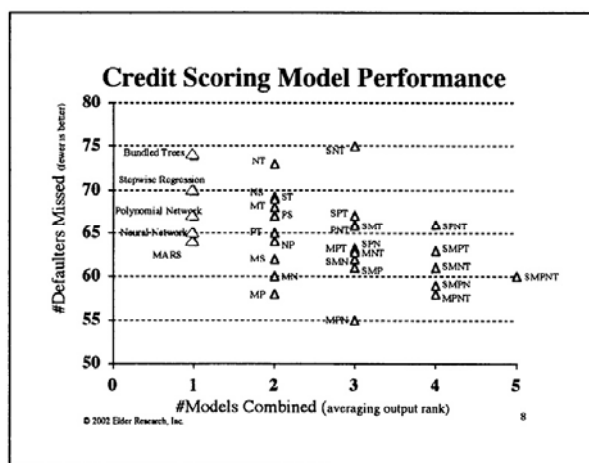
There are a lot of different bundling techniques. Again, the distinction between them is probably less important than just doing them. My favorite way, which is still rather rare, my favorite way of generating variety is to use completely different modeling methods.



This is a surface representation of a decision tree. This is a nearest neighbor surface representation, where a data point looks for its closest known point by some metric, and takes it answer as its estimate.

That gives you similar surfaces to a decision tree, but not rectangular, but convex shapes. This is a piece-wise planer approximation method, a polynomial network, which is similar to a neural network, but smooth and differentiable, and then a kernel estimation surface.

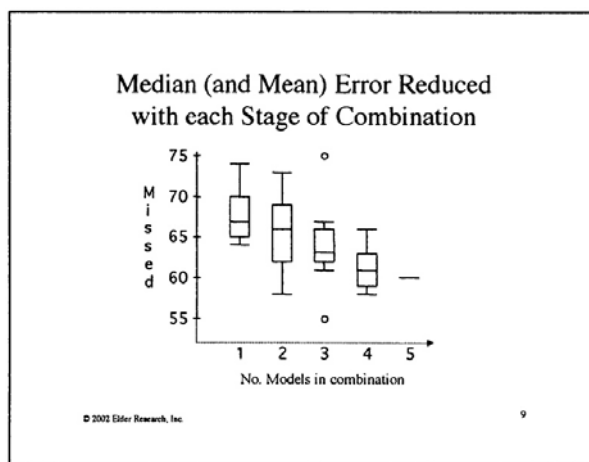
So, you can see that the strength, sort of the vocabulary, of these different methods are different, and they have different types of problems that they will work well on, or different regions of the same problem that they will work well on. So, it is a good source of diversity to have different modeling methods.



One more example, pro bundling, we did a fairly sophisticated long-term project on credit scoring performance, and I am just showing five of the dozen or so techniques that we utilized. Here is a tree. It is scalable, there are properties that can handle categorical variables, real variables, they are fast, they are interpretable, at least in very small dimensions. They only have one problem—accuracy. It is like a really well-written book about a very uninteresting subject.

So, trees alone are up here somewhere, but when you bundle them, it suddenly becomes competitive with much more complex techniques, but still, was the worst of these five. What we did is, we said, okay, let's try all pairwise combinations of these methods. This is out of sample performance, so you wouldn't necessarily have known, well, sure, let's employ MARS and neural nets together and we will do better than either one.

This is the distribution of the pairwise performance, and then, of course, the three- and four- and five-way performance. You can see that there is a trend here. The single best model is a combination of the three network methods. It is possible—in this case, stepwise regression, neural nets and trees—combine to give you something worse than any of the individual models. So, it is not a silver bullet, but statisticians—or statistician wannabees, as I am—you look for the trend, you look for the thing that will help you probabilistically.



Just to highlight that, I will show the box plots of these distributions as the degree of integration of the models, in this case just averaging the outputs, as it increases. You can see that the mean and the median of the error improve with each other model that is

combined. We took this out to a dozen or so. Obviously, at some point, it saturates and has some noise, but the trend was definitely a good one, and it is one that I have seen over and over.

Of course, it is a great research area to say, under what criterion should you include a model and a bundle and how can you estimate error rates and so forth, and that is very much an open issue.

Recent weakened confidence in *Occam's Razor*

- *Nunquam ponenda est pluralitas sin necessitate* ("Entities should not be multiplied beyond necessity") -> Simpler of equally accurately trained models will generalize better.
- "KDD 1998 Best paper" awarded to Domingos for highlighting razor failings, including (besides bundling):
 - Emulating a bagged ensemble improves accuracy (Domingos, 1997)
 - Grafting excess nodes to trees (already perfectly fit) consistently helped generalize (Webb, 1996)
 - Much overfit is from *excessive search* (e.g., Jensen 2000), rather than over-parameterization

© 2002 Eliber Research Inc. 10

Well, this brings up the whole problem of Occam's razor, which basically has been taken to mean that simplicity is going to be a virtue in a sample. Pedro Domingos here got the best paper award a few years ago at the Knowledge, Discovery and Data Mining Conference for highlighting failings of the razor, and highlighting failings of this assumption, including, besides the performance of various kinds of model ensemble techniques, showing that, if you built an ensemble and then estimated it—if you built one model that estimated—that would improve your accuracy over building that one model to start with.

This perverse thing, that I haven't had the guts to read the paper, in 1996, apparently taking trees that were perfectly fit and then grafting excess nodes to them, improved generalizability. That makes no sense, and I also point out some work by David Jensen about how much of overfit isn't really over-parameterization but is excessive search.

Complexity most often controlled by counting and penalizing terms

- Much faster than cross-validation
- Allows one to use all the data for training

but

- A single parameter in a *nonlinear* method can have <1 or >5 effective degrees of freedom.
"The results of Hastie and Tibshirani (1985), together with those of Hinkley (1969, 1970) and Feder (1975), indicate that the number of degrees of freedom associated with nonlinear least squares regression can be considerably more than the number of parameters involved in the fit." (Friedman and Silverman, 1989)
- The final model form doesn't necessarily reveal the *extent of the structure search*.
Ex: The winning KDD Cup 2001 model used 3 variables. But there were 140,000 candidates, and only 2,000 constraints (cases)

Hjorth (1989) " . . . the evaluation of a selected model can not be based on that model alone, but requires information about the class of models and the selection procedure."

→ Need model selection metrics which include the effect of model selection

Eliber Research Inc. 11

This brings up a number of issues with respect to model selection and model complexity. Typically, what has been done most often to control complexity is by

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

counting and penalizing terms.

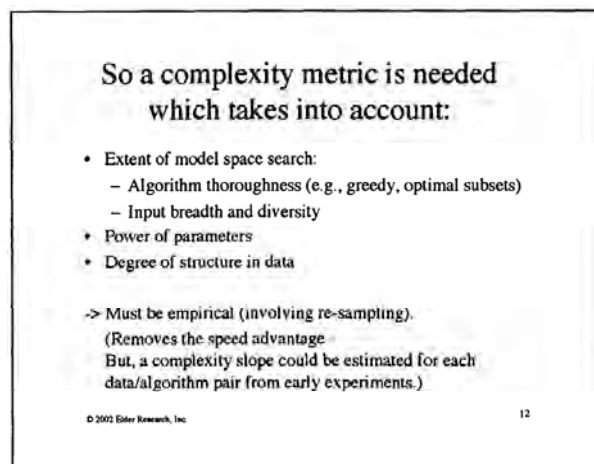
It certainly works for regression, where the number of terms is equal to the number of degrees of freedom that the model is using, and you can generalize a version of cross-validation, which is a function of a penalty times the number of parameters in the error. So, if the error improves with complexity, the complexity goes up and you look for the best trade-off between the two and select a model at that point. It is fast, allows you to use all your data for training, instead of having to reserve some.

It has been long known that single parameters can have fewer or greater degrees of freedom, and I just included some references here from the 1970s and 1980s, where, for instance, if you have a neural net, you can often have more parameters than you have data points and yet, not necessarily be over-fit. It is not making full use of those parameters. The parameters themselves are weak, they don't have full power.

You can have a tree or MARS or something like that, that has the equivalent of three or four degrees of freedom for every parameter. There is a lot more search going on, a lot more flexibility. So, counting terms is certainly not the story.

Also, the search, if you look at a model, you don't know how much work went into getting that model. If you have a three-term model, did you look at 10,000 variables to choose the three or 10 variables to choose those three. That makes a big difference.

Hjorth, in 1989, said the evaluation of an effective model cannot be based on that model alone, but requires information about the class of model and the selection procedure. So, we find ourselves in a situation where, if we truly want to evaluate the complexity—that is, degree of overfit—we have to take into account the model selection process in coming up with a model selection metric.



So, we need to take into account the extent of the search over model space, how thorough the algorithm was. In fact, a number of people hypothesized that, well, we know that greedy step wise regression is suboptimal, but it helps us avoid over-fit. If we looked at all optimal subsets, that would be more likely to over-fit. That is actually false, I believe, but there is the acknowledgment that the thoroughness of the algorithm search is a factor in over-fit.

How diverse and how many inputs there are matter. The power of the parameters, and often overlooked, the degree to which the problem itself may be easy or hard.

There is clear structure in the data. Then, a number of adaptive techniques will latch onto it. If it is much harder to find, you have much more chance for mischief.

So, a model selection technique will have to be empirical and involve re

sampling. We may be able to get back to a place where we can do some of that, but we lose speed with that. We may be able to regain some of it if we do some early estimations and refine our complexity parameter approach from that. So, it is heavily computer-dependent, but we may be able to do it just in some early stages, and more intelligently select models from the vast search that data mining involves.

Data mining, you come up with a final model and people interpret it. I always have trouble with that, because you have looped over a finite data set with noise in it. You have searched over billions of possible combinations of terms. The one that fell to the bottom, or that won, just barely beat out hundreds of other models that had perhaps different parameters, and the inputs, themselves, are highly correlated. So, I tend to think of models as useful and not try to think of the interpretability as a danger, but I know that is not the case with everyone here, in terms of the importance of interpretability.

**Complexity should be measured by the
*Flexibility of the Modeling Process***

- *Generalized Degrees of Freedom, GDF* (Jianming Ye, *JASA* March 1998)
 - Perturb output, re-fit procedure, measure changes in estimates
- *Covariance Inflation Criterion, CIC* (Tibshirani & Knight, 1999)
 - Shuffle output, re-fit procedure, measure covariance between new and old estimates.
- Key step (loop around modeling procedure) reminiscent of *Regression Analysis Tool, RAT* (Faraway, 1991) -- where resampling tests of a 2-second procedure took 2 days to run.

© 2002 Eiler Research, Inc. 13

Well, there are a couple of metrics that have these properties. Jianming Ye introduced generalized degrees of freedom, where the basic idea is to perturb the output variable, refit your procedure, and then measure the changes in the estimates, with the idea that the flexibility of your process is truly complex.

If your modeling process was to take a mean, then that is going to be fairly inflexible. It is certainly more subject to outliers than a median or something, but it is a whole lot less flexible than to changes in the data than a polynomial network or a decision tree or something like that. So, if your modeling procedure can respond to noise that you inject, and responds very happily, then you realize it is an over-fit problem.

This reminds me, when I was studying at the University of Virginia and one of the master's students in my group was working with some medical professionals across the street at the University of Virginia hospital. He had a graph that they were trying to measure heart output strength for young kids with very simple-to-obtain features like the refresh rate for your skin when you press it. That is not the technical word for it, but how quickly it becomes pink again after you squeeze it, or the temperature of your extremities and so forth, and see if they could have the same information that you could get through these catheters that are invasive and painful.

They determined that they could, but I remember a stage along the way when he was presenting some intermediate results on an overhead. This is one of the dangers of graphs on overheads. The nurse and the head nurse and the doctor were going over it and saying, "I see how temperature rises and you get this change." My colleague realized, to his horror, that he had the overhead upside down and backwards. He changed it and it

took them about five seconds to say, “Oh, I see how that works, too. It is completely the opposite.”

So, we want to believe and, if we are extremely flexible with resultant changes to the data, then maybe we are too flexible, too easily fooled, too much over-fit.

So, another method I won't look at in detail, but I will just mention, is Tibshirani and Knight's covariance inflation criteria. Instead of randomly perturbing the outputs, they shuffle the outputs and look at the covariance between old and new estimates. The key idea here is to put a loop around your whole procedure and look at its sensitivity.

I remember that the first person I heard—Julian Fairway—in an interface conference in 1991 doing that in his RAT regression analysis tool, at the time, a two-second analysis took two days to get resampling results on, but I thought it was a very promising approach. I saw him later. He was one of those statisticians trapped in a math department. So, he had gone away from doing useful things and doing more theoretical things out of peer pressure. Anyway, I was excited about this and it was, I think, a good idea.

Generalized Degrees of Freedom

- #terms in Linear Regression (LR) = DoF, k
- Nonlinear terms (e.g., MARS) can have effect of $-3k$ (Friedman, Owen '91)
- Other parameters can have effects < 1
(e.g., under-trained neural networks, "Reron" networks)

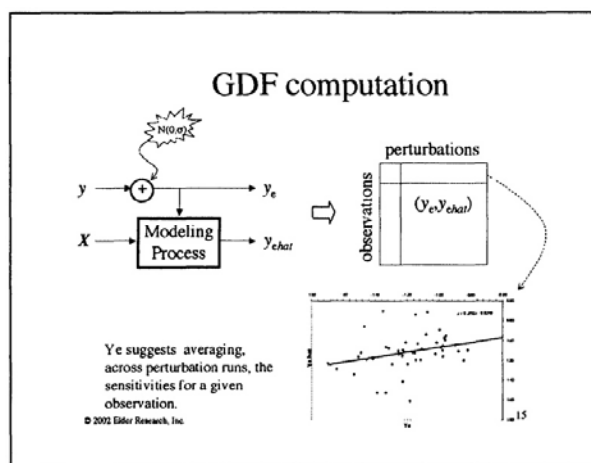
Procedure (Ye, 1998).

- For LR, $k = \text{trace}(\text{Hat Matrix}) = \sum \delta y_{\text{hat}} / \delta y$
- Define GDF to be sum of sensitivity of each fitted value, y_{hat} , to perturbations in the corresponding output, y . That is, instead of extrapolating from LR by counting terms, use alternate trace measure which is equivalent under LR
- (Similarly, the effective degrees of freedom of a spline model is estimated by the trace of the projection matrix, S : $y_{\text{hat}} = Sy$)
- Put a y -perturbation loop around the entire modeling process (which can involve multiple stages)

© 2002 Eder Research, Inc. 14

So, let me explain a little bit about what generalized degrees of freedom are. With regression, the number of degrees of freedom is the number of terms. If we extrapolate from that, you see that you count the number of thresholds in a tree, the number of splines in a MARS or something. People had noticed that the effect can be more like three, as I mentioned before, for a spline or even less than one for some particularly inefficient procedures.

Well, if we, instead, generalize from a linear regression in a slightly different way, if we notice that the degrees of freedom are also the trace of the hat matrix, which is the sensitivity of the output, to sensitivity estimate of changes to the output, then we can use that as a way of measuring. We can empirically perturb the output, and then refit the procedure. This is similar to what is done with splines, I understand.



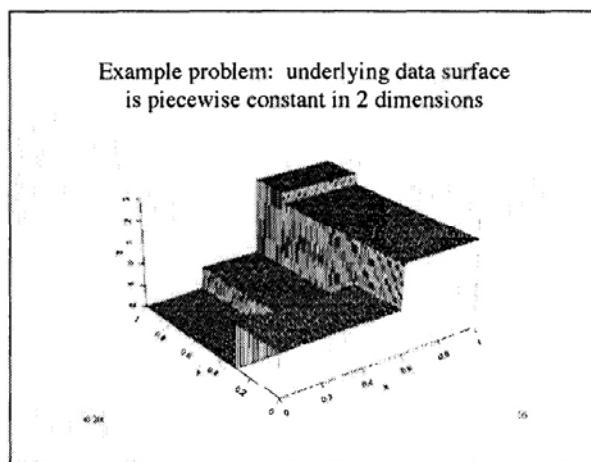
So, put a loop around the process. That is nice, because the process itself can be a whole collection of procedures—outlier detection, all sorts of things can be thrown into this spline. So, graphically, we have a modeling process and we have inputs and we have an output variable. We add some noise to it, and we record the output variable and the new forecast based on that perturbed output variable.

I kind of like the fact that the output y_e also spells the name of the fellow that came up with it.

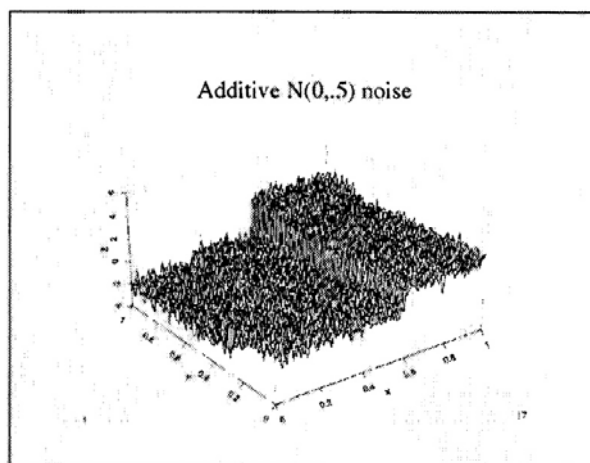
So, you have 100 observations. I know that is a heretically small number for this gathering. You have 100 observations and 50 perturbations. Then, you would record the perturbed output and its change, and you would look at the sensitivity of change in the output and change in the estimate. If I were doing it, I would just naturally—each perturbation experiment, I would calculate a different number.

I do want to point out that Ye, I think, had a good idea, and he took the matrix and sliced it this way and said, well, I am going to look at the effect on observation one of changes over time, which seems to be a little bit more robust than measuring up all of these within an experiment.

Also, interestingly, you can then assign complexity to observations. Now, I excluded that graph from this, but if you are interested, I can show, on a sample problem, where we did that.

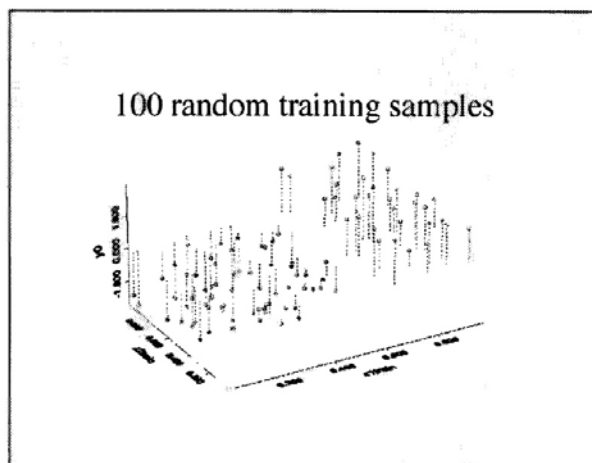


I will conclude, in this last few minutes, here with a very simple problem. This is a decision tree, and this is a decision-making mechanism for the data. Naturally, the tree is going to do pretty well on it. It is amazing how many test problems for trees involve

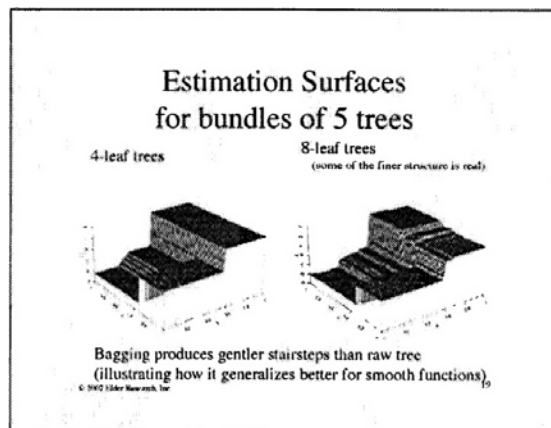


trees as their source.

Here is where we have added noise to it. The noise is at a level that you can see it obscures—pretty much obscures—the smallest structural features, but doesn't obscure others. So, there are some features that are easy to discern and others that are harder, and that seems to be picked up very nicely by the GDS metric.



Now, out of this surface we have sampled 100 training samples, and the other experiments with 1,000 and so forth. I am just going to show you sort of the very initial experiments, and make only a few points. This is very much ongoing.

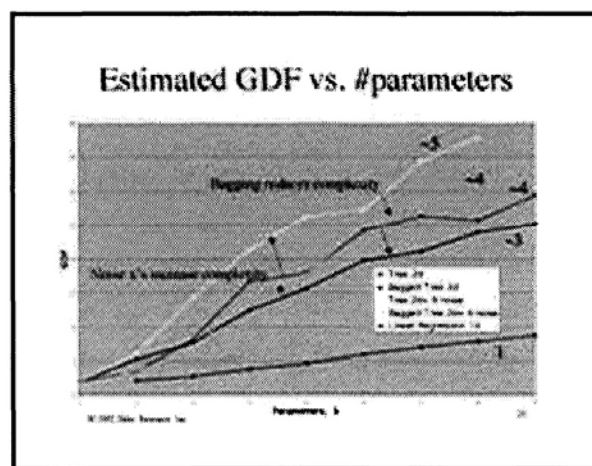


Out of those samples, we built trees and we also built boot-strap and put five trees

together, that went down different amounts. They either had four leaf nodes or eight leaf nodes in this example. You can see, if we built five four-leaf trees and put them together, you still get a tree. You still get something that could be represented as a single tree. It would look rather complex.

So, I can show what that looks like, if you are interested, but you can see here, that the bagging procedure does gentler stairsteps than a raw tree. That tends to be good for generalization, the smoothness. So, bagging helps to smooth trees and that is, I think, the major reason for generalization improvement.

You can see that, when you go to more complex trees, eight leaf nodes would be eight estimation surfaces. There are only five in this data. So, four is sort of under-powerful and eight is over-powerful, and you can see that it completely misses the smallest piece of structure up here. This one picks up on it, but on it, but it also picks up on another structure that isn't there. So, it is the bias variance type of trade-off.



Here is, in essence, the end result. We also did experiments with eight noise variables being added in. So, tree structure depends on two variables, but now you have thrown in eight distracting noise variables. So, how does that affect things. Then, we have, on this slide, the regression as well.

As the number of parameters increases, the measured generalized degrees of freedom for regression is almost exactly what Gary would tell you. It is basically one for one.

Interestingly, here, with the trees, a single tree—this purple line here—it has an estimated rough slope of four, meaning that, on this problem, each split that was chosen with the tree algorithm has roughly the equivalent descriptive power of four linear terms, or it is using up the data at four times the rate of a linear term. If you bag those trees, it reduces to a slope of three. If you add in noise variables, it increases the slope to five, and then, if you bag those trees that are built on noise variables, it brings it back to four.

So, just to summarize, adding in distracting noise variables increases the complexity of the model. The model looks exactly the same in terms of its structure and its size and how long it takes to describe the model. Because it looked at eight variables that could only get in its way, it was effectively more complex.

Then, the other point is that the bagging reduces the complexity of the model. So, the bagged ensemble—five trees—is less complex than a single tree, and that is consistent with Occam's razor idea that reduced complexity will increase generalizability after you reach that sort of saturation point.

Bundling & Complexity Summary

- Bundling competing models almost always improves generalization.
- Different model families is a good source of component diversity.
- If we measure complexity as *flexibility* (e.g., with GDF) the classic relation between complexity and overfit is revived.

The more a modeling process can match an arbitrary change made to its output, the more complex it is.

- Complexity increases with distracting variables.
- It is expected to increase with parameter power and search thoroughness, and decrease with priors, shrinking, and clarity of structure in data. Constraints (observations) may go either way ..
- Model ensembles often have effective complexity *less than* their components.
- Diverse modeling procedures can be fairly compared using GDF

© 2002 Elder Research, Inc

21

To summarize, I have a lot of little points to make. Bundling almost always improves generalization, and I think that different model families is a great source of diversity that you need for the bundling. If we measure complexity as flexibility of the procedure, then we sort of revive or answer—partially answer—some of the problems with general intuition of complexity being related to over-fit. So, the more a modeling process can match an arbitrary change made to its output, the more complex it is by this measure.

Complexity clearly increases with distracting variables. These are variables that are considered, during the model search project, which may not appear at all in the model at the end. Actually, it would have to appear, at least somewhat, to affect the behavior, but the size of the model could be the same between—two candidate models could have the exact same number of parameters and so forth, but one could be more complex because of what it looked at.

The complexity is expected to increase, obviously, with parameter power, the thoroughness of your search, and decrease with the use of priors and shrinking, and if there is clarity in the structure of the data. In our early experiments, I certainly thought the complexity would decrease as you had more data. That seems to be the case with some methods and not with others. It seems to actually increase with decision trees and decrease with neural nets, for instance.

By the way, neural nets, on this same problem, had about 1.5 degrees of freedom per parameter. It could be that local methods somehow are more complex with more data, and methods that fit more global models are not. So, that is an interesting question. Model ensembles usually have effective complexity less than their components by this empirical measure.

The next thing about the GDF is you now can more fairly compare very diverse procedures, even multi-step procedures, as long as you can put a loop around it. That concludes my talk. Thank you.

MS. KELLER-MC NULTY: Questions?

QUESTION: I have one. When you talk about the complexity of flexibility, are you meaning robustness? I am confused about the local methods being more complex.

MR. ELDER: I am, too. It is confusing. The idea behind GDF, the idea behind these empirical measures of complexity is that you want to be able to measure the responsiveness of your procedure. Obviously, you don't want something that, every time you change the task a little bit, it is good for that one thing.

You know, it is the universal health oil. I really have hair problems. Well, it is good for that, it is good for your liver, too. Obviously, it is overfit.

You know, where is that trade-off? So, the measure is, if you arbitrarily change the problem, or you change the problem a little bit, how responsive is it to suddenly get the same accuracy and so forth?

Again, it was a little bit confusing to me, but the trees, they are only concerned about—once they partition the data, they are only concerned about their little area, and they are not paying attention to the big, whereas, models that are basically fitting a global model, data everywhere helps, and it tends to rein it in a little bit, is my best guess. It is certainly an interesting area.

QUESTION: Do you have any extrapolation from your results? [Comments off microphone.]

MR. ELDER: The question, how things might change with many variables and massive data sets. I would like to revisit a problem we did.

We participated in the KDD cup challenge, I guess, two summers ago, and that had 140,000 variables, which is a lot for us, but only 2,000 cases. The final model, the one that won the contest, used three variables. Well, three variables seems like a simple model, but it was a huge search. It would be nice to put that under this microscope and see what happens.

Again, my intuition was that more data would actually help reduce the complexity, but it is kind of wide open.

AUDIENCE: [Question off microphone.]

MR. ELDER: Here, I have just talked about combining the estimates, and some techniques need a little squeezing to get them into the right shape.

Take a tree. You might need to do some kind of robust estimation. If it is a classification problem, you might get a class distribution and use that to get a real value — to turn into a real value so that you can average it in.

What I haven't talked about is, there are whole areas for collaboration amongst these different techniques that we have explored in the past. For instance, neural nets don't typically select variables. They have to work with whatever you give them, but other methods, like trees and stepwise regression and polynomial regressions select variables as a matter of course. Typically, you can do better if you use the subset they select, when you hand it off to the neural nets, and some methods are very good at finding outliers.

There are a lot more methods that can go on between the methods. Up here, I talked about them just in terms of their final output, but yes, you do have to work sometimes to get them in a common language.

AUDIENCE: [Question off microphone.] I am not sure how what you are talking about here answers that.

MR. ELDER: Good point. Criticisms of Occam's razor work with any measure of complexity, is Pedro's point. The one that bothered me the most is about bundled. They are obviously more complex. They just look more complex, and yet, they do better. How can that be?

In fact, it is very easy to get many more parameters involved in your bundle than you have data points and not have over-fit. They are not all freely simultaneously modified parameters, though. So, there is a difference. Under this measure, the things

that are more complex in GDF tend to over-fit than things that aren't. So, it is a vote on the side of Occam's razor.

I haven't addressed any of the other criticisms, but in the experiments we have done here, it is more like you would —

AUDIENCE: Like, for example, when you approximate an ensemble with one model back again, this model is still a lot more complex, because a lot of this complexity is apparent. It just seems that you still get the more complex models in that.

MR. ELDER: But it is apparent complexity.

AUDIENCE: Yes, but when you replace the apparent complexity with the actual complexity, not just measuring to find the number of parameters. [Comment off microphone.]

MR. ELDER: I certainly would be eager to see that. Thanks.

Report from Breakout Group

Instructions for Breakout Groups

MS. KELLER-MC NULTY: There are three basic questions, issues, that we would like the subgroups to come back and report on.

First of all, what sort of outstanding challenges do you see relative to the collection of material that was in the session? In particular there, we heard in all these cases that there are real specific constraints on these problems that have to be taken into consideration. We can't just assume we get the process infinitely fast, whatever we want.

The second thing is, what are the needed collaborations? It is really wonderful today. So far, we are hearing from a whole range of scientists. So, what are the needed collaborations to really make progress on these problems?

Finally, what are the mechanisms for collaboration? You know, Amy, for example, had a whole list of suggestions with her talk.

So, the three things are the challenges, what are the scientific challenges, what are the needed collaborations, and what are some ideas on mechanisms for realizing those collaborations?

Report from Integrated Data Systems Breakout Group

MS. KELLER-MC NULTY: Our discussion was really interesting and almost broke out in a fist fight at one point, but we all calmed down and got back together.

So, having given you the three questions, we didn't really follow them, so let me go ahead and sort of take you through our discussion. When we did try to talk about what the challenges were, our discussion really wandered into the fact that there are sort of two ways that you can kind of look at these problems.

Remember, our session had to do with the integration of data streams. So, you can kind of look at this in a stovepipe manner, where you look at each stream independently and somehow put them together, hoping the dependencies will come out, or you actually take into account the fact that these are temporally related streams of information and try to capture that. The thought is that, if one could actually get at that problem, that is where some significant gains could be made. However, it is really hard, and that was acknowledged in more ways than one as well. That led us into talking about whether or not the only way to look at this problem domain is very problem-specific. Is every problem different, or is there something fundamental underneath all of this that we should try to pull out?

In particular, should we be trying to look at, I am going to say, mathematical abstractions of the problem and the information, and how the information is being handled, to try to get at ways to look at this? What are the implications and database issues, database design issues, that could be helpful here? There clearly was no agreement on that, ranging on, there is no new math to be done, math isn't involved at all, to in fact, there is some fundamental mathematics that needs to be done. Then, as we dug deeper into that and calmed down a little bit, we kind of got back to the notion that, what is really at issue here is how to integrate the fundamental science into the problem.

If I have two streams of data, one coming from each sensor, if I am trying to put them together, it is because there is some hidden state that I am trying to get at. Neither sensor is modeling perhaps the physics of that hidden state. So, how do I start to try to characterize that process and take that into account? So, that really means that I have to significantly bring the science into the problem. So, then, we were really sounding quite patriotic from a scientific perspective.

One of our colleagues brought up the comment that, you know, this philosophy between, am I modeling the data or am I modeling the science and the problem, you know, has been with us for a long time. How far have we come in that whole discussion and that whole problem area since 1985? That had us take pause for a minute, like, where are we compared to what we could do in 1985, and how is it different? In fact, we decided, we actually are farther ahead in certain areas. In fact, our ability to gather the data, process the data, to model and actually use tools, we clearly are farther ahead. A really important issue, which actually makes the PowerPoint comment not quite so funny is that our ability and communication, remote communication, distributed communication, modes of communication, actually ought to work in our favor in this problem area as well. However, the philosophical issue of how to integrate science and technology and mathematics and all these things together, it is not clear we are all that much farther ahead. It is the same soap box we keep getting on.

Then, it was really brought out, well, maybe we are a little bit farther ahead in our thinking, because we have recognized the powerful use of hierarchical models and the hierarchical modeling approach, looking at going from the phenomenology all the way up through integrating the science, putting the processing and tools together. The fact that it is not simply a pyramid, that this is a dynamic pyramid, that if we take into account the changing requirements of the analyst, if you will, the end user, the decision maker, we have to realize that there is a hierarchy here, but it is a hierarchy that is very dynamic in how it is going to change and move. There are actually methods, statistical mathematical methods, that have evolved in the last 10 or 15 years, that to try to look at the hierarchical approach. So, we thought that was pretty positive.

There is a really clear need, as soon as we are going into this mode of trying to integrate multiple streams, to recognize that expertise, the human must be in the loop and the decision process, the decision environment back to the domain specificity of what you are trying to do, is needed. In a couple of the earlier sessions, we actually heard about the development of serious platforms for data collection, without any regard to how that information was going to be integrated, or how it was going to be used, through some more seriously collaborations that I will get into in a second. Maybe we can really influence the whole process, to design better ways to collect information, better instruments, things that are more tailored to whatever the problem at hand is.

I thought there was a really important remark made in our group about how, if you are really just looking at a single data stream and a single source of information, that industry is really driving that single source problem. They are going to build the best, fastest, most articulate sensor. What they are not going to probably nail is the fusion of this information.

If you couple that with the fact that, if you let that be done ad hoc, that you are now going to have just random methods coming together with a lot of false positives, and then we got into the discussion of privacy invasion, and how do you balance all of that,

that we really need the serious thought, the serious integration, multidisciplinary collaboration, to be developing the methods, overseeing the methodological development, as well as being able to communicate back to the public what is going on here. So, I thought that was kind of interesting. So, collaboration, there needs to be very close collaboration in areas like systems engineering, hardware software design, statistics, mathematics, computer science database type things, and basic science. That has to come together. Now, that is not easy because, again, we have been saying that forever that this is how we are going to solve these problems.

Then that comes into play, what are the mechanisms that we can try to do that? We didn't have a lot of good answers there. One idea was, is it possible to mount certain competitions that really are getting at serious fusion of information that would require multidisciplinary teams like this to come together. There was a suggestion that, at some of our national institutes, such as SAMSI, that is Science and Applied Mathematics Institute, one of the new, not solely NSF-funded, but one of the new NSF-funded institutes, perhaps some sort of a focus here. I think that gets back to Doug's comment, which I thought was really good, that regular meetings as opposed to one up workshops is the way we are probably going to foster relationships between these communities. Clearly, funding is required for those sorts of things. Can we get funding agencies to require collaborations, and how do they then monitor and mediate how that happens.

Then, one comment that was made at the end was the fact that, if we just focus in on statistics, and statistics graduate training, there is a lot of question as to whether we are actually training our students such that they can really begin to bite off these problems. I mean, do they have the computational skills necessary and the ability to do the collaborations. I think that is a big question. My answer would be, I think in some of our programs we are, and in others we are not, and how do we balance that?

Just one last comment. You know, we spoke at very high level and just at the end of our time—and then we sort of ran out of time—it was pointed out that if you really think of a data mining area and data mining problems, that there has been a lot done on supervised and unsupervised learning. I think we understand pretty well that these are methods that have good predictive capabilities. However, it seems that the problem of the day is anomaly detection, and I really think that there, from a data fusion point of view, we really have a dearth of what we know how to do. So, the ground is fertile, the problems are hard, and somehow we have got to keep the dialogue going.

Mark Hansen

Untitled Presentation

Transcript of Presentation

BIOSKETCH: Mark Hansen is a professor of statistics at the University of California at Los Angeles, with a joint appointment in design and media arts. His fields of expertise include statistical methods for data streams, text mining and informational retrieval, information theory, and practical function estimation.

Before joining the faculty at UCLA, Dr. Hansen was a member of the technical staff at Bell Laboratories. He specialized in Web statistics and other large databases, directing a number of experiments with sound in support of data analysis.

He has five patents and is the author of numerous publications as well as serving as an editor for the *Journal of the American Statistical Association*, *Technometrics*, and *Statistical Computing and Graphics Newsletter*. He has received a number of grants and awards for his art installation Listening Post. Listening Post produces a visualization of real-time data by combining text fragments in real time from thousands of unrestricted Internet chat rooms, bulletin boards, and other public forums that are then read (or sung) by a voice synthesizer and simultaneously displayed across a suspended grid of more than 200 small electronic screens.

Dr. Hansen received his undergraduate degree in applied mathematics from the University of California at Davis and his master's and PhD degrees in statistics from the University of California at Berkeley.

TRANSCRIPT OF PRESENTATION

MR. HANSEN: [Speech in progress]. That involves artists like Rauschenberg and even Andy Warhol. The idea was to try to pair, then, mostly engineers and artists together to see what kind of useful forms of artistic expression might come out. In a very self-conscious way, I think, the approach was to try to revive this tradition with the arts and, hence, was born this arts and multimedia program.

It was actually an interesting event. The idea, as I said, was to very self-consciously pair artists and researchers together, and this will actually get to streaming data in a moment, I promise. So, what happened was, they organized a two-day workshop where 20 or so media artists from New York City and 20 or so invited researchers in the labs met in the boardroom, then, of Lucent, and each got 10 minutes to describe what they do. I had 10 minutes to kind of twitch and talk about what I do, and the artists got some time to have some very beautiful slides, and have a very big vocabulary and talk about what they do. Then we were supposed to pair up, somehow, find somebody and then put a proposal together, and they would fund three residency programs where the project would get funded.

Ben and I put together perhaps the simplest thing given our backgrounds, him being a sound artist and me being a statistician. We put together a proposal on data sonification, which is a process by which data is rendered in sound, for the purpose of understanding some of its characteristics that may not be immediately obvious in the visual realm. So, instead of visualizing a data set, you might play a data set and get something out of it. This is sort of an old idea, and it seems like everything I have done John Chambers has done many, many years ago. So, I have kind of given up on trying to be unique or novel in any way.

He was working with perhaps the father of electronic music, Max Mathews, at Bell Labs. This was back in 1974. He developed something that Bell Labs at the time gave the title MAVIS, the Multidimensional Audiovisual Interactive Sensifier. The idea was that you would take a data set or take a matrix and you would map the first column, say, the pitch, to the second column, the timbre, the third column, the volume. Then there would be some order to the data somehow and you would just play it. John said you got a series of squeaks and then a squawk, perhaps, if there was an outlier, and that was as far as it went. He said it wasn't particularly interesting to listen to, but maybe there was something that could be done to kind of smoke out some characteristics in the data. Actually, this kind of mapping, when Ben and I were talking, we thought that this kind of mapping might be able to withstand underground bomb blasts and earthquakes. Apparently, this problem was motivated by Suki, who was involved in the Soviet test ban discussions. At least, I am getting this now all from Bill.

I thought I could give you an example of what some of this early sonification sounds like. A friend of mine at the GMD has developed a program on earthquake sonification, and here is what the Kobe quake sounds like, if you speed it up 2,200 times

its normal speed at a recording station in California. [Audio played.] It sort of gets amusing if you listen to other places. Here is what it sounds like, where a few plates come together, but it also happens at the end of the water. [Audio played.] I am told that it has nothing to do with the fact that there is water all around and has everything to do with the fact that you have three plates coming together there, but I am not going to talk this evening about sort of those early examples of sonification. Instead, I am going to start to talk about some recent work we have been doing in analyzing communication streams. Most of this work was done, again, through this Bell Labs/Brooklyn Academy of Music program. Not surprisingly, then, a lot of it was inspired by communication processes or work-mediated transactions.

Our group had been looking at things like call detail records. I am sure Daryl must have talked about that earlier today. I have personally been looking at Web log data, or proxy log data. So, you get records of who requested what file when from the network. Then we sort of slip in the end into like online forms, chat rooms, bulletin boards, that sort of thing, where the fundamental data consisted of who posted, what did they post, which room, what kind of content. If you think about it, in some sense, the model of this Web as being this place where you kind of go out and you download a page is sort of fading in favor of these sort of these more user-generated, or sort of connection-based communication processes. If you think about the number of e-mails per year, Hal Varian at Berkeley estimated that something like 610 billion e-mails are sent a year, versus only about 2 billion new Web pages are made every year.

AUDIENCE: That is the wrong thing to count. You count how many times they are downloaded.

MR. HANSEN: I guess you are right.

AUDIENCE: Half of them are spam.

MR. HANSEN: I am not going to go into my favorite spam story. It is kind of a mixed audience. Then, there are other ubiquitous public forums. IRC cracked its half-million user mark this year.

In a sense, there is something to sort of these user-generated communication streams, and that is what we are going to try to get at with this project with Ben. So, our initial work, which seemed, I guess, a little wonky at the time, but at least produced some cool music—I don't know how practical it was—focused just on Web traffic. The idea was that we were going to develop sort of a sound that you could use to passively monitor a system. So, we were going to look at people's activity on a Web site.

We looked at Lucent.com, which was then organized into a large number of businesses, now not so many businesses. Below each business directory there was a series of products and directories and then, below those product directories, there were white papers and then, below those white papers, you would have technical specifications. So, the deeper you went in the directory structure, the more detailed material you were downloading.

So, the idea was to create a sound that somehow became more expressive as more

people on the site were browsing deeper and getting more interesting things. So, we created a kind of mapping of sorts that generated drones for sort of high-level browsing. So, if you were somewhere in the micro-electronics area—well, you won't be now, but if at the time you were in the micro-electronics area, you would be contributing to the volume of some overall drone.

As you went deeper, you would be contributing to, let's say, a pulse or some other sound that was all at the same pitch. Then, the tonal balance of the sound would vary based on the proportion of people who were at different parts of the site. So, here is the kind of mapping that we used, just to cut to the chase. This is what Lucent.com sounds like at 6:00 in the morning. [Audio played.] Just to give you kind of a lonely feeling. At 6:00 o'clock in the morning, there are probably 15 people rattling around the site. At 2:30 in the afternoon, we get most of our visitors and it sounds more like this. [Audio played.] So, the deal was to make that somehow kind of pleasant and easy to listen to, and it might inform you of something.

The idea of the sound at some level, the unique feature—I see a lot of skeptical faces. This is a crowd of statisticians and you are supposed to have a lot of skeptical faces. I was with David Cox at the spring research conference called The Cautious Empiricists or something. The “cautious” is the important thing. The idea of it was that sound somehow gives us the capability—you can attend to the sound in a way that you don't attend to sort of the visual system, or you can background sound and you can't really do that with the visual system. So, you can have something going on in the background and you can attend to changes in that something, in the musical track, without really having to listen to it hard. The visual system requires you to watch a display or watch a plot. So, we came up with this sort of general map of the Web site activity, and then a graduate student at Tufts wrote me that he didn't really like this tonal balance that we got, he thought it was maybe a little too ravy or a little too something and he didn't really care for it, and he preferred more natural sounds.

So, he created sounds like this— [Audio played.] —to give this, which is telling you something about the network traffic. The patter of the water increases in volume with the more users who are on the system. The one of the bird sounds is incoming mail. So, you can kind of get a sense of what is going on. Anyway, he seemed to think that was more listenable. At some level we decided that these experiments in sonification were interesting, were certainly creating some music that we didn't mind listening to, but they weren't particularly practical. Also, they didn't speak to many people, because very few people care about any one given Web server. I mean, the number of people who would care about the traffic on Lucent.com is quite small. If you think about most Web servers, that is going to be the case. So, we decided that we needed to find something that had perhaps a little more social relevance. So, we decided we would keep to kind of the communications realm and look at online communications.

In some sense, as I pointed to before, with the amount of e-mail traffic and such, the Web is really just a big communications channel. Our thought was, perhaps

aggressively, perhaps we were a little too whatever, but could we characterize the millions of conversations that were taking place right now. You had people who were in chat rooms, hundreds of thousands of people in chat rooms, people posting to bulletin boards. Can you say something about what all these people are talking about. In some sense, these chat rooms and bulletin boards represent new spaces for public discourse. If you take them together, they represent a huge outpouring of real-time data, which is kind of begging to be looked at. There is a lot of structure here. There are sorts of chat sessions that are kind of day-to-day things. In the morning, it is what are you having for breakfast and in the middle of the day it is, my boss is riding my back. At the end of the day it is, this is a great day, I am off to bed. In between, you have got lots of sort of, not just cycles about sort of daily things, what is going on this morning or what is going on at work, but political arguments about terrorism or Afghanistan or something like that. So, our thought was that we would try to create some kind of tools to give us a better understanding, or tap into this big stream and sort of experience this in some way that perhaps is a little bit more accessible to the general public than just a plot or a graph or something like that.

So, here is the kind of data that we are basically looking at, and we have lots of it. So, you get some sense of a source, in this case, suppose all of these are IRC chat rooms. So, you get the room, the name of the room, and then you get the user name and what they posted. We have agents, and I will talk about that in a little bit, who go out and sort of sample from these rooms. So, we will attach to a network and sample from tens of thousands of rooms, to kind of get an overall sense of what people are talking about.

So, the interesting thing about this project is that not only has there been some statistical contact—and I have given talks about some of this stuff—but there has also been the opportunity for public performances or public events around it. The first thing we did with this chat space was a performance at a place in New York City called The Kitchen, which is a fairly well known—people like Laurie Anderson and stuff got their start at The Kitchen.

It is in Chelsea and at that point we were looking at about 100 chat rooms and bulletin boards. We were looking at sort of news chat, sports, community. There was a beautiful room on the care and feeding of iguanas. I have told this story a lot. The beautiful thing about this room is that, after sort of monitoring it off and on for three or four months, they only used the word iguana like five or six times in that period. So, they don't sort of refer to iguanas as iguanas. It is like baby or honey or my little something or other. I found it sort of amusing that you couldn't really tell what people were talking about, if you didn't include something you knew about the room itself.

From there, we were monitoring for topics, looking at activity levels, that kind of thing. Now, the display that we put out, because this was a performance base, it was part of their digital happy hour series. So, we got a big room, perhaps a bit bigger than this, extraordinarily tall ceilings, and we had the picture at the bottom as a sort of layout of the rooms. There were round tables that people sat around because it was a digital happy

hour. There were four speakers, one in each corner of the room, and in the white bar at the top was a very large screen, about 20 feet tall, 20 feet wide. So, the display that we picked involved streaming the text along four lines at the top, and then each line of text was being read by one of the speakers in the room.

I can tell you what that looks like. I have a small movie of very bad quality, but we get better stuff later. Here you get the full lines of text. [Tape played.] The sound is a tone, and then there is a voice that is speaking at that same tone, in a monotone, so you get sort of a chant effect. There is an algorithmic structure that is used to generate the pitches. It was picked according to the length of the post. So, we wanted to have it clear. If it was very short, it would take the voice only a short amount of time. So, that was, first of all, the text-to-speech was horrible. That was kind of the standard Mac text-to-speech voice.

We only had, like I said, we only thought we had about 100 rooms, but we thought the structure was nice, having the text up there, having the text-to-speech to guide you, and having the compositional element helped to keep people's attention. They were sort of watching this thing. At that point, there wasn't a lot of organizational structure put to it. We just sort of randomly selected representative phrases from the individual chat room and let whatever collide, collide, in terms of meaning or content or whatever. So, that seemed to work out reasonably well. So we posed for ourselves another challenge which was, could we in fact display sort of large-scale activity in real-time from not just 100 rooms, but tens of thousands of rooms? As Ben keeps saying, we aspire to listen to it all. The "aspire" word means that we don't actually have to achieve it, but that is where we are headed.

Again, because we were, for some reason or another, extraordinarily lucky for sort of public performances to keep moving this along, we were part of the Next Wave Festival sponsored by the Brooklyn Academy of Music last year in 2001. I will show you some pictures of this. Here, instead of having the one large screen with four lines of text, we created a 7-foot-tall, 10-foot-wide grid of small text displays, fluorescent vacuum displays, about the size of a Hershey bar, each one. There were, like I said, 110 of them and they could show text that we were generating. Instead of having just four voices at a time, we used the speech engine, which would allow us to have up to 45 voices speaking in the room at a time on eight channels of audio.

So, this was the little sign that they put out in front. This is what the installation space looked like. The Rockefeller Foundation kicked in and we were able to build an installation space. You see the hanging grid of small displays, and then the room itself, the silver panels conceal, in some cases, speakers, and in other cases just acoustic insulation, so you don't get a lot of flutter echo from the walls. Here is what each of the little gizmos look like. This is a standard Noritake display and we had a printed circuit board designed with a microcontroller on board, so that we could communicate with this. This is RS45, for those who care. The two wires on the left are carrying communication and the two wires on the right are carrying power. So, you see that these two things are

hanging from the same things that we are talking to them on and powering them on. So, here is another view. We have this very tight text, sort of four lines of 20 characters, and then we also have this big character mode where we can stream text across the screens. Here are some pictures that were taken at the time. This is the back. The back has an LED on it. In fact, in the BAM room, the Brooklyn Academy of Music room, you enter it in the back.

This wedge over here is the doorway, and you would enter in the back. What you would see is the LEDs are the pattern. So, you come into a very sort of abstract space, and then you move around and see all the text. The piece itself was organized in a series of scenes or phases that were all meant to highlight some different aspect of the stream. In some cases, they are quite simple. All they are doing is giving you a tabulation of all the posts by length, and streaming it by, so you not only get a sense of scale, like how much conversation is going on because things are streaming by fairly quickly, but you also get a chance to see the short posts, a lot of hi, hey, hi, and the longer posts were—at that time, there was a whole lot of talk about Afghanistan and John Walker.

I think there was one Friday when we were up at BAM when Wynona Rider was arrested. If only we timed it better, this time we would have been able to see her being sentenced. Anyway, that was a very simple one, but the second scene tries to organize things by content and creates kind of a dynamic version of a self-organizing map. You get large regions that are all referring to the same topic, and the regions grow in response to the proportion of people who are talking about that particular topic.

So, if Afghanistan is popular in the news and lots of people are talking about it, that region will grow quite large, and that depends quite heavily on the text-to-speech that you are using. Then there are other things I won't have time to illustrate. This is the kind of thing that end up coming up from the map scene that I will show you in a minute.

To generate all this, I guess I should give a little bit of talk about the stream itself. We have a series of JAVA and Pearl clients that are on the protected side of the Lucent firewall, that are going out and pulling things from chat rooms and bulletin board. Then, on the other side, in the display space, we have four computers, one running sort of the sounds in the room, one running a text-to-speech engine, one running the individual displays themselves, and then one kind of overseeing all of it. The unfortunate thing is that all of those machines are running on a different operating system. If you can think of another operating system, we would be happy to include it.

So, it is all about interprocessor communication somehow. On the Lucent firewall side, we have two Linux servers and two class C networks that give us the capacity to look like about 500 different IP addresses. The chance that we are going to get somehow spotted and caught and turned off seems small, although, every time I say that I am a little nervous. So, we have upgraded the text-to-speech engine as well. We are using Lucent commercial heavy-duty speech engine, that can give us access to about 100 voices in the room, for this Whitney exhibit that I will show in a minute. [Audio played.]

Can we show the DVD now? I am going to show a couple of examples of the

installation. Then I have some production pictures. The construction just started at the Whitney. I apologize in advance that this is going to be a little hard to see. [DVD shown.] This is just that simple tabulation where, when things are very short, the screens are less bright and the sounds are very soft. I will just show a little of the next scene. This is the one that has the content, or builds up a map dynamically, and here we will have some voices. [DVD shown.]

So, as I said, to get to that point, there is, like I said, a series of scenes that this thing alternates through. Each time, because the stream is live, the text that is spoken and the scenes you experience of it are different, because the data are always changing. We are trying to get the buffering now to about 15 minutes, so that everything you see will have been typed just 15 minutes ago. So, there was a series of things that had to happen to pull things from the chat stream and, given the time, I am not going to go into too much detail. There are things like trying to characterize topic and track features by person and room and do some clustering and what have you.

So, from the Brooklyn Academy of Music, we went on to—actually, the Friday after we closed at the Brooklyn Academy of Music, we were sort of summoned to a morning at the Whitney and an afternoon at the Museum of Modern Art in New York, where we met with curators who were, well, let's talk about having our piece here, which was a little humbling and frightening.

In the end, we were fortunate enough to—well, we are opening at the Whitney Museum of American Art, an expanded version of the piece is opening in just a few days. It is an enhanced display space. So, the display is not, instead of having 10 feet by 7 feet, with 110 displays, which was 11 rows and 10 columns, we now have 11 rows and 21 columns, which spans 21 feet. So, the thing is big, and it is in a soft curve. We had a workshop residency, or development residency with a performing arts organization in Seattle called On the Boards. They gave us like a studio, a kind of stage. You can see part of this grid, the curved grid, and then we got to sort of litter the place with our computers. This is what the thing wound up looking like in the end when it was up in Seattle.

We started construction at the Whitney as of Monday. This is the Whitney lobby. This, right there, is where we get to go, which again is an awesome thing, to think that there will be people walking by. So, here is the inside of the space as construction is going on. The first thing they had to do was put up a wall that is going to have all of our panels, the concealed panels. This is the curved beam from which the text displays are going to be suspended. This is my collaborator, Ben. I am very happy that the beam is up in the air. It was a non-trivial problem to get it attached to the ceiling. This is part of the process. Now, speakers are mounted behind the beam, so that voices are spatialized. This is where I left it when I got on the plane earlier this evening. The carpet guys had just arrived and were putting carpeting in the room. So, it was kind of a lonely little place. I guess where this project goes, we have been looking at kind of a stream of data, a stream of text data. I don't know how much sort of text has been talked about here, but it is

messy. Our audience isn't a technical expert, per se, but the general public, and how you create kind of data analyses bases, in a way, that speak immediately to the general public.

Some other applications that we have been working, we have begun a joint project with Bill Seaman, who was at UCLA and now is at RISD, jointly with UCLA. We are tiling a large room at UCLA with these sorts of condensed networks. So, it is a sensor network, which I have heard people already talk about, but we will have sort of wonky things like speech recognition and temperature and things on these sensors, for an inside of the room. Then, all of them will report wirelessly back to a central place, where we will be dealing with the streams. We have also looked with some Lucent people at perhaps network operations. When we first started this, we were talking to some people in the manufacturing lines.

An interesting application, we were approached by an architect, Rem Koolhaas, to help for a building he was putting up at the IIT. The idea was, he was going to give us access to just streams of facilities data, occupancy sensors, data from boilers and what have you, and that we would create a sound in the foyer of this building. With repeated exposure to the foyer of this building, people would be able to know exactly what was going on in the building, just by the sound when they walked in. This is sort of a look — it is sort of a wonky artistic look at what the building is supposed to look like.

There is a bowling alley in the space and a store, and we were going to get access to all of those data in real-time. Anyway, I guess I should summarize because I need to get on a plane, because tomorrow we have to start putting up, now that the carpet people are done.

So, it began as a collaboration between, somehow, the arts and sciences, and there was an interesting interplay of viewpoints between the two. What I am finding is that there is a community of sort of artist folks who are trying to reach out for collaborations with the scientists. The collaboration is really highly valued and they are looking at and studying, kind of, how collaboration works and why it does, and how you can make it successful and how you promote both art and science through that collaboration, and not have just sort of one side of things. My work with Ben has been motivated, in a way, by new and complex data sources, large quantities of data—I suppose that is the massive part—and there is a strong time component. I guess that is the streaming part.

Ultimately, the goal is to create kind of new experiences with data and, in particular, to create public displays of data that somehow speak to the general public.

With that, I will thank you for spending your digesting time with me.

AUDIENCE: How long are you at the Whitney?

MR. HANSEN: Three months. We open the 17th and we are up until March 9. If anyone—I have invites to the opening, if anyone would like to come. The opening is on the 20th, the party.

AUDIENCE: [Question off microphone].

MR. HANSEN: That is exactly the—that kept coming up again and again. We had a formal critique, actually, with some curators at MOMA and some artists, and that

was extremely important to them, that it be live, and that it be—that the process from data to display was legible, and you weren't kind of tampering with it.

There was some notion of unbiased-ness that came out there that they didn't really have the words for, but it was definitely there. There is no filtering. It is funny. Even if there is no filtering, the bad words don't get people. It is the really bad thoughts, somehow. Chat is like really bad talk radio, but it is even more anonymous, because you don't even give your voice. So, you can type just any horrible thing that is off the top of your head. That is what seems to get people, when something comes across that is just like really hateful. It is not even clear to me how you would filter for that kind of hateful stuff, because it is not just the four letter words or whatever, which would be easy.

AUDIENCE: In terms of interaction with the public, what sort of things have — [off microphone.]

MR. HANSEN: There were some things, actually. We have another scene that we have just put in. So, this documentation video has four scenes, one that goes dee, dee, dee, dee, dee, one that has the spinning and the talking. There is another where we kind of blast along the streams, and then we have another that just gives a listing of the user names. We have added a few more. One of them was looking at how—every few hours we look at the most frequent ways in which people start their posts. Inevitably, aside from welcome back, which we kind of toss out—everyone gets stop lists. So, we toss our welcome back.

After that, it is I'm or I am, is the most frequent way that people start their posts. So, we have started kind of a litany of those as one of the scenes. Our host in Seattle, who is this sort of semi-jaded art curatorial type, was in tears over this scene. I wasn't prepared for it. You know, you kind of present this in a kind of human way because, at the end of the day, it is about people communicating. If you present this in a reasonably straightforward way, I think it has an impact, and that sort of surprised me. I should say, in Seattle, a very strange thing happened. So, we were there for three weeks. The first week we were just setting up. The second week, we were open to the public, and the third week we were open to the public. The third week, we got written up in the *Seattle Times* and what have you, but we started sort of marching up this crazy attendance curve. Like, the Wednesday, we had like 50 people and the Thursday it was 90 and the Friday was 150 and the Saturday was 311 and the Sunday it was 652, who came and just sat in this room for an hour, hour and a half, at a time. It blew my mind. People would come up afterwards and tell me what they thought it was doing and what they thought about it, and that was very surprising to me, that it would be sort of well received like that. It made me very nervous, at the same time.

AUDIENCE: [Comment off microphone.]

MR. HANSEN: I heard someone mention anomaly detection earlier. You people talked, could you use this to scoop up lots of chat and then find a terrorist or something. I think our approach has been to like sample and give a big picture, like what is sort of the broad—I don't know that there is any practical use for it, really. I mean, there is a lot of

data analysis that I think is interesting to pursue but, like practical, I don't think so.

AUDIENCE: I guess I disagree with that. If you think of an analyst that has to try to assimilate all the information that is coming in, if you are actually moving in a direction— [off microphone.] —options that they have, to make it easy for— [off microphone.]

MR. HANSEN: We thought about it. For those sorts of systems, like for the sonification, we thought, would be a natural for that, because you could hear a process drifting before an alarm would be set off in a statistical process, like in a control chart of some kind. So, we thought about that and we kind of toyed with that. Then we were quickly derailed with this text stuff and kind of went off in a different direction. I think that is an application, that you will be able to hear shifts in the background. Even something you are not directly attending to, you will be able to hear kind of shifts. So, for real-time process monitoring, I think it has got some applications for sure.

AUDIENCE: [Question off microphone]

MR. HANSEN: We do that all the time with our laptops and say, oh, this is a problem.

AUDIENCE: I would point out that the accelerator— [off microphone] —if it deviates one little bit, you notice it. If it is something phenomenal, you would hear that.

MR. HANSEN: There is—I mean, we do a lot of analysis in the—we do a lot of information gathering with our auditory system in our regular life. We start up the car and immediately we know if there is a problem. I mean, we are doing that all the time. The question is, can you apply that in a data sense.

AUDIENCE: I was wondering if you had spoken with some of the people who are working on sonification of things like Web pages, and mathematics.

MR. HANSEN: We have been to a couple of these ICAT meetings. So, there is an international community for auditory display and they have a meeting every year in some very exotic places. When it was in Finland, I remember there was a session—it was crushing to see how primitive that technology was, about how blind people were forced to kind of mouse over a page until they hit the radio button or something. It was horrifying to see what kind of state of the art there was at that point. That is a huge area for work that I don't know who—David and I were at some digital libraries meeting at Dimax.

I think one of the things that—we were supposed to propose things that people should be doing as more and more data libraries are keepers of more and more digital data. One of the things we were pushing for was assistive technologies like that. Horrifying is the word, the kinds of things that people have to do. Maybe I am more sensitive to it, because my mom is slowing losing sight. I am trying to get her to use speech recognition and things like that, and it seems like a really good research area.

AUDIENCE: Different kinds of voices, I didn't hear any different voices— [off microphone] —voices in different language.

MR. HANSEN: It is the typical kind of male engineer response when you go, well, why aren't there any women voices.

So, we asked the people who make the text-to-speech engines. Well, we have had a lot of problems with her. They can't like get it. I mean, they have a lot of the—there are two sort of voice qualities. At the high-quality end, they only have the male voice. At the low-quality end, they have several males and several females. We really wanted the high-quality one, because it just sounds so much better. They have one now that we just started getting working as we went to Seattle. We are hoping we can get it wedged into the Whitney show. That was one criticism that we had. When you get these 45—even though they are British inflected—when you get these 45 voices going at once, it is a very male space, and sometimes it can be very heavy. The female voice is quite nice. It sounds something like an NPR announcer. She just keeps crashing. She will like crash the—we had a problem with the male voice initially—actually, this is a nice story. We had a problem with the male voice and that is that it would stay up for—we had a set of test data. We weren't running it on live data. We had a set of test data. Inevitably, after two hours it would crash. Just before we were going to Seattle, this kept us debugging and working and figuring out. We had, you know, the engineers from the Lucent speech thing. I mean, they were like in the studio until like 2:00 and 3:00 o'clock in the morning.

In the end, it was the word, abductor. There was something about the word abductor that brought the whole thing down. They had to bring in some new thing or whatever. I thought it was beautiful that it was the word, abductor. It kept us in the studio for a very long time. There was a fix, and we think something like that can fix the female voice, but as of last Thursday—because these things always happen on Thursday—the last of the text-to-speech people at Bell Labs were laid off. We are hoping that we will be able to get something like that going. They have French, they have Italian. They have Spanish. We stayed away from other languages, because I can barely speak English. So, I can barely do what I need to do and see that it is working right in English, much less in these foreign languages. There is that capacity. If, somehow, we find someone with a better grasp of languages, we can try that.

AUDIENCE: This being a collaboration with artists, was there any piece that made it really difficult to understand certain parts, given the real mathematical sorts of things —

MR. HANSEN: We are a slightly weird pair. I took a lot of art classes at undergraduate. My collaborator took a lot of computer science classes as an undergraduate. To the extent that kind of—the stats on the computer science I can find someplace to overlap. We did have some very difficult discussions around the concept of sampling. In fact, this came up at the critique, where the curators kept using the word privilege, why are some data points privileged over some others. It is not that they are privileged—it is sort of a hard thing to get over. We had some really sort of rocky evenings where we had to explain, we don't need to take all of the data and throw it at some collection port. UDP protocol has no guarantee. So, packets will get dropped all over the place. So, rather than sending sort of a million messages at this poor port and just grabbing whatever sticks, we could send as many as we need—that was a concept

that was just really hard to get through. Then, like I said, there was this privileging concept, and legibility seems really important. Initially, there wasn't a lot of trust—not my collaborator, but the curators and stuff—what is this all doing. Something like this would be too easy to twist. If we only went to like Aryan Nation Web sites, the thing would have a totally different character than it does now.

So, there has been—the other thing I have noticed, and I am sorry to be kind of yammering—but the other thing that I have noticed is that these media artists are a lot more savvy technically than we give them credit for, maybe not statistically, but software and hardware wise, they run circles around them. Not my collaborator in particular, but a lot of them will run circles around us. That is kind of why—so, my position in UCLA that I am starting in April, is joint between media arts and statistics, and I will be teaching joint classes. I think that it will be interesting to have media arts students with stats students, in the sense that the stats students aren't going to have the same kind of computing skills that the media arts students will, and the art students just won't know what to compute.

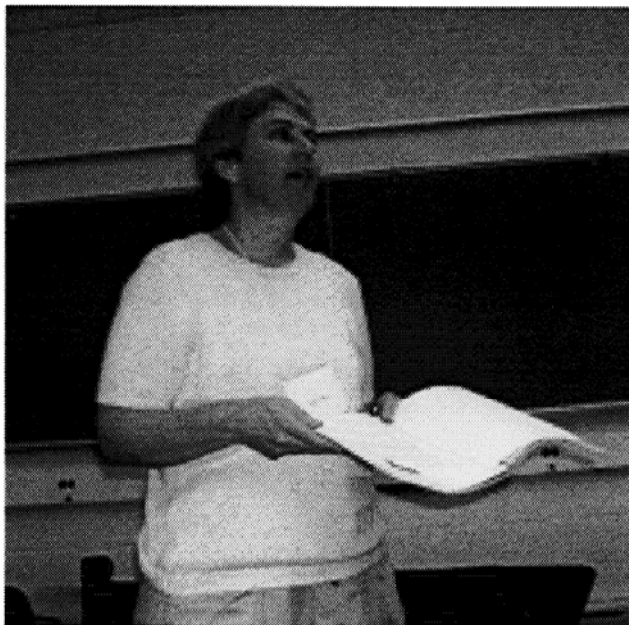
So, it is going to be kind of an interesting interplay, I think, of approaches to problem. Introducing to both a group of media arts students and statistics students the concept of a database and how a database works and all that, there is a big thrust now about database aesthetics, not just the politics of databases, but there is an aesthetic to them as well. So, I think that that is going to be kind of interesting. I suppose the last comment I want to make is that my collaborator has this interesting—everything should be doable, and that kind of pushes me a little farther. Of course, we should be able to string these 231 displays and, of course, we should be able to update the entire grid 25 times a second. That has been the other thing, that of course we can do it and we just haven't hit kind of the limits yet.

Thank you for your time.

Wendy Martinez, Chair of Session on Network Traffic

Introduction by Session Chair

Transcript of Presentation



Wendy Martinez is a scientist in the Probability and Statistics Division at the Office of Naval Research.

TRANSCRIPT OF PRESENTATION

MS. MARTINEZ: Welcome back to the second day of the workshop on massive data streams. Without further ado, I will introduce the first speaker, who is Bill Cleveland. I am not going to take up any of his time. So, we will turn it over to him.

William Cleveland

FSD Models for Open-Loop Generation of Internet Packet Traffic

[Abstract of Presentation](#)

[Transcript of Presentation and PDF Slides](#)



BIOSKETCH: William Cleveland is a distinguished technical staff member at Bell Laboratories. He specializes in Internet engineering, visualization, model building, visual perception, consumer opinion polling, and Bayesian statistics.

His professional service has included editorial positions with *Technometrics*, the *Journal of the American Statistical Association*, the Wadsworth Probability and Statistics Series, and *The Collected Works of John W. Tukey*. He is a former member of CATS.

Dr. Cleveland received his PhD in statistics from Yale University. He is the author of two books on visualization and analysis of data: *The Elements of Graphing Data* (Chapman and Hall, 1994) and *Visualizing Data* (Hobart Press, 1993).

ABSTRACT OF PRESENTATION

FSD Models for Open Loop Generation of Internet Packet Traffic

William S.Cleveland, Bell Laboratories (with Jin Cao and Don X.Sun)

Abstract: The packet traffic on an Internet link is a marked point process. The packet arrival times are the point process, and the packet sizes are the marks. The traffic arises from connections between pairs of computers; for each pair, the link is part of a path of links over which files are transferred between the computers; each file is broken up into packets on one computer, which are then sent to the other computer where they are reassembled to form the file. Packets arriving for transmissions on the link enter a queue. Many issues of Internet engineering depend heavily on the queue-length distribution, which in turn depends on the statistical properties of the packet process, so understanding and modeling the process are important for engineering. These statistical properties change as the mean connection load changes; consequently, the queuing characteristics change with the load.

While much important analysis of Internet packet traffic has been carried out, comprehensive statistical models for the packet marked point process that reflect the changes in statistical properties with the connection load have not previously been developed.

We introduce a new class of parametric statistical models fraction sum-different (FSD) models for the packet marked point process and describe the process we have used to identify the models and to then validate them. The models account for the changes in the statistical properties through different values of the parameters, and the parameters are modeled as a function of the mean load, so the modeling is hierarchical.

The models are simple, and the simplicity enhances the basic understanding of traffic characteristics that arise from them. The models can be used to generate synthetic packet traffic for engineering studies; only the traffic load and certain parameters of the size marginal distribution that do not change with the load need to be specified. The mean load can be held fixed for the generation or can be varied. FSD models provides good fits to the arrivals and sizes provided the mean connection load—the mean number of simultaneous active connections using the link—is above about 100. The models apply directly only to traffic where packets on the link input router delay only a small fraction of the packets, about 15 or less; but if delayed traffic is needed, it can be very simply generated by putting the synthetic model traffic through a queue.

C code is available for generation as well as an implementation in the widely used NS-2 simulation system.

TRANSCRIPT OF PRESENTATION

1

FSD Models for Open-Loop Generation of Internet Packet Traffic

J. Cao, W. S. Cleveland, and D. X. Sun
{cao, wsc, dxsun}@bell-labs.com

Further reading
<http://stat.bell-labs.com/InternetTraffic>
google: bell Labs internet traffic

2

Outline

Background

- Internet technology
- Internet traffic, packet arrivals and sizes, a marked point process
 - previous work on the statistical properties
 - importance of modeling the statistical properties for engineering the Internet

Results

- new class of statistical models
- provide excellent fits to packet marked point process
- use of models in Internet engineering studies
 - provide basic understanding of packet traffic for qualitative conclusions
 - reverse a critical conventional wisdom after traffic
 - generate synthetic packet traffic for quantitative conclusions
- code for generation: (1) C programs (2) implementation in NS-2, widely-used network simulator from LBL

Tools and Strategy (Not Discussed)

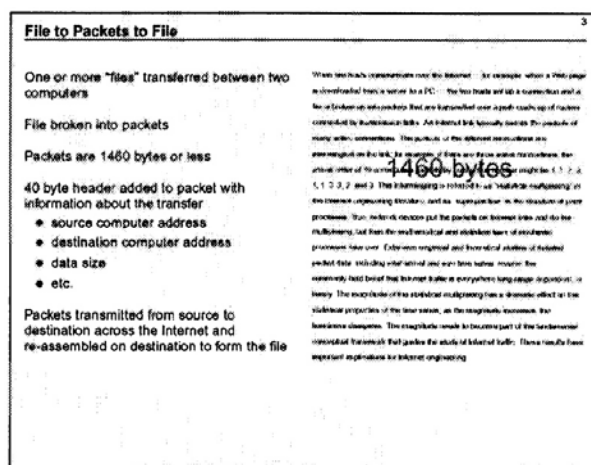
- S-Net hardware/software system for Internet traffic study

MR. CLEVELAND: Thanks. I am going to be talking about statistical models for Internet packet traffic. The models contribute in adding to the basic understanding of Internet packet traffic, which was very much in need of understanding, and also provide a mechanism for open-loop generation of packet traffic for engineering studies.

So, what I am going to do is, I am going to start out by giving you just some information about Internet technology. I need to do this. Otherwise, the issues that I will be raising and tacking into modeling just won't be clear. So, we will go through that. I will talk, then, about packet arrivals and sizes, which constitute the traffic, as we have modeled it. It is a mark point process. I will describe previous work in looking at the statistical properties of this traffic, and also I will describe the importance of the modeling of these properties for engineering the Internet.

I am going to tell you about a new class of statistical models that provide very good fits to the packet mark point process. As I said, they do provide help in a basic understanding of packet traffic. They also help to reverse a very critical central conventional wisdom about packet traffic, and can be used to generate traffic for quantitative conclusions.

We have C code, and we have also implemented this in a very widely used NS-2 network simulator that is used by the Internet research community.



So, the technology. Internet communication consists of transfers of files between pairs of computers. It is transferred in the following way. The file, sitting on the source computer, is broken up into packets. These packets are 1,460 bytes or less.

Here is an abstract of a talk I gave—

AUDIENCE: Bill, is there any discussion of increasing the— [off microphone.]

MR. CLEVELAND: No, that is the max, that is the rules.

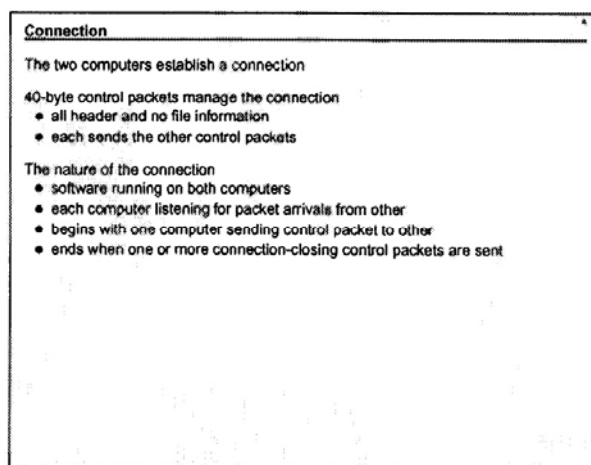
AUDIENCE: It is such a Stone Age thing. Technology has changed.

MR. CLEVELAND: Okay, well, the answer is no. There is no discussion of that whatsoever, because we have got millions of computers out there sitting in peopled homes where that is the max. So, there is a lot of inertia. One doesn't hear that, no.

Okay, so, what happens is, the file is broken up into these packets, 1,460 bytes or less. Here is an abstract of a talk of reasonable size. That just fits into 1,460 bytes. So, if you want to calibrate how much information goes in, it is a reasonable abstract's worth of information.

To this packet, a 40-byte header is added. That header has a lot of information about the transfer. It has the source computer address, the destination address, the size of the data going into the packet—the amount of information in the packet, and a host of other variables.

The packets are transmitted from the source to the destination, across the Internet, and then they are reassembled on the destination computer.

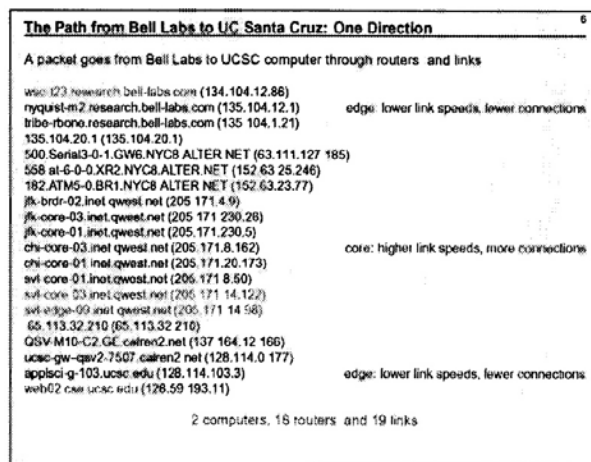
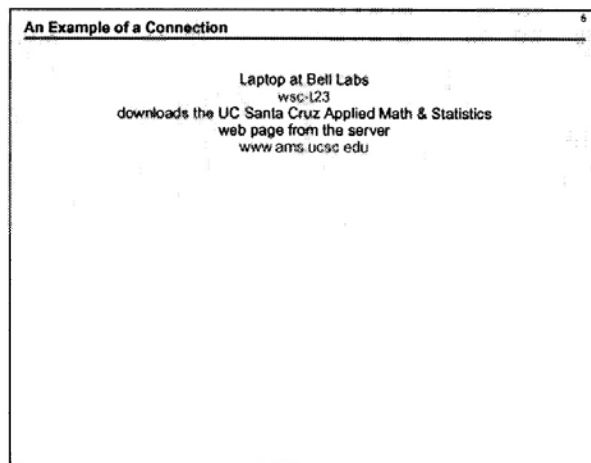


Now, to carry out this transfer, the two computers establish a connection. That is the word that is used. In addition, 40-byte packets, which are all header, and no data, are

used to manage the connection.

Each sends the other control packets. The nature of this connection is just simply software daemons running on the two computers, listening for arrivals from the other computer.

It begins when one computer sends a packet to the other saying it would like to open up the connection, and it ends when control packets are sent back and forth between the two computers, agreeing to close the connection.



Let's take a look at an example here. So, I am sitting at Bell Laboratories with my laptop, actually, this one right here, and I download the UC Santa Cruz applied math statistics Web page, because I am going to be going there to give a talk and I need to see directions to drive down from Berkeley. So, I download that Web page and the connection is set up. Here is what the connection looks like. So, here is my laptop sitting here at Bell Laboratories.

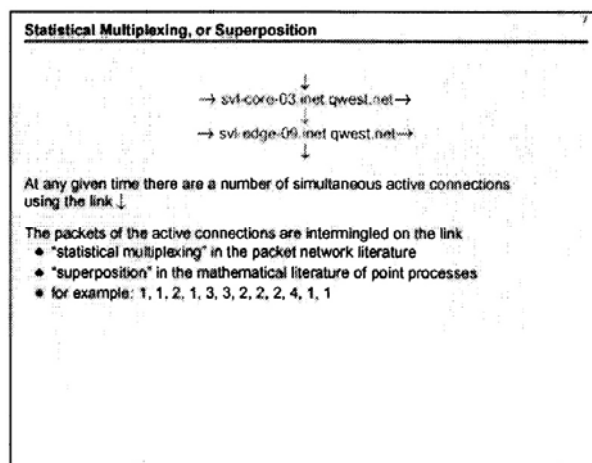
A packet coming from my laptop and going to UC Santa Cruz goes through this series of routers. So, what we are seeing here are devices which are routers, and every successive device here is a link, over which the package is sent.

So, we start out here at my computer. Here is a first router at Bell Laboratories, and then we continue on and hit WorldCom routers, then we go through Crest routers, and then we hit the California Educational Network, and then finally get to UC Santa Cruz.

So, there are two computers here—my laptop and their server—there are 18 routers in between us, and there are 19 links connecting these devices.

Now, when it first starts out, the packet is at the edge of the Internet. The link speeds are low. There are fewer connections using the links at any given time, just less traffic generally.

As we move into the core and hit the service providers, the link speeds are going up, the usage of the links goes way up. So, this is the core, higher link speeds, more connections. Then, as we travel on farther, getting close to UC Santa Cruz, we are at the edge, lower link speeds, fewer connections.



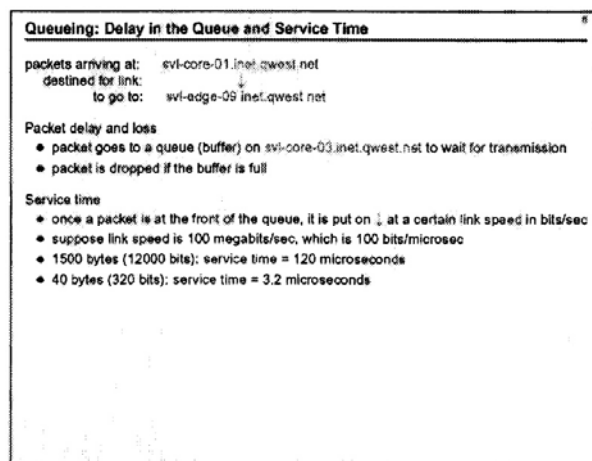
What I want to do, I want to take a look at two routers here. This one, svl-core, and svl-edge, and let's take a closer look at them.

So, we have packets coming in on different links, and then we have packets going out here on this link, and then this particular link here that goes to the next router that is in the pack from my packets.

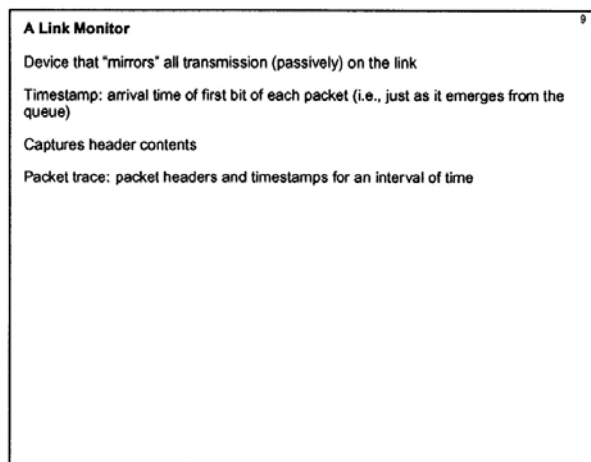
Now, at any given time, on that link, there are a number of simultaneous active connections using the links, not just mine, of course. There are others sharing the link.

The packets of the active connections are intermingled on the links. So, if we — say, down here, let's number the connections one through N . So, a packet comes from one and then another from one, and then from two and from one and three and so on. So, they are intermingled on the link.

In the packet network literature, they describe this as statistical multiplexing. In the statistical literature, they refer to it as superposition. You might think it would be the other way around, but it is not. I will tend to use statistical multiplexing in this talk.



Now, the packet is arriving at the first svl-core router, destined for this router over this link in question. They go into a queue, which is a buffer, to wait for transmission. Of course, if there are no other packets in the queue, they just go right off. There is no waiting. If the buffer is full, the packet is dropped.



Now, once a packet gets to the front of the queue, it is put on the link at a certain link speed. Actually, link speed is the wrong word. I don't know why anybody ever invented that word, because it is not the speed of the link. I mean, things go down the link at the speed of light.

It is actually the speed of the device. It is the speed at which the device puts the bits on the links. So, it is really a device transmission speed.

Now, supposedly the link speed is 100 megabits a second. This is a common speed that you would find in Ethernet networks, local networks, perhaps at your university or work location.

So, 100 megabits per second, that is 100 bits per microsecond. So, if we take a maximum-size packet, which is 1,500 bytes or 12,000 bits, the service time to put the packet on the link is 120 microseconds.

For these control packets, the smallest possible packets, the service time is 3.2 microseconds. So, we have got a queue. That is a key issue in Internet technology. We will get back to that shortly.

Now, if we want to understand Internet traffic, one extremely useful and effective measurement scenario is to attach a device to a link. So, we want to understand the packet traffic on a link. We attach a device to the link that mirrors all the transmission.

When a packet arrives—that means when the first bit goes on the link, you write a time stamp, so you know its arrival time, and you capture the header contents. Now, some people might want to capture the content. We don't, and most people doing Internet traffic research don't capture content. We don't want to know what is in the packets. A lot of it is encrypted anyway. It is not our business, and it doesn't contribute to the kinds of engineering questions that need to be asked. Well, some of it actually would, but we still don't get it. It is too tricky.

Our Study: Packet Trace Database 10

Traces from 17 links

17 trace durations: 1 hour to 4 years

Link speeds: 100 megabits/sec, 156 megabits/sec, 622 megabits/sec, and 2.5 gigabits/sec

Highest packet rate \approx 80,000 packets/sec

For this presentation

- 2072 sub-traces from the 17 links
- 15 sec to 5 min in duration for within-trace stationarity

A packet trace consists of the headers and the time stamp. That is the term that is used. So, the data are a packet trace on the link over some period of time. Now, in our study, we have put together traces from 17 links around the Internet. The durations of these traces from the 17 links are from anywhere from one hour to four years.

The link speeds go from 100 megabits per second to 2.5 gigabits per second. The highest packet rate of collection is this 2.5 gigabit per second link, which is the link of a service provider out on the West Coast. The rate there is 80,000 packets per second.

So, for this presentation, I am going to be showing you a couple of pictures that show 2,072 subtraces from this collection of traces. They are broken up from 15 second to 5 minute intervals duration, because that is a standard way we analyze it, because we want to break things up to keep characteristic stationary within the interval of study, and then watch how that changes through time.

QUESTION: How did you pick the links?

MR. CLEVELAND: We picked the links to get a diversity of traffic rates. So, when we started studying them, we started at the edges. The first data we got we collected ourselves, and we started analyzing that.

There were lots of characteristics that were quite interesting, but we realized that, if we wanted to really understand the traffic, we had to be in places where the amount of multiplexing was much higher, because we started seeing things higher, as the number of active connections increased.

So, that was a major factor. We also picked the links—sorry, some of these data are already existing measurement programs. It is not as if we went off and measured a service provider.

We began establishing relationships with people who actually do Internet traffic measurement, and we knew we needed to get to links where, as I said, we knew the links were high. Also, we needed to get data that were extremely accurate, because we pushed the time stamp accuracy to a very high degree, and much that we do is highly dependent on that.

QUESTION: So, everything you are analyzing is a high-accuracy time stamp?

MR. CLEVELAND: The accuracy actually varies. It is sufficiently high, yes, for the kinds of things we need to do. We threw out a lot of data, like data from Harvard. Maybe I shouldn't say that. Anyway, people were collecting data at Harvard, and we looked at it and we said, this just won't do.

The UNC data doesn't have especially accurate time stamps for us to do our work, and we knew that when we set it up. We knew it would not have it.

I mean, there are other purposes you can use these data even without the highly accurate time stamps. There are a lot of other tasks that can be carried out if your time stamps are actually quite inaccurate, because there are other things to be done.

AUDIENCE: You also— [off microphone.]

MR. CLEVELAND: The packet headers contain information about the order. I mean, they have to, because when it gets on the receiving computer you have to collect them and put them back in the correct order. So, the packet headers carry all that information. So, we have that, yes.

AUDIENCE: [Remark off microphone.]

MR. CLEVELAND: I am sorry, I am only talking about one specific problem here in terms of when one analyzes Internet traffic. I mean, there are a whole host of problems that one can attack. I am telling you about one specific, very important, but one specific problem. So, these variables don't represent all the variables that one can get off the packet header.

Traffic Variables for a Link: $a_u, t_u, q_u, c, p_i, b_i$ 11

Packet arrivals and sizes are a marked point process

Arrival number u : $u = 1$, first arriving packet; $u = 2$, second arriving packet, . . .

| | |
|---|--|
| Point process | Marks |
| <ul style="list-style-type: none">• a_u: arrival times (sec)• $t_u = a_{u+1} - a_u$: inter-arrival times (sec) | <ul style="list-style-type: none">• q_u: sizes (bytes) |
| Connection load (number) | Packet and byte counts |
| <ul style="list-style-type: none">• c, measure of magnitude of multiplexing (number)• at a given time: the number of active transport connections• over an interval: the average number over the times in the interval | <ul style="list-style-type: none">• divide time into equal intervals, $i = 1, 2, \dots$ (e.g., 10 ms)• p_i = the number of a_u arriving in the ith interval (packets/sec)• b_i = sum of sizes of packets arriving in ith interval (bits/sec) |

Measurements of the variables from packet traces.

All right, the packet arrivals and sizes are a mark point process. The arrival number—I will let the u be the arrival number, $u=1$, that is the first packet, the second packet is $u=2$, and so forth.

So, I will let a_u be the end arrival times, and $t_u = a_{u+1} - a_u$ are the end arrival times.

I will let q_u be the sizes. So, q_u is the size of the packet arriving at time a_u . Now, we are going to need some other variables here, also, in addition to this.

I need a measure of multiplexing, a magnitude of multiplexing, and here is one measure we use. There are several of them, but here is one. It is just simply the number of connections that are active at a given point in time. Now, over an interval of time, if we want a summary measure for a trace—say a 15-minute trace—then you just take the average of the number of active connections over the 15 minutes. So, that will be c .

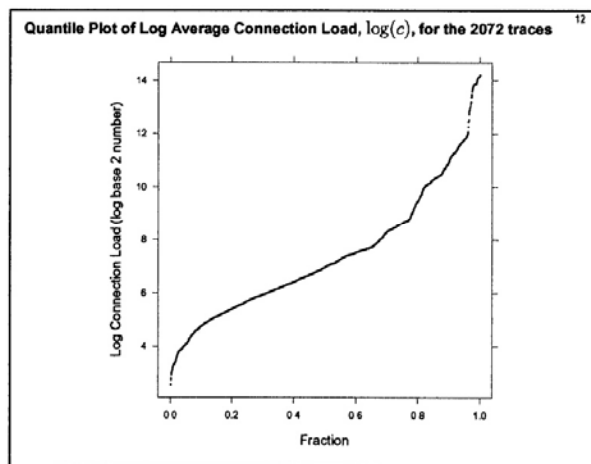
AUDIENCE: So, you can't know that in real time.

MR. CLEVELAND: Actually, you can know it in real time. You have a time. The devices like firewalls have to keep state on connections. So, you do your best to try to figure it out. You can do better off-line, of course, but you just do your best if you have to do online things like security.

So, here are the traffic variables that we are going to be studying today, a few of many.

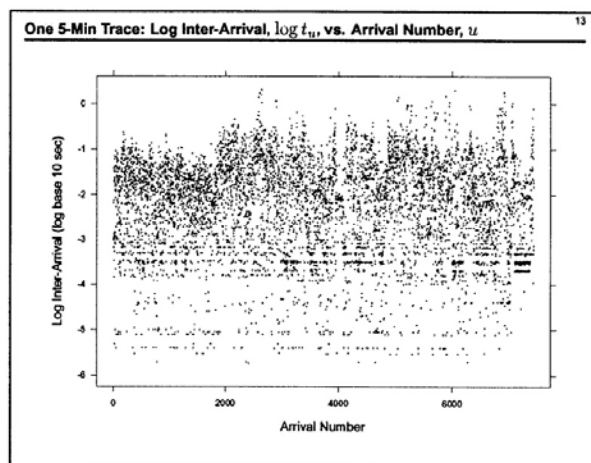
I told you there were 2,072 traces I was going to be describing to you. Let's take

a look at the connection loads.



So, I have got the log of the average active connection load for 2,072 traces. So, this is log base two, and this is a quantile problem.

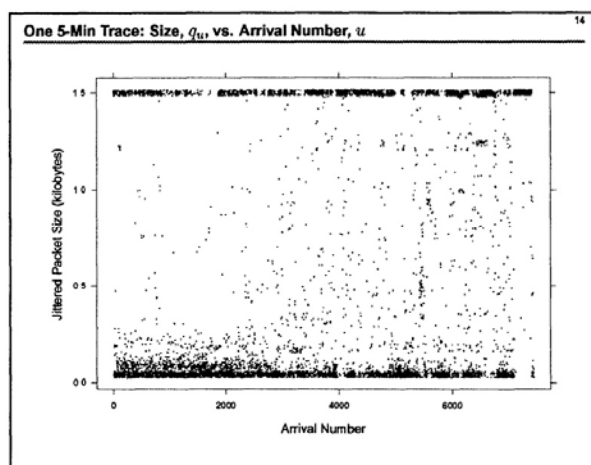
So, we are going from 2^4 , actually 2^3 , about 8, up to 2^{14} , so that is 16,000. So, we are going from an average of 8 connections, active connections, at any given time up to 16,000. So, we have got a very wide range of traffic rates. Again, that was a goal in being able to get data for this purpose here.



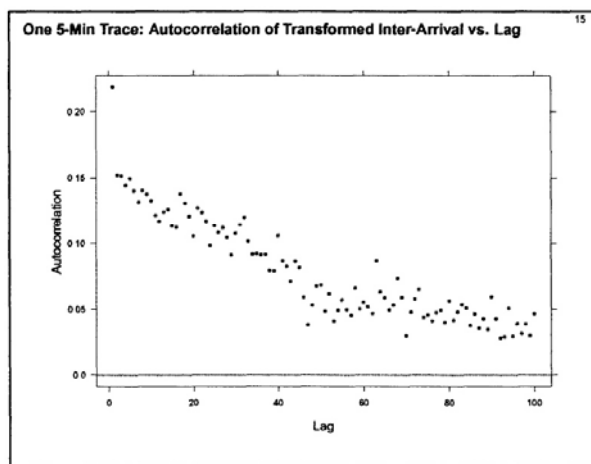
Here is one of these traces, a 5-minute trace. So, I have taken the log base 10 now, or the end arrival time, and apply it against the arrival number.

So, it is something like 7,000 packets arriving during this particular 5-minute interval on this link. So, you see the log goes from a little bigger than -6 . So, 10^{-6} would be a microsecond. So, the smallest end arrival time is—well, it is actually 3.2 microcycles. It is the arrival time of the smallest packet on the link. It goes up to about a second.

So, we are going through nearly six orders of magnitude in these end arrival times. We do have accuracy in this particular case down to that level.



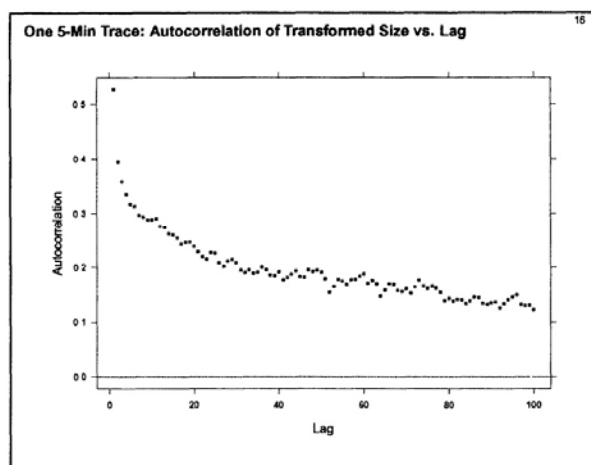
Here are the sizes, the packet sizes plotted against the arrival number, except I have had to take the 40-byte packets, of which there are many, and the 1,500 byte packets, of which there are many, and jitter them a bit, so that we get some better resolution on the plot.



So, here they are, plotted against time as well. Now, if you look at it you say, gee, that actually looks fairly random.

If you look a little closer, though, you see they are actually bunching up together, aren't they. So, we get end arrival times that seem to come in little bursts here, and there are sort of striations that you can see. So, just from this time plot, you see that there must be time relationships, time correlation. Well, it looks noisy. Anyway, let's stop torturing ourselves. Let's look at the autocorrelation function. Here is the autocorrelation function of the log end arrivals.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



What you see is that the correlations are all positive, and they fall off very slowly. So, we don't have too much correlation at lag one, but we still have a fair amount left at lag 100. If we looked at lag 500, we would still see a fair amount of correlation. The same is true of the packet sizes.

One 5-Min Trace: Time Dependence 17

Long-range dependence

- $t_u, q_u, p_i,$ and b_i
- autocorrelation positive
- autocorrelation decreases slowly with lag k , like k^{2d-1} for $0 < d < 0.5$

Arrivals would be Poisson if the inter-arrivals were

- independent
- identically and exponentially distributed

The arrivals a_u are not Poisson

The sizes q_u are not independent

The sizes in the air arrivals are long-range dependent. That is a critical factor of the Internet traffic that has a major impact on the engineering of the Internet.

So, let's take this one 5-minute trace. I am sorry, I actually didn't define for you the byte count. Actually, I am not going to be looking at those variables, but I do have an important comment to make about them.

So, the packet counts, you take an interval, say, of 10 milliseconds, and you count the number of arriving packets in 10 milliseconds through time. So, you are getting counts rather than arrivals and sizes.

For the byte counts, a similar process, except that, instead of just counting the number of packets during the interval, you add up the sizes of all the packets arriving during that interval. So, packet counts and byte counts, they are all long-range dependent, sizes and arrivals by packet counts.

The autocorrelation falls off slowly, like K^{2d-1} between 0 and .5. Keep in mind the arrivals would be Poisson, if the end arrivals were independent and identically and exponentially distributed.

The arrivals are not Poisson because they are neither independent, nor are they, at least on this particular 5-minute trace, nor are they exponential, and the sizes aren't independent. So, nothing is Poisson and independent.

Why Do We Need to Model the Traffic Marked Point Process? 18

Quality of Service

Packet queuing delay and service time add to transfer time

Packet loss adds to transfer time because drops often cause source to reduce packet sending rate

Delay and loss

- depend on the statistical properties of the packet marked point process
- much larger for long-range dependent traffic than for Poisson and independent

One QoS Problem: Bandwidth Allocation
How much traffic in λ bits/sec can be put on a link of speed C bits/sec for fixed loss and delay criteria?
Need statistical model of traffic to study this

Now, why do we need to model the traffic mark point process? The packet queuing delay and service time add to the transfer time of files on the Internet. So, when you click on a Web page and it takes an immense amount of time to arrive, it can well be that packets are being delayed in routers along the path. It could also be that the server is just slow. That is another source of pages slowing down on the Internet, but the congestion, when it gets bad, it gets really bad. So, that is a major factor in how long it takes you to get a Web page.

Packet loss, if it is full and the packet gets dropped, it is even worse. The sources detect this in many cases, and start slowing down the sending rates of the packets.

Now, all this depends on the queuing characteristics. The queuing characteristics depend on the statistical properties of the mark point process.

The queue links are much longer for dependent, long-range dependent traffic than they are, for example, for Poisson and independent.

For example, one QoS problem is bandwidth allocation. If I have traffic at V bits per second—sorry, if I have a link speed of L bits per second, how much traffic in bits per second can I put on that link for a fixed loss and a fixed delay as quality criteria? So, this is a problem that needs to be attacked, and depends heavily on the statistical properties of the traffic.

Previous Results 19

Largely a history of packet and byte counts

- long-range dependent
- driver for intuition, theory, and empirical study
- with enough aggregation, a gaussian time series
- a data reduction method
- widespread wavelet modelings
- fractional Brownian motion a popular model

Arrivals and sizes, the marked point process

- almost no empirical study as time processes
- long-range dependence of inter-arrivals and sizes eventually established
- a few wavelet modelings
- no comprehensive generation model

The history of Internet traffic study has largely been one of the study of packet and byte counts. As I said, they are long-range dependent. This has been historically the driver for intuition theory and the driver for empirical study. With enough aggregation,

of course, the packet byte counts are a Gaussian series.

So, they make some things easier. It is also data reduction methods, so you go from gigantic files to smaller files, and for some people, that is an advantage.

There has been widespread wavelet modeling of packet byte counts, and fractional Brownian motion is a popular model.

Arrivals and sizes, the mark point process, the real thing that the routers see, very little empirical study as fine processes.

Enough that the long-range dependence has been established, as you can see almost immediately when you look at the data, a few wavelet modelings, but no comprehensive generation model, and that is what we set out to fix.

Multiplexing, what happens when the rates go up? The conventional wisdom arose, even though there was almost no study of what actually happens, empirical study and conventional wisdom arose that said that the long-range dependence was unabated or even magnified.

20

Previous Results

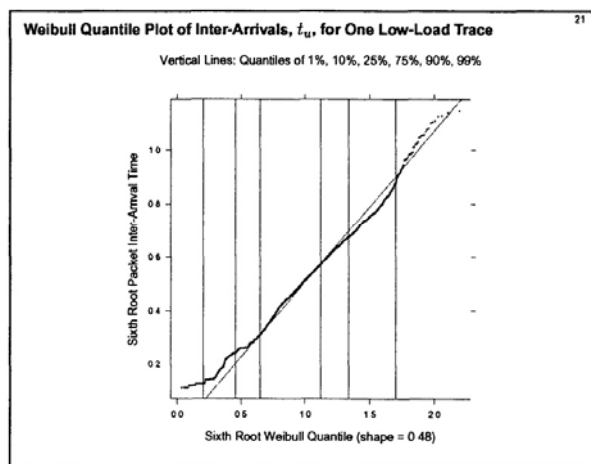
Multiplexing: the effect of increasing connection load on packet traffic statistics

- some theoretical study
- virtually no empirical study

Conventional wisdom: long-range dependence unabated or even magnified

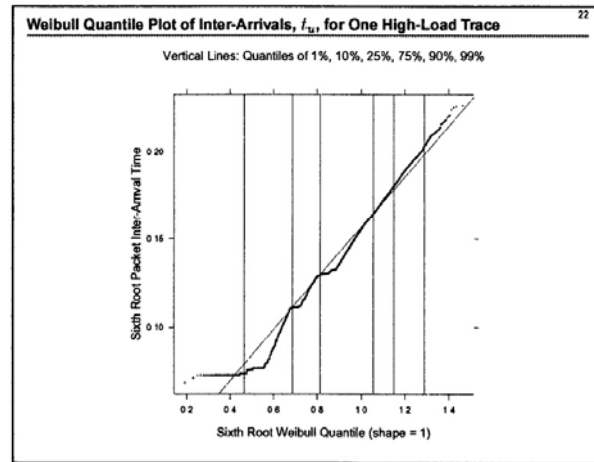
IEEE Communications Magazine, 2000:
... traffic on Internet networks exhibits the same characteristics regardless of the number of simultaneous sessions on a given physical link.

For example, in IEEE *Communications* magazine in the year 2000, there was a statement by somebody talking about architecting the future Internet, traffic on Internet networks exhibits the same characteristics, regardless of the number of simultaneous sessions on any given physical network. That was the conventional wisdom that dominated engineering design, both network design and device design.



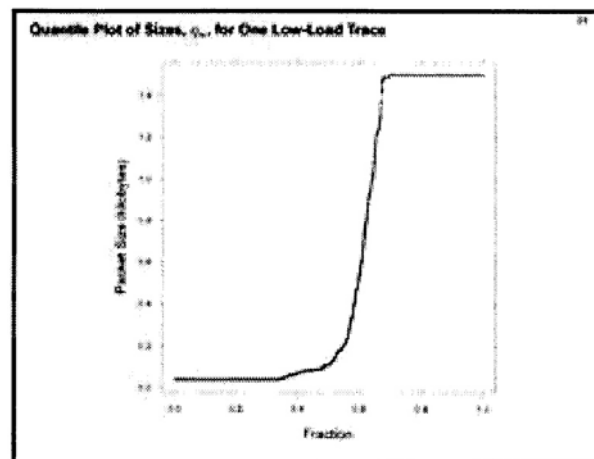
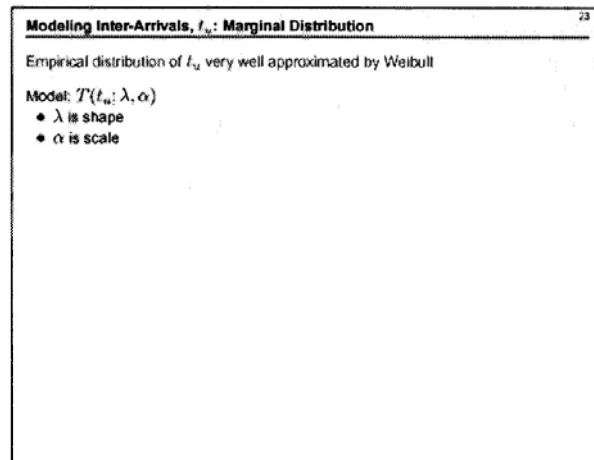
So, let's start doing some modeling. Here is a Weibull quantile plot of the end arrival times of one particular low-load trace.

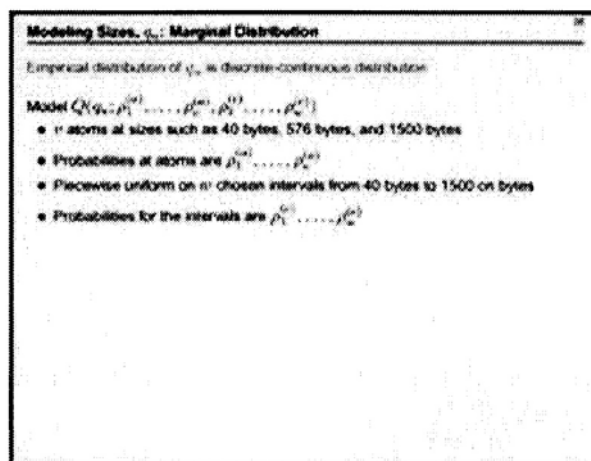
So, here is the sixth root taken because they are highly skewed and that sort of messes up the plot, so here is something to just make the plot work and make the data more nearly symmetric.



So, sixth root of the packet end arrival times. We could have taken a log section, but six through of the arrival Weibull quantiles with a shape parameter of .48. So, we estimated the shape parameter, found the Weibull quantiles of that shape parameter, and made a plot. So, this looks pretty good, actually, as these things go.

This is a model. You have to be forgiving, of course, because when you have an arbitrarily large amount of data, of course, nothing ever fits. So, Weibull in fact, turns out to be an excellent approximation of the distribution of the end arrival times.





Here is a Weibull quantile of sizes for one very high-load trace. Now we start to see a little bit of deviation down here at the low end of the distribution, not enough, though, to jeopardize modeling these things by Weibull.

By the way, there is a minimum end arrival time. You can't have an end arrival time any less than the transmission time of the smallest packet appearing on the link. So, that is a truncation.

So, you might say, strictly speaking, it is a truncated Weibull, truncated at the bottom, but the amount of truncation is extremely small. This vertical line here shows that it is 1 percent of the data. So, you have actually got a very small amount of data, in this particular case, being terminated.

So, Weibull turns out to be an excellent approximation. By the way, the shape parameter in this case is one, which is an exponential.

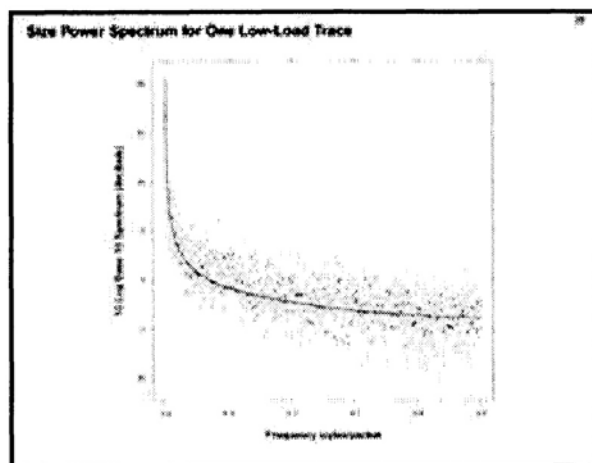
So, Weibull distribution as a model for the end arrival times. So, t, t_w, λ is the shape and λ will be the scale.

How about the sizes? The sizes, here is the same thing, a quantile plot of the sizes themselves. You see there is something like 35 percent of the packets are the 40-byte packets, control packets, and something, about the same fraction, of the packets are 1,500 bytes.

Then, sort of down through here, things are reasonably uniform, and then there is a bit of a turn-around here.

This is this one low-load trace. Here is the high-load trace. The same sort of thing, except that we see now there is an accumulation of packets of 576 bytes.

So, this is a case where somebody is even more behind the times and has configured some device, either a server or a client, so that the maximum packet size allowable on the connection is 576 bytes and not 1,500. So, it can even be worse.

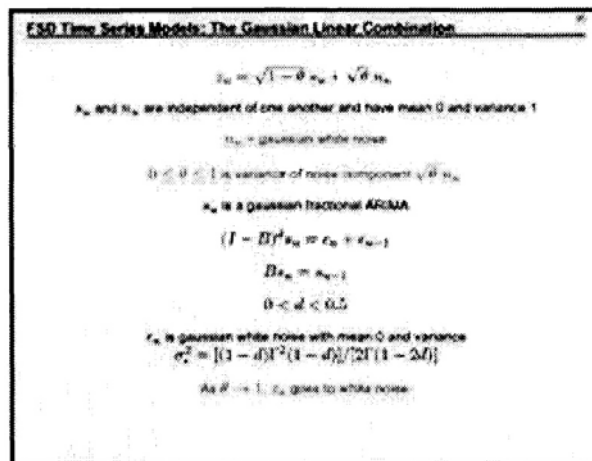


In any case, the same sort of things, the accumulation at a small number of fixed points in the distribution and then, elsewhere, sort of reasonably uniform up here, and then some curvature down here.

So, the marginal distribution of the sizes of the model as a discrete continuous distribution—I guess everything is pretty much discrete in continuous distribution, but one picks out particular sizes—say 40 bytes, 576 bytes, what we just saw here 1,500 bytes.

Actually, the way we model is to take things to be uniform on a set of variables from 40 bytes to 1,500 bytes and, oftentimes, just one interval suffices.

For the purposes at hand, if you take the packet size distribution to be uniform, it is a little crude. You find that things don't really change too much in simulation. If you want to do a little better job of containing things, then you certainly can do that. So, something on the order of three or four intervals is usually just fine and you get an exceedingly close fit.



Now, to model these data, here is what we do. Let's suppose that x_u is a time series to be modeled, and I want to let F be the marginal cumulative distribution, and bias some unknown parameters.

I am going to let $G(z)$ be the cumulative distribution function of the normal, with mean zero and variance one.

So, what we did was, we said, all right, we got a good handle on the marginal distribution and now we are going to transform the time series. We are going to transform it so that it is marginal with Gaussian. Well, that is slightly tricky because that

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

factor size distribution has got some atoms. It is true. So, you just do a little bit of randomization and you are off with the Gaussian.

So, here is it. So, half of the data where F is a fit of distribution and then G^{-1} of that, and that gives you Gaussian.

Now, that time series has got a Gaussian margin, a normal margin, but we can't suppose that that is a Gaussian time series.

Of course, this has to be checked, it has to be true, but the idea is that we transform it to try to get our best shot at turning it into Gaussian.

Of course, to generate the x_u , once we have a model for z_u to generate x_u , we transform z_u and then transform it back to the original scale.

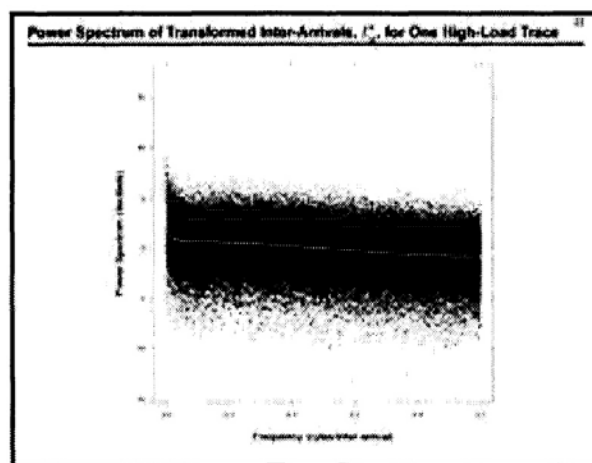
Everything works if I am taken to a Gaussian time series. Then, of course, everything is perfectly legitimate.

Now, why do we need this transformation? The reason is that the end arrival times and the sizes are grossly non-Gaussian and grossly nonlinear. So, if you attack them directly with the wavelet modeling, you are going to see immense complex structure.

The transformation actually removes most of that compound structure, as it turns out.

So, t^* will be t_u transformed to normality, and q^* will be q_u transformed to normality.

Now, here is the power spectrum. We take those transformed variables. Here is the power spectrum for a low-load trace of the sizes.



What you see is, there is a rapid rise at the origin of the power spectrum. This is the long-range dependence exhibiting itself. That is the way it comes out of the power spectrum, is a rapid rise at the origin.

The Parameters of the FSD/FSD-MA(θ) Models for q_{in} and t_{in}

| | |
|---|--|
| <p>Parameters:</p> <ul style="list-style-type: none"> • size marginal <ul style="list-style-type: none"> - skew probabilities $p_{i,j}^{(k)}$ - interval probabilities $p_{i,j}^{(k)}$ • inter-arrival marginal <ul style="list-style-type: none"> - Weibull shape λ - Weibull scale α • size dependence <ul style="list-style-type: none"> - fractional difference exponent $\beta_{i,j}$ - noise variance $\sigma_{i,j}^2$ • inter-arrival dependence <ul style="list-style-type: none"> - fractional difference exponent $\beta_{i,j}$ - noise variance $\sigma_{i,j}^2$ - moving-average coefficient ρ <p>Generation of packet marked point process from model by specifying ρ and ρ/λ</p> | <p>$\beta_{i,j}^{(k)}$ and $\beta_{i,j}^{(k)}$ are constant across links and loads and are both taken to be 0.41</p> <p>ρ/λ change with the link but not appreciably with the load on the link</p> <p>ρ used only for fitting and not for generation</p> <p>λ, $\beta_{i,j}$, and $\sigma_{i,j}^2$ and α modeled as a function of the connection load, r</p> |
|---|--|

This is the low-load trace. Here is the high-load trace. You see there is the same rapid rise at the origin, but things have flattened out immensely. Keep in mind that is the spectrum of white noise.

What we found was that an exceedingly simple model described this behavior. Once transformed to Gaussian, we found a very simple model described the relation structure for long-range dependence of the sizes.

Do These Models Fit

EXTENSIVE VALIDATION

Methods

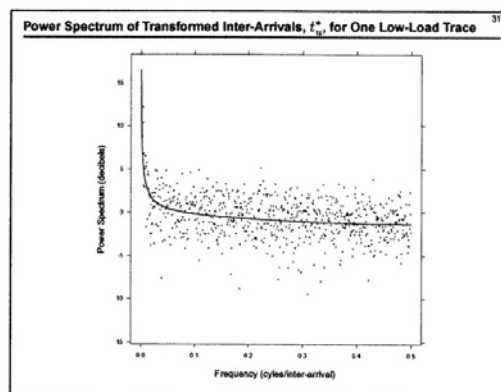
- model checking: visualization of data, model fits, and residuals
- model multiplexing
 - superpose low-load synthetic traces to get high load trace
 - compare results with synthetic high-load trace
- model duplication of observed scaling effects in byte and packet counts
- open-loop study of bandwidth allocation problem
 - live traces
 - synthetic traces

Domain of validity

- moderate to high aggregation, about $r = 80$ connections and above
- low positive queuing delay
 - delayed packets approximately 15% or less
 - but can get delayed traffic by putting through a queue

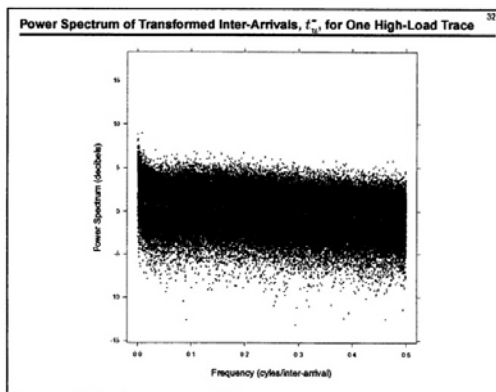
It is a linear combination of two time series, one of them white noise, v , and one of them a very simple, long-range dependent series, mixed according to this parameter θ .

So, θ goes between zero and one. If θ is one, then we get nothing but white noise. If θ is zero, then we get nothing but this long-range dependent series. Otherwise, we are mixing the two together.



That, of course, is what is happening here. As we mix in more of the white noise, the spectrum at higher frequencies heads toward flat.

No matter how much we add, if there is still a little long-range dependence left, eventually we wind up going to infinity at the origin.



FSD-MA(1) for Transformed Inter-Arrivals, t_u^*

Replace white-noise η_u by first order moving-average

$$z_u = \sqrt{1-\theta} s_u + \sqrt{\theta} \eta_u$$

$$\eta_u = \zeta_u + \beta \zeta_{u-1}$$

ζ_u is gaussian white noise with mean 0 and variance $(1 + \beta^2)^{-1}$

If $\beta = 0$, FSD-MA(1) is an FSD

As $\theta \rightarrow 1$, z_u goes to MA(1)

So, we never quite get rid of long-range dependence, but its influence is reduced more and more through time. For the end arrivals, it is a similar story, except that the end arrivals head off to, it turns out, the first order of moving average process.

The Parameters of the FSD/FSD-MA(1) Models for q_u and t_u

| | |
|---|---|
| <p>Parameters</p> <ul style="list-style-type: none"> • size marginal <ul style="list-style-type: none"> - atom probabilities $\rho_1^{(a)} \dots \rho_n^{(a)}$ - interval probabilities $\rho_1^{(i)} \dots \rho_n^{(i)}$ • inter-arrival marginal <ul style="list-style-type: none"> - Weibull shape λ - Weibull scale ϵ • size dependence <ul style="list-style-type: none"> - fractional difference exponent $d_{(s)}$ - noise variance $\theta_{(s)}$ • inter-arrival dependence <ul style="list-style-type: none"> - fractional-difference exponent $d_{(i)}$ - noise variance $\theta_{(i)}$ - moving-average coefficient β | <p>$d_{(s)}$ and $d_{(i)}$ are constant across links and loads and are both taken to be 0.41</p> <p>ρ's change with the link but not appreciably with the load on the link</p> <p>β used only for fitting and not for generation</p> <p>λ, $\theta_{(s)}$, and $\theta_{(i)}$, and ϵ modeled as a function of the connection load, ϵ:</p> |
|---|---|

Generation of packet marked point process from model by specifying ϵ and ρ 's

So, you need a little bit of extra stuff in there to account for the observed

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

behavior. So, instead of the n_u being white noise, it is the first order moving average, with actually quite a small θ .

Actually, the other reason why we need to throw this in is that estimating the parameter θ is a bit sensitive to the β . So, it has to be in there for the estimation purposes.

Then, when it comes time to actually generate the traffic, we just ignore β .

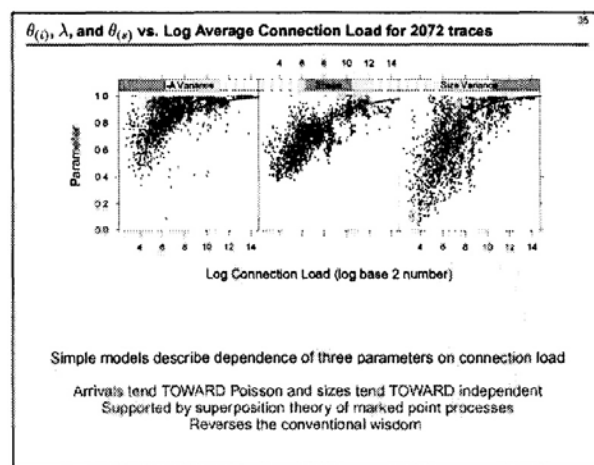
AUDIENCE: So, you ignore β ?

MR. CLEVELAND: Yes. β tends—when we estimate β , we have had no β bigger than .3. That is short-term correlation and it just isn't salient for the queuing characteristics. You could put it in, but we have done that, and you look at the results and you look at the simulation results and queuing out and so on, and it just looks identical. So, after a while you say, forget it. Everything is supposed to be as simple as possible, you know, Occam's razor always works.

So, we have these generation models. There are a number of parameters here, but actually we have been able to reduce this down so that, when you do packet generation for engineering studies, the only thing you have to specify is the size distribution, you have to say what that is, and then you have to pick a traffic frame, c , the average number of back-up connections.

What we found was that the shape parameter of the Weibull and those two θ 's for the end arrivals and the sizes, change with the connection load.

What we decided to do is to sort of fit the behavior. So, for example, here are the three parameters plotted against the log of the connection load.



So, here is the θ for the end arrivals changing with the connection load heading up toward one, because those end arrivals are tending to independent, and the same thing with the sizes. So, we fit these curves and they become a function of c in the model.

By the way, this immense variability here is because we have taken the intervals to be small. If we took the intervals to be larger, then the variability would go down, but the problem is that now we are beginning to risk non-stationary, because the connection load changes.

So, if the connection load changes appreciably, then we actually get different physical characteristics.

So, the phenomenon is really the curve going through. By the way, this wasn't fitted. This is actually what you might call a theoretical curve that we put together, to see how everything worked out, to see if the models were consistent. As you can see, it is doing a very good job of fitting the data.

In any case, one thing about the change in parameters with load sets is that the end arrivals are tending toward independent. That is what this is telling us, as the connection load goes up.

The shape is tending toward exponential. Sorry, the end arrival marginal is tending toward exponential. The shape parameter is tending toward one. So, that means that the end arrivals are tending toward Poisson, and the end arrivals are tending toward independent. So, this reverses that conventional wisdom that held for so long on the Internet. It is just not true.

By the way, this is supported by the super-imposition theory of mark point processes. So, some of you might say this should be no surprise. It is not quite that. Obviously, it wouldn't been an issue. It doesn't necessarily have to be the case that the assumptions regarding the Poisson are true.

We did an immense amount of validation of these models. We did a huge amount of what you might call traditional model checking.

You take these fits. Of course, these fits are blindingly simple. So, there isn't any issue about gobbling up huge numbers of the degrees of freedom. We really only have a few parameters in effect. Actually, we have got more parameters for the size marginal distribution than we do for the whole rest of the model.

Still, we did a lot of plotting of bits and data and residuals and things like that. We did other things. We did consistency checks, you might say. In fact, that is what I was referring to when I showed you that curve there.

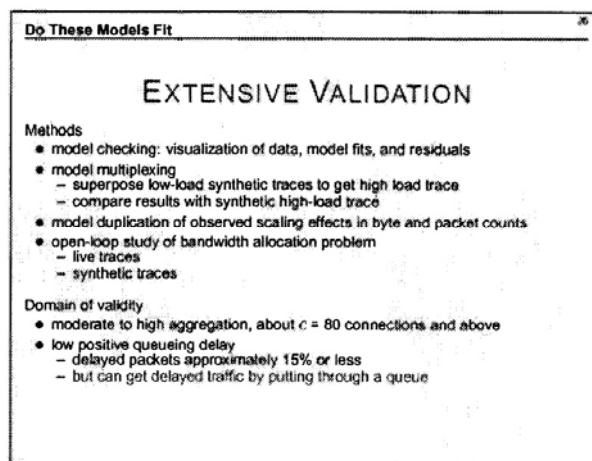
What we did was, we said, well, look, if these models work, we ought to be able to take them and generate a whole bunch of traffic streams at very low rates, and then multiplex them from the generated model so that you get a high rate, and then fit parameters and look at it, and that ought to correspond to what the model says it should be for that high rate. We did that and it worked out fine. You saw the results on that plot.

In running our queuing studies for quite some time now we have been doing it side by side, by sticking live traces in. We stick the models in with match to those traces, and any queuing results are consistently coming out to be about the same.

What we are hoping is that eventually people will allow us to use that and not get more data. We have, sort of sitting off somewhere on the West Coast, data on that same link, that same high-speed link, but now with 300 times as much traffic.

So, it is 300 gigabytes of data, and we would just as soon people would let us get away with extrapolating to a gigabyte of traffic rather than having to analyze the data to prove, again, that everything fits. But we will see what happens.

In any case, what about the domain of validity from all this extensive validation? The models fit, provided the number of active connections is about the same as a number of, say, eight.



Once you start to get too low a load, the communication protocols that are doing the transfers are starting to make themselves—their footprint is very visible in the statistical characteristics. You are just chasing endlessly after detail that you are never really going to account for in any reasonable way. You really need to see it at low rates like that, at access links, for example. Then, somehow you just have to come to grips with modeling directly the communication protocols. There is no point in trying to do the statistics.

All of this actually supposes that there is a small—I mean, the modeling was done under a supposition of a small amount of queueing. So, it really tells you what are the statistical characteristics of the power packets arriving at the queue as opposed to coming out of the queue. However, if you said, you know what? I actually need packets where there is delay—you know, they have been delayed in a queue. Then, it is that easy. You can just take the output of the model and put it through just a simple personal computer.

So, for generation purposes, let's say, there is no problem creating packets that have felt the effects of more than just a small percentage of packages being related in a queue. That is pretty much it. I think I am just at the end here.

MS. MARTINEZ: I think we have time for at least one or two questions, and the next speaker can come up and switch.

AUDIENCE: It seems like if everybody sent files that were of some small size, you would seek Poisson right off the bat.

The sort of two-part questions are, is the reason it is not Poisson at a lower rate just because long files are reduced, and if that is the case, as people start downloading bigger and bigger files, will the Poisson go away?

MR. CLEVELAND: It is not just the file size. Actually, that was thought for a long time. It was thought that the long-range dependence on the Internet was exclusively as a result of, as you are saying, the heavy-tailed file size distribution.

There is another thing which is generating, which may actually be more important. There is another raging debate going on right now about this issue. So, there is another conventional wisdom under attack right now. So, it is not a sole creator of long-range dependence.

Another reason for it is the transfer protocol. It tends to ramp up its sending rate and then drop off and then ramp up its sending rate and drop off. That creates a kind of persistence of long-range dependence as well.

That may well turn out, in the end, to be the more salient factor. So, we will see. There is going to be a special issue of papers in *Statistical Science* on Internet traffic, and

we will have a paper on this topic. So, let's see what the authors come up with.

AUDIENCE: Bill, just thinking about your results, I was sort of struck that you are characterizing traffic during normal loads.

When you engineer a network, you engineer it so that it performs well at heavy loads, and you tend not to care what happens in normal loads, because you are engineering from an extreme point of view.

I am sort of curious, it sort of seemed like, to get traction with these results, you really have to move into the case where you are in overload.

MR. CLEVELAND: Steve just said something extremely important. He says, well, look, you are trying to design a network, and it is when you get up to high-load that it matters, and you modeled it when it was at low-load. So, what are we going to do about high-load.

The answer is, well, our modeling says everything works in the model in describing the arrival process. The key thing here is that it models the way the router sees it as it is about the enter the queue, is really what we have modeled, although strictly speaking, the data aren't that.

Actually, this is another thing. We took data where the queuing wasn't actually large. So, what we are saying is, we did it in the queue. We really did modeling as it comes into the queue.

Now, you want to push your queuing as far as you can. Everything will be okay in terms of putting that into a simulation study, what we have modeled, so long as the drop is small, and that is where matching will be the case for higher-speed load.

So, you run an open-loop queuing, you stuff our stuff in, in the queuing simulation, and everything is fine so long as the packet loss is a small fraction.

Today, packet loss is not likely allowed to be high on service provider networks at all. It is likely to be in numbers like .1 and .5 percent. The question is, how high can we get before that happens, and that is what the simulation tells you, and that is what nobody knew, by the way.

Service providers just didn't know how to—I am sorry, I am taking too long for this question. Anyway, let's talk about it offline.

AUDIENCE: As sort of a follow-up question, ultimately the engineering issue is how do we relate the size of the queue to the speed of the link. When you have higher-speed links you need longer queues.

MR. CLEVELAND: The size of the queue is also dictated by quality of service criteria. You can't make the queue arbitrarily large. Otherwise, the line packs a long time and then your customer comes and says, you owe me a lot of money because you broke the rules.

So, the queuing—I mean, there are two things. There is packet loss, and then there is how long you delay. For example, if you want to do voice over IP, you can't delay the packets too long and create a lot of jitter, and the queuing does that.

AUDIENCE: The question really is, is what you are doing informing you of that?

MR. CLEVELAND: Yes, absolutely. We have solved the bandwidth allocation problem under the assumption that the packet loss is going to be low. I will be talking about that at Los Alamos on February 24. So, I would like to invite all of you to come and hear my talk.

MS. MARTINEZ: Okay, let's thank our speaker again.

AUDIENCE: [Question off microphone.]

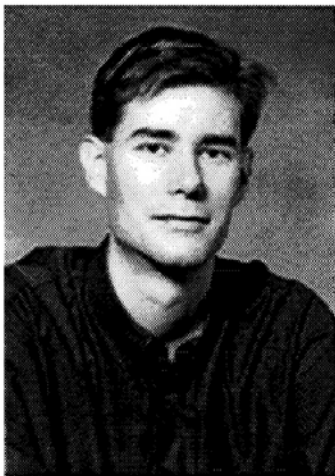
MR. CLEVELAND: That is a good question. Hopefully, 2003, probably summer and fall. The authors, none of them are here, but anyway, it is up to the authors at this point.

Johannes Gehrke

Processing Aggregate Queries over Continuous Data Streams

[Abstract of Presentation](#)

[Transcript of Presentation](#)



BIOSKETCH: Johannes Gehrke is an assistant professor in the Department of Computer Science at Cornell University. He obtained his PhD in computer science from the University of Wisconsin at Madison in 1999; his graduate studies were supported by a Fulbright fellowship and an IBM fellowship.

Dr. Gehrke's research interests are in the areas of data mining, data stream processing, and distributed data management for sensor networks and peer-to-peer networks. He has received a National Science Foundation Career Award, an Arthur P.Sloan Fellowship, an IBM Faculty Award, and the Cornell College of Engineering James and Mary Tien Excellence in Teaching award. He is the author of numerous publications on data mining and database systems, and he co-authored the undergraduate textbook *Database Management Systems* (McGraw-Hill, 2002, currently in its third edition), used at universities all over the world.

Dr. Gehrke has served as program co-chair of the 2001 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, tutorial chair for the 2001 IEEE International Conference on Data Mining, area chair for the Twentieth International Conference on Machine Learning, co-chair of the 2003 ACM SIGKDD Cup, and he is serving as Program co-chair of the 2004 ACM SIGKDD Conference.

Dr. Gehrke has given courses and tutorials on data mining and data stream processing at international conferences and on Wall Street, and he has extensive industry experience as a technical advisor.

ABSTRACT OF PRESENTATION

Processing Aggregate Queries over Continuous Data Streams Johannes Gehrke, Cornell University

In this talk, I will describe techniques for giving approximate answers for aggregate queries over data streams using probabilistic “sketches” of the data streams that give approximate query answers with provable error guarantees. I will introduce sketches and then talk about two recent technical advances, sketch partitioning and sketch sharing. In sketch partitioning, existing statistical information about the stream is used to significantly decrease error bounds. Sketch sharing allows one to improve the overall space utilization among multiple queries. I will conclude with some open research problems and challenges in data stream processing.

Part of this talk describes joint work with Al Demers, Alin Dobra, and Mirek Riedewald at Cornell and Minos Garofalakis and Rajeev Rastogi at Lucent Bell Labs.

TRANSCRIPT OF PRESENTATION

MS. MARTINEZ: Our next speaker is Johannes Gehrke, and he is from Cornell University, and he is going to talk somewhat on the same topic.

MR. GEHRKE: Hello. I am Johannes Gehrke from Cornell University. My last name is German. I get these calls from telemarketers who say, hello, may I please speak to Mr. Jerk.

So, my talk is going to be a little bit different from the previous talk. I am going to talk about some technical details, but I thought I would give you a broad picture of some of the techniques that I am going to talk about here, where they apply, and how they fit in the larger picture.

So, what has been our work motivator, especially since 9/11, is this notion of information spheres. This notion of information spheres actually comes really from the intelligence community. There is a local information sphere within each intelligence agency that presents challenging problems, and there is a global information sphere that would like to enable intelligence agencies to work together in a collaborative way. So, these local information spheres, really, are within each local agencies or even businesses, and there you have some of the problems that are addressed here in this session.

You have to process high-speed data streams. You have to have variation of thousands of triggers you might set on events as they might happen, and you also have to worry about storage and archiving of the data. In addition, you also have this global information sphere where you would like to share data or collaborate between different intelligence agencies. At the same time, you have legal restrictions that do not allow you to share the actual data. So, privacy preserving computations in the setting are a very important part.

So, let me give you sort of a little bit of a background on this work. So, in a local information sphere, really what we are building is this distributed data stream event processing data mining system. The technical challenges are, if we would like to process physically distributed high-speed data streams, it has to be scalable because we do not really anticipate, we don't really know what kind of questions others are going to ask. There will be a high-speed archiving component there. We have a graph-based data model, and we also look into some other issues like data mining model management, and built a support for especially data provenance.

Since I am going to talk about the problems with high-speed data streams from a data management point of view, let me first step a step back and ask, well, why can't we use existing approaches? Why can't we use existing approaches? If you think about databases, people have created this multi-billion-dollar industry with Oracle and IBM and Microsoft in this market. Why can't we use traditional database systems to process high-speed data streams? That is the question to us.

So, it is looking at sort of using natural approaches that you would take. So, the first thing would be, so database systems have this technical component which is called a trigger. A trigger is basically like a condition that you can set on a table. If this condition is true, then a certain event is going to happen. This might be a notification; this might be another insertion for another table, what have you. The only problem is that these triggers, they really—there is a trigger developed usually for every insert into the table. So, if you do a linear number of triggers, you can see that your performance degrades

linearly with the number of triggers. There is some documentation out there, like in the Air Force JDI, but they really still require index maintenance on each publication.

So, what would be another possibility, if we look at the bottom thing? You could use bulk loading for archival in this local information sphere. There is another component which is very important here, if you want to archive data, but if you think about current database systems, they are really not made for high-speed append-only relations. For example, Oracle has something, what is called a transportable table space. The transportable table space is where you take an existing table, you dump it out in its existing physical format. You can read it, actually, very fast. If you think about it, well, how did I get it first into this physical existing format? To do that, actually, I first have to load the table and then dump it out and then reload it again. Actually, this is really a reload of the existing table. You know, modeling construction—actually, Pedro Domingos is going to do a very nice talk about online construction of data mining on this.

Then, the last point that I think is actually very important as we talk here about streams and monitoring applications is this notion that I think, in order for this technology to be acceptable by the public, we actually have to make advances from a technology point of view. That means that we actually have to have techniques that allow, for example, to collect data and, at the same time, be able to build models, but not being able necessarily to refer this data back to the individual. So, we really need to have techniques for private data sharing in this global information sphere. So, these are sort of the two main goals of our research processes, and I am actually going to concentrate now on some techniques in the first part—this is distributing streamlining and monitoring in a processing system.

So, really to the ultimate of my talk is I am going to give you a little bit of background of modeling considerations. I am going to actually switch again to sort of a high-level mode and talk about sort of one exciting application of these kinds of techniques in sort of a different setting, which is a network setting. Then I am going to talk a little bit about the actual techniques we are using, which are actually random projections. They are also called sketches, which allow us to process network data streams at high-speed rates, and with very small space.

So, the high-level model that we are considering—this is a model that the previous speaker talked already much about. We have a bunch of routers. So, there is some kind of network operation center, and we can't afford to actually stream all the data through the central operations center. This might be a measurement allowance. These might be other questions that you might ask about the status of the triggers, that may be based on certain events, like a service attack. The kind of queries that I am going to concentrate on here, and my techniques, are these kind of data stream join queries. These data stream join queries, conceptually, the way you can think about it is that there are sets of events happening at these different routers distributed through the network.

What you would like to know is how many of these events actually match. So, I am a database person, so another way of thinking about this is that you have conceptually three different relationships that are distributed at three different routers. What I would like to find out is the site of the join between these numbers, that is, the number of elements that actually match these three different data streams.

Again, my objective is to do this without actually sending all the data to a central site. So, that really is the goal. I would like to do this with a very small amount of space.

The main technique that I am going to use is that I am going to say that, well, if you are looking for trends, or if you want to analyze behavior of this network, you are not caring about the count of single packets. What you care about is high-level trends. Therefore, approximate answers are sufficient for a large class of applications. I am not saying for all applications. For a large class of applications, you want to have fast classification of actual high-level events and, in this case, approximate answers are good enough. Actually, a lot of applications are out there. There is one application which is out there, which is sensor networks, and actually, later in the talk, I am going to switch and talk just a little bit about that work, because I think it fits very nicely here into the scope.

So, in computations over streaming data, the model really is that I have this synopsis—essentially, you have this stream processing engine, yet these different streams are streaming through the stream crossing engine, and the stream crossing network is physically distributed over different parts of the network. You have this synopsis of the relation of the different streams that you are building. Really, the stream processing engine really keeps some small-space summaries of these relations, and there is a certain workload— [off microphone] —so that is the model. So, the computational model is a single pass, each parameter is examined once, and a fixed order that it arrives. At the same time, we would like to use a very small space. Small space can actually be in sort of two dimensions. It can be small in the length of the stream as well as small in the size of the domain of the attribute which you would like to match. I am going to give you an example, actually, what I mean by this. So, this is the setting that we are talking about, and let me give you sort of an idea of the class of queries to consider.

The class of queries to consider is some aggregates over join. So, this is database technology, so let me tell you what really a join is. So, the join between these two different relations are one through R , is the following. In the simplest case, you have one attribute between sort of a pair of relations. So, there is this linear chain of relations. In either pair, you would like to find only those records that match actually on this pair of attributes. For example, one or two may join here on this little attribute. Then, R_2 and R_3 , again, might be joining on a different attribute. What you would like to find out is — you would like to find out some sort of aggregates over this join. For example, in the simplest case, how many doubles are actually in this join. We would like to do this, again, in a distributed fashion in a very small amount of space in an online scheme. So, another way of specifying these queries is the count of this relation here is just the sum of the ones where the doubles on the records of the output of this join. Another way of thinking about this is sort of the job product of the frequency vectors, and actually, I will give you an example of this as well.

So, this is the class of queries. Let me give you some examples of these queries and actually how this works. So, for example, a little two-way join. We would like to find the size of this join. So, how does this work? You have these two streams, F and G , that are streaming in. Essentially, what I would like to do is, I would like to build these frequency vectors here on the top right—namely, F and G —where, for each value in the domain I have the frequency that it actually occurred. The size of the join is the dot product of these two frequency factors. That is what I would like to ask it. Again, I would like to do this without shipping the actual frequency vectors to a central site, because the domain of these attributes might be extremely large. That is what I would like to work on. In this case, if you get the dot product of F and G , you get the single case

for these two streams. This would be a two-way join somewhere. So, now, again, I match up the two events, the two events, the two relations at these two different sites, but I am summing over a third attribute, or a third variable.

So, now, here, for streaming for F and G , I have two different variables A and B . I match them up on A and then, for those, wherever I have a match, I sum over B . Again, this is now the sum of the join, which I modify the vector of G' , which is now 30, 10 and 40, and the overall answer is 180. The frequency matrix in G now has the frequency of the actual attribute value. Then I also have, for each of the values in B . So, it is now a two-dimensional matrix. For every value in G , how often does it appear for every attribute value in A .

Again, I want to do the match on A and I want to sum over B . That is the set. Again, another way of thinking about this, I match these two relations on A . Every match has a certain value of B . I want to sum over all of these values. That is the answer to the query that I would like to compute.

You can also extend this to more than two streams. Here are three streams, stream F , G and H . Now, stream G has two different attributes, again, A and B . A is the connection to F , and B is the connection to H .

What I would like to point out is, what is the size, again, of the join between all these three relations. Actually, again, in the middle you have this matrix, in this case it is just 24. So, this is the setting I would like to compute. So, think about it in a distributed setting. I would like to find out the number of matches of attribute values.

So, there is a lot of previous work that I think actually some of the experts here in this area who have done a lot of work on this here in the audience. So, there has been a lot of work on conventional data summaries. One of the problems that they have is that they do not really work well for foreign key joins, or they require a lot of space, or they can't cross across— [off microphone.] I can't say that this technique that I am going to talk about is the main technique that should be used in all these applications, but is one technique that works well for this class of query.

Let me give you an introduction to sketches, and then let me actually tell you about some work that we did, sketch partitioning, sketch sharing, if time permits. Again, let's go back to the examples. Again, the setting is that there are distributed data streams. I would like to find out the number of matches. That is the simple query. Again, these are my examples. I have these streams coming in. What I would like to find out is the count of the join. The join is the number—excuse me, those doubles were actually matching pairs.

So, the main idea that we are going to use is called a sketch, which was developed by Mateas and Lewis in a paper in 1996. This is a work where they centered this in a single join in 1999. So, the idea is the following. As you have seen in the previous slide, you can have these frequency vectors, and I can actually compute the size of the join exactly. So, one way of doing the join would actually be, well, I computed the frequency vectors at every point in the network, shipped these frequency vectors to a single site, and then compute the size of the join.

Now, the problem is, if I have an attribute, something like, say, my key address, this attribute has a huge domain. So, these frequency vectors can actually be extremely, extremely large. So, the problem is, depending on the size of the domain, there might be a polynomial, or the size of the stream at the polynomial, let's say, in the size of the

domain, actually. So, in this case, these frequency tables are really too large to ship to a central site. So, the main idea is that I would like to summarize these frequency vectors and the way I can summarize them is as follows. I take the frequency vector and project it onto a random vector. This is what is called a sketch. Let me tell you how this is done. So, you build this random vector of ± 1 value. Then what you do is—the size of the domain is N . What you do is, you take the frequency vector and you multiply with this random vector.

Now, this product that comes out is a single number. I do this on both F and G . Now, if these random vectors have certain properties, let's look at what comes out and I am going to multiply these two sketches. Now, when I multiply these two sketches, the expectation is now F times these two in the middle. Now, I have these two items in the middle, and take the frequency vector and take them to each site. Because these are plus minus vectors, classification independent, such as the diagonal actually has one, and the expectation of the diagonals is zero. Then I can actually get the size of the join out. What this means is that, I have now a construction that, in expectation, gives me exactly the size of the join with a single number on both ψ 's.

Again, with this random projection, what this allows me to do is summarize this frequency vector into a single number. I have the expectation that I am going to take the product of these two numbers and actually get exactly the statistic that I would like to estimate. Again, this is not a technique that we developed. This is a well-known technique from the algorithms in theoretical literature, actually.

So, let's see how this actually does. So, we have this vector at site one, site two, site three. Let's assume that our domain set is three. Let's assume it is $\{-1, +1, -1\}$.

Then what I do is, for F , for the sketch of the stream F , I just take the frequency vector of 3.2 and multiply with this vector $+1$ or -1 . I get -4 . I do the same thing on G . I get -5 . I take the product of them. I get 20, which is sort of about 13, which you can see, is a little bit off, but in explication it is actually correct. So, you can see, the variance might be a problem. Actually, I will show you actually a couple of hopefully interesting techniques how to actually reduce the variance.

Again, the only property you need from this site actually is estimates of the variance. Four is independence, and there are techniques out there to generate them with small seeds where the ψ vector actually is not stored. Well, what I have to do at each of the sites, I have to store another ψ vector of ± 1 's. That, again, is big as the size of my frequency vector. So, what have I gained.

The main idea, again, is that I can generate these entries on the ψ vector through a very small seed, basically thrust through a small function, which I give the actual attribute value in the domain, and it spits out ± 1 .

So, how is this actually working? There is a seed S that generates the ψ family, and if I have a stream 1, 2, 1, 3, the main thing I do, I sort of take my function H , take my seed and I give it seed and one, what outcomes, ± 1 , take the same for the next function, take the seed in one, put in the function, what comes out is ± 1 . The following or the corresponding entry in my vector.

As you can see, I can do this online in a streaming factor. As I add my individual elements here, what comes out is actually online in a one pass algorithm exactly the sketch for F and XF . The counters now, you can actually see, they only need log space actually in the size of the domain, because really they are not—and the size of the

stream. So, as you can readily see, the estimation of this count from single step is too noisy. So, there are sort of standard techniques where you take medians of averages to actually reduce the error and to increase the confidence. So, basically what you do is, you have these seeds here in the middle. You take F and G . For the same seeds, you do sketches of F and you do sketches of G . Then you take these independent copies of X , you average them, you take the median of the average to get the confidence up.

So, again, this is not my contribution. This is sort of a nice set of techniques that I am now going to use in a streaming fashion to get high-quality estimates, or the estimates. Again, in a picture, how does this actually work? The estimation of this count is, you take this stream, you build a sketch, as the total stream in a single pass online.

You take these two sketches, you multiply them. You do this for a bunch of sketches, you take median averages, and what comes out is an estimator that is unbiased and has a low variance. So, that is sort of a warm up of how to use sketches.

MR. NYCHKA: Where does the probability statement come from? Is that exact? You have $\pm\epsilon$ with probably of $1-\delta$.

MR. GEHRKE: That comes from basically how many sketches I average. This comes basically from the number of sketches that I use here. What I do is, I take the average of a number of sketches and this gives me basically a small amount of error because the variance of the average is much smaller than the average of the individual sketches. Then, to get the confidence, I actually take the median of a bunch of these averages. That only comes from this picture here.

So, let's do a simple extension. A simple extension, let's look at multiple joins first. First of all, you can already see that conceptually easily I can do sum. What I do is, instead of just summing $+1$'s or -1 's, I actually take the actual value of these and I compute it into one of these sketches that I have. Then, if you actually look at the math — again, I am not going to do this—what comes out is actually an estimate of the sum exactly.

So, again, for sum, the only difference is that I build a sketch. When I build the sketch, instead of just adding X_i 's, I also add all the attribute value that I need.

Let's look at another example. Here, I have three different streams. So, how does this work? Again, I take now two different ψ families, one between F and G , and another one between G and H . I can use the same construction and, again, sort of F and G and H factor out nicely, and C_x^i and transports in the middle, because of independence, again, a factor out and what I have here in explication is exactly the size of the join.

This is schematically what is happening. From the left-hand ψ , I build ψF , around the middle I built ψG , which now I have to take the product of the corresponding ψ 's for the one attribute and the other attribute and add it to this corresponding sketch. Now you can sort of see this actually extends through multiple joining. You can actually see this variance has this sort of unfortunate property that actually grows up exponentially with the number of joins. It is sort of a really bad property, if you look at it.

Again, in expectation, I still get exactly what I am trying to estimate, but my variance is really getting extremely large and hopefully, I have a little bit of time, and I will tell you now about some techniques how to reduce the variance, and then hopefully some techniques how to do multiple characterization for a set of sketch queries.

So, again, to bring this point home, the sketch making algorithms is really as simple as this, on a stream. So, I take each double, for each double, I take my function H

and my seed, look at the attribute value in the domain, find out whether I am +1 or -1, and add that to my sketch. It is a relatively simple operation that I can do online, on a stream, at very high speed. There are actually very nice and parallelizable, very nice computing in a distributed setting.

For example, at the stream level, I can compute sketches for each stream independently in a distributed way at each point in the network. At a sketch level, I can even maintain each sketch independently. Even at the data level, I can partition the sketch into fragments, and then the sketch of the whole stream is the sum of the parts. Actually, I am going to use this next, when I am going to show you a technique that actually shows you how to reduce this variance.

You can see, so far it looks nice. In expectation, I get exactly what I want, but my variance really has this bad property that really seems to blow up completely, especially with the number of joins, because there is this bad exponential factor there in front of the variance. If you look at the technique, how I get my confidence and error bounds, this sort of decreases my variance by some linear factor. I actually have to pay for this linear factor with the number of copies that I am going to get. So, this is not really exactly what I am going to—this is not going to help me against this exponential growth that I have in front of my variance.

I think I am going to skip over some experiments that basically show that sketches are really good, and let me just give you sort of an intuition of one of the techniques that you can use to actually reduce the variance.

So, the problem really is, if you have a large variance, you really get bad estimation guarantees. It is really so bad that your variance, for example, can be just much, much larger than the quantity that you are actually trying to estimate. So, basically, you don't really get anything. So, what would be one solution? One solution is to use this previous technique of keeping lots and lots of copies, trying to drive the average down through some conventional techniques.

If you think about what we are trying to do, we are really trying to estimate the size of the join. The size of the join is sort of pay-independent. If, for one type of event I have many matches and for another type of event, there are few matches, this is really getting to my estimate, because my estimate is for the number of matches of events. What this actually tells me is that maybe by actually looking at what is happening inside the join, this should give me some insight on how potentially to drive down the variance.

So, here is the idea. Let me give you an example. For example, consider this gray down here, this count of F and G , and these are F_i and G_i on the frequency vectors. These are the frequency of the different attribute values for attribute values one through four.

What I do is build a single sketch, but the variance is this huge number, you can see, 35,000. So, it is just ridiculous, and this technique doesn't really help me here at all. It is 357,000, yes. So, what would be the idea? The idea is that maybe what I can do is, I can use the concept that I can split the domain, the different parts.

I compute the drawing on that part of the domain and, in the end, I just add these two sketches up. Hopefully the variance on these different partitions is much, much smaller than the actual overall variance of the way I constructed it over the whole part.

What I can do is I can split the sketch up into F_1 and F_2 and G into G_1 and G_2 . I can start X_1 and X_2 . Then, my final sketch is just the sum of these two sketches. Now,

this now assumes that I have some previous knowledge about the distribution of the stream, actually, of the attribute values for the two parts of the stream, and you might be able to use some historic knowledge, of you can actually do an estimation of this.

Conceptually, what is happening here, it is a little bit hash joined. You have put the relation into two packets. The sum of your join is the sum of the one packet plus the sum of the other packet. What is happening is, if I do this partitioning here, I compute the variance on one part and the variance on the other part. What this actually allows is to drive the error down by a factor of 2.4.

Actually, we have seen a dramatic decrease about an order of magnitude on real data, because real data is actually usually quite skewed. What this allows us, it allows us to have many sketches for the heavy part of the distribution and few sketches for the loose part of the distribution. So, the main idea is really that, by splitting the main of the join attribute, we get a reduction in error. So, this is not a nice sort of organization problem that you can formulate. So, I mean, you want to partition the domain into two parts, or $2K$ parts, such that it minimizes the sum of the variances, and you get the sum of the variances, and the variance is sort of a product between cell join sizes. We can actually show that you should actually allocate this space in proportion to the variance, and we only have to look at partition as sort of the order of the ratios of the two frequencies, and it actually follows from sort of a nice theory that actually comes from a decision tree construction of a data mining problem called Raymond's theorem. So, it gives you this fast solution to this problem. Actually, you can do some more extensions. You can do KR splits and multiple joins, etc.

I think I am running out of time. Let me give you just some idea how this actually performs. For example, here, we are using some census data as sort of an example of real data. You can see, as the number of partitions increases, the relative error here goes down by about a factor of two. Here, again, a join on two different attributes, actually, it is sort of a correlation between the attributes. Again, the variance goes down by about a factor of two. The aggregate goes down by a factor of two.

So, I have maybe about two minutes left. What I want to do, in the end is, I want to show you another application of these kinds of techniques, again, in a streaming network environment. That example is sensor networks. You probably have seen there is sort of an evolution happened. Actually, one of the speakers yesterday talked about the same thing, namely, that small scale embedded devices are going to be everywhere.

In 1965, Wilson did an estimate of 10^{16} or 10^{17} ants on earth. In 1997, we produced one transistor per ant. You can really see what kind of computational power this actually puts in the space surrounding us within a few years. So, what we really need are techniques to deal with the slew of data that is streaming at us from these different sensors. I would like to give you one idea of one potential solution or one potential approach for how you would handle this data.

Traditionally, apparently how these sensor networks manage a program, the idea is that you have some kind of program that addresses these sensor. Say, sensor 17 sends something to sensor 25. Then, it sends something to sensor 83 and then sends it back.

We believe that actually the right way of programming these sensors would be through declarative queries. What you should have is these queries or these large scale sensor networks that give you the extraction of a single virtual database. Let me just take one minute. So, the more distributed database system, and you can now ask, what is the

distribution of chemical X in this quadrant, or in which area is the concentration of chemical X higher than the average concentration in this area. The nice thing is that really you have this declarative high-level tasking that shields away from network cracking and the system optimizes resources.

Again, this is a streaming problem because these sensors continuously generate data. Some of the techniques that I actually talked about and similar techniques can be used in this example. So, I can just give you sort of an example. So, what user would have, a user would have some kind of mini-GUI and array a set of network resources.

I think I sort of went over a lot of stuff. Let me just—there are sort of lots of problems in the sketching model. I think actually there are several people here in the audience who have also done fundamental work on the sketches.

Instead of running more over, let me just ask if you have any questions and thank you.

AUDIENCE: It seems possible in some cases you could get additional acceleration by allowing your two data sources to do a little bit of negotiation between themselves before they start sending off lots of information. For instance, they could initially show each other what their most frequent items are and, if they tend to agree on different items— [comment off microphone.]

MR. GEHRKE: This is actually a very nice observation. One possibility is, for example, if you look at where the variances are— [off microphone]. Another extension of this idea would be to look at actually streams that are somewhat annotated.

Instead of just sending blocks of data over to you, maybe I can do some computation on my side, or the originator of the stream, where there is some annotation.

Annotate the stream, for example, with saying this is partially sorted, or maybe you could sum it, that would actually allow the computer of the actual computer— [off microphone.]

Edward Wegman

Visualization of Internet Packet Headers

[Abstract of Presentation](#)

[Transcript of Presentation and PowerPoint Slides](#)



BIOSKETCH: Edward J. Wegman is the Bernard J. Dunn Professor of Information Technology and Applied Statistics, the chair of the Department of Applied and Engineering Statistics and the director of the Center for Computational Statistics at George Mason University. He received his MS and PhD in statistics from the University of Iowa. He spent 10 years on the faculty of the Statistics Department at the University of North Carolina.

Dr. Wegman's early career focused on the development of aspects of the theory of mathematical statistics. In 1978, he went to the Office of Naval Research (ONR), where he was the head of the Mathematical Sciences Division. In this role, he was responsible for a variety of cross-disciplinary areas, including such projects as mathematical models of biological intelligence, mathematical methods for remote sensing, and topological methods in chemistry. Dr. Wegman was the original program director of the basic research program in ultra-high-speed computing at the Strategic Defense Initiative's Innovative Science and Technology Office ("Star Wars" program). As the SDI program officer, he was responsible for programs in software development tools, highly parallel architectures, and optical computing.

Dr. Wegman came to George Mason University with a background in both theoretical statistics and computing technology, with knowledge of the considerable data analytic problems associated with large-scale scientific and technical databases. In 1986, he launched the Center for Computational Statistics and developed the MS in statistical science degree program. More recently he has been involved with the development of the Institute for Computational Science and Informatics and the new PhD program in computational sciences and informatics at George Mason University.

He has been consultant to a variety of governmental and private sector organizations, organized some 15 major workshops and conferences, and served as associate editor of the *Journal of the American Statistical Association*, *Statistics and Probability Letters* and *Communications in Statistics*. He presently serves on the editorial boards of the *Journal*

of Statistical Planning and Inference, the Naval Research Logistics Quarterly, the Journal of Nonparametric Statistics, and Computational Statistics and Data Analysis.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

ABSTRACT OF PRESENTATION

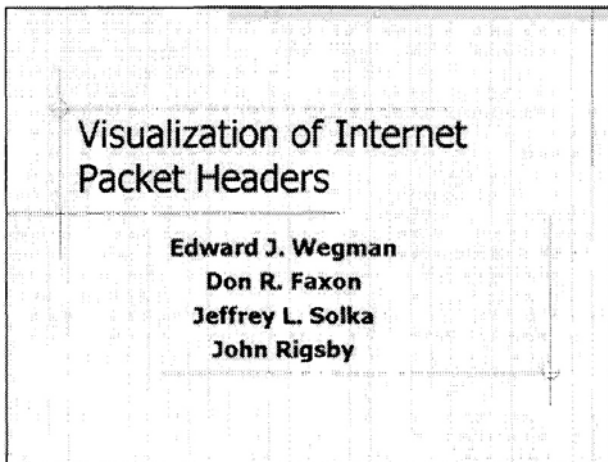
Visualization of Internet Packet Headers

Edward J. Wegman, George Mason University (with Don R. Faxon, Jeffrey L. Solka, and John Rigsby) .

Abstract: We have launched a project with the agreement of the University's CIO to capture all header information for all Internet traffic in and out of the University. This includes TCP, UDP, SNMP, and ICMP packets. We have installed sniffer and analysis machines and are capable of recording up to a terabyte of traffic data. Preliminary experiments within our small statistics subnet indicate traffic of 65,000 to 150,000 packets per hour. Indications are that we will have terabytes of data traffic daily university-wide, 35–40 megabytes of header traffic per minute, or approximately 50–60 gigabytes of header information per day in the larger University context. Much of the packet traffic is administrative traffic from routers. Ultimately, we are interested in real-time detection of intrusion attacks so that analysis methods for streaming data are necessary. In this talk I will describe our project, including some background on TCP/IP traffic, indicate some recursive methods capable of handling streaming data, illustrate a database tool we have developed, and give some suggestions for visualization procedures we are in the process of implementing. This report is very much a preliminary report. In data mining, 80% to 90% of the effort involves getting the data in shape to analyze, and this project does not deviate from this pattern.

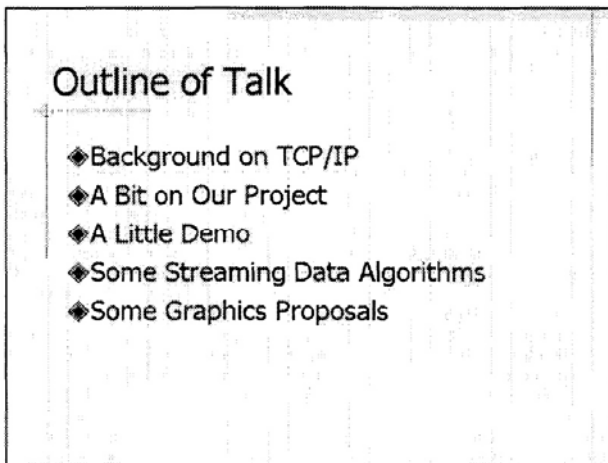
TRANSCRIPT OF PRESENTATION

MS. MARTINEZ: Our next speaker is Professor Ed Wegman from George Mason University, and he is going to continue to talk about Internet traffic and how can we analyze it.



MR. WEGMAN: I must say, it is always a formidable challenge to come after Bill Cleveland, who was my colleague once upon a time in Chapel Hill, when we were both quite a lot younger. Bill was looking at Internet traffic from sort of the global perspective. What I would like to do is have a discussion of our sort of internal structure. We are particularly concerned with issues of intrusion into our systems. Don Faxon and I work at George Mason University. Jeff Solka and John Rigsby are colleagues of ours that work at the Naval Surface Warfare Center.

One of the issues of interest is, how intrusions in a military setting are different from intrusions in an academic setting, and is there sort of a qualitative or quantitative characteristic difference between the Internet traffic in these two settings? So, we are working relatively closely with these guys.



What I would like to do is give a little background on TCP/IP. I realize probably a lot of people in the audience already know a lot of this. This is, you know, partly, for me, but I hope that some people may not know as much about the nuts and bolts of it, which are relatively important in understanding how intrusions are done.

I would like to talk a little bit about our project, do a little demo. Then, I would

like to talk about streaming algorithms and maybe some graphics proposals that we have.

Scope of the Problem

- ◆ Most of us have seen IP addresses
 - More precisely IPv4 address
 - An IPv4 is a 32 bit number usually represented as 4 dotted fields
 - field1.field2.field3.field4
 - These IP addresses uniquely identify a machine.
 - In theory, there are 4,294,967,296 addressable machines

Now, Bill talked about IP addresses, but didn't go into any detail. Technically, these are called IP version 4 addresses.

IPv addresses are 32-bit numbers, usually represented as four dotted fields. Field one, two, three or four, or sometimes called octet one, two, three and four. In general, these IP addresses uniquely identify a machine, although that is not entirely true, because IPs can be dynamically assigned. So, they may not uniquely identify a machine.

There is also a number called a MAC number, which is a manufacturer's number, which does essentially uniquely identify the machine. So, in principle, it is better to find the MAC number than the IP address, if you are trying to identify and locate individual machines.

Each of these fields is 256. It is an eight-bit field. So, if you multiply that you, you come out with approximately 4 billion addressable machines.

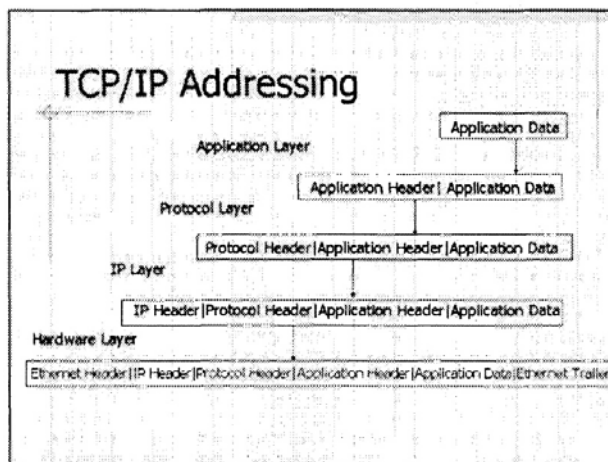
Types of Networks

- ◆ Class A – field1 identifies the network, fields2-4 identify the specific host
 - field1 is smaller than 127, e.g. 1.1.1.1
- ◆ Class B – field1.field2 identifies the network field3.field4 identifies the specific host, field3 sometimes used for subnet
 - Field1 is larger than 127, e.g. 130.103.40.210
- ◆ Class C- field1.field2.field3 identifies the network, field4 the host
 - E.g. 192.9.200.15

There are a number of different types of networks—Class A, Class B, Class C. Most of us are probably associated with Class B networks, where the first two fields identify the network, and the second two fields identify specific hosts. Field three is often used as the identifier of the subnet.

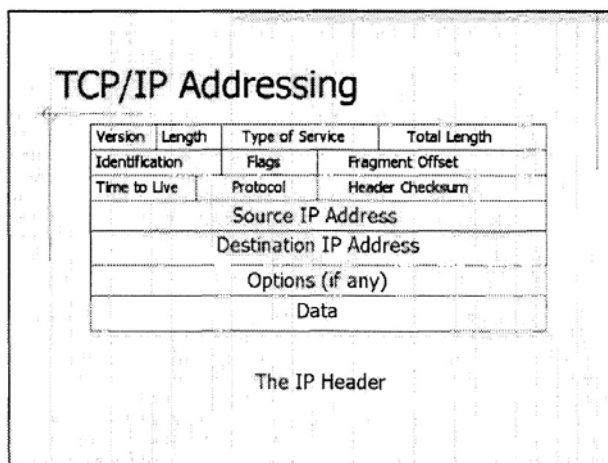
In a Class A network, field one is always smaller than 127, 127 is the loop-back number. So, there are relatively few, there can only be 126 or so Class A networks, and there are a few. I noticed in Bill's presentation—Bill Cleveland's presentation—none of the networks were identified as Class A networks.

You can also have Class C networks, in which field four identifies the host and the first three fields identify the particular network. So, Class C networks are relatively small networks, they only have 256 hosts in the network.



Just a word on IP addressing. The TCP/IP addressing is normally regarded as a layered system. So, at the highest level, you have sort of the application layer, which is the data itself. Attached to that, application header. Attached to that is a protocol header. The protocol that we are typically familiar with is TCP/IP, although there are three or four other kinds of protocols that are relatively common.

The discussion that Bill was talking about principally is the IP layer, which is the IP header, and then finally the hardware layer attached to this. So, in order to get the application data, you have all this other stuff, which is basically metadata about routing and so on, and what kind of stuff is in the IP address.



So, here is the basic structure of the IP header. It is the version, IP version four is the common version. IP version six is in the wings. The length, the type of service, total length, some identification, flags, fragment offset, time to live. Each packet has a maximum possibility of 255 lifetimes, and each time it transits a router, the time to live gets decremented. So, if it goes through too many routers, it ceases to exist. There is information about the protocol and critical information of source IP and destination IP and other options.

TCP/IP Addressing

◆ Some Flag Types

- ACK – used to acknowledge receipt of a packet
- PSH – data should be pushed to application ASAP
- RST – reset
- SYN – synchronize connection so each host knows order of packets
- FIN – finish the connection

One of the things that is of significant importance is the flag. Of course, what really is important is the data, but that almost sort of gets lost in the, I guess, noise. Some flag types are acknowledgment flags ACKs, PSH flags that say the data should be pushed into the application as soon as possible, a reset, a SYN, or synchronization of a connection, and a FIN, which is the finish of the connection.

TCP/IP Addressing

| HOST 1 | HOST 2 |
|---------|---------|
| SYN | SYN/ACK |
| ACK | |
| PSH | |
| PSH | |
| PSH | |
| ACK | ACK |
| FIN | PSH |
| ACK | FIN/ACK |
| ACK | PSH |
| FIN/ACK | FIN |

Possible TCP Session

So, a possible IP session might be that the Host 1, or the computer that is seeking data will send out a SYN packet. These are the 40-byte packets that Bill is talking about. Host two will send out a SYN/ACK, will acknowledge the fact that it has received this packet. So, one possibility for discovering intrusion is if you are receiving a lot of SYNs with no ACKs, no SYN/ACKs, then somebody is probing the system, and they are not probing the right ports, so they are not getting any acknowledgments.

So, one of the ways of discovering that you are being probed and things maybe aren't quite what they should be is looking for the ratio of SYN that is in ACK packets.

Once there is an acknowledgment, then the host one acknowledges the fact that it received the SYN/ACK packet, and then started sending out the data. So, essentially, those are the PSH things. When that stuff is received, the host two sends the acknowledgment. It pushes out some data. The host one will acknowledge it and say, I am done, and the host two can send out a FIN/ACK, which is the final acknowledgment. It is an acknowledgment of his FIN.

Host 2 may not be done, so he may send out some more data. Finally, you get to an acknowledgment of that. Then finally, the sign off handshake is that Host 2 will say, I am done, send the FIN packet, and then the other host will send the FIN/ACK, and that

ends the session.

Clearly, this is a very abbreviated kind of session. If you are getting Internet traffic from e-mail or, for example, from Web pages, there is maybe lots more of this going on, but this is sort of the typical prototype.

IPv6

- ◆ An IPv6 address is a 128-bit address arranged as 8 groups of 16 bit numbers separated by colons
 - e.g. EFDC:BA62:7654:3201:AFDC:BA72:7654:3210
- ◆ Leading zeros may be omitted
 - e.g. 1060:0000:0000:0000:0006:0600:200C:3268 = 1060:0:0:0:6:600:200C:3268
- ◆ Any sequence of single zeros and colons may be replaced by a double colon
 - 1060::6:600:200C:3268
- ◆ All IPv4 fit in ::****.****
 - 130.103.40.5 in IPv6 is ::8267:2805
 - Also hybrids are allowed ::130.103.40.5
 - Note 130 in decimal is 82 in hex, 103 is 67 in hex, 40 is 28 in hex, and 5 in decimal is also 5 in hex

Just to give you a scale of things, an IPv version 6 address is a 128-bit address, arranged as eight groups of 16-bit numbers, separated by colons. So, it is hexadecimal and so a typical IPv6 address may be configured something like that. In general, leaving zeros may be omitted. So, instead of writing all these zeros in, you can just put the final zero in, and so you can compress the address quite a lot by omitting leading zeros. If you have a sequence of zeros, then that can be replaced by a double colon. So, the address can be compressed even further.

Now, it turns out the last two fields are sufficient to fit in current IPv4 addresses. So, for example, an address that is 1310345 can be compressed as this. The way that is, is that 130 in decimal is 82 in hex, so the 130 becomes 82, the 103 in decimal becomes 67 in hex. So, this part is the first two fields of an IPv4 address, and then the second part, 2805, corresponds to the party in five in the IPV 4 address.

Since everything else would be leading zeroes, we can simply put the double colon in. So, this is the IPv6 version of the IPv4 address, and there are sort of hybrids allowed, so that you can put in the single point instead of a colon to get something.

IPv6

- ◆ How many hosts are possible in IPv6?
 - $3.4028236692093846337460743177 \times 10^{38}$
 - Less a few reserved addresses
- ◆ IPv4 has basically 4 billion
- ◆ Visualization of everything is hard even in IPv4

Now, a question is, how many hosts are possible? One reason I am sort of going through this is that, if you are interested in visualization, you are interested in how many

things you can see. If you sort of multiply all that junk out, you get 3.4 times 10^{38} , which roughly means that every 30 atoms can be having its own IPv6 address. So, there are a lot of these addresses.

AUDIENCE: It is still a factor of 30 short.

MR. WEGMAN: I was thinking on the way in, because I have sort of stupid random thoughts while riding the subway, that one of the things that, you know, in Star Trek they have these transporters. They got it wrong in Star Trek because they always had these pattern buffers. See, this would be a sufficient amount to address every molecule in your body or, in fact, every atom in your body. So, if you had an IPv address for every molecule or atom in your body and you transmitted just the location and type of atom it was, you could clearly, as long as you had a streaming algorithm, you could clearly implement the transport. In fact, I was thinking even further that, if it lost a few packets, I wouldn't mind, you know? [Laughter.]

So, in IPv4, there are basically 4 billion, and so, visualization of everything is hard to see, even with IPv4.

Ports

- ◆ There are some $2^{16} = 65,536$ ports for each host
 - Some standard services use standard ports
 - e.g. ftp – 21, ssh – 22, telnet – 23, smtp – 25, http – 80, pop3 – 110, nfs – 2049, even directv and aol have standard ports.
 - Unprotected (open) ports allow possible intrusion
 - Scanning for ports is a hacker attack strategy

Now, in addition to the issues of sort of looking at all possible Internet addresses, each machine has 2^{16} or about 65,000 ports. So, looking at ports is an interesting thing. 65,000 is something that is visualizable. So, looking at ports is an interesting thing from the point of view that there are some standard ports, but attacks—intrusion attacks— often attempt to go after ports that are not properly closed off or not properly unused. So, one thing that can happen is that ports can get scanned to see if something is misallowed with those things.

So, looking at attacks that scan ports and seeing if you can do that visually is interesting. I put some popular ports down. FTP is 21, simple-mail is 25, HTTP is typically 80, it is also 8080. Top three servers, or mail servers are 110. NFS is 2049. One of the things that I thought was interesting is that even Direct TV and AOL have standard ports that they use.

tcpdump

- ◆ **In order to analyze network traffic data, data are captured by programs called "sniffers"**
 - tcpdump is such a program
 - Sniffers capture all or part of the data flowing through a given point
 - At GMU, we have been allowed to install a sniffer outside the firewall capturing all packet header data flowing in and out of GMU
 - Total traffic in and out of the class B network at GMU is in the multi-terabyte range
 - 35-40 megabytes of header information per minute, 50-60 gigabytes per day
 - Even within the relatively small statistics subnet, we see 65,000 -150,000 packets per hour (during final exams when traffic is low)

So, in order to analyze the traffic data, you have to basically capture the traffic data. This is done—at least in our case—with something called sniffers. I think Bill had a less provocative name for such things. Anyway, these things sit outside the firewall in our case. We have implemented a couple of these things and I will show you a little bit about this. The idea is that the program will capture something like TCP Delta's program that captures the header information of all data flowing through a given point.

Ours is implemented in such a way that all traffic in and out of George Mason University is routed through—not routed through, but also sent to this machine. The sniffer is actually a covert machine. It is not visible on anybody's system. You can't tell that it is there, basically.

So, our network at George Mason University is a Class B network, and just to give you some calibration, the data in and out, the traffic in and out of the Class B network, which is a sort of medium-sized network, is in the multi-terabyte range daily. We can collect, when we do, on the order of 35 to 45 megabits of header information per minute, something like 50 to 60 gigabytes of header information per day. Even within our relatively small statistics subnet we have eight faculty members that sit on statistics subnet, plus a secretary or two.

Last week, just in getting ready for this, we were collecting a little data just to see what kind of things were going on. Last week was final exam week, so we didn't have as much activity as we usually do. Usually only the faculty that are giving final exams are around. Even in this relatively small subnet, we were getting 65,000 to 155,000 packets per hour.

Observations

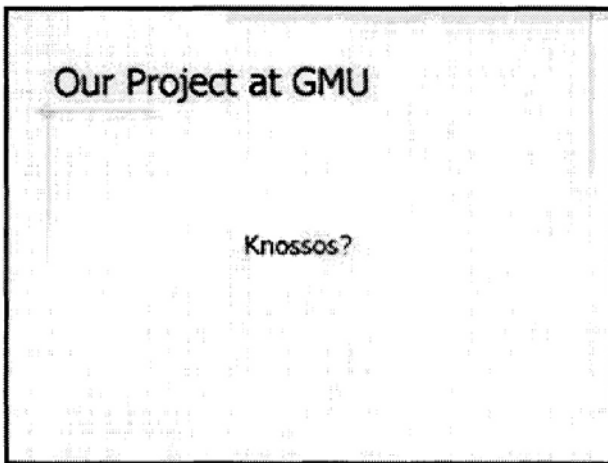
- The scale of traffic, although discrete, is for many purposes essentially continuous.
- Storage of all header data is not possible. We have terabyte storage capability, but streaming algorithms and methods are essential. Recursive algorithms are essential.
- Fortunately, not every computer in the system talks to every other computer, but even visualization methods are stretched to their limits.
- Nature of traffic changes during the day.

So, some observations. With the scale of traffic, even though things are usually thought of, in computer terms, as being discrete, for many purposes, they are essentially continuous. If I tried to look at all IP addresses, somehow I couldn't do that in any reasonable discrete form. If I tried to do graph theory associated with that, it would be tough.

Clearly, the storage of all header data is not possible. We have a terabyte storage capability and that would run out in not too long a time. So, streaming algorithms and methods are essentially, and essentially, recursive algorithms are of great interest. I think, in general, for streaming data, that is something we would like to make a comment on.

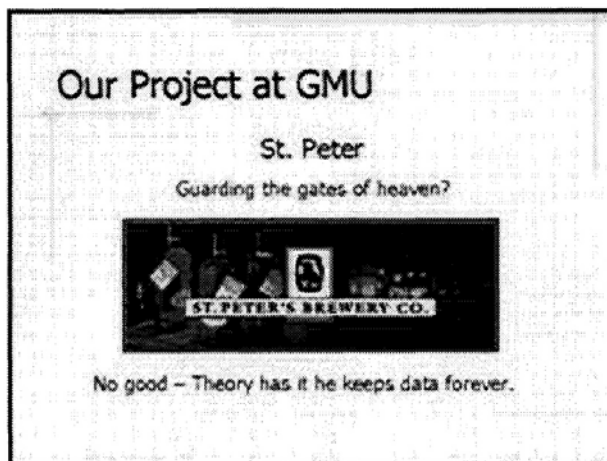
The good news is that not every computer system talks to every other computer system. So, the networks are relatively sparse. Even so, the visualization methods typically are stretched to the limit. Of course, the nature of traffic changes during the day. Within George Mason University, we have very little traffic between about 3:00 and 5:00 in the morning, comparatively little, very heavy traffic up to about 10:00 o'clock when people are sorting out, I guess, their e-mail. Traffic tapers off a little bit. George Mason has a lot of evening students, so traffic builds up again relatively heavily in the evening. As students go to their dorms and surface naughty Web sites, we get a lot of traffic after 10:00 o'clock.

By the way, I worked for a while at the Bureau of Labor Statistics. The Bureau of Labor Statistics has a filter that cuts off any IP that seems to have nasty stuff on it. One of the discussions, when I was working at BLS, was that the *Washington Post* Web site was cut off because it apparently had offensive material in it.

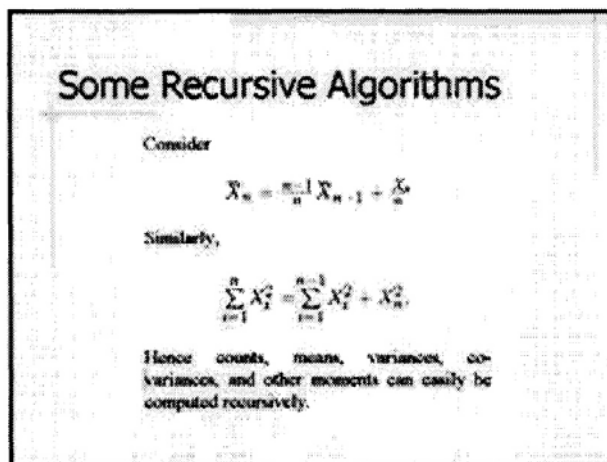


My friend, Don Faxon, who is working with me on this project, went to the Isle of Crete during the summer. My wife was in the hospital and I couldn't go to give a talk, so I sent him to give a talk. He started calling this project Knossos. I don't know why, because that is a city in the Isle of Crete. I was not fond of this name.

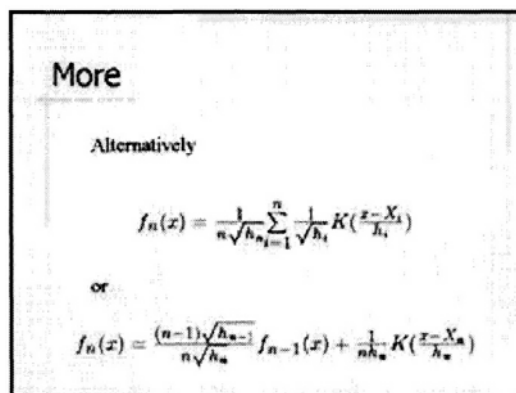
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



I thought, as long as you are going with Greek mythology, certainly Cerberus would probably be a good name for this. Cerberus is the three-headed dog that guards the gates of hell. The comment is, after all, they do call it a firewall, and the question is, which side of hell is on our gateway—is it on our side or is it on the outside?



I thought maybe a better name is to call it St. Peter because he guards the gates of heaven. But for a streaming data set, this is no good, because in theory, St. Peter keeps the records forever. So, you are potentially in trouble. He stores too much data.



So, I decided it would be Project Santa Claus, keeping with the season. He wants to find out who is naughty or nice, but he discards the data after a year, so he is clearly a streaming data analyst.

So, let me toggle over to show you a little bit about this. This is just a little bit of stuff that we have put together. It has some interesting things in it. You can go to a glossary and you can look up any of these acronyms that you can imagine what they might actually mean. You can do things like look up port numbers. So, if you are interested in finding out which port is which, you can do such things. You can look up route zones. So, if you want to know what a suffix is and where that place is, you can do that. One that has been advertised a lot lately is “tv.” So, if you want to know where “tv” actually is, it is in Tuvalu, wherever that is. So, you can do some things like that.

We have two stations, Ariadne, which is the sniffer, which sits outside the firewall, has roughly a terabyte of storage capability, and it runs the TCP dump and other kinds of stuff. You can find out some stuff about that, if you are interested. Theseus is the analysis station which sits in side of our lab. By the way, as Bill pointed out, these are sort of sensitive issues. The content of what is being looked at is an interesting privacy issue. There is an assumption of privacy in general on the Internet, although it is not so clear that that is maintained. We could, in principle, track the actual stuff that people are downloading and certainly the IP addresses that they are downloading. One of the interesting things in the state of Virginia is that faculty are not allowed to look at X-rated Web sites, but students are, and so are secretaries. It is against the law for me to look at anything bad, unless I have a project like this.

Steve Marin was asking me why I do this instead of using somebody else's database. Let me show you a little bit of sort of what data looks like. So, here is some data that comes out of—this is what Amy would call sort of Level 2 data, I guess. This is sort of semi-raw data that comes out of—that is slightly processed coming out of something like TCP dump.

We can look at substructure data. One variable of interest is the source manufacturer, and source serial number. So, we can track things to specific machines. This is the time stamp that is put on by our machine, when the packet comes through it. So, we do have a time stamp associated with it.

As somebody pointed out the other day, no talk of Ed Wegman's would be complete without some kind of graphic parallel coordinate display. So, here is a scatterplot matrix of things that we might be interested in. We could increase the size of the pixels. One of the interesting things is to go to the parallel coordinate display. Source manufacturer is an interesting thing.

If we don't know the MAC number of the machine, it continues to record the IP address. So, these things over here on the right-hand side are things where we couldn't identify the MAC number of the machine, but those are things that have an IP address. Just for purposes of discussion, we can get rid of those things, and we see a couple of things that are of interest.

AUDIENCE: Ed, could you just be clear about that? Do you mean you don't have a MAC number?

MR. WEGMAN: We don't have the MAC number for it. We have the IP but we don't have the MAC. So, here are a couple of things that are of interest.

So, the thing I just colored in green, let me do one other quick little thing here. We can drag this down. So, the thing I just colored in green, you will notice, is one machine source manufacturer and one source serial number. We only have one machine and one serial number. That, in fact, is our router. What we have here is, this is delta

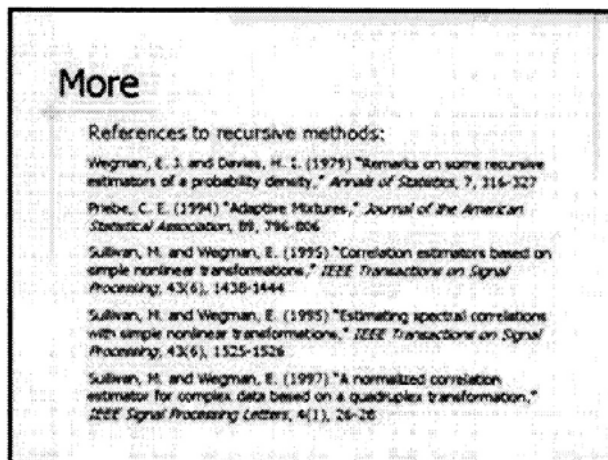
time, this is time. The frame is essentially, in an IP there is a sequence number.

So, what is happening is there is a lot of data coming in. There is a batch of data coming in here, and it is coming in through this time frame. So, these are the sequence numbers that are associated with the data that is coming in.

Here is another probably Web page that is coming in, and here are the frame numbers associated with that. Here is a third item, and the frame numbers that are associated with that. So, those are big files.

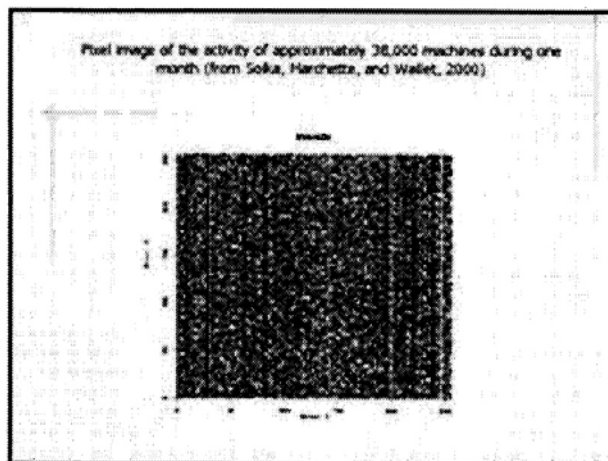
The red thing also has only one IP, one source manufacturer number, and one source serial number. So, that turns out to be our Web server. So, sad to say, we get less traffic. People aren't interested in us as much as they could be, I guess, but occasionally we get some hits. Just to give some other interesting things, we have a lot of Dell machines and we have a number of Gateway machines. So, we can identify those.

This is all stuff that is internal to our subnet. So, that is a quick tour of something that you could do. I am going to exit Santa and go back to the PowerPoint.



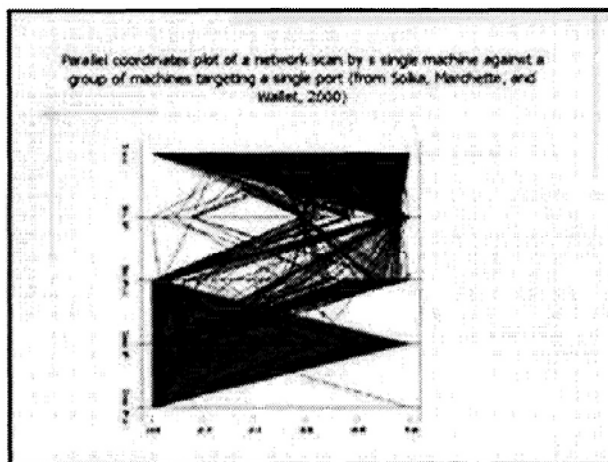
So, I wanted to say a word on recursive algorithms, because recursive algorithms are the algorithms that you basically need to deal with streaming data.

Clearly, things like means and moments can be formulated recursively. The idea with recursive algorithms is that you don't want to store data any longer than you have to. So, you want to keep whatever quantity you have and do an update. So, things like counts, means, variances, covariances, other moments, are easily computed recursively.



Probably less well known is that there are recursive forms of kernel density

estimators. So, for the people at Rice University who think that probability density estimation was invented, there are some recursive forms of kernel estimators.



Additional References

Marchette, D. J. (2001) *Computer Intrusion Detection and Network Monitoring*, New York: Springer-Verlag

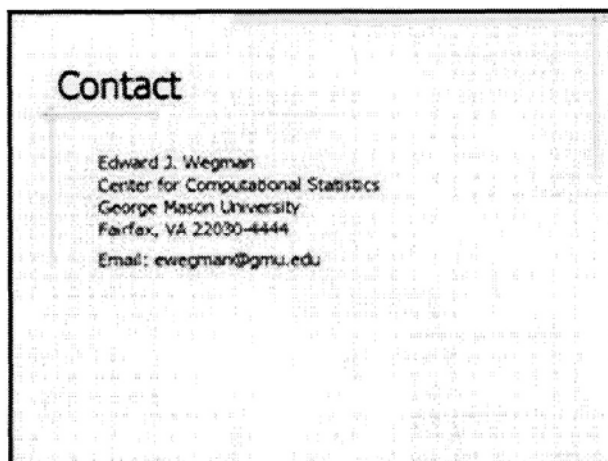
Stevens, W. R. (1994) *TCP/IP Illustrated, Vol. 1*, Reading, MA: Addison-Wesley

Leiden, C. and Wilensky, M. (2000) *TCP/IP for Dummies (4th Edition)*, New York: Hungry Minds

Solka, J. L., Marchette, D. J. and Wallet, B. (2000) "Statistical visualization methods in intrusion detection," *Computing Science and Statistics*, 32, 16-24

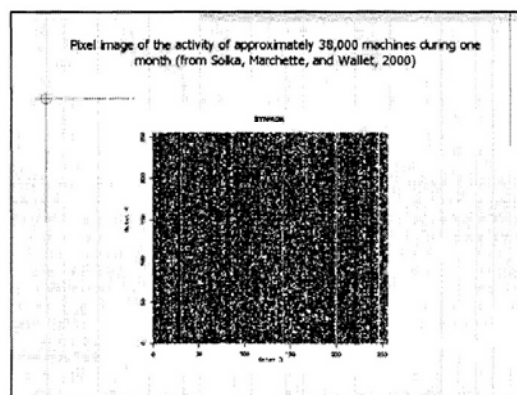
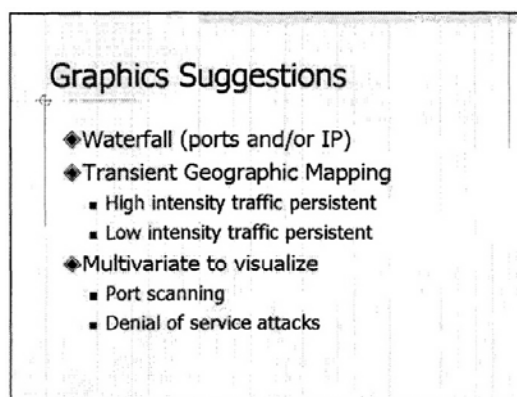
This is one and this is another one. These have essentially the same properties as ordinary kernel density estimators, the same kinds of convergence rates. They can be extended to multivariate settings as well. So, if you are interested in streaming data and collecting density information, you can do that this way.

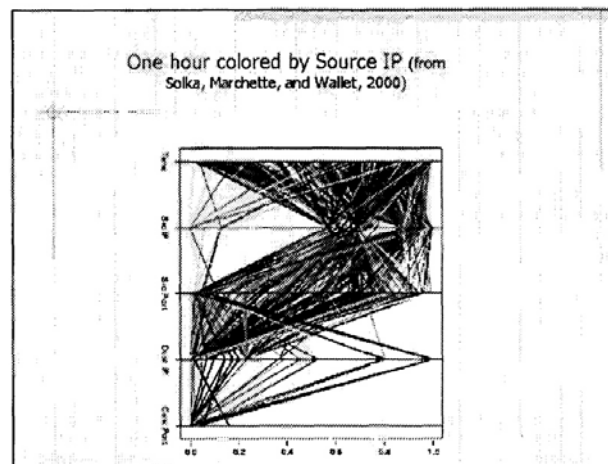
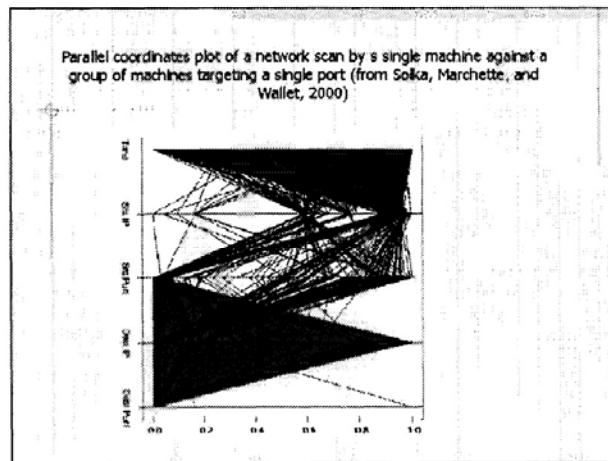
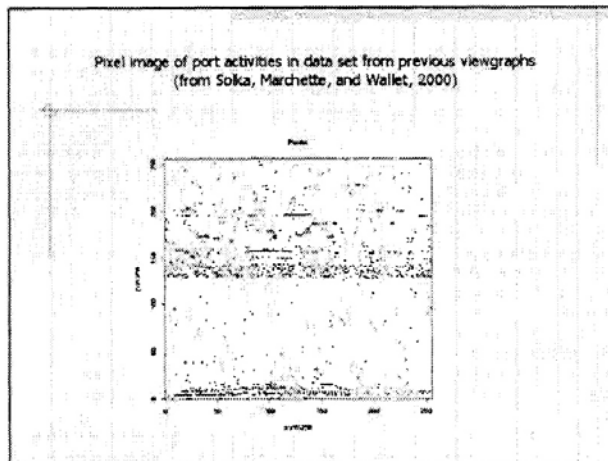
I thought Daryl Pregibon brought up an interesting idea yesterday, which is the idea of exponentially weighted averages. If you have something that is sort of an exponentially weighted average, that takes some data and creates an object, then you can do this recursively, and you can do this with almost any old object. So, if you have a starting object and you have some data that updates it in the proper way, you can do streaming data with this as well.



I just wanted to give you a couple of references. This paper, I told someone the other day of the rusty staple principle which is, if the staples on your reprint are rusty, it is time to cycle the research, because clearly everybody has forgotten about it. I did some work with Ian Davies in the late 1970s that did this stuff on recursive density estimation, and gave exact rates of almost—Carey Priebe, who is in the audience here, did the stuff on adaptive mixtures, and adaptive mixtures has a formulation that is a recursive formulation as well. I had a student a few years ago, Mark Sullivan, who did stuff on correlation and spectral estimators in a recursive fashion, so, computationally efficient stuff.

Some of the things we are planning to do is some waterfall graphics, where we scan ports and IP addresses. One of the things we are planning to do is transient geographic mapping.





We are interested in both high-intensity traffic being persistent and low-intensity traffic being persistent. The idea is that people who are trying to intrude often do it very, very quickly. So, what we would be interested in doing is looking at a geographic map that had the locations of IPs that are very short-lived.

We, for example, at George Mason have had a lot of trouble with people in mainland China probing our systems.

There are just a couple of graphics that I will hurry through. Again, as someone pointed out, one has to have a parallel coordinate plot.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

Additional References

- Marchette, D. J. (2001) *Computer Intrusion Detection and Network Monitoring*, New York: Springer-Verlag
- Stevens, W. R. (1994) *TCP/IP Illustrated, Vol. 1*, Reading, MA: Addison-Wesley
- Leiden, C. and Wilensky, M. (2000) *TCP/IP for Dummies (4th Edition)*, New York: Hungry Minds
- Solka, J. L., Marchette, D. J. and Wallet, B. (2000) "Statistical visualization methods in intrusion detection," *Computing Science and Statistics*, 32, 16-24

So, some additional references—and a lot is due to Dave Marchette, who is also in the audience, who wrote this wonderful book on computer intrusion detection and network monitoring—Stevens and this Leiden-Wilensky book, are both useful in terms of understanding TCP/IP addressing and so on.

Some of the pictures that I just hurried through were from Solka, Marchette, Brad Wallett, all three of whom were my students.

Acknowledgements

Our work is supported and surveillance equipment provided by the AFOSR.

Dr. Wegman's work is also supported by DARPA's ISP program through a subcontract with the Johns Hopkins University, Carey E. Priebe principal investigator and by ONR

Special acknowledgement goes to David Marchette whose diagrams we liberally borrowed in slides 5-8.

Contact

Edward J. Wegman
Center for Computational Statistics
George Mason University
Fairfax, VA 22030-4444
Email: ewegman@gmu.edu

Just for acknowledgment, our work here and the work on surveillance is supported by a critical infrastructure grant from the Air Force. I also work on the DARPA ISP program with Carey Priebe as the principal investigator and, as I said, Dave Marchette plays a key role. So, that is it.

MS. MARTINEZ: While we are switching, we have time for one question.

AUDIENCE: Ed, it doesn't strike me—looking at these packets, it doesn't strike me as a visualization problem. So, I am sort of curious why you framed it that way.

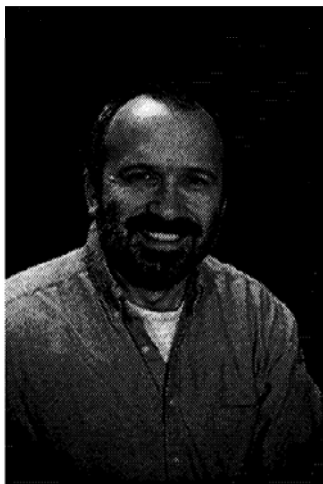
MR. WEGMAN: I am not sure why you don't think it is that. So, I guess I am curious the other way. I guess I see a lot of things as visualization problems.

One of the issues that we are particularly interested in is having a monitoring system that can detect, very, very quickly that we are getting intrusions into the system. We want to be able to cut off things quite rapidly, I guess. I am not sure that is an adequate address any more than my answer to Steve Marin. I guess I am one of those guys who likes to do it myself.

Paul Whitney

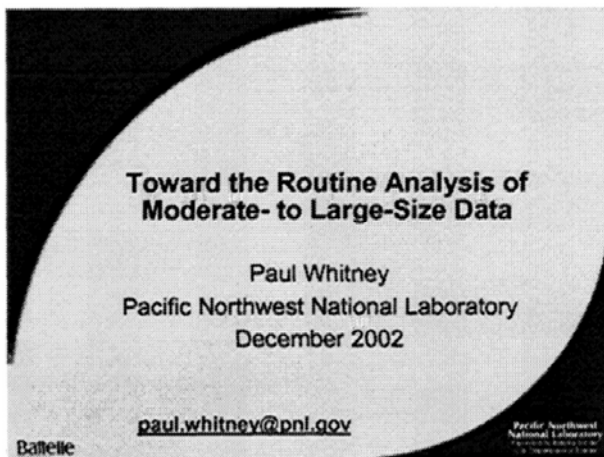
Toward the Routine Analysis of Moderate to Large-Size Data

[Transcript of Presentation and PowerPoint Slides](#)



BIOSKETCH: Paul Whitney is a scientist in the Statistical Sciences Group at Pacific Northwest National Laboratory. His research interests currently include the analysis of data objects associated with the contents of the Internet and data analyses associated with computer simulations. He has developed information retrieval methods, exploratory analyses algorithms, and software for these, notably for image and text data. He has contributed to the development of a variety of information visualization methods. Currently, Dr. Whitney is exploring data analysis challenges associated with agent-based models and developing methodologies for exploring and retrieving structured information such as transactions and relations.

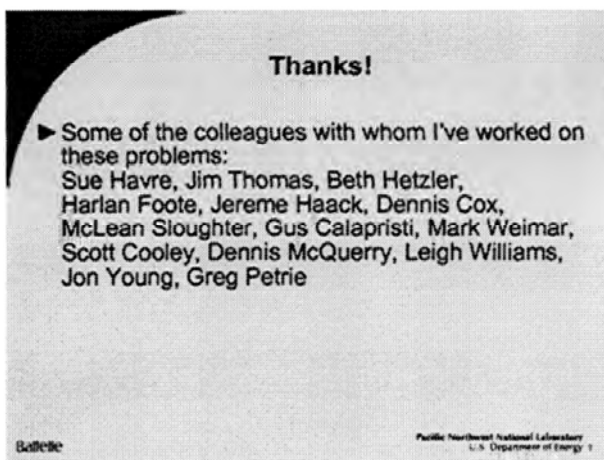
TRANSCRIPT OF PRESENTATION



MR. WHITNEY: Good morning. I don't know that I am going to start out as ambitious as looking at the whole network.

I am from Pacific Northwest National Laboratory, a group of about 40 statisticians there. We do all kinds of data analysis. There is this phrase that describes a fair amount of what we do, probably because of the situations we live in, and it is fighting our tools. They just don't quite do everything you want them to do. We have to do a lot of custom work.

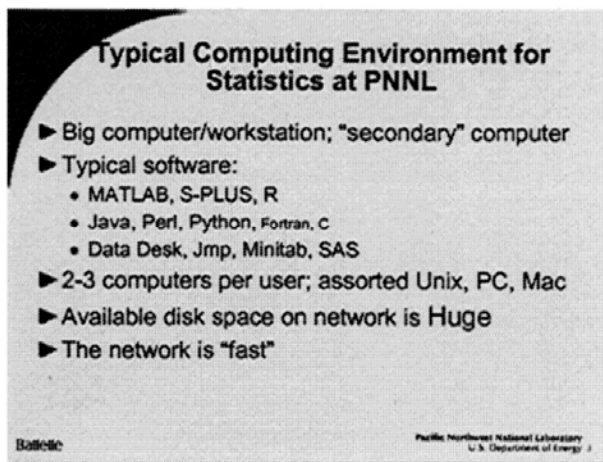
I really wish I would have recorded the sound of my hard drive swapping, just to play for you guys. I hear it all the time. It is really a key thought. If you have heard your own hard drive swapping, just remember that we are going through some of these analyses, some of the stories of what we have done.



I have a lot of colleagues. Here is a list of some that are related to the stories I am going to tell. I think one person here is in the audience. I think there are 15 people on that list, and 4 of them are statisticians. It turns out that there is a fellow there who is a risk analysis for things like nuclear reactors. There is a fellow there who is a remote sensor, some software engineers. There are people here who I really don't know what their technical background is, but we do have useful technical interaction. It just never quite comes up, what they did in school. Here is some data. It also happens to be where I am from.

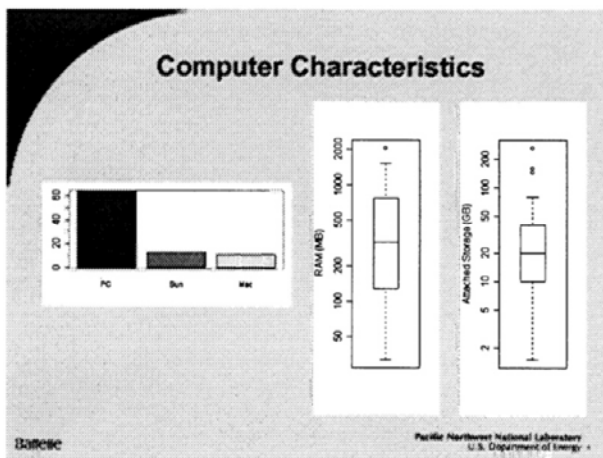
This is the Columbia River. This is some satellite image I downloaded from SPOT. It was a few years ago and I might have the name wrong. It is low resolution, but

you can see a lot of what is going on. This is the city here. There is the Snake River going into the Columbia, the Yakima you can't quite make out. There it is. I work right about there. This is agriculture, and it is blurry, probably not because the measurement is that bad, but you get an image in one form, you put it in another form, you put it into PowerPoint, and God knows what has happened by now, but there are a lot of those circles for irrigation. The reason that you don't see anything like that here is that that is the Hanford site. They don't do agriculture there. There are a few installations actually here that we will look at. I haven't got a good idea about why there is this demarcation. I suspect a fence. This is Rattlesnake Mountain.



Okay, I am going to describe some data analysis stories. Just for background, I would like to describe what our computing environment looks like, just so you can get some idea about the size and difficulty of challenge that we are facing here. Forty-some odd people, two to three computers per person, typically a newer one, an older one, some Unix, some Mac, some PC.

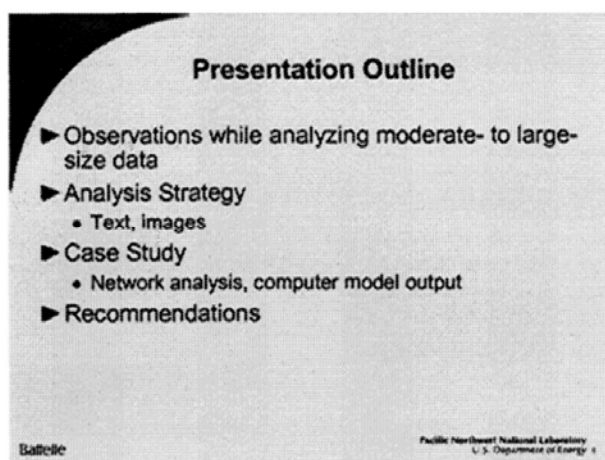
Our bread-and-butter software tends to be these guys here, because of the flexibility they give you. Some other languages, less and less Fortran and C over time. I don't think I have written anything in those languages in a long time. Then, packages, just depending.



We have got the potential for a lot of storage, the AFS share kind of thing, and the network is good. It is reliable. You know, a lot of PCs. The demographics associated with Macintoshes are interesting. Here is the RAM size. This is getting back to the swapping thing. There is one lucky dog who has got a couple of gigabytes of RAM and,

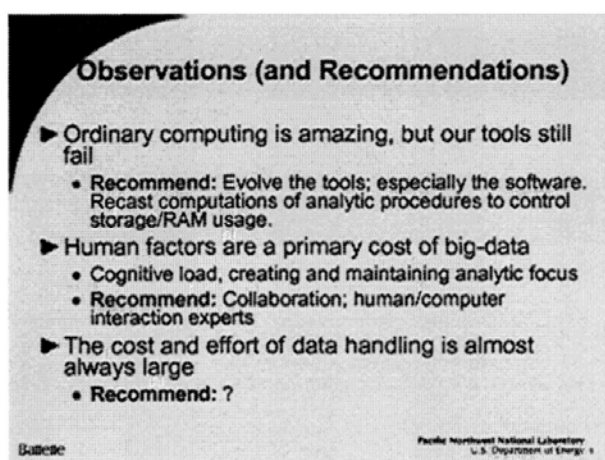
you know, my computer is right there, and then my portable is right there, pretty typical, good stuff, when you think about it.

Also, another thing to keep in mind is the computing model. It is the single work station computing model is what we tend to deal with on a day-to-day basis.



Here is an outline. I thought I would start at the end. So, the computers are good. They are great, and stuff still breaks. The reason stuff breaks has to do, I believe, with the software model that we are using. The realization of that is nothing deep, and the solution is just work, that you have to keep track of the RAM you are using, and that will have a lot of implications, I think, in the future.

It is an observation echoed very strongly by Ed, and Johannes is worried about memory, also, very explicitly in his talk.

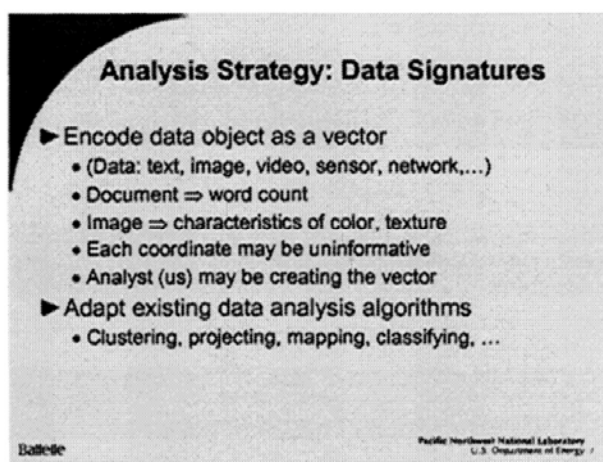


This failure of tools happens not only on streaming data or massive data, but ordinary data sets of a size you could presumably fit in memory will cause the computers to die with the tools that we typically use.

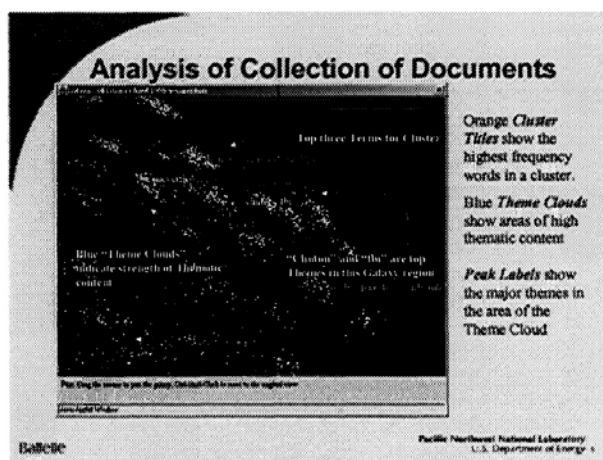
Then, there is another type of complexity that is a whole lot slipperier. I think it is hard to describe, but if you think about the potential complexity underlying a large data set, and how you start to get your arms wrapped around it, it starts to become kind of daunting. For instance, let's pretend that you did a cluster analysis of a homogenous data set, and you had some nice formula that said what the optimum number of clusters might be, and if you can get that down to 3,000, because that is the optimum, well, you are just not done. That is still way too many for the human to absorb.

So, somehow the complexity isn't just in the data set. It is in the communication

of the information to the user, and that is a tricky thing, a very slippery thing. We spend way too much time handling data. I am sure that happens to everybody, and I don't have a deep recommendation there to get to. It is just that we work through it like everybody else does.



So, here is what we do. Here is a good strategy to consider, and think of the data as being something like that image of the Hanford area. First off, you have got this data. It is in a digital format, but we want to be able to use our standard data analysis tools for it. So, the first thing that we do, we make a vector out of it. There is a ton of ways to do that, and there is a lot of good work out there that can be used.



For instance, if you are dealing with a collection of documents, this isn't what you would use, but you could use as fundamental information just word frequency counts. The world has moved on from that, but that is a nice measurement to think about. If you are working with an image or a collection of images, you could imagine looking at textures, edges, various bits of color.

It turns out that both of those things, while being simple and way-too-briefly-stated there, they eventually can be made to work very well. It is kind of surprising and gratifying.

The characteristics of those are indicated here in these two bullets. One is sort of a social characteristic, the bottom one, and the upper one is pretty interesting. Each coordinate in that vector that you are using to construct a signature can be very uninformative. For instance, if you are making a vector of word frequencies to represent a document, and you have multiples of those because you have multiple documents, the

number of times that the word signature appears, that is a pretty low level of information about the objects in question.

It turns out, you put that together and you can start to do useful data analysis. Similarly, with an image: A color histogram, by itself, might not tell you a lot, but you put that together with other information and you start to get somewhere.

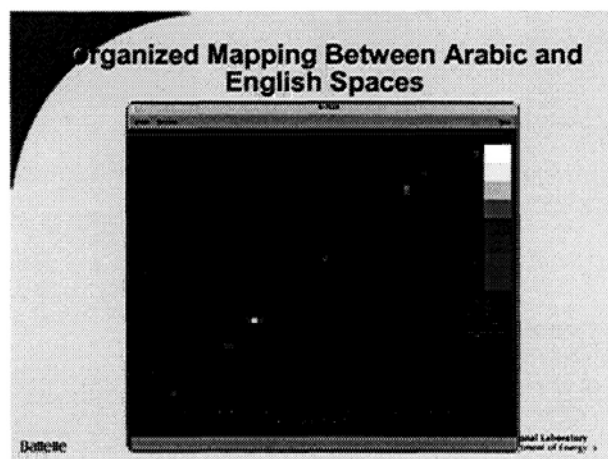
This social characteristic is important. Guys like me can start making up sensors, virtual sensors, basically. You can, too. That basically changes the nature of our profession and our jobs as data analysts, I think, in a good way.

Finally, why would you care about a strategy like this? Well, people have been coming up with data analysis algorithms forever. It is great. If you take a strategy where you encode some data object as a signature vector, that means that you can borrow algorithms from all kinds of staff packages, neural net packages, whatever, to try things. So, it is an effective strategy, basically, for running through a lot of analysis potential.

Here is a way too busy showing of a collection of documents. Let me back off that one and go to something a little simpler. Suppose you got 400 documents. You want to know what is in there. And you have only got 30 minutes. So, you can't read them all, that is out of bounds at this point. Well, there are tools out there that will basically look at those, summarize what, say, the key words are, and do a visual clustering of the collection. Then, even better, start to label the various regions in ways that you can hopefully begin to understand what the generic contents of that collection are.

So, the technology. You make one of those signature vectors. It turns out you need the coordinates to be meaningful.

You do a little non-metric multidimensional scaling. There is a lot of artistic license in this particular one, and you go. So, it is a good functional thing, and you can start to build analytic functionality on top of that as well.

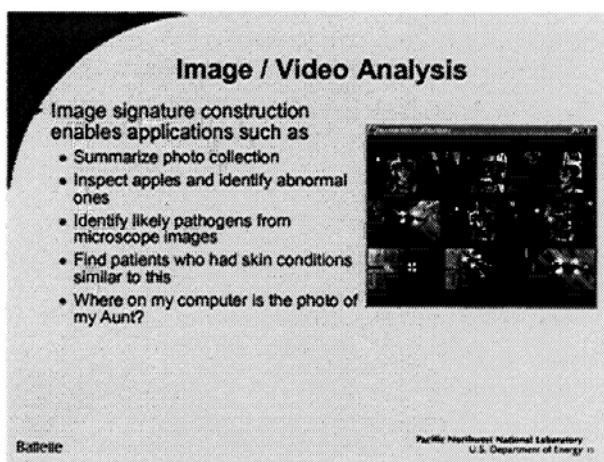


This is a dangerous picture in the sense that I have never been able to explain it very well in a public forum, or a private forum either, probably. The picture suggests there is some sort of relation going on. Let me indicate broadly what it is. You have matched pair data. Think sophomore statistics here. It turns out that one measurement is one of these text vectors for the English version of the document, and another measurement is the text vector for the Arabic version of the document.

You have got, then, in some sense this regression problem you can do from English to Arabic or Arabic to English. That part, by the way, has just become standard data analysis. It is regression and this is a very loosey-goosey regression display. By the

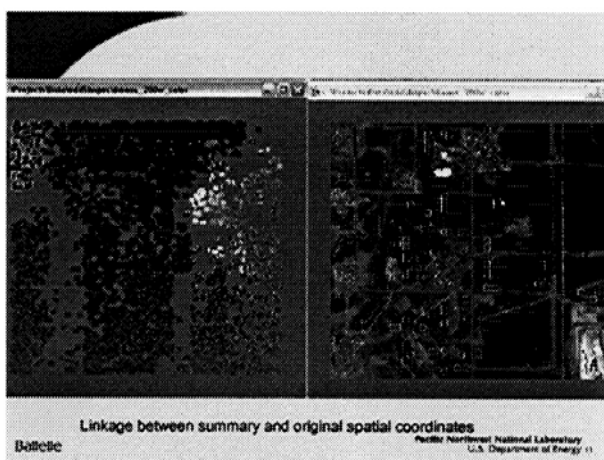
way, it does show that there is some potential for mapping there. The reason you might care is that maybe you don't speak one of those languages. So, you say, okay, I am going to calculate my vector for this—you know, I speak Arabic and somebody gave me this darned English document. So, I will calculate my vector for it, I will plug it into the regression formula, I will find what Arabic part of the space it lies in and then say, oh, that is just like these other documents.

So, it gives you a way to do that kind of problem, again, based on this simple idea of calculating these signatures, using standard data analysis procedures, and then mapping back to the problem space.



Another type of data object, image and video. It turns out that there are walls of books of how to do image analysis. So, we don't have to do that. The science is done.

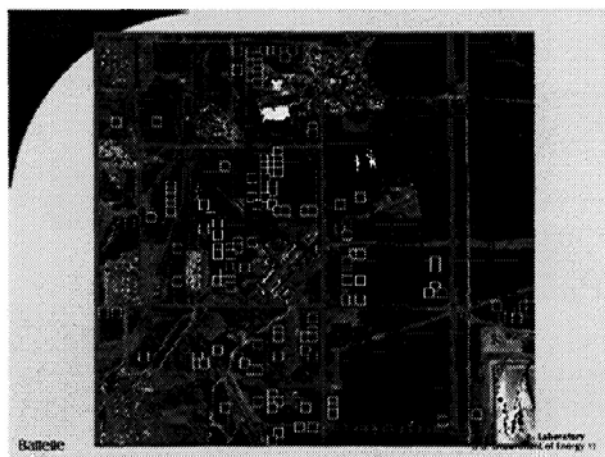
What they don't have done is things like, well, I have 10 of these images in my shoe box. I would like to sort them out. I have never actually organized my photo collection. How do I go about doing that? Well, if they are digitized, you can calculate a vector representation for each of those images. You can do one of those visual clustering type things and do a multi-dimensional scaling and show basically this organized collection of images, and that is one way to go.



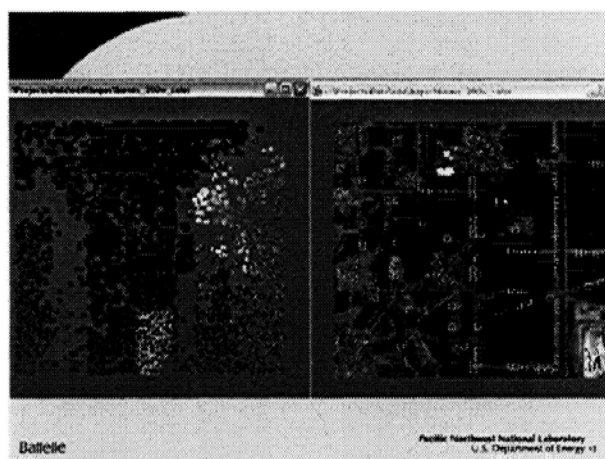
What this shows is something similar for a small videoclip. I am assuming 80 percent of you can recognize what videoclip this is. The calculations are really simple here, that led to this picture.

We did one of those signature vectors for each frame. We did a cluster analysis for the signature vectors. We took the picture with the vector nearest the representer

vector, and then just show it and say, well, there is a label you can slap on that tape that gives you some idea of the content in a quick and dirty fashion.



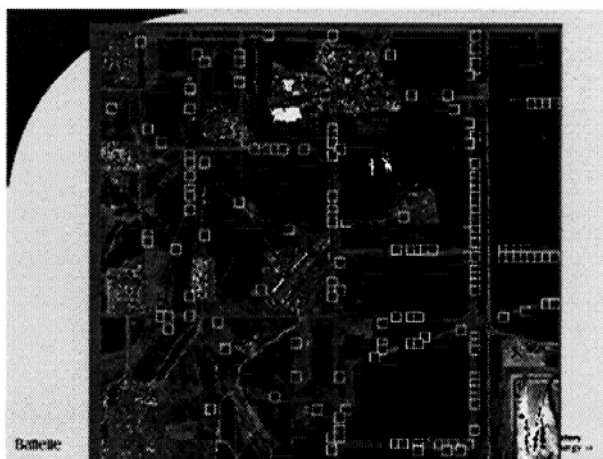
The idea just goes on. Here is the Hanford site, a little part of it. The 200 area is what it is affectionately known as. It is an ICONUS shot of it, so it is one of the bands of that. You can see, there are roads, some buildings, God only knows what, and so on.



What this picture is, is what happens if you calculate a little vector that describes the content, just sort of there and there and there and there, and get a little subpicture around that vector location and say, well, I am going to do a multidimensional scaling type thing for those vectors.

To indicate the content, I will just show the little pictures near each vector. Then you say, okay, I have got this. Have I got anything interesting from an exploratory analysis or even classification point of view, even though it was an unsupervised thing? So, you know the data base has been around for a while and this data has been around for a while, so you can start doing brushing ideas.

So, you grab a bunch of these little pictures here in this region of this multidimensional scaling thing and see what you grab over here, just to get an idea of whether you have done anything useful. Okay, we have got a lot of these sort of benign regions, fields, as it were. Actually, it would be more like gravel fields there.



Let's grab another region and see what we get. Well, we got roads, when you look at it.

So, it becomes—you develop the potential for building some interesting data analytic tools with that strategy, and it kind of gets you to the place of this, you know, all of these things, it is either the statistician manifestation or statistician hubris, that everything is data.

You have got all of these objects. You have got this strategy. You can use it and see what happens. So, let's talk about some more data.

A slide titled "Assorted Data Types" with a list of data types: Unstructured text, Image, Sensor, Categorical, and Video. It includes a survey example: "The DOW rose 32 points..." and "Survey example: Rate your work space from 1 (poor) to 10 (excellent); 1 2 3 4 5 6 7 8 9 10". There is a box labeled "Internet data" and a large box at the bottom that says "Everything is Data". The slide is from Battelle and Pacific Northwest National Laboratory.

Let's back up a second. So, you have got network traffic. You have got economic models. I was going to also talk about an error reporting data base, very diverse data objects.

A slide titled "Analyzing Network Traffic Data" with a list of challenges and constraints. The challenges include summarizing contents of streaming network contents, working with packet payloads, and characterizing contents "going by" at a particular point. The constraints include privacy, data handling (format of network traffic, assorted tools like tcpdump and tcpshow), and streaming data. The slide is from Battelle and Pacific Northwest National Laboratory.

Network traffic, you have seen some information on the format of that, but we have got a strategy for summarizing the content, potentially. So, we will show how that begins to play out.

An economic model, it turns out we spent some time analyzing the content of the output of an international economy, just a simulation of it. We have no idea, actually, if the model was any good. There is probably a joke in there somewhere. It is, again, a whole other data type and how do you get your arms around it. Well, let's just dive in.

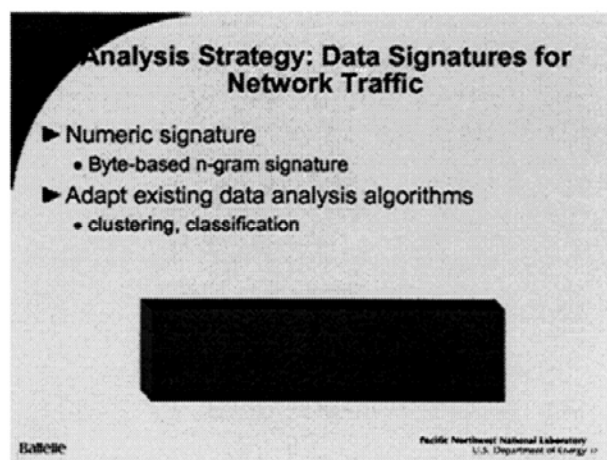
We are focusing on the content. I mean, there is a lot of work on the packet. You have seen a lot of that here today, and some indication of why you might be interested in that from a network design point of view, but we wanted to look at the payloads.

Our model is, we were going to look at the contents going by at a particular point. So, you imagine, if this is a network cable, I am just going to keep track of what goes by here, something that has that type of mental model.

There are tons of challenges. You have the ethical and legal issues associated with privacy. Data handling is always a problem. There are tools out there to help you get by that, but you have to learn how to deal with the tools. I am sure folks have gone through that learning curve here as well.

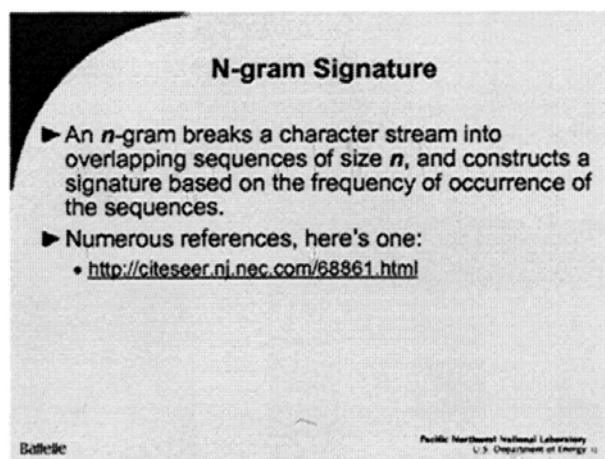
Then, it is streaming data, and we have kind of been challenged by data sets on the order of the size of memory in our computer, given our current tools. So, streaming data is a whole other level of difficulty.

So, what do we do? Well, we have got a strategy. So, let's try our strategy and see what happens. What can you do for a signature on streaming network data?

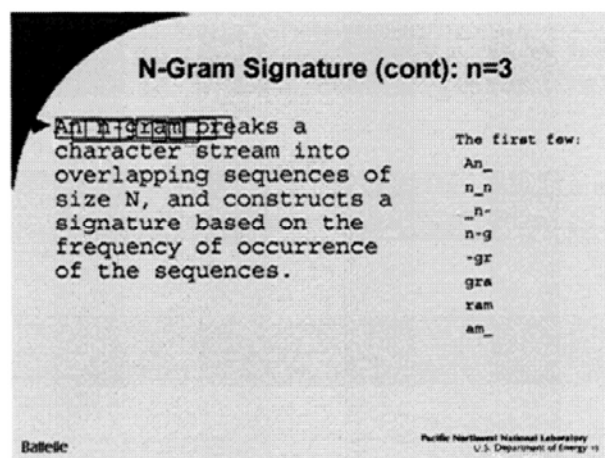


It is not just text going by and it is not just images going by. It is who knows what going by in those payloads. So, you have immediately got a challenge of, well, I can't just read the image analysis literature. I can't just go to the computational linguistics literature. It is everything. I have got to do something that handles everything.

Well, you know, there are some really straightforward things that have worked in the past for text that have at least the mathematical extension to a bite, to just digital data. So, we decided to look at byte-based n-grams and a couple of simple means of summary.



N -grams are fairly well known. I thought I would take a minute, just in case there were a couple of people that didn't know what those guys were.



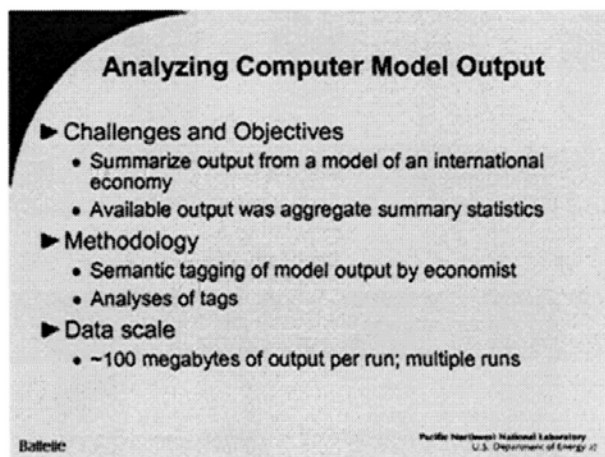
An n -gram signature for a text document is basically a frequency table of successive runs of characters in the document. So, if you have got this type of text, and you are doing a three gram [on the phrase “an n -gram”], then you have $A-N$ -space, you have got N -space- N , N -space-dash, and so on. You just accrue those things.

It turns out a fellow named Danoscheck did some very nice work showing that you could use those as a basis of a signature to distinguish among language types. Then, subsequently, folks figured out that, son of a gun, you could actually do a fair job of information retrieval based on those crude measurements.

This, again, emphasizes a point, that this, as the basis of the coordinate vector, $A-N$ -space, isn't awesomely informative, all by itself, and the vector $G-R-A$ is not buying you a lot either, all by itself. If you take that weak collection of measurements together, you can start to do stuff. There is nothing new about this.

This second bullet was our *T* stumbling block. My computer kept making that noise, basically, during this exercise, using the types of tools I showed you. It turns out that if we just took a little time and rewrote one of the clustering algorithms to just not use so much memory, we did much better. Basically, the compute time went from impossible to, okay, that is a cup of coffee.

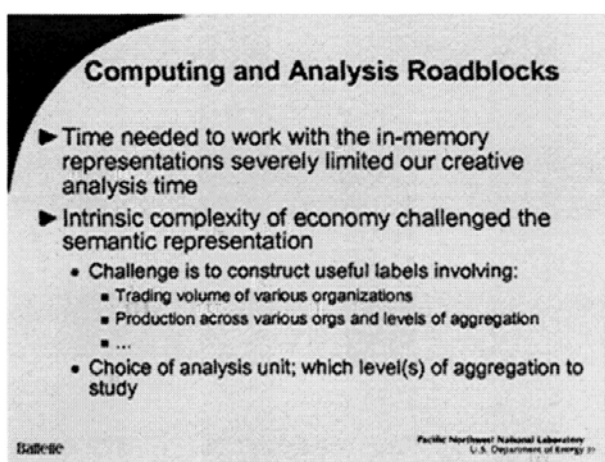
I am going to go right to the end. Ed wrote a very nice paper that appeared in the mid-1990s that laid out one way to organize your computations to fit on work stations. One of my favorite lessons learned was that my work station is pretty good. I can solve a lot of problems with my work station.



Even taking that advice, and just working with, say, a 100-megabyte data set in a 500-megabyte work station, and the kinds of tools that we typically use, stuff happened. Bad stuff happened. The computer made the noise. We charge our time by the hour. It is bad.

It turns out that this would be good. A bounded RAM would be really good, and recursive formulations, there are a lot of things out there. Use statistics are good, the common filter is good.

Once you know what you are looking for, you can do a Web search and start finding good theory out there in the computer science literature in the database area, saying how to organize your calculations to achieve this. I think they are just getting started there as well. There is a ton of good work that a lot of people could do.



Let's think about that together here a second. There is some, as I said, some work out there. There is some commercial software, actually, that is thinking along these lines,

probably some I don't even know about.

One of our spin off companies took some time to worry about keeping good control, the amount of RAM used in their calculations, but there is theory that you could do, too.

Some Relevant Theory and Strategy for Facile Data Analysis Computing

- ▶ $O(N)$ is good for computational complexity
- ▶ Implications for massive data:
 - Stay away from "traditional" hierarchical clustering, full SVD and multi-dimensional scaling algorithms; $O(N^2)$ for the distance calculations and matrix ops
- ▶ Even with $O(N)$, <sound of a hard disk swapping>
- ▶ $O(1)$ RAM would be good; explicit control over RAM use might be better

Ballistic Pacific Northwest National Laboratory U.S. Department of Energy

I mean, you could do the relative efficiency of a bounded RAM statistic versus an unbounded RAM statistic for various problems.

You could imagine re-architecting some of our standard tools, R , say, to start taking advantage of some of these ideas.

Data Analysis in Bounded Memory

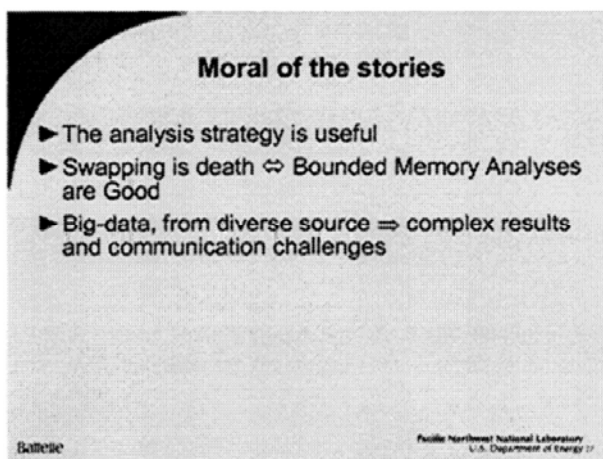
- ▶ U-statistics $\sum \phi(x_i, \dots, x_{i+j})$
- ▶ Kalman filter
- ▶ Some more detailed theory beginning to be developed
 - Queries for data streams
 - Clustering in controlled amount of RAM

Ballistic Pacific Northwest National Laboratory U.S. Department of Energy

Data Set Size Bestiary

- ▶ Small
 - Fits easily in memory, as does the analysis and analysis products (10 megabytes or fewer).
 - Our tools run smoothly.
- ▶ Moderate
 - On the order of the size of memory (100-1000 megabytes).
 - Trouble begins; but projects are possible
- ▶ Large
 - Fits on attached storage (10s of Gigabytes)
 - OK, now we're definitely in trouble. Analytic focus becomes critical.
- ▶ Big Trouble
 - Distributed, or worse. Bigger than a terabyte.
 - We're hosed. Primary challenge is organizing the data set so that some question can be answered.
- ▶ Streaming
 - Probably various rates to consider here; a whole new set of animals

Ballistic Pacific Northwest National Laboratory U.S. Department of Energy



There is a lot of good work a lot of people could do that would get us not only to moderate data sets, but I think get us all going in a streaming data set sense.

This second recommendation is slippery. I don't think I have made a great case for it. If you think about the complexity that ought to be inherent in some of these larger data sets, and how much trouble we have communicating some basic ideas, I believe there is a lot of effort that we need to expend in that area, and it is going to take a lot of folks, in part, just because they are all the type that we are going to be communicating with, and in part, because the statistics community isn't going to have all of those answers.

MS. MARTINEZ: It is lunchtime. One question.

AUDIENCE: What type of clustering algorithms have you had the best luck with, and have you developed algorithms specifically to deal with streaming data?

MR. WHITNEY: I tend, just by default, to use a K-Means, and it works pretty good. It is simple, it is fast. We played with variants of it, a recursive partitioning version, with various reasons why you would recurse.

The version that I ended up writing for that network problem wasn't for streaming data. It was for data basically defined so you could pass through and repass through it and repass through it. So, we had an explicit data structure that represented that type of operation. I hope to use that as a basis for lots of other algorithms, though.

Leland Wilkinson, Chair of Session on Mining Commercial Streams of Data

Introduction by Session Chair

Transcript of Presentation



BIOSKETCH: Leland Wilkinson is an adjunct professor of statistics at Northwestern University and senior vice president of SPSS, Inc. His research interests include statistical graphics and statistical computing.

Dr. Wilkinson received his AB from Harvard College in 1966, his STB from Harvard University in 1969, and his PhD from Yale University in 1975. In addition to his statistics background, Dr. Wilkinson has also served as a lecturer, visiting scholar, and professor of psychology at Yale University, the Israel Institute of Applied Social Research, and the University of Illinois at Chicago.

One of Dr. Wilkinson's many accomplishments is the development of SYSTAT, a statistics and statistical graphics package that he designed in the late 1970s and then incorporated in 1983. An early feature of SYSTAT was MYSTAT, a free statistical package for students. SYSTAT and SPSS became the first statistical software companies to market full-featured Windows versions of their software. In 1995, SYSTAT was sold to SPSS, which was in turn sold to Cranes Software International in 2002.

TRANSCRIPT OF PRESENTATION

MR. WILKINSON: All right, this session is on mining commercial streams of data, with Lee Rhodes, Pedro Domingos and Andrew Moore. Only one of our speakers is from a company although the other two, as you know, are involved in developing procedures, high-dimensional searches, and mining and other areas that are highly relevant to what is done by businesses.

I just want to highlight the three major market shares of applications of streaming data analysis, and these are quite large. Monitoring and process control involves such applications as General Electric with its turbines worldwide. There are many, many turbines and, to shut down a turbine, can cost millions of dollars per day in their system. So, they need to maintain continuous multivariate data stream monitoring on those turbines, and they have real needs for display and alert and analysis capabilities.

E-commerce goes without saying. We all know pretty much where that lies. Many are putting e-commerce data and Web logs into databases, but Amazon and other companies are analyzing these in real-time.

Financial is another huge area for streaming data. I thought I would give you a quick illustration of how that gets used.

This is a JAVA application called Dancer that is based on the graphics algebra, and the data we are putting into it now, we happen to be offline, of course, but this data feed is simulating a natural stream coming in.

These are Microsoft stock trades, and these are coming in at roughly 5 to 10 per second. On the right, you see the list of trading houses, like Lehman Brothers, and so on. These trades, the symbol size is proportional to the volume of the trade. Up arrow is a buy, down arrow is a sell order, and then a cross trade is a rectangle. These traders want to be able to do things like alter time, back it up, and reverse it. Those of you who have seen the TiVo system for TV, video, know that these kinds of manipulations of time can be critical.

This application, by the way, is not claiming this as a visualization. It is actually doing the calculations as soon as the real-time feed comes in. Notice all the scaling is being done on the fly. You can speed up the series. If you speed this up fast enough, it is a time machine, but I won't go into that. I will show you just one more aspect of real-time graphics, and these are the kinds of graphics that you plug into what the rest of you guys do.

When you develop algorithms, you can plug them into graphic displays of this sort. This one simulates the way I buy stock. Actually, I don't buy stock for this reason. It is just a simple exponential forecast.

You can see the behavior. This is trading in Oracle and SBSS. This type of a forecast represents exactly what I do and probably some of you as well which is, as soon as it starts going up a little bit, buy.

What is being done here, the model is being computed in real-time. So, you get, in this kind of a system, anywhere from 10 updates a second to 10,000 data events per second, and 90 percent of the effort in developing software in this area is in the data handling. How do you buffer 10,000 events per second and then render in roughly frames per second using the graphic system? So, the rendering system is a lot simpler than the actual data handling system.

So, now we are going to see some presentations that will highlight how these systems work, and we will begin with Lee Rhodes from Hewlett-Packard, who will tell you about data collection on the Web.

Lee Rhodes

A Stream Processor for Extracting Usage Intelligence from High-Momentum Internet Data

Transcript of Presentation

Technical Paper

BIOSKETCH: Lee Rhodes is chief architect for analysis in Hewlett-Packard's Management Software Organization. During his career at HP he has led numerous research and development efforts in a broad range of technology areas, including fiber optics, integrated circuit design, high-performance graphics software and hardware, CPU architecture, massively parallel processing systems, multimedia and video streaming software, communications software systems, and two- and three-dimensional visualization software.

Since 1996, Mr. Rhodes has been heavily involved in the development of operational systems software for the communications network service provider industry. He has invented a suite of technologies in the area of real-time analysis software that enables Internet service providers to quickly extract key statistical information about subscribers' usage behavior that is critical to security, network operations, capacity planning, and business product planning. He assembled and managed the R&D team that developed this technology into a commercially successful product.

Mr. Rhodes' formal educational background includes a master's degree in electrical engineering from Stanford University where his emphasis was on integrated circuit design and solid-state physics. His undergraduate degree was in physics from the California State University at San Diego.

TRANSCRIPT OF PRESENTATION

MR. RHODES: This should go flawlessly because of our advance planning. First, I want to tell you how honored I am to be here, and I want to thank the organizers of this conference for putting this setting together. I particularly want to thank Lee Wilkinson for being a great mentor and friend and guiding me along the way.

I am going to talk about a stream processor that we have developed at HP that is currently on the market. We sell this product. Before you do any kind of measurements in terms of what we are talking about—I just want to be clear that we should be calibrated.

This is not science. This is engineering. Our role, my role, at HP is to develop advanced software. Our statistical sophistication is very low. I am learning and, with the help of Lee Wilkinson, I have learned an immense amount. I hated statistics when I was in college, but now, I am really excited about it. So, I am really having fun with it.

In isolation, much of the technology you will see here has been written about before in some form. Nonetheless, I think you will find it interesting for you.

The context of this technology is that we develop software for communications service providers. So, this software—and particularly Internet, although not exclusively Internet providers—those are our customers.

How we got started as a start-up within HP about five years ago was exclusively focused on the Internet segment, and particularly broadband Internet providers. We are finding that the technology we built is quite extensible to neighbor markets, particularly telephony, mobile, satellite and so forth.

Now, the network service providers, as I am sure you know, have some very serious challenges. The first one is making money. The second one is keeping it.

In terms of making money, Marketing 101 or Business 101 would tell you that you need to understand something about your customers. The real irony here is that few Internet service providers do any measurements at all about what their customers are doing. In fact, during the whole dotcom buildup, they were so focused on building infrastructures, that they didn't take the time, or invest in, the systems that would allow them to understand more about customer behavior.

That even goes for the ISPs that are part of big telephone companies. Of course, telephone companies have a long history of perusing the call detail records and understanding profiles of its customers.

There are some real challenges here, not only understanding your customers, but understanding what the differentiating services are. It is very competitive. What kind of services are going to make money for you.

Another irony is pricing this stuff. It is not simple. It is not simple now, and it will get even more complex, because of this illusion that bandwidth is free. That won't survive. It is not free.

So, there have to be some changes and, as I go through the talk a little bit later, I think you will see why pricing is such a challenge, particularly for broadband. Certainly, you want to keep your own subscribers are part of your network, but you are also concerned about use, fraud, theft, and other kinds of security breaches.

Now, when you go and talk to these service providers, they own the big networks. What you find is like in any big organizations. They have multiple departments and, of

course, the departments don't communicate very well. This is not a surprise. We have the same problem at HP.

Nonetheless, the sales people are interested in revenue. So, they are really interested in mediation systems which collect the data about the usage of other subscribers so that they can bill for it in some way. This is an emerging trend and will continue to be.

They are interested in not just bytes, but they are interested in what types of traffic it is, time of day. For instance, they want to be able to track gamers, say, to a local gaming host on the network, because their network bits are cheaper than peering agreements out on the open networks. So, understanding who the people are who are using games and so forth would be of interest to them.

Product development. These are the folks who come out with the new services. So, they need to have some sense of, well, is this going to make money or not, what is attractive.

Network operations needs understanding of utilization and performance on a day-by-day basis. They tend to be very focused on servers, on machines, on links, to make sure they are operating properly.

Product planning is often in a different department. These are the ones who are interested in future capacity, how can I forecast current behavior forward to understand what to buy and vend.

Realize that a lot of quality of service, if you can call it that, on the Internet, today is accomplished by over-provisioning. So, if I have bodacious amounts of bandwidth, nobody tends to notice. Of course, IP is particularly poor at quality of service, but there is work being done to do that.

So, the technology challenges for the service provider, there are many, but here are some of the few key ones.

They would like to capture the data that would service all these different needs once. They are expensive to capture usage data, and the tendency is, among vendors such as HP, is to go in and say, oh, great, we have this widget. We will just sample your key core routers with SNP queries and get all this valuable data for you.

Of course, every other vendor comes in and wants to do the same thing. So, they end up with 50 different devices querying all their routers and virtually bring the routers down.

Economic storage and management of the Internet usage data is a severe problem. Of course, they want the information right away and, of course, it has to scale.

So, I am talking about some of my back of the envelope kind of analysis of this problem of data storage and analysis challenges.

Starting with—this is what I call a cross over chart. What I did is very simplistic calculations saying Internet traffic, particularly at the edges, is still growing at about doubling about every, say, 12 months. At times it has been faster than that. Over the past several years, it seems to be pretty stable.

One of the interesting things is that the traffic in the core of the Internet is not increasing as fast as it is at the edges, and a lot of that has to do with private peering agreements and caching that is going on at the edge, which is kind of interesting.

The next thing I plotted was aerial density of disk drives. In the disk industry, this is one of their metrics, is how many millions of bits per square inch of magnetic surface

can they cram onto a disk. That has been doubling about in a range of 15 months. So, it is a little bit slower. Then Moore's law, which doubles about every 18 months.

So, the axes had no numbers on them. They don't need it. It doesn't matter where you originate these curves, you are going to have a cross over.

If this continues to grow at this rate, then at some point the—choose your measure. The traffic on the Internet is going to exceed some value. I think we can help with this one by better collection strategies and using statistics.

AUDIENCE: I have to admit, I am really confused here by comparing Internet traffic volumes to disk drive densities.

MR. RHODES: It is just a very simplistic assumption. It says that, if I am receiving traffic and I need to store information about that traffic that is proportional to the non-traffic, I have got to put it someplace.

AUDIENCE: What does it mean that they are equal?

MR. RHODES: I am just saying choose a value. Suppose you can store so many trillion or terabytes of data today. If the ability to store economically their data doesn't increase as fast as the traffic increases and the need to store it, you may have a problem.

AUDIENCE: So, where is the traffic coming from, if people can't store it?

MR. RHODES: That is on your own machines. Remember, the Internet is still growing. There are people joining.

Now, the other crossing is Moore's law, which says if the traffic continues to increase faster than Intel can produce CPUs that keep up with it, or Cisco can produce processors that keep up with it, you just have to add more horsepower.

AUDIENCE: Well, isn't the traffic consumed? If I am watching a video, I consume that traffic, I don't store it.

AUDIENCE: Some people might want to store it.

MR. RHODES: Okay, at the service provider, they are not storing the actual traffic. What they are interested in are the summary records, which are called usage data.

The usage data are summaries of flows. At least, that is very common in the service providers. It is a fraction of the actual traffic, but as a fraction, it stays about the same. So, as a service provider, the tendency—and this may seem strange—is to serve all of it. Those who have telecom backgrounds sometimes save their call detail records (CDRs) for seven years. Sometimes there are regulatory requirements.

Saving the Internet traffic, number of summary records for a session which you might have on the record, is far higher, orders of magnitude higher, than a single phone call. If you make a phone call, one record is produced. If you sit hitting links on the Internet, you are producing sometimes hundreds of sessions, as far as the way these sessions are recorded.

The second graph is also a back of the envelope calculation. This is based on some measurements that we have done, which is the storage required.

Now, presume that you wanted to store each of these just usage records. One of the factors that we have measured on broadband Internet is the number of what we call flows. These are micro flows, really, per second per subscriber in a broadband environment. It is around .3, and varies, depending on time of day from about .1 up to about .3.

Now, you multiply that through times the size of a storage record, and they don't want to store just the flow information, they usually also need to put information like the

subscriber ID and some other key information. You assume a couple hundred bytes per record. All of a sudden, you are talking about pedabytes or exabytes of storage, if you want to store it for any kind of period.

So, these represent different numbers of subscribers, different scales. The dark red one is about a million subscribers and, as a service provider we are working with today that saw this coming and realized that they had a problem.

The other one is also a back of the envelope calculation, time to process this stuff. Say you get it all into a database. You have got to scan it once. That can take a long time.

There, I just projected different database systems, depending on how many spindles and how sophisticated you want to get, in terms of how many records per second can you process, and how much money do you want to spend on it.

So, we are talking about years, sometimes, if you wanted to scan the whole thing. So, there is a problem here and it has to do with inventory, if you just have too much inventory of data. Handling it is a severe problem.

So, this is a somewhat tongue-in-cheek illustration, somewhat exaggerated to make a point, but a lot of our major customers are very used to having very big data warehouses for all their business data. Data warehouses are tremendous assets. As soon as you start trying to plug these into the kinds of volume we are talking about, it no longer makes that kind of sense.

What we have developed—this is just a short cut—is a sense of how can we capture information on the fly, and build not just a single model, but hundreds or thousands of small models of what is going on in the network.

Then, we have added the capability of essentially a real-time look-up, where the user here can, using a navigation scheme, can select what data they want to look at and then they look at, for instance, the distribution statistics of that intersection.

This is the product—I promise I am not trying to sell anything, but I just want to say this is the architecture of the product that is the foundation of this. It is called Internet manager. It is an agent based technology. These represent software agents. It is these three things together here, encapsulator, rule engine, and a distributed data store.

In a large installation, you can have scores to hundreds of these, and the whole idea is putting a lot of intelligence right up close to the source of this high speed streaming data.

We have different encapsulators. These are all plug-ins. The encapsulator is like a driver. It basically connects whatever the unique source, type, or record type or whatever that these various sources produce to internal format. Then, this is a rule engine, which I won't talk about. Basically, the flow is generally to the right, although this is somewhat simplistic, so it represents a kind of pipeline.

So, these rule engines process rules, and they scale in three dimensions. One is the horizontal parallelization, which you have with many agents. The second is the size of the machine you put these one. The third is you can defer certain rules downstream. So, you can spread your intelligence processing.

Now, a lot of times, and where we initially got started, was supplying basically data in database form, or file form, to various other business systems like rating, billing, reporting operations and so forth. That is how we got started.

Now, to give you an idea of what the data—here is an example of one format of hundreds that we read. This is a net flow, version five record format. You can see all the

different types of information that comes out. Basically, it is summary information of the headers that you have been hearing about in the previous talks.

Source destination addresses, source destination ports, bytes, packets, and a lot of very valuable information here.

It is of a flow. A flow is a group of packets that is matched to a source and destination IP address and a source port, sometimes even an added destination port, and sometimes even a source port. So, it is really nailed down to the particular transaction that is going on.

So, what do we do with this? Each of our engines, each one of them, can pull in records from anywhere from around 50,000 to 100,000 per second. The first task is to normalize these, collect them and normalize them. The second task is the normalization and I like to think of as a vector, which was also spoken of earlier.

This is a set of arbitrary attributes. Think of them as columns in a database, but it comes in as a single record and actually can be variable in the number of attributes, and dynamic.

Now, once these come in, and we can have multiple streams coming in, usually we know quite a bit about these streams. We might have a stream coming in from an authentication service like a DHCP or combination DHCP, sometimes radius, sometimes DNS, that basically authenticates a user.

So, the service provider knows it is a legitimate describer, as well as the usage information coming from the router itself.

What we call them is normalized metered events. It is sort of the most atomic information about usage. So, these entities come in just like a record, and they are processed in this rule change, and a stream processing engine can't have loops. So, no four statements and stuff like that. We basically can't afford it.

It travels down and you can have F&L-type statements. The other interesting thing is, we have a statement where each of these, as the data is processing through, it looks at each of the field based on what rule it is—and this is all configurable, what rule you put in, about several hundred.

There is an association with a data tree. One of the things that this data tree can be used is in sorting. As the ME travels through, decision are made, there is a natural selection going on based on a certain field. Then we can do simple summing, for instance.

So, summing on a variable or even a group of variables is very straightforward, doing very much the joint something that was spoken about earlier. This all occurs in real-time.

The other use of this data tree, we call it—and it doesn't have to be just a tree, it can be a number of different points—is each one of these triangles is a structure that can have an arbitrary container, and we can put data in it.

So, one of the ways that we do stream correlation in real-time is that we effectively have like a switch, where we can select information coming from what we call a session correlation source.

It will load information into the tree that is used for matching, and then virtually all you have to do is now, as the new entities come through, they correlate dynamically to information that you want. For instance, it could be the IP address to a subscriber, or you could do all different kinds of correlation.

Now, in any one engine you could—so, I am using a symbolic representation of what you just saw, is this little triangle of a tree here, and you can have multiple ones.

So, we can do fan out. So, you can have a single source because the same data needs to go to different applications and needs to be processed by different sets of rules. So, you can parallel them, going to different applications, or you can put them into sort of sequential themes for more sophisticated rule processing.

So, the work that I have been doing has developed what I call capture models. So, as this data is flying by, I would like to collect more than just a sum. In fact, I would like to capture distributions of these variables or other kinds of characteristics. I think there are lots of things that you can do—Jacobbeans, I haven't seen the need for that—but there is the opportunity.

A capture model can have child models associated with it, but one of the rules of the capture model is that the NME that goes in left goes out of the right, because you can have a series of these in a row. So, you can have multiple of these capture models plugged together.

I tend to look at this like a matrix. Inside any of these capture models you have a--inside there is a matrix where you have a number of different variables that you can track. If you are doing binning, then the other axis is the bins. So, now you can put these, instead of doing just simple summing, now you can do sorting of your data, and it feeds right into this capture model.

You can put them in layers and do sequential summing. So, you create all these little matrices, and they are not very big, a few kilobytes, the largest eight to ten kilobytes. So, you can have thousands of them.

Now, the end-to-end architecture looks something like this, where you may have some free staging, for instance, some basic correlation going on. Then you put it directly into the models. That is one thing our customers are doing, or you can have these models directly on the raw data.

So, you can be binning of it and making decisions as the data is flying by. What we do, then, is we store just the models. Of course, the nice thing about these capture models is that they don't really grow with volume. The number of them is proportional to the size of your business problem that you are trying to deal with. Then, on the right here, you have the clients.

This is an example—it is not a great example, but it is one example of a distribution that we collected. I don't have a good example of truly real-time, but this kind of data can be collected in real-time. It represents the usage of subscribers over a 30-day period. This thing is just constantly updating as the data is flying by.

Red represents the actual number of subscribers and the red axis is the amount of their usage. Now, this is a broadband Internet. So, you will see, I have a subscriber out here with 23 gigabytes of usage for that period, all the way down to tens or hundreds of bytes. So, there is a huge dynamic range.

If you think about it, like electric utilities or other types of usage services you might have, very few of them have this kind of wide, dynamic range. Now, I fitted this, and it fitted pretty nicely to a log normal. Plotting this on a linear axis doesn't make a lot of sense.

In fact, what we do in the distribution models is do logarithmic binning. This data fits that very, very nicely. It is very probable in terms of binning.

Now I can see up to 90 percent of my subscribers now. There are two plots here. This is the subscribers at a particular usage, and this is the traffic that they create. One of the things it took me a while to figure out is why this right-hand side of this is so noisy. Notice it is jumping around quite a bit.

Part of that is not just noise. Part of that is the fact that subscribers only come in unit quantities. So, a subscriber at 10 gigabytes of usage also creates big deltas out at the right-hand edge of this.

The other reason is the actual binning. So, they may not fall in a particular bin. So, you will see some oscillation and it is actually easier to see the oscillation between bins on this graph.

I did some—after reading Bill Cleveland's book, I tried the QQ plot, but I did a reverse QQ plot because I have already got bytes on my X axis, and these are the standard normal quantiles on the left. What is interesting is that the fit on this is very, very good, over about four orders of magnitude.

I didn't bother doing any fancier fitting at the top or the bottom of this. The users at the bottom are using more than the models would predict, of course, and at the high end, they are using less. I find that, in looking at about a dozen of these from different sites, that the top ones slop around a bit.

This is an asymmetry plot, which you read a lot about in the press. Actually, here, it is quantified. You can look at, for instance, that 20 percent of the subscribers, the top 20, are using 80 percent of all the traffic. That happens to be the way this distribution fell out. What they don't talk about is, 80 percent of the users are only using 20 percent, which is the obverse of that, which means they have got a real severe pricing and fairness problem, but I won't go into that.

Some extensions of this basic technology we are doing now, and actually deploying with one of our customers, is using this kind of technique for security, abuse, fraud and theft. We are doing a lot of learning in how to do this, but I am convinced that, once you have a distribution of a variable and you can normalize it, say, over some longer period of time for the standard population, then you can very quickly see changes in that distribution very quickly.

If all of a sudden something pops up, like a fan in, fan out, which is the number of destination IP addresses, or destination ports all of a sudden explodes, then you know someone is scanning ports.

These terms mean different things, but in the service provider industry, fraud and theft are different. Theft is when they are losing money. Fraud is only when someone is using your account, because you are still paying. Then, abuse is basically violation of the end user agreement that you signed when you signed up with the service provider.

Now, the other thing I am working on is dynamic model configurations, where you can dynamically refocus a model, a collection model, on different variables, different thresholds, what algorithms are actually used and so forth, do that dynamically.

That allows you to do what I call drill forward. So, instead of having to drill down always to the history, you see something anomalous. It is likely to come back. This is not like looking for subatomic particles.

So, if someone is misbehaving, more likely it will occur again, and you want to zoom in on that and collect more data, and more detailed data. So, that is what I call drill forward.

Now, some back burner stuff, something that is interesting—I haven't found a real business application for this—now that I have got this multidimensional hypercube, so to speak, of all these collections of models, and each one of these circles can be different kinds of models, it sort of represents, or can represent, the business of what the service provider's customers are doing.

I thought it would be kind of interesting to take that and do a reverse transform of it, and then create a random stream of usage events that looks exactly like what your subscribers would look like. It is random, but it has exactly the same distribution behavior as the stuff coming in, and it would be multiple distributions. I figured out the algorithms for this, but I haven't found anybody that needs it yet.

So, some of the paradigm shifts that I find are challenging when I talk to service providers is really, the knee jerk reaction is, oh, I want to store everything, and it is just prohibitively expensive.

I find that I have to be a business consultant and not just a technologist when talking to people. What is the business you are in, do you really want to keep this stuff for this long. This belief that I have that you have to analyze this high-volume data as a stream, and not trying to store it first, do it on line, in the stream, can reduce it first.

Then, consider drilling forward rather than always wanting to drill back into the history. Drill forward for more detailed analysis.

We are very interested in collaboration with research laboratories. We have research licenses for this software with qualified laboratories that would like to take advantage of this kind of a real engine. Some of the things that I think would be very interesting is capture model development for us in other kinds of purposes that I will never even think of.

Certainly, we need more robust statistical approaches, and visualization work, how to visualize this stuff.

The last thing I want to bring up, this is a client. It is not hooked to the network, so you can't see the real-time graphs changing. You can see, this is actual data. You can see, for example, this is a broadband supplier. If I looked at data, say, from one hour, it is pretty noisy and, as you increase the time, in about 30 days, it turns into a real nice shape.

This is what I would be doing here if I were hooked to the network, is navigating along the different axes of that hypercube, where I am choosing different service plans or user pricing plans and so forth, that the service provider has chosen.

The last thing here, I actually took advantage of the fact that I have a distribution of usage for a population and, if I know their pricing function, I can compute the value of the traffic, and do that virtually instantaneously.

That is what this does. I am not going to demonstrate it, but basically I can help the product planners for the service provider figure out what the volume of the traffic is, without having to go back through millions and millions of records and basically try to model their whole subscriber base. Basically, you have it all here. Thank you very much.

MR. WILKINSON: While Pedro Domingos is setting up, we have time for a question.

MR. CLEVELAND: Lee, I just would ask if you could give some idea of where you have set this up so far.

MR. RHODES: In terms of commercial deployments? We have a number of

pilots.

MR. CLEVELAND: Are most of your experiences at the edges with 80SL and cable?

MR. RHODES: Yes, most of these are the same with edges. So, it is 80SL cable and we did one backbone, media backbone service provider—well, they dealt with commercial clients. So, they had a few thousands, but very large pipes.

I would say most of our—in fact, our current deployment that we are working on is a very large sized service provider in Canada.

A STREAM PROCESSOR FOR EXTRACTING USAGE INTELLIGENCE FROM HIGH-MOMENTUM INTERNET DATA

Lee RHODES

The data streams of the Internet are quite large and present significant challenges to those wishing to analyze these streams on a continuous basis. Opportunities for analysis for a Network Service Provider include understanding subscriber usage patterns for developing new services, network demand flows for network operations and capacity planning functions, and early detection of network security breaches. The conventional analysis paradigm of *store first, then analyze later* has significant cost and latency issues when analyzing these high-momentum streams. This article presents a deployed architecture for a general purpose stream processor that includes dynamically configurable Capture Models that can be tailored for compact collection of statistics of the stream in real time. The highly configurable flow processing model is presented with numerous examples of how multiple streams can be merged and split based on the requirements at hand.

Key Words: DNA; IUM; Real-time statistics; Statistical pre-processing.

1. INTRODUCTION

In 1997 a small R&D group was formed inside of Hewlett-Packard's Telecommunications Business Unit to develop Internet usage management software for Network Service Providers (NSPs). The services offered by these NSPs ranged from Internet backbone to Internet access. The range of access services included residential and commercial broadband (cable and xDSL), dial-up, mobile data, as well as numerous flavors of hosting and application services. Early on our focus was the processing of usage data records (e.g., NetFlow® or sFlow®) produced by Internet routers. However, it quickly broadened to include convergent voice Call Detail Records (CDRs) as well as the ability to collect and process data from a very broad range of sources such as log files, databases, and other protocols. The diverse technological histories (and biases) of the different segments of the communications industry created for us interesting challenges in creating a software architecture that was

Lee Rhodes is Chief Scientist/Architect, IUM/DNA, Hewlett-Packard Co. (E-mail: lee.rhodes@hp.com).

©2003 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America
Journal of Computational and Graphical Statistics, Volume 12, Number 4, Pages 927–944 DOI: 10.1198/1061860032706

robust, quickly adaptable, and scalable in order to meet the broad range of requirements for fast data collection and analysis. It was in the process of meeting this challenge that the concept of a general purpose stream processor emerged.

There are two related suites of technologies discussed in this article. Internet Usage Manager (IUM) (<http://openview.hp.com/products/ium/index.html>) is the platform technology that provides the basic stream processing capability. Dynamic Netvalue Analyzer (DNA) (<http://openview.hp.com/products/dna/index.html>) technology extends the IUM platform to enable stream statistical analysis capabilities.

2. BUSINESS CHALLENGES FOR THE NSPs

With the recent spectacular collapse of some major NSP players, the challenge within the industry of creating a profitable return on their sizable infrastructure investments could not be more visible (McGarty 2002; Sidak 2003). During the technology buildup of the late 1990s, many of the NSPs invested heavily in rapid expansion of network capacity at the expense of infrastructure for metering and analyzing subscriber behavior. In the frenzy of the technology hype and market buildup with cheap capital readily available it was easy to believe that bandwidth was free, should be free, or would become free. Why bother to measure usage? And given the wide publicity of the ever expanding bandwidth of optical fiber, a superficial examination of the issues could lead one to that conclusion. Unfortunately, the real bandwidth limitations are not in the Internet backbone, but in the access networks (the “last mile”), where the upgrade costs are high. It is ironic that even those ISPs that were operating units of larger, well-established telecommunications companies, which had developed extensive telephone subscriber data collection and analysis capability over the past 20 years, did not make substantive investments in measuring and understanding subscriber usage behavior. This is rapidly changing today.

The business motivations for understanding usage behavior on the revenue side include various usage-based charging models for billing and subscriber segmentation for marketing and product planning. Additionally, because IP-based services are still young, having a statistical basis for trial-testing potential pricing models for these services is certainly better than having no data at all. The motivations on the expense and investments side are equally strong. Without the ability to measure or analyze the impact of various usage behaviors, whether they are legitimate or not, the tasks of network management, security, performance, and capacity planning reduce to a guessing game.

Some of our early R&D work focused on the usage mediation and tracking in support of billing for Telstra's BigPond™ Cable and DSL Internet services (<http://www.bigpond.com>). In Australia, as was the case in many parts of the world outside the U.S., the cost of international transit fees, based on bandwidth usage, represented a significant variable cost that was constantly increasing on a per subscriber basis but not transferable to subscribers who were billed, at that time, only on a flat, all-you-can-use pricing model. Data collected at Telstra from nine different DSL and cable broadband Internet services revealed that the distribution of subscriber usage can be fitted very closely by a lognormal (with a shape

factor of ~0.7). The top 20% of subscribers generate ~80% of all traffic. The top 5% of the subscribers generate ~50% of all traffic. Another way to look at this is that 95% of the subscribers can end up subsidizing the top 5%. Simple flat rate pricing plans for unlimited usage broadband services will naturally force the NSP to charge high monthly fees, which naturally restricts the economic accessibility and uptake of the broadband services.

3. SOURCES AND TYPES OF DATA

Data sources can also be grouped by different device types, which vary considerably by the specific application. Device examples include network equipment (routers, switches, and gateways), application servers (Web, e-mail, game servers), general purpose computers, network probes, and database management systems (DBMS). We have found it useful to classify the types of data sources into usage, session, and reference categories based on how the data needs to be processed. For many real-time sources of data we have defined the term metered event (ME) as an atomic data structure that encapsulates information about or relevant to usage of a service at a specific point in time or within a specific window in time.

3.1 USAGE MEs

Usage MEs contain metadata, which are data about data. At the lowest level of collection usage MEs are often grouped into small records where each of the fields contain basic statistics about an atomic usage event such as a single phone call or an Internet data transfer. Typical fields that are often found in usage MEs are source, destination, usage volume, start time, and end time. Depending on the context and applications involved, a usage ME may also include fields such as service type, quality of service level, termination conditions or error codes.

In telephony a common usage ME is the Call Detail Record (CDR) that is produced by the originating switch and records key information about the calling number (source), the called number (destination), and the length of the call in minutes (usage) among other fields. It is from CDRs that telephone companies construct their billing records and perform extensive analysis of subscriber behavior.

In the Internet context a single ME might capture the usage details of a large file download or a small GIF image of a button. Because Web pages can be containers for references to many other web pages or objects, clicking on a few pages of a complex Web site can result in hundreds of ME events. Considering this “session” of browsing a Web site as roughly comparable to a telephone call, it is easy to see that the number of MEs generated will be considerably higher than the single CDR produced from a phone call.

3.2 SESSION MEs

Session MEs provide accounting and state information about the user originating a

particular stream of traffic at a point in time. Internet session MEs are dynamic and usually create an association between an IP address (or cookie) and a responsible account ID or subscriber ID. Depending on the service definition, a session ME may also provide information about session state, for example, logged on or off, authorization level, location, and so on. In telephony, the usage information and session information are often combined into the same record. In the Internet, however, these data are acquired from different sources and must be time-correlated together in near-real time. Sources for session data for Internet services include DHCP, DNS, and DDNS, as well as authentication, authorization, and accounting (AAA) services such as those provided by RADIUS.

3.3 REFERENCE DATA

Reference data, defined by the NSP, is merged with the real-time streams of incoming MEs in order to facilitate additional downstream processing and analysis. Network operational examples include network topological, physical, or routing information (e.g., Autonomous System Numbers). Business examples include subscriber segmentation and classification information useful by product planning. Security examples include thresholds or patterns useful for identifying abuse, fraud, hostile, or attack traffic.

The actual collection and interpretation of MEs from real devices is complex because of its diversity and the legacy of old devices still in use. In the future, this arcane and time-consuming development process could be facilitated by the adoption of abstract event and services models such as those being developed by Jeff Meyer (see <http://www.circumference.org>) for the IPDR (<http://www.ipdr.org>). The proposed model has a simple three-layer structure. The top layer is the data model, which defines the service or data represented in the ME. This is preferably a machine-readable file (e.g., W3C XML-Schema), however, for legacy reasons a human-readable document can do the job. The middle layer is the data encoding model, which defines how the data are represented as a serialized stream of bits. The bottom layer is the transport model, which defines how to get the data from point A to point B. The transport model is often a hierarchy of protocol layers, but includes concepts of file-based exchange, streaming data, and other transports.

4. DATA STREAMS AND RIVERS

The data streams of the Internet are huge. Even though usage MEs will be a couple of orders of magnitude less, the volume of usage events can still present significant design challenges for general purpose collection systems. We have begun to characterize these ME flow volumes from data that have been shared with us from several NSPs. We have measured ME streaming rates of 0.2 to 0.5 MEs/subscriber/sec for Cisco IOS® NetFlow (<http://www.cisco.com/go/netflow>) enabled routers. For a moderate-sized network supporting 1 million subscribers, 0.3 ME/sub/sec represents an input rate of 300K MEs/sec. NetFlow version 5 uses UDP packets of 30 ME records of 48 bytes each plus a single header of 24 bytes. At 50 bytes average per ME this is a line speed of about 120Mb/s. But stored into a database

it can represent approximately 4 Terabytes per day (allowing 3X for the inefficiencies of relational DB storage). This is assuming, of course, that you are willing to pay for a database that can handle a continuous input stream of 300K records per second and perform useful analysis work at the same time.

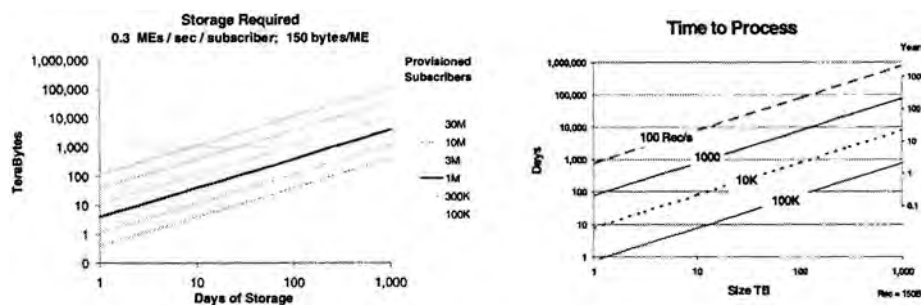


Figure 1. Simplistic approaches of store first then analyze later when applied to high-momentum streams can lead to very large storage requirements (high infrastructure costs) and long analysis latencies.

How long do you wish to keep these records? The left side of Figure 1 is a simple back-of-the-envelope calculation (BOTEC) that computes the database storage requirements as you scale up in number of subscribers and in length of storage time. The heavier line represents the 1 million subscriber case above. Three months of storage at 3 million subscribers is already a petabyte!

The other consequence of these large datasets is the time required to process them. The chart on the right above is another BOTEC that illustrates the time it would take to do a single pass of a dataset as a function of the dataset size in terabytes and the record processing speed of the database. Another way to think about this is to consider the ratio of the continuous input record rate to the record processing rate once the data have been stored into the database. If the query requires complex processing while it is completing its scan it may not be much faster than the input data rate. At a ratio of 1:1 it will take as long to perform a single pass on the data as it took to capture it. This could be unacceptably long to obtain some of the key results hidden in the data.

For high-momentum streams it is common to have hard-coded pre-processors that perform either simple aggregation or sampling to reduce the data down to a rate that can be absorbed by conventional databases. However, the use of either of these techniques involve making major a priori assumptions about the nature of the data and what potential queries will be made on the reduced data.

Assuming, for a moment, that the NSP can afford the DBMS infrastructure required to capture all of this raw data, advanced data reduction techniques have been developed for obtaining quick approximate answers from large databases. The paper edited by Hellerstein (Hellerstein et al. 1997) provides an excellent survey of these techniques, which include singular value decomposition, wavelet, regression, log-linear, clustering, index tree, and sampling. But the choice of these techniques also heavily depends on the nature of the data

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

and queries anticipated. Unfortunately, Internet data can have a high number of dimensions, the variables can be highly skewed in both frequency and value, and some of the events or patterns of high interest can be very rare (e.g., a slow address scan by a potential intruder). To make matters worse, with the constant evolution of viruses and worms, the priority of what is important to examine is constantly changing. These complicating factors make the selection of data reduction techniques somewhat of an art form.

Broadband service providers find themselves between a rock and a hard place. They need much richer information about their subscriber usage behavior with strong business rationale on both the revenue and the cost side. The rock is the very high cost of building and managing these large datasets. The hard place is that most general purpose data analysis tools presume that the data to be analyzed exists or will exist in a database. No database, no analysis.

What if you could extract some meaningful information about a data stream before you had to aggregate and commit it to hard storage? This idea, by itself, is not exactly new. But what is needed in a number of these high-momentum, complex data stream situations is a high-performance, flexible, and adaptive stream processing and analysis platform as a *pre-processor* to long-term storage and other conventional analysis systems. In this context, high performance means the ability to collect and process data at speeds much faster (>10X) than most common database systems; flexible implies a modular architecture that can be readily configured with new or specialized components as needs evolve; adaptive implies that certain key components can change their internal logic or rules on-the-fly. These changes could be as a result of a change in the input stream, or a detected change in the reference data from the environment, or from an analyst's console. Starting in 2000 we set out to build a platform with these goals in mind. The remainder of this article discusses the progress we have made.

5. IUM HIGH-LEVEL ARCHITECTURE

Figure 2 is a high-level view of the IUM architecture. Streams of data flow left to right. The purple boxes on the left represent different sources of raw data within a service provider's network infrastructure. The blue boxes on the right represent the target business applications or processes that require distinctly different algorithms or rule sets applied to the streams of data. The gold triad of a sphere, rectangular prism, and a cylinder represent a single instance of an IUM server software agent that we call a Collector. Each Collector is capable of merging multiple streams of input data and producing multiple output streams, each of which can be processed by a different set of rules.

The basic unit of scalability is the Collector. The first dimension of scaling is horizontal (actually front to back in the graphic) in that different input streams can be processed in parallel by different Collectors on the left. The second dimension of scale can be achieved through the processing speed of the hardware hosts. The third dimension of scale can be achieved by using pipelining techniques that partition the overall processing task for the various target applications into smaller sequential tasks that can execute in parallel. The

IUM platform has been implemented in Java, which enables multiplatform operation. The architecture has been designed with high modularity and configurability from the start. Upon start-up each of the Collectors obtains its own configuration from a central configuration server and then builds itself with the proper components required.

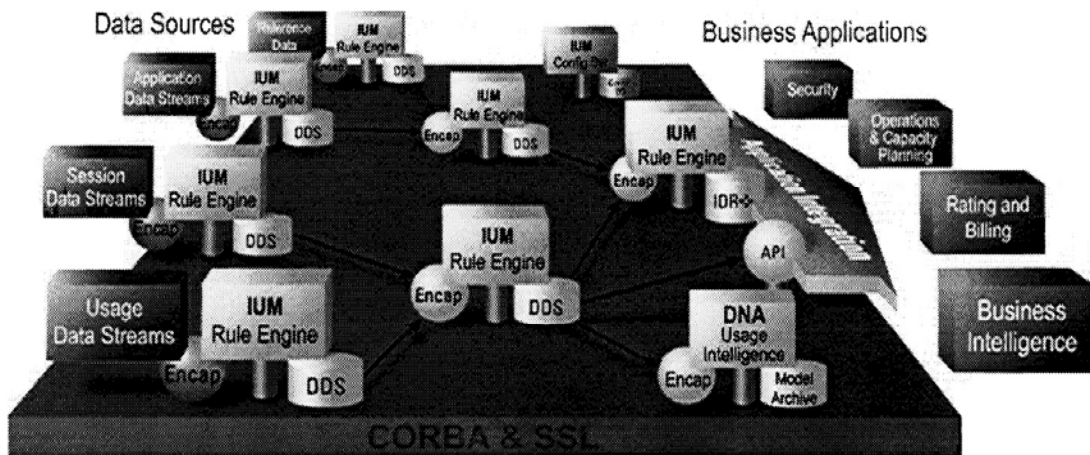


Figure 2. HP's Internet Usage Manager (IUM) enhanced with Dynamic Network Analysis is a distributed agent architecture.

6. STREAM COLLECTION AND NORMALIZATION

The input streams are captured from the source devices by plug-in Encapsulator (Figure 3) components represented in two different colors. The gradient shaded purple to gold ones are configured to interpret the data from specific source device types. The gold encapsulators are configured to read normalized data.

Over the past few years we have developed many preconfigured encapsulators for a wide range of devices mentioned above as well as different collection modes that include real-time streams, (e.g., NetFlow, sFlow, DDNS), polled data (e.g., SNMP), files and directories, and databases (via JDBC).

Raw data streams are captured by an Encapsulator...

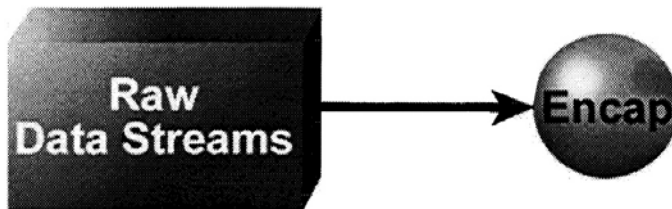


Figure 3. Encapsulation plug-ins are tailored to collect different types of input streams.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

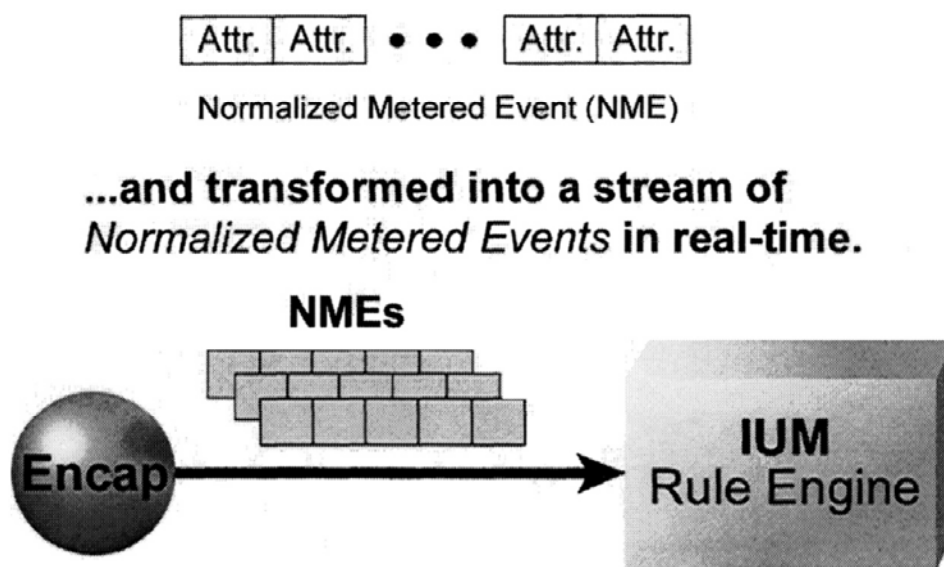


Figure 4. The events of the stream are converted into Normalized Metered Events and passed directly to a rule engine.

Once collected, the data of an ME are normalized into a common data structure called a Normalized Metered Event (NME, see Figure 4), which is an array of attributes that contain different data types similar to a DB record. The only attribute that is required is an end-time stamp. All the other attributes can be configured to suit the processing needs of the application. Unlike DB record schemas the number and type of attributes can change dynamically as it undergoes processing. In this stream processing context new attributes can be dynamically “adorned” to the NME and be used as intermediate variables, which travel along with the NME in the stream, and then disposed of when no longer needed. (The word “travel” is only a metaphor. The NME object doesn’t actually move; its position in the stream is tracked by passing a small reference or “pointer” to the NME object from rule to rule.)

7. STREAM RULE PROCESSING

Once normalized, the NMEs move directly into a Rule Engine, which has been specifically designed for merging, processing, and splitting streams.

The input streams can be independent or related (such as a usage stream and a session stream). In the rule engine (Figure 5) there can be multiple rule chains that operate on the input streams. It is possible to have a single stream processed by multiple rule chains each producing distinctly different output streams, or multiple similar streams processed by a single rule chain, or combinations of the above.

There are several forms that the output streams can take: (1) During processing of an NME, attributes within the NME are added, modified, or deleted and the result NME is forwarded immediately to a downstream Collector for further processing. (2) A cyclic aggregation time interval is configured into the Collector and rule chains are configured to

produce aggregates of particular attributes. At the end of the aggregation time interval the aggregates are flushed either to a persistent store for recovery operations, or the aggregates can be fed directly to a following Collector for further processing (or both). (3) Certain rules allow a real time query of current results. This is particularly valuable to gain insight into ongoing statistics. For example, examining the current empirical distribution of a variable. This kind of functionality is enabled with DNA discussed later on.

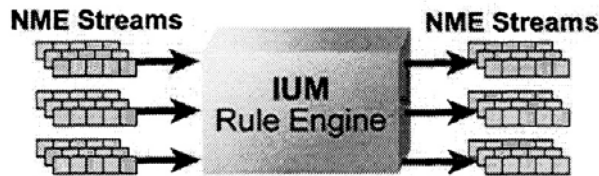


Figure 5. The rule engine can be viewed as an in-line agent that performs the work of merging, analyzing, and splitting streams. The output streams of a rule engine use the same NME structure, which enables the creation of a fabric of chained stream processors.

8. RULE CHAINS AND ASSOCIATED DATA STRUCTURES

A rule chain is sequence of configured operations on the flowing NMEs of an input stream. Example rules include standard three-address arithmetic and logical operators of the form $op(a1, a2, a3)$ where the first two parameters represent NME attributes as operands, and the third attribute is the result. Flow control rules include conditionals (e.g., if then else) that can change the path of the rules based on the NME attributes, but no looping rules—not a good idea in a stream! In addition we have developed a rich set of lookup rules, filter rules, adornment rules, and other special rules for more complex operations. The rules are written in Java and there is a developer's kit that enables users to create their own rules. However, the current rule library is extensive with more than 100 rules, which is satisfactory for most applications.

The right side of Figure 6 illustrates a simple rule chain on the left with a single if then else rule followed by some sequential operations.

Although the processing logic applied to an NME as it flows through is contained in the rule, there is also a reference to a data structure that travels with the NME and can also be changed by the action of the rule logic. This data structure can hold state information that can interact with subsequent NMEs traveling through the Rule Engine and can be flushed to a datastore at regular intervals. This parallel data structure is usually organized as a tree shown on the right.

Hash rules are an example of the special rules that operate on the references to the data tree. Each node of a level of the tree associated with a Hash rule can contain a hash table (Cormen, Leiserson, and Rivest 1999; Knuth 1973) where the entries contain a key and a pointer to a child node. A hash rule is configured to use an attribute of the NME as a key and then either chooses the successor data node based on the result of the hash if it exists, or create one if it does not exist. Each data node can be an n-way branch (a binary tree

structure is used in the diagram for graphical simplicity). New nodes of the tree get created as new values appear in the data. If the last rule in the chain is configured as an aggregation rule, and the Collector is configured to flush the leaf nodes flushed at periodic intervals, the resulting dataset would be the same as an SQL aggregating group-by operation except the grouping (essentially routing) occurs as the NMEs flow, not as a batch operation. The following table contrasts the stream processor rules that produce the same operation as the batch SQL statement on the right.

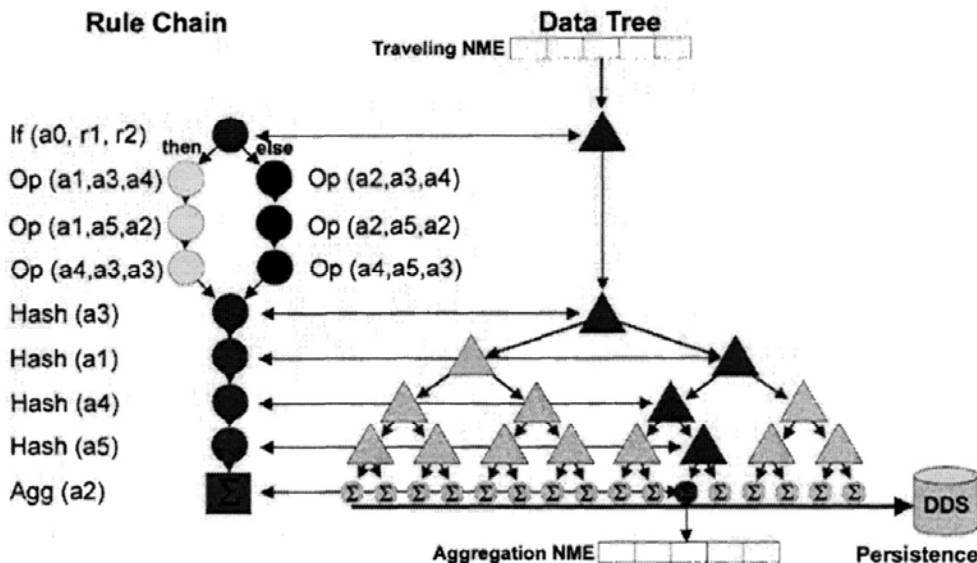


Figure 6. A simplified view of how rule chains can interact with dynamic data structures. A single rule engine can have multiple of these rule chains in various series and parallel configurations.

| Stream Processor Rules | Batch SQL Statement |
|------------------------------------|---------------------------------|
| (in a specific rule chain context) | Select SrcIP, DstIP, sum(usage) |
| Hash SrcIP, DstIP; | From <table> |
| Aggregate sum(usage); | Group By SrcIP, DstIP |

Because there can be multiple rule chains, as explained earlier, the specific rule chain where the hash rules appear is analogous to the “From <table>” clause in SQL. However, in a stream processing context we have the opportunity to perform operations that would be much more cumbersome in an SQL environment. This will be further discussed in Section 9.3

An n -level hash in this structure is analogous to a hyper-cube in the sense that the leaf nodes of the tree can be mapped to coordinates of an n -dimensional hyper-cube. However, there are differences. One is that the tree structure can be sparse in that nodes are created only when the combination of data actually exists in the input stream. Another difference is that the cell of a hyper-cube usually contains only one value. In this data structure the nodes are containers that can contain their own complex data structures internally.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

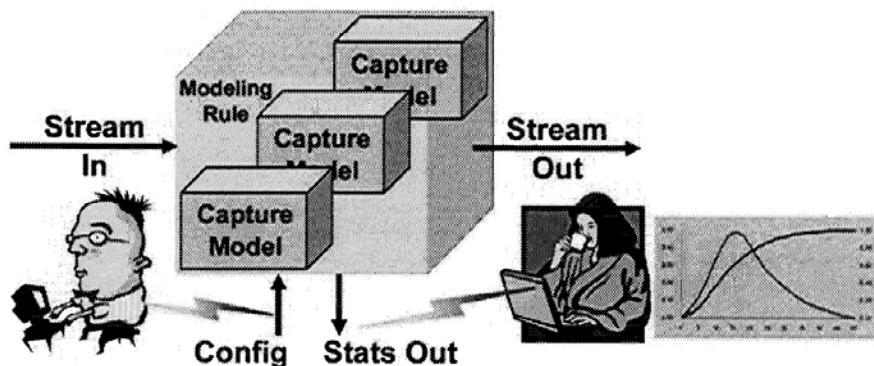


Figure 7. Capture Models are sophisticated rules that can perform complex, binning, filtering, and associative processing. These capture models can be dynamically configured to examine different statistical properties of the incoming stream.

The tree structure can be also used to create a result stream that is an associative merge of two different input streams. Suppose we have two streams A and B. For stream A we mark each NME so that instead of traveling through the entire tree an A-NME travels to the coordinate node of the tree specified by A's hash attribute values and drops off its associative attributes values at that node, which are stored there until they are replaced with more current data for that coordinate, again from Stream A. Stream B is processed as discussed earlier. As a B-NME travels through the tree it is routed to the coordinate node specified by B's hash attribute values. As it passes by, the B-NME picks up the associative data, which was previously dropped by the A-NME. In this way, a multidimensional set of associations between streams can be performed in real-time.

Other forms of associations can be performed by look-up rules, which are designed to perform fast specialized lookup algorithms for associations with reference data. An example of this is finding the longest qualified prefix of an IP address given a reference routing table.

9. STATISTICS FROM STREAMS

DNA builds on the platform discussed above and extends the stream processing capabilities in several ways.

9.1 CAPTURE MODELS

As the stream of NMEs pass through a node in the tree above it is possible to collect richer statistics as well (Figure 7). The motivations are several, but a capture model of a few KB can extract selected characteristics of a large stream very economically.

Capture Models (or just Models) are similar to Rules. Models are contained in a special Modeling Rule that acts as a manager and container for multiple models. When a Modeling Rule is inserted into a rule chain it will spawn capture models into the associated data nodes of the tree as they are created. Consider a node of the hash tree as a representation of the intersection of a set of business coordinates such as *customer*, *service*, and *geography*. Each

node can contain multiple Capture Models, which can collect different views of the data passing by. One Capture Model might be an adaptive histogram on one variable, another could be a TopN Model of a different variable from the stream.

One way to think of a Capture Model is that its input is a stream of vectors (NMEs) and its output can be a matrix of values defined by the Capture Model:

$$\text{Input NME} = \{t_s, t_e, a_3, \dots, a_n\},$$

$$\text{Result data: Capture Model } \mathbf{M} = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,m} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & \cdots & v_{n,m} \end{pmatrix},$$

$$v_{i,j} = \text{model internal vector variable, } 1 \leq j \leq m,$$

$$v_{i,\bullet} = \text{model internal vector variable element, } 1 \leq i \leq n.$$

Capture models can be configured with an integration interval (minutes to days) that defines the amount of time that statistics are collected. At the end of the integration interval, the result matrix is usually flushed to a persistent store or to another downstream rule engine.

There is no fundamental restriction on how a capture model is designed as the output can be any data structure that can be contained in a Java Object. For example, creating a correlation matrix model would be relatively straightforward. Defining conventions, like the matrix form above, has allowed us to create additional functionality such as model aggregation mentioned in the following.

The most common capture models include log distributions, linear distributions, TopN, History (time series), and other specialty models for security and capacity planning flow analysis.

As an example, the distribution capture model performs dynamic binning on the values that fly by for a configured attribute. For improved accuracy, particularly for rare events, the model defines two vector variables, sum and hits, both of which are dimensioned by the number of bins. The order of the above matrix becomes $n \times 2$, where n represents the current number of bins. The bins need not be contiguous and are only created based on actual data values that appear in the stream.

The result of the distribution capture model, when queried, is again an NME. The first several attributes define the coordinates of the model and then a single object attribute that is a compact form of the empirical distribution of the variable. This result NME can be output either at flush time of the aggregation tree or obtained by a real time query from the DNA client application.

9.2 CAPTURE MODEL AGGREGATION

A large stream with a lot of variables can create a lot of models based on how you choose to configure the DNA Collector. A single distribution model consumes about 2KB with 100 bins. To monitor bandwidth distribution characteristics of stream flows at 10,000 points in your network amounts to only 20MB of memory, but how does one examine 10,000 distributions, (or 100,000 distributions)? This leads to the concept of model aggregation.

Returning to the navigation discussion, the tree structure could be leveraged again and produce one type of model aggregation where capture models would reside at each of the interior nodes of the tree in addition to the leaf nodes. These interior models create a hierarchy of models where an interior model in an upper level of the tree represents the aggregate statistics of all the child nodes below it. Order is important, however. Using the "*" to represent the aggregation of all the coordinate values for a dimension you could create navigation coordinates like (a1.*.*), (a1.a2.*.*), or (a1.a2.a3.*), where the coordinates are in top-down order. This does not allow aggregations of the form (*, a2, a3, a4), which diminishes this strategy's usefulness. Instead we have provided an internal query capability within the DNA Collector server that can traverse the in-memory tree and collect data from nodes with an arbitrary query of the form (*, a2 *op* x, *, a4 *op* z), where *op* is a qualifying operator.

So far these aggregations have been inside a particular Collector. A large deployment may have hundreds of Collector agents widely distributed geographically. The second mechanism we have developed for model aggregation allows the model data from widely dispersed DNA Collectors to be merged as long as some basic rules are followed. The ability of a model to be aggregated with other models depends on the definition of the model. History Models and Distribution Models can be combined as long as the data was collected during the same aggregation time interval and the events of the different models are independent. An example is two sets of subscriber usage distributions collected per hour in San Francisco and Los Angeles. As long as the subscribers generating traffic in San Francisco are not the same subscribers generating traffic in Los Angeles and both datasets are from the same day and hour of the day the distributions can be aggregated. To facilitate this kind of model aggregation, the above dimensions of statistics data collection are marked with an independence parameter. This simple facility protects the user from accidentally creating model aggregations that would be meaningless.

9.3 DRILL FORWARD

One of the important capabilities of these Capture Models is that they can be dynamically configured by the user, or some other agent, including the type of model and all of its configuration parameters. This leads to an important concept in stream analysis I call *Drill Forward*.

Most of us are familiar with the concept of *Drill Down* when dealing with multidimensional online analytical processing (MOLAP) or relational online analytical processing (ROLAP) analysis systems. Clicking on a bar of a bar chart creates a new window of the historical detail displaying the next level deeper components that made up the selected bar.

Note the word *historical*, because the presumption of drill down is that there is a database of history behind the data you see. Unfortunately, constructing such a history of data for massive data streams may not be economically practical, or take too long for the reasons discussed previously.

Drill forward is simply a different name for what we all do when something draws our attention. We focus in and look more closely, discarding a vast majority of the other data pummeling our senses. But we are moving forward in time not backward. When we are dealing with massive data streams, the same technique can be used to investigate patterns.

In a stream-processing context, a few key variables could be monitored to establish normative behavior. If there is a sudden change (exceeding a percentile threshold or the change in shape of a distribution, etc.) the rule logic could be dynamically restructured to collect more detailed data about a reduced, but focused subset of the stream where the exception occurred. For example, the appearance of certain traffic patterns may be a precursor to a hostile attack on the network. If this particular pattern occurs, it is desirable to collect additional detail on that substream. A simple example of this can be accomplished with a conditional hash rule, which is a variation of the hash rule above:

```
Hash IPaddress;  
If (port memberOf TrojanPortList) flag=true; else flag=false;  
ConditionalHash (port) {  
    If ( (flag == true) || (IN_HASH == true) ) keep; else drop;  
}  
Aggregate (bytes)
```

In this example, if a single event flowing to (or from) a particular IP address shows traffic activity on one of a list of Trojan ports, a flag triggers aggregation of traffic by port in addition to aggregation by IP address. Once this port has been hashed into the table data continues to be collected for this port for a defined interval of time because it already exists in the hash table. This avoids having to collect high granularity data all the time for all substreams resulting in significant data reduction and efficient processing of these data in downstream systems.

As another example, assume a capture model has been configured to measure the distribution of the number of unique destination addresses per subscriber for outgoing traffic on a routine basis. A large spike of activity at the 99th percentile may signal a subscriber performing address scans on the network. Based on this abnormal event the capture models can be reconfigured with filters to focus in on only the portion of the distribution where the spike occurs, then start exporting additional information about the suspect traffic such as protocol and destination port, which will help identify the type of traffic. The ability to establish normative distributions of various characteristics of a stream and then dynamically explore deviations from the norms adds considerable analysis capability. This technique is ideal for detecting and exploring patterns in a stream, but not for discovering once-in-a-lifetime events.

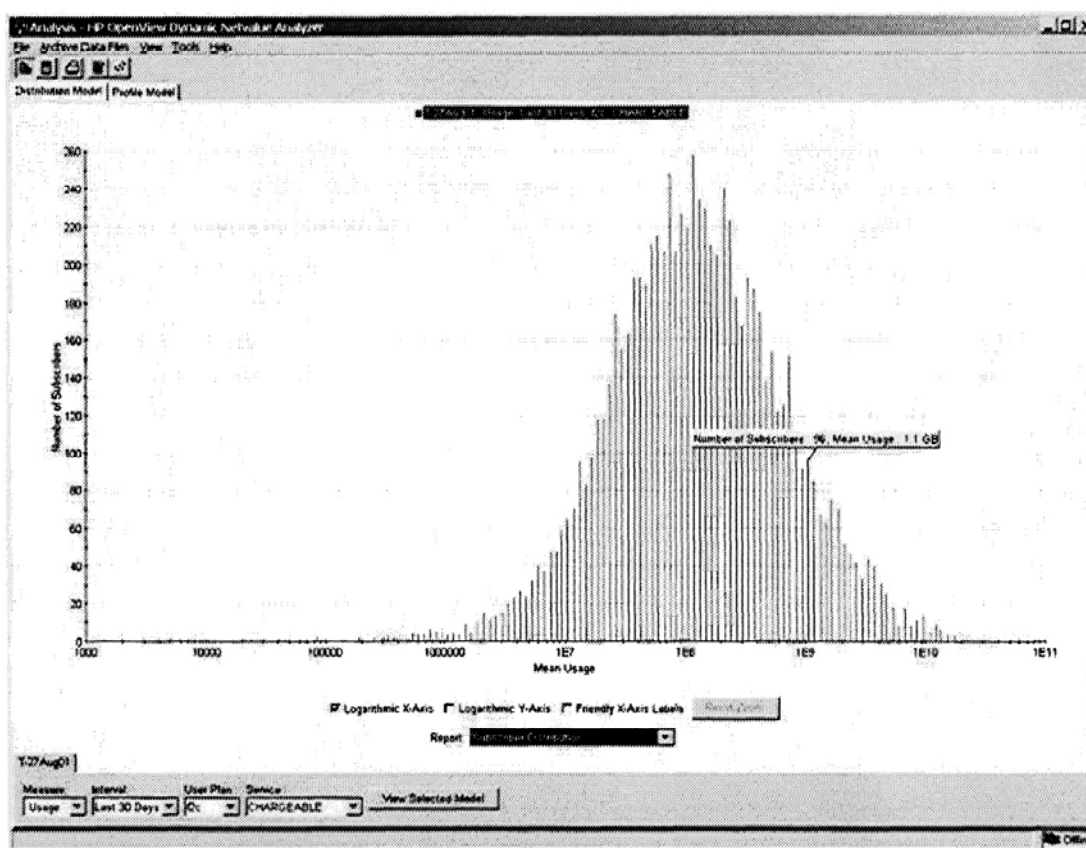


Figure 8. Capture Models can be configured for real-time queries, which enable interactive snap-shot views of the statistical data captured in memory. The above screen-shot reveals the lognormal distribution of subscriber usage.

9.4 USER INTERACTION WITH STREAMING MODELS

The collection and processing of these streams forms the foundation, but users need graphical and visual tools for exploring this space. Wilkinson (1999) has done some extraordinary work in this area. This is a challenging area in its own right and where we will be investing more R&D going forward. The DNA technology suite includes both a browser-based client and a Java application client for more sophisticated viewing and analysis.

Figure 8 is a real data example of the analysis screen examining a subscriber usage distribution. This kind of data can be pulled up from a DNA server using the real-time query mechanism mentioned earlier.

What is interesting is that this usage distribution follows a lognormal distribution over five orders of magnitude (90KB/mo to 22GB/mo) with a shape factor of ~ 0.67 .

Transforming this into a CDF is trivial (Figure 9, top), which gives marketing folks information on how to segment their subscribers based on usage. The graph on the bottom is a percentile-percentile plot of percent subscribers using what percent of the overall traffic. This graph shows that this distribution follows the 80:20 rule, the top 20% of subscribers generate 80% of the traffic. The top 5% generate 50% of all traffic!

To demonstrate how capturing statistics from a stream can generate valuable business

insight, Figure 10 is from the DNA financial modeling tool that uses empirical distribution data collected from the DNA server to compute the estimated dollar value of subscriber traffic modeling different pricing scheme scenarios.

Given

| | | |
|---------|---|---|
| b | = | bytes of usage per month |
| $s(b)$ | = | density function: # subscribers at b |
| $\$(b)$ | = | pricing function: \$ paid by a subscriber with total usage b for the month. |

The revenue in dollars for all subscribers with monthly usage between b_0 and b_1 is

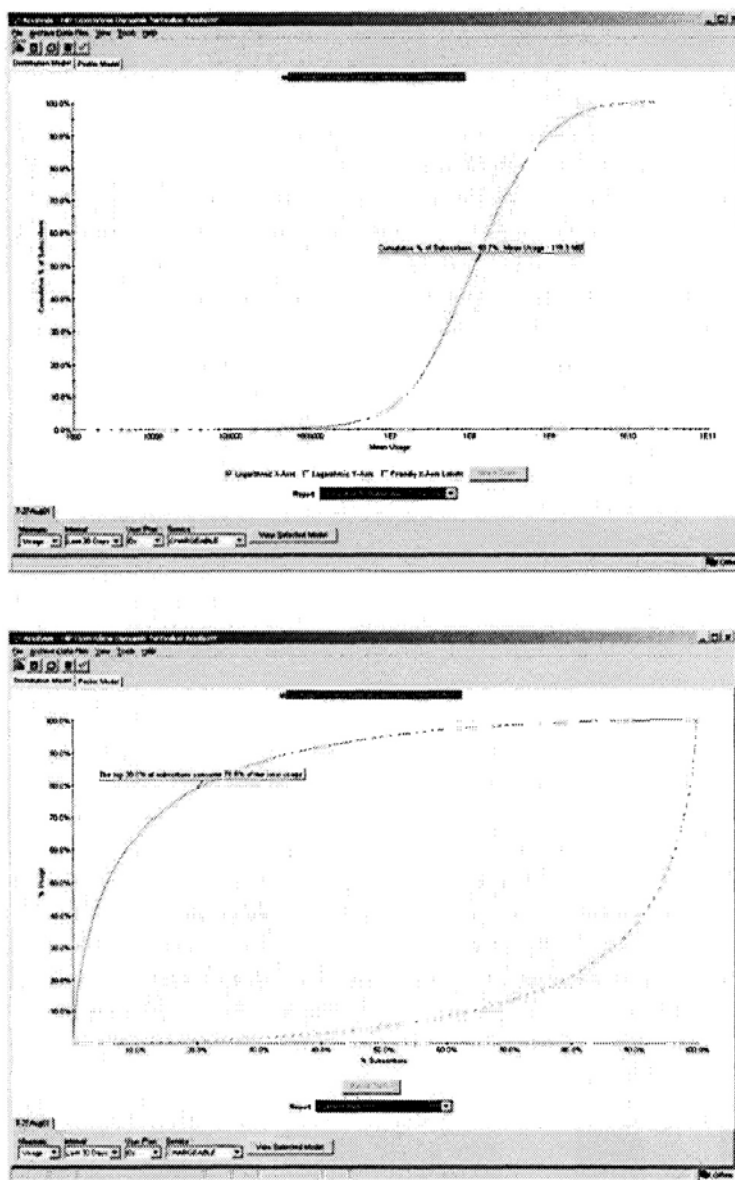


Figure 9. From the empirical distribution, multiple parameters can be derived and various transforms applied.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

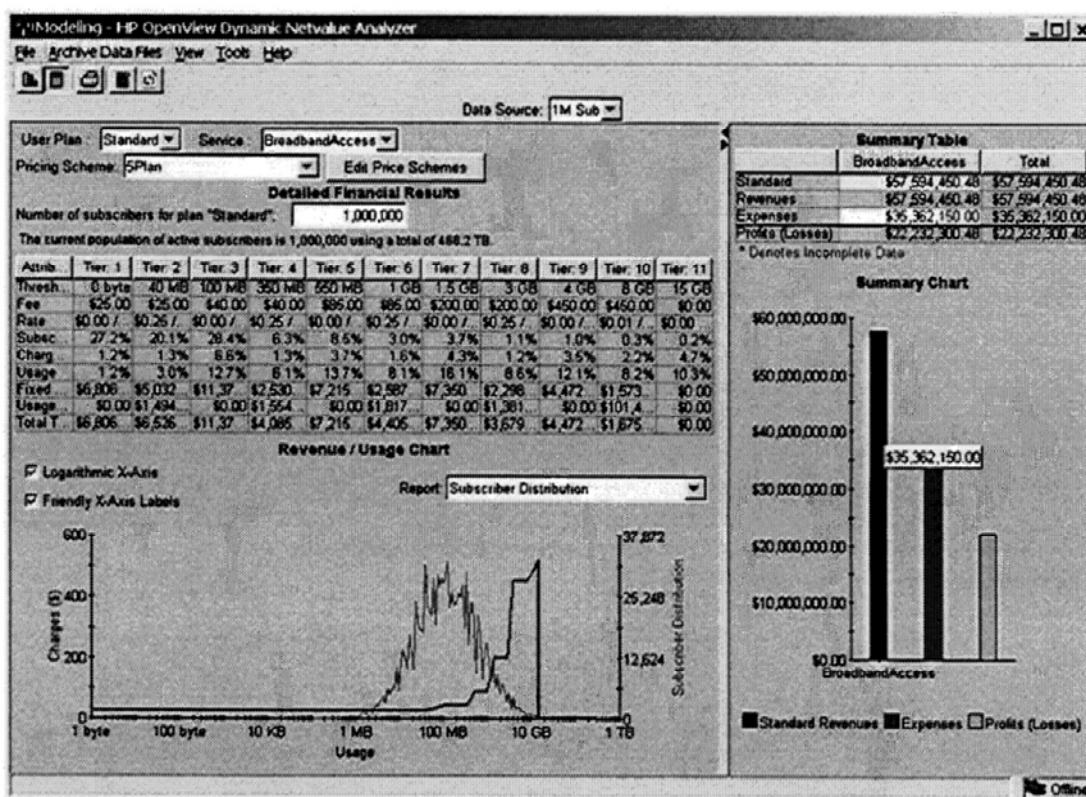


Figure 10. Certain types of traditionally tedious computations can be performed very quickly once the underlying empirical distribution of the data is known. This tool takes advantage of that fact for performing interactive financial analysis of the value (in currency) of a stream based on pricing models input by the user. The empirical distributions can either be extracted in real-time or archived for later comparison and analysis.

$$R(b_0, b_1) = \int_{b_0}^{b_1} s(b)\$(b)db.$$

Because of the compactness of the models this kind of computation can be performed by the client in a few milliseconds, which enables “what-if” modeling based on actual or forecast-extended distribution models of subscriber usage behavior.

Other tools currently in development include network analysis and forecasting for capacity planning as well as a suite of security analysis tools. A more complete discussion of how these more advanced tools take advantage of streaming analysis will be the subject of follow-on papers.

10. SUMMARY

High-momentum data streams and rivers can be expensive to store and require long processing times to analyze using the traditional *store first*, then *analyze later* techniques. Although some types of analysis will always require this time-proven approach, we are discovering that a great deal of valuable insight can be extracted from these streams *prior*

to other data-reduction processes and commitment to storage and yield significant cost and latency reductions as well. Key paradigm shifts that we have had to make (and are still making) in our own thinking have been in areas such as dynamic interaction with stream-oriented programming languages, distributed stream processing architectures and visualization of streams.

ACKNOWLEDGMENTS

I would like to thank some key individuals who have supported me and contributed to this program: Leland Wilkinson, Senior VP, SPSS, who has personally given me strong encouragement to get this material published; Eric Buatois, HP VP, who helped fund early research on these concepts; Jeff Meyer, HP Chief Architect for IUM, who has been the thought leader and creator of many of the key concepts of the IUM platform; Ying He, HP Software developer, who has always been open to changes and yet more changes and has contributed extensively to the DNA server architecture; Eric Peterson, HP Software developer, a great communicator and developer who is primarily responsible for the DNA front-end architecture; Scott Lamons, HP R&D Project Manager, who has been a tremendous asset to the smooth workings of our team and a strong supporter of the program. Cisco IOS® NetFlow is a patented technology of Cisco Systems, Inc. (<http://www.cisco.com>). sFlow® is a registered mark of InMon Corporation (<http://www.inmon.com>).

[Received April 2003. Revised October 2003.]

REFERENCES

- Cormen, T.H., Leiserson, C.E., and Rivest, R.L. (1999), *Introduction to Algorithms*, Cambridge, MA: The MIT Press.
- Hellerstein, J.M., et al. (1997), "The New Jersey Data Reduction Report," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 20, 4.
- Knuth, D.E. (1973), *The Art of Computer Programming* (vol. 3), Reading, MA: Addison-Wesley.
- McGarty, T.P. (2002), *The Imminent Collapse of the Telecommunications Industry*, The Merton Group, <http://www.mertongroup.com/Collapse%20of%20Telecom%202002.pdf>.
- Sidak, J.G. (2003), "The Failure of Good Intentions: The WorldCom Fraud and the Collapse of American Telecommunications After Deregulation," *Yale Journal on Regulation*, http://www.aei.org/docLib/20030403_SSRN_ID335180_code021001500.pdf.
- Wilkinson, L. (1999), *The Grammar of Graphics*, New York: Springer-Verlag.

Pedro Domingos

A General Framework for Mining Massive Data Streams

[Transcript of Presentation and PDF Slides](#)

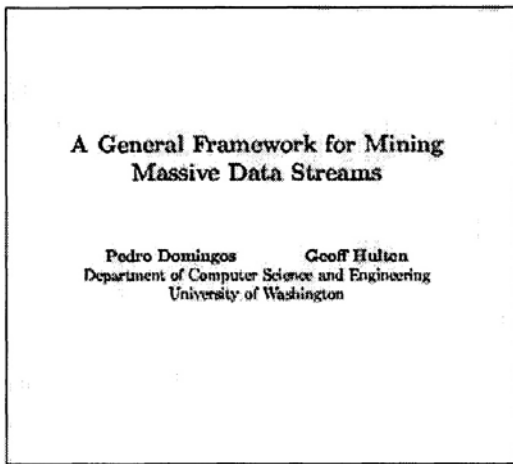
[Technical Paper](#)



BIOSKETCH: Pedro Domingos is a professor in the department of computer science and engineering at the University of Washington. He received a master's degree in electrical engineering and computer science in 1992 from the Instituto Superior Técnica (IST) in Lisbon and a second master's degree in 1994 and a PhD in 1997 in information and computer science from the University of California at Irvine. He spent 2 years as an assistant professor at IST before joining the faculty of the University of Washington in 1999.

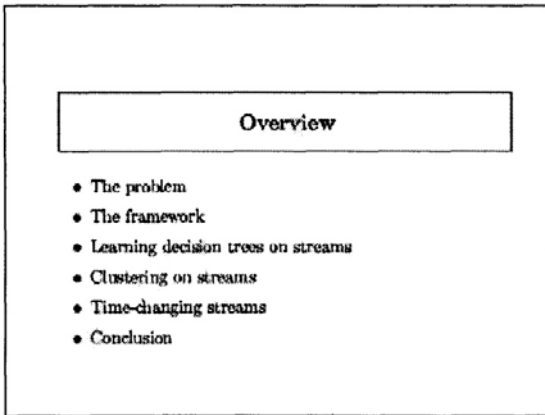
Dr. Domingos is the author or coauthor of over 100 technical publications in topics related to machine learning and data mining. He is also the associate editor of *JAIR*, a member of the editorial board of *Machine Learning*, and a cofounder of the International Machine Learning Society. Dr. Domingos was program co-chair of KDD-2003, and has served on the program committees of American Association for Artificial Intelligence (AAAI), International Conference on Machine Learning (ICML), International Joint Conferences on Artificial Intelligence (IJCAI), Knowledge Discovery and Data Mining (KDD), the World Wide Web Consortium (WWW), and others. He has received an NSF CAREER Award, a Sloan Fellowship, a Fulbright Scholarship, an IBM Faculty Award, two best paper awards at KDD, and other distinctions.

TRANSCRIPT OF PRESENTATION

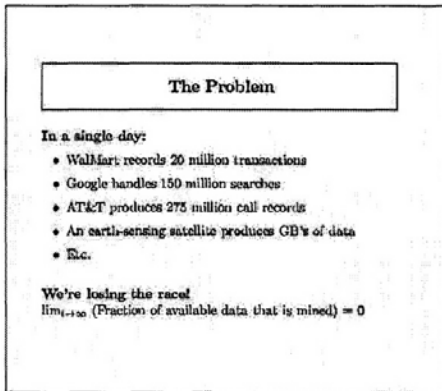


MR. DOMINGOS: My talk is about a general framework for mining data streams. This is joint work that I have done with Jeff Hockney at the Department of Nuclear Science and Engineering at the University of Washington.

So, this talk is basically about building things like classification models, regression models, public information models and messages.



Here is what I am going to do. First, I am going to describe what the problem is that we are trying to solve. Then I am going to present the general framework we have for solving this problem. Then I will describe an example application of this framework. [Comments off microphone]. Then I will conclude with some comments.

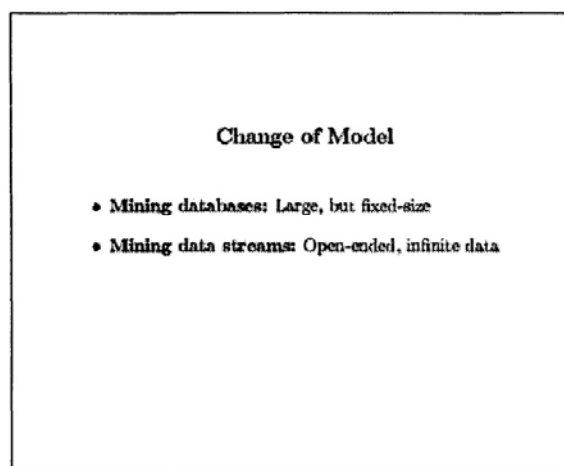


About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

So, what is the problem we are trying to solve? A lot of the classification questions and so forth, algorithms, at least when most of them were developed, you know, the number of data points wasn't very large.

Over the last 10 years, people have made a lot of progress in the sense that there is a lot of hard work— [off microphone] —decision tree learners on data sets. I think there is great achievement, several orders of magnitude. The speed at which the data rates are going up actually exceeds the speed at which we are speeding up our algorithms. So, we are losing the rates. I have a few random examples that I probably don't need to go through because people already know most of them. The bottom line is that, in most domains, in any given day, you can easily collect tens of millions or hundreds of millions of records.

For example, if you want to do a decision tree, doing a decision tree on, say, 20 million records, even with today's best algorithms, takes more than a day. So, in spite of all the progress that we have made, we are actually losing the race. The fraction of the available data that we actually use to build our models is actually dwindling to zero as time goes forward. So, there is something wrong here. We need to do something about it. The thing that we need to do about it is one of mining databases to one of mining data streams.



You know, databases and data streams are different in many respect, but the idea that I have in mind here is that the database can be very large, but it is of a fixed size. At the end of the day, you know that you have so many records, you have so many samples that you can learn from. In the data scheme, it is open ended. In essence, we have infinite data. How would we modify any of the algorithms if we actually had Internet data available, because that is what we have in the data stream. We get another 100 million, and another 100 million records. So, we need to change our model of what we are doing from databases to data streams. What I am going to describe here is a framework to address these decisions. So, what are they?

- Desiderata**
- Small constant time per record
 - Fixed amount of main memory
 - At most one scan of data
 - Results available anytime
 - Results equivalent to standard algorithm
 - Ability to handle time-changing phenomena

First of all, our algorithms need to require only a small amount of time per record. If the time requirement goes up with the number of data points that you have in your past, then you are lost. Sooner or later, you run out of breath. Also, we need to be looking more at statistics than main memory. Clearly, storing all the data in a main memory is not an option.

We want to do a scan of the data. We can't assume that you are going to be able to store your data and go back and look at it. So, we want to be able to do everything we want, by looking at each data point, at most, only once. Notice I say at most. Maybe we can do things in even less than one scan of the data, and that is part of what I am going to be talking about here.

We also want the net results available at any time. Again, the traditional model is that you collect your data and then you run your algorithms on it and then, at some point in the future, your algorithm isn't running and you have your model. That isn't going to work here, because you have to wait forever. So, you want to have a model that gets better and better as time goes by but, at any given point, you can push the button and see what you already have, given the data that you already looked at.

Another very important thing is the following. We would like to ensure the results that we get are, impossible, equivalent to what you would get if you were actually just running your regular algorithm on a regular, but infinite-sized database with a computer with infinite resources. It is very easy to satisfy those requirements if you compromise the quality of the results that you are producing. The whole challenge is to actually guarantee those things while producing, say, decision trees or things that are not different from the ones that you get if you ran the algorithms that we know, and that we have, and whose properties we know.

Finally, we also want to be able to handle time changing phenomena. In a typical database, we just assume that the data is IID, so it doesn't matter what order they are in. In large data streams that last over months or years, very often, the phenomenon that you are looking at is actually changing over time. So, the model that you ran a year ago isn't necessarily valid now, and you also want to be able to deal with that.

So, these are the criteria that we want to satisfy. In fact, when we started doing this work a couple of years ago, it sort of sounded overly ambitious. Where are we going to have to compromise? However, to our amazement, we found that we were actually able to satisfy all those criteria. In fact, what we now have is, we have a framework that meets all those criteria, and we have successfully applied it to several algorithms, including decision tree learning, network learning, image clustering.

Current Status

- Developed framework that satisfies all desiderata (Hulten & Domingos, KDD-2002)
- Successfully applied to
 - Decision tree learning (Domingos & Hulten, KDD-2000; Hulten et al., KDD-2001)
 - Bayesian network learning (Hulten & Domingos, KDD-2002)
 - K-means clustering (Domingos & Hulten, ICML-2001)
 - EM for mixtures of Gaussians (Domingos & Hulten, NIPS-2001)
- Mines billions of examples per day
- Currently developing general-purpose library

We are able, using this, to deal with millions of examples a day using existing hardware. For example, the best decision trees we were able to mine perhaps a million examples per day, and now we are able to mine on the order of a billion examples per day.

What we are doing now is, we are developing the general purpose library such that, if you write your algorithms using this library, you won't have to worry about scalability. You can just write them the same way as you always wrote them, and they will run on data streams without your having to focus on it. You just focus on the problem that you are trying to solve. So, this is not available yet, but it is like our next target that we are dealing with. So, I am not going to have time in this half-hour to actually talk about how we meet each of those problems. I will just focus on the most salient ones. Those are the problems with trying to ensure that you learn the same model, in real time, as quickly as possible, that you would run on infinite data, and a little bit at the end about what happens when the data-generating process is not sufficient.

The Framework

Key Questions

- How much data is enough?
- What if the data-generating process is not stationary?

How Much Data is Enough?

Hoeffding bounds:

- Real-valued random variable x with range R
- n independent observations, mean \bar{x}
- With prob. $1 - \delta$, true mean of x is within ϵ of \bar{x}

$$\epsilon = \sqrt{\frac{R^2 \ln(2/\delta)}{2n}}$$

Basic idea: Bootstrap this to whole complex models

What do I mean by discretion and how much data is enough? In a traditional learning algorithm, what you do is, the algorithm is a sequence of steps of some kind, and each of these steps look like the data in some way. Think of your favorite algorithm and this is how they work. In the traditional setting, each of these steps, in general, looks at the whole data before doing what it wants to do. Now, in a data stream setting, that is not going to work because you would never get past the first step, because you would wait forever for all your data to arrive.

So, what you need to do is, you need to make the decision about how much data to use in each step. In essence, you want to use as little data as possible in each step, so that your model gets better as quickly as possible. I want to have the best possible model as soon as possible, and this means minimizing the amount of data that I am using at each step.

Now, if I reduce the amount that I use at each step, I probably will pay a price for that in terms of the quality of the model that I get. So, the goal of what we are doing here, in a sense, one of the basic features of our framework, it enables you to minimize the amount of time that it takes to learn on the data stream, while guaranteeing that you are still getting the same result that you would get if you waited for all the data up to eternity to arrive.

Some of you may be wondering, well, how can that be possible? To persuade you that that can be possible, I would like to give you the basic idea of what we are doing. Think about the simplest possible model that anybody would want to build. Say you just want to know what the average of some quantity is. What is the number—for example, in the network application, I want to know what the average number of packets is that you get from some source, or whatever you need to find. Suppose that I have, say, a billion samples of this. I don't necessarily need to look at that billion, if I assume that they are IID, to form a good estimate of the mean. This is what pollsters do when they use 3,000 people as a sample for, say, 200 million. I mean, there are many kinds of results that you can use to decide how much data you need. One that we have used a lot is a hooking balance, and what happens in a hooking balance is this. Suppose we use variable X and range R . Then we have N independent observations.

What this guarantees is that, with a probability of $1 - \delta$, the variable is within ϵ of X , where ϵ is given by this expression. It depends on the range. It goes down with the number of data points that you have, and it actually only goes up with the log of one over the area. So, what this formula does is that, suppose you want to find the mean of some

variable, but you are willing to have that mean be off by ϵ , by at most ϵ , with the probability at most δ . Then, I will tell you how many data points you need to gather. Then, after that, you don't have to worry. It doesn't matter if you have a trillion points or infinite points. You don't need to look at them, because you already have the quantity that you want with the tolerance that you want. So, the ϵ and δ are the quality parameters that you have given.

In essence, all we are doing, at least in the basis of our framework, all that we are doing is this. We are bootstrapping this idea to not just one quantity, but to a whole model, for example, a whole density estimation model with all of its parameters, or a whole decision tree with all of its nodes.

So, we take these kinds of things for the individual things that you are doing and, by doing some analysis, you actually come up with a guarantee on the quality of the whole model.

So, in summarizing one slide, here is the approach that we have. It goes in three steps. The first step is that you have an upper bound on the time complexity of your algorithm, on how long it takes to run, as a function of the number of examples that you use in each step.

General Approach

1. Derive upper bound on time complexity,
as function of #examples used in each step
2. Derive upper bound on loss between finite-data
and infinite-data models, as function of #examples
3. Minimize time bound subject to limits on loss

Effectively mines infinite data in finite time.

In many cases, it is just the sum of the number of examples that you use in each step, multiplied by something that is a constant. This is actually quite easy to do. So, we figure out how the running time of the algorithm varies with the number of examples that you use at each step.

Part two, we divide the upper bound on the loss between the model you get with finite data and the model that you would get with infinite data, as a function of the number of examples that you use when you are doing this with finite data. So, you give me your loss function, the loss function that you are going to use to compare to different models. So, if two models are very different by your loss function, then you know your loss function should have a large value. This could be easier or harder, depending on the algorithm of lost function. We figure out how this loss varies when you are comparing what you would get with infinite data, with what you are getting using a finite number of examples.

The final step is to minimize this. So, to minimize this, subject to the user, given constraints on this, meaning that we have a constrained optimization problem. What we are trying to do is to minimize the running time of the algorithm as a function of the number of examples we can expect, subject to the constraint that the loss that you get at

the end of the day, the loss between finite and infinite data, is at the most ϵ , with the probability of at least $1-L$. So, this is the general approach that we have.

Example: Decision Tree Induction

Time: Proportional to #examples used to pick each test

Loss: Probability that decision trees disagree on example

Solution: Use min. #examples at each step that guarantees loss bound

One of the surprises that we found is that we are actually able to do this for very broad classes of algorithm. We studied industry decision trees, and then we did it for clustering, and generally we saw that we could generalize this to many different things.

Notice that, in some sense, what we are effectively doing with this is learning from infinite data in finite time, because we are getting, in finite time, the same results with the same tolerance that you would get if you waited forever, until your data stream gave you infinite data.

So, that is our general idea. There are lots of different methods that I am not talking about here, but that is the basic idea of our framework. Let's see how it applies to the case of decision tree induction, which is actually a particularly simple one, perhaps in some ways not the most interesting one, but it is the first one that we did, and the easiest one to talk about.

So, in decision tree induction, what happens with the time? Remember, the first step was to figure out how the time varies within the examples that are used in each step.

In the decision tree induction, which I assume most people here are familiar with, what happens is that you have a series of nodes. Each node tests the value of some attribute. Depending on that value, it sends it to another node, and at the least you have a classification.

The basic problem in decision tree induction is deciding which tests, which attributes you pick to test in each node.

The amount of time that we need to do that is proportional to the number of examples that you have. What you basically need to do is, you need to gather some sufficient statistics that have to do with how often each value of each attribute goes with each class.

AUDIENCE: Is the number of nodes fixed?

MR. DOMINGOS: No, the number of nodes is not fixed. So, the first thing we are going to have to do with our algorithm is that potentially, as time goes to infinity, the size of your tree could grow infinitely as well. In practice, it usually doesn't, but in principle, it could. So, when you have a model that is of unbounded size, potentially, the model takes an infinite amount of time to learn.

What we are concerned with is what is the time that each step takes, and we are going to try to minimize, in this case, the sum of those times, which means that you run

the model faster.

So, notice, traditionally, and even in some of the very fast optimized algorithms for learning decision trees on very large databases, they use all the data to take each test. If you unleash those algorithms on a data stream, in some sense, they never even pick, because they keep waiting for the rest of the data to come up, before they make their first decision. Intuitively, you know, we should be able to make that decision after we see so many examples, and then go on to the next node, and that is exactly what we did.

So, what about the lost function? How are we going to compare the decision tree we are building with finite data with the decision tree that we would build with infinite data. You can do this in many ways. We are actually going to use a very stringent criteria which is, I am going to take the loss as being the probability that the decision trees disagree on the random example. So, this is much more demanding than just we find that the tree be at least nearly as accurate as the other tree, or that they make nearly the same decisions.

For the two trees to be an example, I am going to require that, from the point of view of that example, the two trees are indistinguishable, meaning the examples see the exact same sequence of tests, and winds up with the same class prediction at the end of the day. So, this is going to be my loss function.

AUDIENCE: It doesn't seem that the tests matter, if it gets to the same answer.

MR. DOMINGOS: Maybe it does, maybe it doesn't. For some applications, it matters. The point is that, once you get guarantees on this loss function, you get guarantees on all the others, since we can do this at no extra cost.

So, the final step, of course, is to try to minimize the limits on this loss. Without going over a lot of details, what is going to happen here is, what we are going to do, we are going to use the minimum number of examples to fit each test at each node such that, at the end of the day, the probability of disagreement is going to be bounded, and we are going to see the algorithm, and then we are going to see one example of the kinds of guarantees that we can get through the algorithm.

So, here is our algorithm for mining massive data streams. We call it DRPT. Again, at a very high level, it has two arguments. The stream, which is an infinite sequence of examples, and δ is the limits that we want to impose on the disagreements between the two trees. We want to learn on that stream as fast as we can, subject to the concerns that the disagreement between that tree and what we would have with infinite data is, at most, δ .

I beg your pardon, this δ is actually not the disagreement between the whole trees. We are going to call that Δ . This is the probability of getting each test wrong, and we are going to see how the two relate. It is the probability of actually making the wrong decision on any given step.

```
Procedure VFDT(Stream,  $\delta$ )  
-----  
Let HT = Tree with single leaf (root).  
Initialize counts  $n_{ij}$  at root.  
For each example (x, y) in Stream  
  Sort (x, y) to leaf using HT.  
  Update counts  $n_{ij}$  at leaf.  
  Compute Gain for each attribute.  
  If Gain(Best attr.) - Gain(2nd best) >  $\epsilon$ , then  
    Split leaf on best attribute.  
    For each branch  
      Start new leaf and initialize counts.  
Return HT.  
-----
```

So, how does our algorithm work? We start with a tree that just has the roots. We initialize to suggest the count of the number of times that each value generated with each attribute I occurs with each class, K .

Now, for each example, X is the attributes and Y is the class. We sort that example through a leaf. We are going to have a partial tree at any given point. Then we update the concept technique. Then, the information gain on a linear or any other matters, we compute it for each attribute. Then, if the data of the best attribute is better than the data of the second attribute by at least epsilon, when epsilon is a function of this δ , then at this point we know that, with probability $1-\delta$, we are picking the same attribute that we would be picking if we waited until we saw infinite data. So, at that point, we just split on that leaf, and now we start sending the new examples in the stream down to the children of that leaf.

So, we start with the root. After some of our examples, we go through the roots to the leaves, and then the examples get routed to its children, and then we collect examples of those guys and at some point we pick the test data, and we keep building it in this way.

So, notice, if you think about it for a second, this satisfies everything that we were talking about. It does require an amount of memory that is independent of a number of examples that you have seen and so forth. It is any time, because at any time you have a partial tree built and so forth. We have a few questions here.

AUDIENCE: [Question off microphone.]

MR. DOMINGOS: No, we deal with continuous inputs. In the particular network we are talking about here, we actually— [off microphone.] Again, we can deal with continuous attributes.

AUDIENCE: [Comments off microphone] —at many levels?

MR. DOMINGOS: Yes. That is actually not a problem with the levels that you have in the attributes. You just have more statistics to start with.

AUDIENCE: It appears to be a greedy tree-building algorithm. From that standpoint, it would perhaps lead to an inferior tree when compared to one that is pruned back with something else.

MR. DOMINGOS: The basic algorithm that we are trying to use isn't really a decision tree algorithm. So, we are doing the same thing. [Comments off microphone.] You can stream data forward. So, this is the basic algorithm. The main claim that I made to you is that, running a decision tree in this way, in some sense, you will learn it in as little time as you can.

What you will get is a tree that disagrees very little with the tree that you get on

infinite data. Well, how come I can make that claim? There are actually various kinds of results that we can show here. The one that I am going to show you is select a better bound. To get that better bound, we have to make one assumption about the data, which is almost always true, and that is why I am presenting it here.

We can also get results without making that assumption. The assumption I am going to make is that some fraction of your data, at any given level of the tree, winds up in a leaf. Again, in all the dozens of domains that I have applied decision trees, I have never seen one where this doesn't happen. What this means is that the tree is somewhat unbalanced. You don't have all your leaves at the last level. If that happens, we will have to use a different kind of guarantee than the one I am going to give here.

Again, for purposes of analysis, I am just going to assume that that probability is the same at every level, which it doesn't need to be, and I am going to call it P , the leaf probability.

Guarantees on VFDT's Performance

Disagreement between two decision trees:
 $\Delta(DT_1, DT_2) = P_x[\text{Path}_1(x) \neq \text{Path}_2(x)]$

Let

HT_δ VFDT tree given δ , infinite sequence of exs.
 DT Batch DT given infinite exs.
 p Leaf probability

Theorem: $E[\Delta(HT_\delta, DT)] \leq \delta/p$

You will recall that in our last measure there was some disagreement between some trees, and that I define as the probability, that the path of the example through the first tree differs from the path that the example takes in the second.

The two trees that I am going to compare is the decision tree that I learned using the algorithm that I just saw, with parameter δ , and the decision tree that I would learn in batch mode that I would learn using an ordinary decision tree algorithm, with infinite memory and infinite time, waiting to see infinite samples.

Here is what we can guarantee. We can guarantee that the expected disagreement between the two trees is bound by δ/p . So, the δ number is the value that we use at each node. I guess what this means is, if I get a smaller δ , call like this δ the one that you guarantee. You know the leaf probability. Then immediately you know which lowercase δ you need to use at each local decision to get the result you want.

Obviously, the lower δ , the smaller δ gets, the lower the leaf gets. Not surprisingly, the lower the leaf probability, the bigger the tree you are going to grow. So, the smaller it has to be to get to the big δ .

AUDIENCE: It is noteworthy that HT_δ is also on an infinite sequence.

MR. DOMINGOS: I will get to that in a second. Actually, I will get to that right now. One correlate of this theorem is that, on finite data, what our tree is going to be is a subtree. That is the best you could ever hope for, again, with the same probabilistic guarantee. Then, it is like a two-step process to see how that correlator comes about.

Without actually proving the theorem here, it is actually quite easy to give you

some idea of the basic intuition as to why we are able to have a guarantee like that.

Theorem: $E[\Delta(HT_i, DT_i)] \leq \delta/p$

Why?
Prob(paths differ) $\leq \delta \times$ Path length
Prob(ex. falls in leaf at level i) $= (1-p)^{i-1}p$
 $E[\Delta(HT_i, DT_i)] \leq \sum_{i=1}^{\infty} (\delta i)(1-p)^{i-1}p = \delta/p$

Corollary:
Finite HT differs from some subtree of DT , by at most δ/p

Bottom line:
Near-perfect agreement even with few examples
E.g.: $p = 1\%$, 725 exs/node \Rightarrow Disagreement $\leq 0.01\%$
Linear increase in #examples
 \Rightarrow Exponential decrease in disagreement

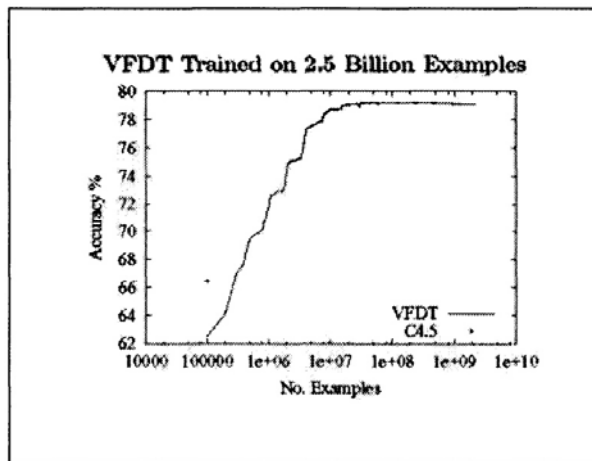
The reason is actually just the following. Notice that if you have an example going down a certain path in the two trees, and the probability of that example bumping into another one that is different is bounded by δ because we designed it that way.

So, if the path is of a certain length, L , then by the union of the probability that the data disagrees anywhere in the path is, at most, δ times L . This is, in fact, a loose bound. So, this is the problem when you get two paths of L data. Now, the other side of this is, if we have a leaf probability of P , then the probability that a random example falls into a leaf at level I is the probability that it doesn't fall into a leaf at any of the P levels, so it is $(1-p)^{I-1}p$, which is the probability that it falls into a leaf at this level.

Although we need to consider computer expected disagreement, this is just the sum of the probability of these two guys over all possible levels. The expected disagreement is the sum of all possible levels of the probability that the example falls into that level, and the expected disagreements for that level.

So, if you take this expression and you sum it, you just come out with the δ that we have. Like I said, there is this nice corollary that applies to, in finite time, basically what you get is a subtree of the decision tree that you would get with infinite data, again, with at most this disagreement.

The bottom line here, though, is the following, that people in competition line theory have a lot of bounds that have this flavor, but they are incredibly loose. They are nice for intuition but, from a practical point of view, they are useless.



The results that we are getting here is very different, because we are not looking at the size of the models to consider. We are actually only looking at the concrete number of decisions that you make. The really nice thing, is that the number of examples that you need actually only goes up logarithmically with a δ that you want to assure.

Putting it another way, as the number of examples grows linearly with time as you see more of the stream, your guarantees get exponentially better. The fact that your guarantees get exponentially better means that, with realistic numbers of examples, we can get disagreements that are extremely low. So, this is just an example to make things easy.

Let's say that your leaf probability is 1 percent, which basically means that your tree is huge, because only 1 percent of your examples runs into a leaf at any given level. So, this is a very large tree. Let's suppose that you have 725 examples per node. It is not a very large number, if you think of the data stream of millions of numbers coming at you. This is enough to guarantee disagreement of less than 1 in 10,000, .01 percent.

So, here is one graph of results out of the many that we have produced, just to give you a flavor of what we are able to do. This is the FPT framework, 2.5 billion examples. Using an ordinary PC, it took us a couple of days to do this. In fact, the time to learn is completely dwarfed by the time just required to read the data from this.

So, the best application of this algorithm is actually not unlike the kind of stuff that people have been talking about all day today, where you never even study an example. The algorithm actually takes an order of magnitude less time to run than it actually takes to just read them from this.

So, we see a standard decision-tree-running algorithm, and this is it running on the maximum number of examples that we could store in memory, which was 100,000. It had about 66 percent accuracy. Without going into details of the data set, it was composed of 100 attributes. So, this is a 100-dimensional data set.

What we see here is that the FDT, it is able to expecting more and more from the data right up to 100 million examples. Actually, between there and the 2.5 billion, we don't get anything, a slight amount of increase. You could get God knows how many, but the FDT wouldn't have any trouble with that. Again, this is just one result with a very large number. Let me just mention the issue of time changing data.

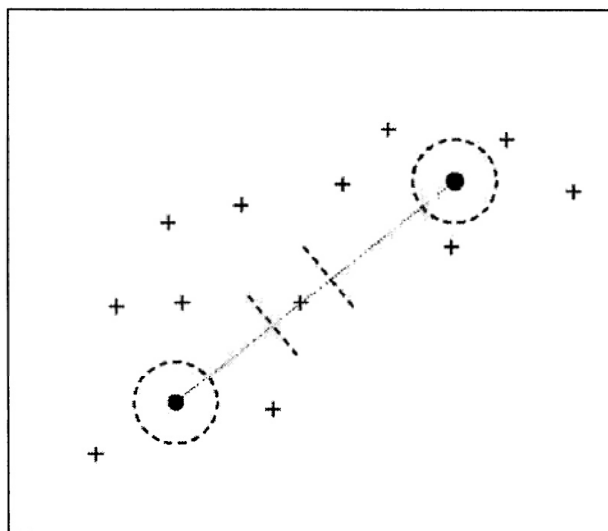
Everything I have talked about here assumes what those bounds assume. What those bounds assume is that the data are independent. In fact, the Hawking model isn't identically distributed, but it is significantly independent. What this means is that you data has to be generated by a stationary path, but then it has to be exchangeable. It can't be changing over time.

As long as that happens, in some sense, we don't need memory, because the stream itself is a memory, and that is kind of what makes our algorithms work, is that at some point you don't have to see more, because it doesn't contain any more information with respect to your model than you have already seen.

However, in a lot of applications of interest, what happens is that the data generating hypothesis is stationary. It is changing over time. So, what do we do in that case?

Example: *K*-Means Clustering

- Two sources of error in *k*-means:
 - Sampling error
 - Assignment error
- Compound these over successive iterations
- Final error is function of #examples used in each iteration
- Minimize these, subject to loss bound



The VFKM Algorithm

- Runtime quantities required for optimization
- Run *k*-means with min. #examples that could satisfy Hoeffding
- If Hoeffding satisfied, we're done
- If not, re-run with new min. #examples at each step

Time-Changing Data Streams

- Standard technique: Sliding window
- CVFDT: VFDT + Example forgetting
- Old examples subtracted from sufficient stats
- If new best split, grow alternative subtree
- When new subtree more accurate, switch it in
- Similar to windowing, but with constant cost per example, instead of $O(\text{window size})$
- Uses old information while still useful

Again, our framework handles this for very broad classes of algorithms. For an easy explanation, I am just now talking about how we do it in the context of decision trees. So, the standard technique to handle time changing data is to use a sliding window, or use some kind of weighted behavior examples.

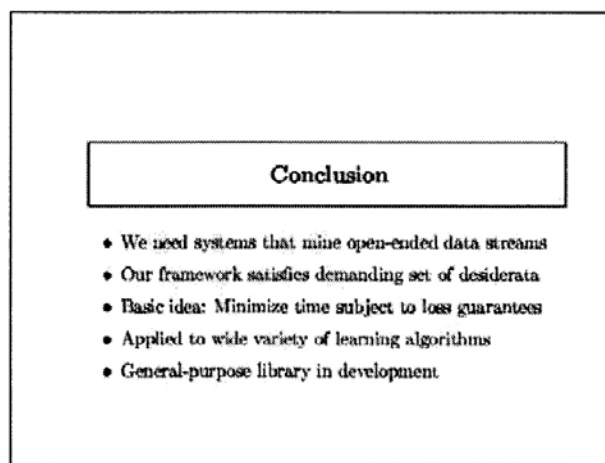
So, let's look at what is considered the sliding window case and generalizing it to the— [off microphone.] What you do is that, the model that you have at any given point is always the model run on— [off microphone.]

The problem we have is that, at the data rates that we are looking at, you don't want to have to start a sliding window. It would just be like that. Your sliding window might be a day. A day is like 100 million examples. We can't store that. Also, you don't want to relearn your model every time a new example comes up in the sliding window. However, what you can do in our framework, almost every learning algorithm under the sun basically runs by computing some sufficient statistic, the decision tree algorithm that we just saw. So, all that we need to do, in order to effectively run on the sliding window, is to be able to forget examples.

That means that, if we want to run on a sliding window, when a new example comes up, as well as adding that example to the sufficient statistics, we subtract the oldest example in the window from those statistics. If the data is stationary, then the new examples and the old ones are equivalent and nothing changes. However, if the data is not stationary, then what happens is that at some point, what looks like the best is no longer the best way. At that point, what we start doing is growing a new subtree, and we leave those two until the new one is doing better than the old one. So, we actually keep the old stuff around as long as it is still useful.

Suppose that the route suddenly becomes outdated. You don't want your performance to be bad because it is the entire tree. So, we let that tree stay there. We start growing a new tree and, when the new one becomes better than the old one, we switch it again. Again, I am glossing over a lot of details here, but this is the basic idea.

Let me just conclude. I will skip over some of the future work that we are thinking of doing. I hope I have convinced you—and you probably don't need convincing—that we need systems that mine open-ended streams of data.



What I described here is a framework that satisfies a very stringent set of algorithms that learn on massive data streams.

You know, the basic idea behind this framework is to minimize the amount of time that you need to grow a model, or you build a model on the data stream subject to guarantees that that model is going to be almost the same as you get with infinite data.

We have applied it to a wide variety of learning algorithms by now, and we have actually developed now in a library that hopefully we will make available in a few months or a year, that anybody who is interested in doing classification or clustering or regression estimation can use these primitives.

What we guarantee is that, as long as you access to data is encapsulated in the structures that we give you, it will scale to large data streams without your having to worry about it.

AUDIENCE: Could you share your thoughts on 4.5 versus—at 100,000 or 105. So, it is clear that you are ultimately going to win, and in the larger applications, that is the way to go clearly.

Is there a concern that you are not as efficient as sort of spitting out data than you would be— [off microphone.]

MR. DOMINGOS: Quite so. The reason it is doing better is because— [off microphone.]. The reason that basically C.45 does that is it is using each example multiple times, whereas we are only using each sample once.

So, you can modify algorithms by using each sample multiple times. So, what we are seeing here is the crudest version of this algorithm.

The other thing that we can do is, we can bootstrap our algorithm with C.45 running on the number of examples that we are picking at random, and then start for there.

C.45 actually makes lots of bad decisions. So, at the end of the day, we are going to offset the window. There is more stuff that I can tell you about it.

A GENERAL FRAMEWORK FOR MINING MASSIVE DATA STREAMS

Pedro Domingos Geoff Hulten

Department of Computer Science and Engineering

University of Washington

Box 352350

Seattle, WA 98185–2350, U.S.A.

{pedrod, ghulten}@cs.washington.edu

Abstract

In many domains, data now arrives faster than we are able to mine it. To avoid wasting this data, we must switch from the traditional “one-shot” data mining approach to systems that are able to mine continuous, high-volume, open-ended data streams as they arrive. In this extended abstract we identify some desiderata for such systems, and outline our framework for realizing them. A key property of our approach is that it minimizes the time required to build a model on a stream, while guaranteeing (as long as the data is i.i.d.) that the model learned is effectively indistinguishable from the one that would be obtained using infinite data. Using this framework, we have successfully adapted several learning algorithms to massive data streams, including decision tree induction, Bayesian network learning, k -means clustering, and the EM algorithm for mixtures of Gaussians. These algorithms are able to process on the order of billions of examples per day using off-the-shelf hardware. Building on this, we are currently developing software primitives for scaling arbitrary learning algorithms to massive data streams with minimal effort.

1 The Problem

Many (or most) organizations today produce an electronic record of essentially every transaction they are involved in. When the organization is large, this results in tens or hundreds of millions of records being produced every day. For example, in a single day WalMart records 20 million sales transactions, Google handles 150 million searches, and AT&T produces 275 million call records. Scientific data collection (e.g., by earth sensing satellites or astronomical observatories) routinely produces gigabytes of data per day. Data rates of this level have significant consequences for data mining. For one, a few months' worth of data can easily add up to billions of records, and the entire history of transactions or observations can be in the hundreds of billions. Current algorithms for mining complex models from data (e.g., decision trees, sets of rules) cannot mine even a fraction of this data in useful time.

Further, mining a day's worth of data can take more than a day of CPU time, and so data accumulates faster than it can be mined. As a result, despite all our efforts in scaling up mining algorithms, in many areas the fraction of the available data that we are able to mine in useful time is rapidly dwindling towards zero. Overcoming this state of affairs requires a shift in our frame of mind from mining databases to mining data streams. In the traditional data mining process, the data to be mined is assumed to have been loaded into a stable, infrequently-updated database, and mining it can then take weeks or months, after which the results are deployed and a new cycle begins. In a process better suited to mining the high-volume, open-ended data streams we see today, the data mining system should be continuously on, processing records at the speed they arrive, incorporating them into the model it is building even if it never sees them again. A system capable of doing this needs to meet a number of stringent design criteria:

- It must require small constant time per record, otherwise it will inevitably fall behind the data, sooner or later.
- It must use only a fixed amount of main memory, irrespective of the total number of records it has seen.
- It must be able to build a model using at most one scan of the data, since it may not have time to revisit old records, and the data may not even all be available in secondary storage at a future point in time.
- It must make a usable model available at any point in time, as opposed to only when it is done processing the data, since it may never be done processing.
- Ideally, it should produce a model that is equivalent (or nearly identical) to the one that would be obtained by the corresponding ordinary database mining algorithm, operating without the above constraints.
- When the data-generating phenomenon is changing over time (i.e., when concept drift is present), the model at any time should be up-to-date, but also include all information from the past that has not become outdated.

At first sight, it may seem unlikely that all these constraints can be satisfied simultaneously. However, we have developed a general framework for mining massive data streams that satisfies all six (Hulten & Domingos, 2002). Within this framework, we have designed and implemented massive-stream versions of decision tree induction (Domingos & Hulten, 2000; Hulten et al., 2001), Bayesian network learning (Hulten & Domingos, 2002), k-means clustering (Domingos & Hulten, 2001) and the EM algorithm for mixtures of Gaussians (Domingos & Hulten, 2002). For example, our decision tree learner, called VFDT, is able to mine on the order of a billion examples per day using off-the-shelf hardware, while providing strong guarantees that its output is very similar to that of a "batch" decision tree learner with access to unlimited resources. We are currently developing a toolkit to allow implementation of arbitrary stream mining algorithms with no more effort than would be required to implement ordinary learners. The goal is to automatically achieve the six desiderata above by using the primitives we provide and following a few simple guidelines. More specifically, our framework helps to answer two key questions:

- How much data is enough? Even if we have (conceptually) infinite data available, it may be the case that we do not need all of it to obtain the best possible model of the type being mined. Assuming the data-generating process is stationary, is there some point at which we can “turn off” the stream and know that we will not lose predictive performance by ignoring further data? More precisely, how much data do we need at each step of the mining algorithm before we can go on to the next one?
- If the data-generating process is not stationary, how do we make the trade-off between being up-to-date and not losing past information that is still relevant? In the traditional method of mining a sliding window of data, a large window leads to slow adaptation, but a small one leads to loss of relevant information and overly-simple models. Can we overcome this trade-off?

In the remainder of this extended abstract we describe how our framework addresses these questions. Further aspects of the framework are described in Hulten and Domingos (2002).

2 The Framework

A number of well-known results in statistics provide probabilistic bounds on the difference between the true value of a parameter and its empirical estimate from finite data. For example, consider a real-valued random variable x whose range is R . Suppose we have made n independent observations of this variable, and computed their mean \bar{x} . The Hoeffding bound (Hoeffding, 1963) (also known as additive Chernoff bound) states that, with probability at least $1 - \delta$, and irrespective of the true distribution of x , the true mean of the variable is within ϵ of \bar{x} , where

$$\epsilon = \sqrt{\frac{R^2 \ln(2/\delta)}{2n}}$$

Put another way, this result says that, if we only care about determining x to within ϵ of its true value, and are willing to accept a probability of δ of failing to do so, we need gather only $n = \frac{1}{2} (R/\epsilon)^2 \log(2/\delta)$ samples of x . More samples (up to infinity) produce in essence an equivalent result. The key idea underlying our framework is to “bootstrap” these results, which apply to individual parameters, to similar guarantees on the difference (loss) between the whole complex model mined from finite data and the model that would be obtained from infinite data in infinite time. The high-level approach we use consists of three steps:

1. Derive an upper bound on the time complexity of the mining algorithm, as a function of the number of samples used in each step.
2. Derive an upper bound on the relative loss between the finite-data and infinite-data models, as a function of the number of samples used in each step of the finite-data algorithm.
3. Minimize the time bound (via the number of samples used in each step) subject to user-defined limits on the loss.

Where successful, this approach effectively allows us to mine infinite data in finite time, “keeping up” with the data no matter how much of it arrives. At each step of the algorithm, we use only as much data from the stream as required to preserve the desired global loss guarantees. Thus the model is built as fast as possible, subject to the loss targets. The tighter the loss bounds used, the more efficient the resulting algorithm will be. (In practice, normal bounds yield faster results than Hoeffding bounds, and their general use is justifiable by the central limit theorem.) Each data point is used at most once, typically to update the sufficient statistics used by the algorithm. The number of such statistics is generally only a function of the model class being considered, and is independent of the quantity of data already seen. Thus the memory required to store them, and the time required to update them with a single example, are also independent of the data size.

When estimating models with continuous parameters (e.g., mixtures of Gaussians), the above procedure yields a probabilistic bound on the difference between the parameters estimated with finite and infinite data. (By “probabilistic,” we mean a bound that holds with some confidence $1-\delta^*$, where δ^* is user-specified. The lower the δ^* , the more data is required.) When building models based on discrete decisions (e.g., decision trees, Bayesian network structures), a simple general bound can be obtained as follows. At each search step (e.g., each choice of split in a decision tree), use enough data to ensure that the probability of making the wrong choice is at most δ . If at most d decisions are made during the search, each among at most b alternatives, and c checks for the winner are made during each step, by the union bound the probability that the total model produced differs from what would be produced with infinite data is at most $\delta^*=bcd\delta$. For specific algorithms and with additional assumptions, it may be possible to obtain tighter bounds (see, for example, Domingos & Hulten (2000)).

3 Time-Changing Data

The framework just described assumes that examples are i.i.d. (independent and identically distributed). However, in many data streams of interest this is not the case; rather, the data-generating process evolves over time. Our framework handles time-changing phenomena by allowing examples to be forgotten as well as remembered. Forgetting an example involves subtracting it from the sufficient statistics it was previously used to compute. When there is no drift, new examples are statistically equivalent to the old ones and the mined model does not change, but if there is drift a new best decision at some search point may surface. For example, in the case of decision tree induction, an alternate split may now be best. In this case we begin to grow an alternative subtree using the new best split, and replace the old subtree with the new one when the latter becomes more accurate on new data. Replacing the old subtree with the new node right away would produce a result similar to windowing, but at a constant cost per new example, as opposed to $O(w)$, where w is the size of the window. Waiting until the new subtree becomes more accurate ensures that past information continues to be used for as long as it is useful, and to some degree overcomes the trade-off implicit in the choice of window size. However, for very rapidly changing data the pure windowing method may still produce better results (assuming it has time to compute them before they become outdated, which may not be the case). An open direction of research that we are

beginning to pursue is to allow the “equivalent window size” (i.e., the number of time steps that an example is remembered for) to be controlled by an external variable or function that the user believes correlates with the speed of change of the underlying phenomenon. As the speed of change increases the window shrinks, and vice-versa. Further research involves explicitly modeling different types of drift (e.g., cyclical phenomena, or effects of the order in which data is gathered), and identifying optimal model updating and management policies for them. Example weighting (instead of “all or none” windowing) and subsampling methods that approximate it are also relevant areas for research.

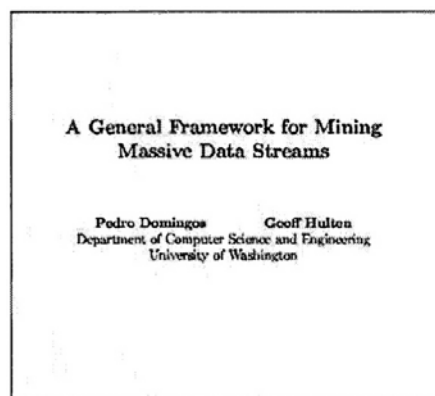
4 Conclusion

In many domains, the massive data streams available today make it possible to build more intricate (and thus potentially more accurate) models than ever before, but this is precluded by the sheer computational cost of model-building; paradoxically, only the simplest models are mined from these streams, because only they can be mined fast enough. Alternatively, complex methods are applied to small subsets of the data. The result (we suspect) is often wasted data and outdated models. In this extended abstract we outlined some desiderata for data mining systems that able to “keep up” with these massive data streams, and some elements of our framework for achieving them. A more complete description of our approach can be found in the references below.

References

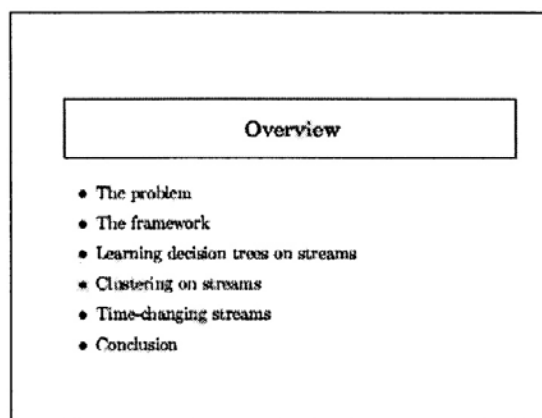
- Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 71–80). Boston, MA: ACM Press.
- Domingos, P., & Hulten, G. (2001). A general method for scaling up machine learning algorithms and its application to clustering. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 106–113). Williamstown, MA: Morgan Kaufmann.
- Domingos, P., & Hulten, G. (2002). Learning from infinite data in finite time. In T.G.Dietterich, S.Becker and Z.Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, 673–680. Cambridge, MA: MIT Press.
- Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58, 13–30.
- Hulten, G., & Domingos, P. (2002). Mining complex models from arbitrarily large databases in constant time. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 525–531). Edmonton, Canada: ACM Press.
- Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 97–106). San Francisco, CA: ACM Press.

TRANSCRIPT OF PRESENTATION

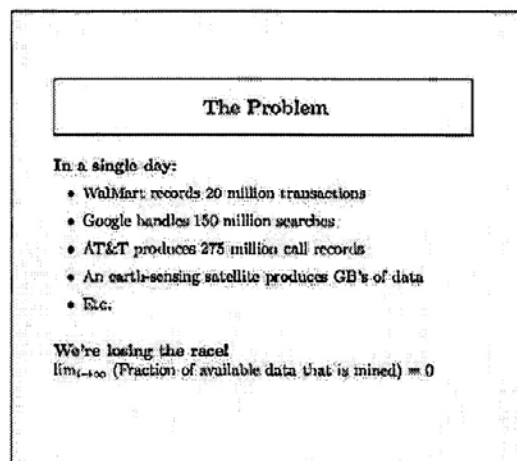


MR. DOMINGOS: My talk is about a general framework for mining data streams. This is joint work that I have done with Jeff Hockney at the Department of Nuclear Science and Engineering at the University of Washington.

So, this talk is basically about building things like classification models, regression models, public information models and messages.



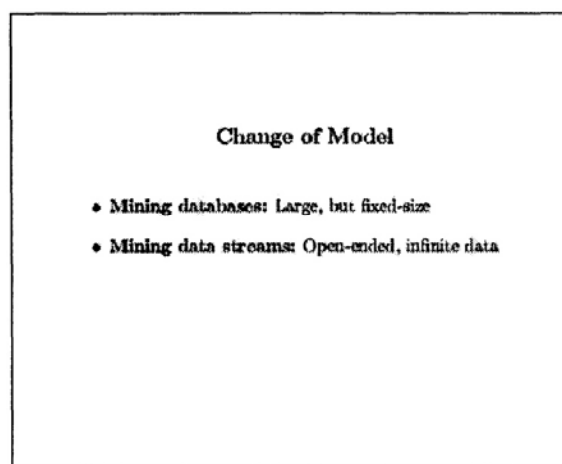
Here is what I am going to do. First, I am going to describe what the problem is that we are trying to solve. Then I am going to present the general framework we have for solving this problem. Then I will describe an example application of this framework. [Comments off microphone]. Then I will conclude with some comments.



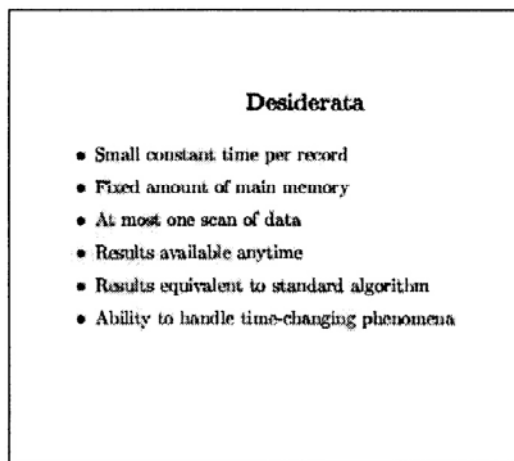
So, what is the problem we are trying to solve? A lot of the classification questions and so forth, algorithms, at least when most of them were developed, you know, the number of data points wasn't very large.

Over the last 10 years, people have made a lot of progress in the sense that there is a lot of hard work— [off microphone] —decision tree learners on data sets. I think there is great achievement, several orders of magnitude. The speed at which the data rates are going up actually exceeds the speed at which we are speeding up our algorithms. So, we are losing the rates. I have a few random examples that I probably don't need to go through because people already know most of them. The bottom line is that, in most domains, in any given day, you can easily collect tens of millions or hundreds of millions of records.

For example, if you want to do a decision tree, doing a decision tree on, say, 20 million records, even with today's best algorithms, takes more than a day. So, in spite of all the progress that we have made, we are actually losing the race. The fraction of the available data that we actually use to build our models is actually dwindling to zero as time goes forward. So, there is something wrong here. We need to do something about it. The thing that we need to do about it is one of mining databases to one of mining data streams.



You know, databases and data streams are different in many respect, but the idea that I have in mind here is that the database can be very large, but it is of a fixed size. At the end of the day, you know that you have so many records, you have so many samples that you can learn from. In the data scheme, it is open ended. In essence, we have infinite data. How would we modify any of the algorithms if we actually had Internet data available, because that is what we have in the data stream. We get another 100 million, and another 100 million records. So, we need to change our model of what we are doing from databases to data streams. What I am going to describe here is a framework to address these decisions. So, what are they?



First of all, our algorithms need to require only a small amount of time per record. If the time requirement goes up with the number of data points that you have in your past, then you are lost. Sooner or later, you run out of breath. Also, we need to be looking more at statistics than main memory. Clearly, storing all the data in a main memory is not an option.

We want to do a scan of the data. We can't assume that you are going to be able to store your data and go back and look at it. So, we want to be able to do everything we want, by looking at each data point, at most, only once. Notice I say at most. Maybe we can do things in even less than one scan of the data, and that is part of what I am going to be talking about here.

We also want the net results available at any time. Again, the traditional model is that you collect your data and then you run your algorithms on it and then, at some point in the future, your algorithm isn't running and you have your model. That isn't going to work here, because you have to wait forever. So, you want to have a model that gets better and better as time goes by but, at any given point, you can push the button and see what you already have, given the data that you already looked at.

Another very important thing is the following. We would like to ensure the results that we get are, impossible, equivalent to what you would get if you were actually just running your regular algorithm on a regular, but infinite-sized database with a computer with infinite resources. It is very easy to satisfy those requirements if you compromise the quality of the results that you are producing. The whole challenge is to actually guarantee those things while producing, say, decision trees or things that are not different from the ones that you get if you ran the algorithms that we know, and that we have, and whose properties we know.

Finally, we also want to be able to handle time changing phenomena. In a typical database, we just assume that the data is IID, so it doesn't matter what order they are in. In large data streams that last over months or years, very often, the phenomenon that you are looking at is actually changing over time. So, the model that you ran a year ago isn't necessarily valid now, and you also want to be able to deal with that.

So, these are the criteria that we want to satisfy. In fact, when we started doing this work a couple of years ago, it sort of sounded overly ambitious. Where are we going to have to compromise? However, to our amazement, we found that we were actually able to satisfy all those criteria. In fact, what we now have is, we have a framework that meets all those criteria, and we have successfully applied it to several algorithms, including decision tree learning, network learning, image clustering.

Current Status

- Developed framework that satisfies all desiderata (Hulten & Domingos, KDD-2002)
- Successfully applied to
 - Decision tree learning (Domingos & Hulten, KDD-2000; Hulten et al., KDD-2001)
 - Bayesian network learning (Hulten & Domingos, KDD-2002)
 - K-means clustering (Domingos & Hulten, ICDM-2001)
 - EM for mixtures of Gaussians (Domingos & Hulten, NIPS-2001)
- Mines billions of examples per day
- Currently developing general-purpose library

We are able, using this, to deal with millions of examples a day using existing hardware. For example, the best decision trees we were able to mine perhaps a million examples per day, and now we are able to mine on the order of a billion examples per day.

What we are doing now is, we are developing the general purpose library such that, if you write your algorithms using this library, you won't have to worry about scalability. You can just write them the same way as you always wrote them, and they will run on data streams without your having to focus on it. You just focus on the problem that you are trying to solve. So, this is not available yet, but it is like our next target that we are dealing with. So, I am not going to have time in this half-hour to actually talk about how we meet each of those problems. I will just focus on the most salient ones. Those are the problems with trying to ensure that you learn the same model, in real time, as quickly as possible, that you would run on infinite data, and a little bit at the end about what happens when the data-generating process is not sufficient.

The Framework

Key Questions

- How much data is enough?
- What if the data-generating process is not stationary?

How Much Data is Enough?

Hoeffding bounds:

- Real-valued random variable x with range R
- n independent observations, mean \bar{x}
- With prob. $1 - \delta$, true mean of x is within ϵ of \bar{x}

$$\epsilon = \sqrt{\frac{R^2 \ln(2/\delta)}{2n}}$$

Basic idea: Bootstrap this to whole complex models

What do I mean by discretion and how much data is enough? In a traditional learning algorithm, what you do is, the algorithm is a sequence of steps of some kind, and each of these steps look like the data in some way. Think of your favorite algorithm and this is how they work. In the traditional setting, each of these steps, in general, looks at the whole data before doing what it wants to do. Now, in a data stream setting, that is not going to work because you would never get past the first step, because you would wait forever for all your data to arrive.

So, what you need to do is, you need to make the decision about how much data to use in each step. In essence, you want to use as little data as possible in each step, so that your model gets better as quickly as possible. I want to have the best possible model as soon as possible, and this means minimizing the amount of data that I am using at each step.

Now, if I reduce the amount that I use at each step, I probably will pay a price for that in terms of the quality of the model that I get. So, the goal of what we are doing here, in a sense, one of the basic features of our framework, it enables you to minimize the amount of time that it takes to learn on the data stream, while guaranteeing that you are still getting the same result that you would get if you waited for all the data up to eternity to arrive.

Some of you may be wondering, well, how can that be possible? To persuade you that that can be possible, I would like to give you the basic idea of what we are doing. Think about the simplest possible model that anybody would want to build. Say you just want to know what the average of some quantity is. What is the number—for example, in the network application, I want to know what the average number of packets is that you get from some source, or whatever you need to find. Suppose that I have, say, a billion samples of this. I don't necessarily need to look at that billion, if I assume that they are IID, to form a good estimate of the mean. This is what pollsters do when they use 3,000 people as a sample for, say, 200 million. I mean, there are many kinds of results that you can use to decide how much data you need. One that we have used a lot is a hooking balance, and what happens in a hooking balance is this. Suppose we use variable X and range R . Then we have N independent observations.

What this guarantees is that, with a probability of $1 - \delta$, the variable is within ϵ of X , where ϵ is given by this expression. It depends on the range. It goes down with the number of data points that you have, and it actually only goes up with the log of one over the area. So, what this formula does is that, suppose you want to find the mean of some

variable, but you are willing to have that mean be off by ϵ , by at most ϵ , with the probability at most δ . Then, I will tell you how many data points you need to gather. Then, after that, you don't have to worry. It doesn't matter if you have a trillion points or infinite points. You don't need to look at them, because you already have the quantity that you want with the tolerance that you want. So, the ϵ and δ are the quality parameters that you have given.

In essence, all we are doing, at least in the basis of our framework, all that we are doing is this. We are bootstrapping this idea to not just one quantity, but to a whole model, for example, a whole density estimation model with all of its parameters, or a whole decision tree with all of its nodes.

So, we take these kinds of things for the individual things that you are doing and, by doing some analysis, you actually come up with a guarantee on the quality of the whole model.

So, in summarizing one slide, here is the approach that we have. It goes in three steps. The first step is that you have an upper bound on the time complexity of your algorithm, on how long it takes to run, as a function of the number of examples that you use in each step.

General Approach

1. Derive upper bound on time complexity, as function of #examples used in each step
2. Derive upper bound on loss between finite-data and infinite-data models, as function of #examples
3. Minimize time bound subject to limits on loss

Effectively mines infinite data in finite time.

In many cases, it is just the sum of the number of examples that you use in each step, multiplied by something that is a constant. This is actually quite easy to do. So, we figure out how the running time of the algorithm varies with the number of examples that you use at each step.

Part two, we divide the upper bound on the loss between the model you get with finite data and the model that you would get with infinite data, as a function of the number of examples that you use when you are doing this with finite data. So, you give me your loss function, the loss function that you are going to use to compare to different models. So, if two models are very different by your loss function, then you know your loss function should have a large value. This could be easier or harder, depending on the algorithm of lost function. We figure out how this loss varies when you are comparing what you would get with infinite data, with what you are getting using a finite number of examples.

The final step is to minimize this. So, to minimize this, subject to the user, given constraints on this, meaning that we have a constrained optimization problem. What we are trying to do is to minimize the running time of the algorithm as a function of the number of examples we can expect, subject to the constraint that the loss that you get at

the end of the day, the loss between finite and infinite data, is at the most ϵ , with the probability of at least $1-L$. So, this is the general approach that we have.

Example: Decision Tree Induction

Time: Proportional to #examples used to pick each test

Loss: Probability that decision trees disagree on example

Solution: Use min. #examples at each step that guarantees loss bound

One of the surprises that we found is that we are actually able to do this for very broad classes of algorithm. We studied industry decision trees, and then we did it for clustering, and generally we saw that we could generalize this to many different things.

Notice that, in some sense, what we are effectively doing with this is learning from infinite data in finite time, because we are getting, in finite time, the same results with the same tolerance that you would get if you waited forever, until your data stream gave you infinite data.

So, that is our general idea. There are lots of different methods that I am not talking about here, but that is the basic idea of our framework. Let's see how it applies to the case of decision tree induction, which is actually a particularly simple one, perhaps in some ways not the most interesting one, but it is the first one that we did, and the easiest one to talk about.

So, in decision tree induction, what happens with the time? Remember, the first step was to figure out how the time varies within the examples that are used in each step.

In the decision tree induction, which I assume most people here are familiar with, what happens is that you have a series of nodes. Each node tests the value of some attribute. Depending on that value, it sends it to another node, and at the least you have a classification.

The basic problem in decision tree induction is deciding which tests, which attributes you pick to test in each node.

The amount of time that we need to do that is proportional to the number of examples that you have. What you basically need to do is, you need to gather some sufficient statistics that have to do with how often each value of each attribute goes with each class.

AUDIENCE: Is the number of nodes fixed?

MR. DOMINGOS: No, the number of nodes is not fixed. So, the first thing we are going to have to do with our algorithm is that potentially, as time goes to infinity, the size of your tree could grow infinitely as well. In practice, it usually doesn't, but in principle, it could. So, when you have a model that is of unbounded size, potentially, the model takes an infinite amount of time to learn.

What we are concerned with is what is the time that each step takes, and we are going to try to minimize, in this case, the sum of those times, which means that you run

the model faster.

So, notice, traditionally, and even in some of the very fast optimized algorithms for learning decision trees on very large databases, they use all the data to take each test. If you unleash those algorithms on a data stream, in some sense, they never even pick, because they keep waiting for the rest of the data to come up, before they make their first decision. Intuitively, you know, we should be able to make that decision after we see so many examples, and then go on to the next node, and that is exactly what we did.

So, what about the lost function? How are we going to compare the decision tree we are building with finite data with the decision tree that we would build with infinite data. You can do this in many ways. We are actually going to use a very stringent criteria which is, I am going to take the loss as being the probability that the decision trees disagree on the random example. So, this is much more demanding than just we find that the tree be at least nearly as accurate as the other tree, or that they make nearly the same decisions.

For the two trees to be an example, I am going to require that, from the point of view of that example, the two trees are indistinguishable, meaning the examples see the exact same sequence of tests, and winds up with the same class prediction at the end of the day. So, this is going to be my loss function.

AUDIENCE: It doesn't seem that the tests matter, if it gets to the same answer.

MR. DOMINGOS: Maybe it does, maybe it doesn't. For some applications, it matters. The point is that, once you get guarantees on this loss function, you get guarantees on all the others, since we can do this at no extra cost.

So, the final step, of course, is to try to minimize the limits on this loss. Without going over a lot of details, what is going to happen here is, what we are going to do, we are going to use the minimum number of examples to fit each test at each node such that, at the end of the day, the probability of disagreement is going to be bounded, and we are going to see the algorithm, and then we are going to see one example of the kinds of guarantees that we can get through the algorithm.

So, here is our algorithm for mining massive data streams. We call it DRPT. Again, at a very high level, it has two arguments. The stream, which is an infinite sequence of examples, and δ is the limits that we want to impose on the disagreements between the two trees. We want to learn on that stream as fast as we can, subject to the concerns that the disagreement between that tree and what we would have with infinite data is, at most, δ .

I beg your pardon, this δ is actually not the disagreement between the whole trees. We are going to call that Δ . This is the probability of getting each test wrong, and we are going to see how the two relate. It is the probability of actually making the wrong decision on any given step.

```
Procedure VFDT(Stream,  $\delta$ )  
  
Let HT = Tree with single leaf (root).  
Initialize counts  $n_{ijk}$  at root.  
For each example (x, y) in Stream  
  Sort (x, y) to leaf using HT.  
  Update counts  $n_{ijk}$  at leaf.  
  Compute Gain for each attribute.  
  If Gain(Best attr.) - Gain(2nd best) >  $\epsilon$ , then  
    Split leaf on best attribute.  
    For each branch  
      Start new leaf and initialize counts.  
Return HT.
```

So, how does our algorithm work? We start with a tree that just has the roots. We initialize to suggest the count of the number of times that each value generated with each attribute I occurs with each class, K .

Now, for each example, X is the attributes and Y is the class. We sort that example through a leaf. We are going to have a partial tree at any given point. Then we update the concept technique. Then, the information gain on a linear or any other matters, we compute it for each attribute. Then, if the data of the best attribute is better than the data of the second attribute by at least epsilon, when epsilon is a function of this δ , then at this point we know that, with probability $1-\delta$, we are picking the same attribute that we would be picking if we waited until we saw infinite data. So, at that point, we just split on that leaf, and now we start sending the new examples in the stream down to the children of that leaf.

So, we start with the root. After some of our examples, we go through the roots to the leaves, and then the examples get routed to its children, and then we collect examples of those guys and at some point we pick the test data, and we keep building it in this way.

So, notice, if you think about it for a second, this satisfies everything that we were talking about. It does require an amount of memory that is independent of a number of examples that you have seen and so forth. It is any time, because at any time you have a partial tree built and so forth. We have a few questions here.

AUDIENCE: [Question off microphone.]

MR. DOMINGOS: No, we deal with continuous inputs. In the particular network we are talking about here, we actually— [off microphone.] Again, we can deal with continuous attributes.

AUDIENCE: [Comments off microphone] —at many levels?

MR. DOMINGOS: Yes. That is actually not a problem with the levels that you have in the attributes. You just have more statistics to start with.

AUDIENCE: It appears to be a greedy tree-building algorithm. From that standpoint, it would perhaps lead to an inferior tree when compared to one that is pruned back with something else.

MR. DOMINGOS: The basic algorithm that we are trying to use isn't really a decision tree algorithm. So, we are doing the same thing. [Comments off microphone.] You can stream data forward. So, this is the basic algorithm. The main claim that I made to you is that, running a decision tree in this way, in some sense, you will learn it in as little time as you can.

What you will get is a tree that disagrees very little with the tree that you get on

infinite data. Well, how come I can make that claim? There are actually various kinds of results that we can show here. The one that I am going to show you is select a better bound. To get that better bound, we have to make one assumption about the data, which is almost always true, and that is why I am presenting it here.

We can also get results without making that assumption. The assumption I am going to make is that some fraction of your data, at any given level of the tree, winds up in a leaf. Again, in all the dozens of domains that I have applied decision trees, I have never seen one where this doesn't happen. What this means is that the tree is somewhat unbalanced. You don't have all your leaves at the last level. If that happens, we will have to use a different kind of guarantee than the one I am going to give here.

Again, for purposes of analysis, I am just going to assume that that probability is the same at every level, which it doesn't need to be, and I am going to call it P , the leaf probability.

Guarantees on VFDT's Performance

Disagreement between two decision trees:
 $\Delta(DT_1, DT_2) = P_x[\text{Path}_1(x) \neq \text{Path}_2(x)]$

Let

HT_δ VFDT tree given δ , infinite sequence of exs.
 DT Batch DT given infinite exs.
 p Leaf probability

Theorem: $E[\Delta(H T_\delta, D T)] \leq \delta/p$

You will recall that in our last measure there was some disagreement between some trees, and that I define as the probability, that the path of the example through the first tree differs from the path that the example takes in the second.

The two trees that I am going to compare is the decision tree that I learned using the algorithm that I just saw, with parameter δ , and the decision tree that I would learn in batch mode that I would learn using an ordinary decision tree algorithm, with infinite memory and infinite time, waiting to see infinite samples.

Here is what we can guarantee. We can guarantee that the expected disagreement between the two trees is bound by δ/p . So, the δ number is the value that we use at each node. I guess what this means is, if I get a smaller δ , call like this δ the one that you guarantee. You know the leaf probability. Then immediately you know which lowercase δ you need to use at each local decision to get the result you want.

Obviously, the lower δ , the smaller δ gets, the lower the leaf gets. Not surprisingly, the lower the leaf probability, the bigger the tree you are going to grow. So, the smaller it has to be to get to the big δ .

AUDIENCE: It is noteworthy that HT_δ is also on an infinite sequence.

MR. DOMINGOS: I will get to that in a second. Actually, I will get to that right now. One correlate of this theorem is that, on finite data, what our tree is going to be is a subtree. That is the best you could ever hope for, again, with the same probabilistic guarantee. Then, it is like a two-step process to see how that correlator comes about.

Without actually proving the theorem here, it is actually quite easy to give you

some idea of the basic intuition as to why we are able to have a guarantee like that.

Theorem: $E[\Delta(HT_\delta, DT_\infty)] \leq \delta/p$

Why?
Prob(paths differ) $\leq \delta \times$ Path length
Prob(ex. falls in leaf at level δ) $= (1-p)^{\delta-1}p$
 $E[\Delta(HT_\delta, DT_\infty)] \leq \sum_{i=1}^{\infty} (\delta i)(1-p)^{i-1}p = \delta/p$

Corollary:
Finite HT differs from some subtree of DT_∞ by at most δ/p

Bottom line:
Near-perfect agreement even with few examples
E.g.: $p = 1\%$, 725 exa/node \Rightarrow Disagreement $\leq 0.01\%$
Linear increase in #examples
 \Rightarrow Exponential decrease in disagreement

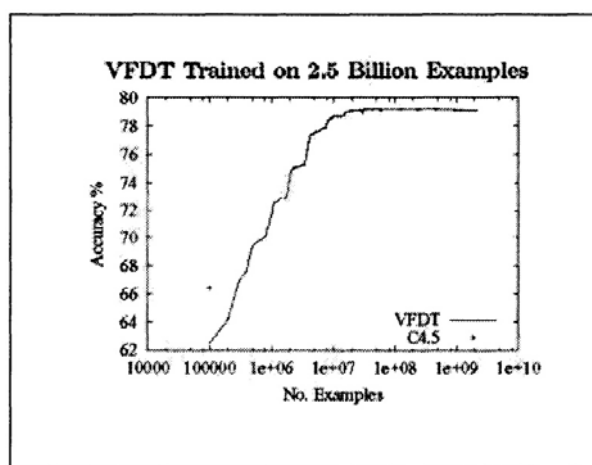
The reason is actually just the following. Notice that if you have an example going down a certain path in the two trees, and the probability of that example bumping into another one that is different is bounded by δ because we designed it that way.

So, if the path is of a certain length, L , then by the union of the probability that the data disagrees anywhere in the path is, at most, δ times L . This is, in fact, a loose bound. So, this is the problem when you get two paths of L data. Now, the other side of this is, if we have a leaf probability of P , then the probability that a random example falls into a leaf at level L is the probability that it doesn't fall into a leaf at any of the P levels, so it is $(1-p)^{L-1}p$, which is the probability that it falls into a leaf at this level.

Although we need to consider computer expected disagreement, this is just the sum of the probability of these two guys over all possible levels. The expected disagreement is the sum of all possible levels of the probability that the example falls into that level, and the expected disagreements for that level.

So, if you take this expression and you sum it, you just come out with the δ that we have. Like I said, there is this nice corollary that applies to, in finite time, basically what you get is a subtree of the decision tree that you would get with infinite data, again, with at most this disagreement.

The bottom line here, though, is the following, that people in competition line theory have a lot of bounds that have this flavor, but they are incredibly loose. They are nice for intuition but, from a practical point of view, they are useless.



The results that we are getting here is very different, because we are not looking at the size of the models to consider. We are actually only looking at the concrete number of decisions that you make. The really nice thing, is that the number of examples that you need actually only goes up logarithmically with a δ that you want to assure.

Putting it another way, as the number of examples grows linearly with time as you see more of the stream, your guarantees get exponentially better. The fact that your guarantees get exponentially better means that, with realistic numbers of examples, we can get disagreements that are extremely low. So, this is just an example to make things easy.

Let's say that your leaf probability is 1 percent, which basically means that your tree is huge, because only 1 percent of your examples runs into a leaf at any given level. So, this is a very large tree. Let's suppose that you have 725 examples per node. It is not a very large number, if you think of the data stream of millions of numbers coming at you. This is enough to guarantee disagreement of less than 1 in 10,000, .01 percent.

So, here is one graph of results out of the many that we have produced, just to give you a flavor of what we are able to do. This is the FPT framework, 2.5 billion examples. Using an ordinary PC, it took us a couple of days to do this. In fact, the time to learn is completely dwarfed by the time just required to read the data from this.

So, the best application of this algorithm is actually not unlike the kind of stuff that people have been talking about all day today, where you never even study an example. The algorithm actually takes an order of magnitude less time to run than it actually takes to just read them from this.

So, we see a standard decision-tree-running algorithm, and this is it running on the maximum number of examples that we could store in memory, which was 100,000. It had about 66 percent accuracy. Without going into details of the data set, it was composed of 100 attributes. So, this is a 100-dimensional data set.

What we see here is that the FDT, it is able to expecting more and more from the data right up to 100 million examples. Actually, between there and the 2.5 billion, we don't get anything, a slight amount of increase. You could get God knows how many, but the FDT wouldn't have any trouble with that. Again, this is just one result with a very large number. Let me just mention the issue of time changing data.

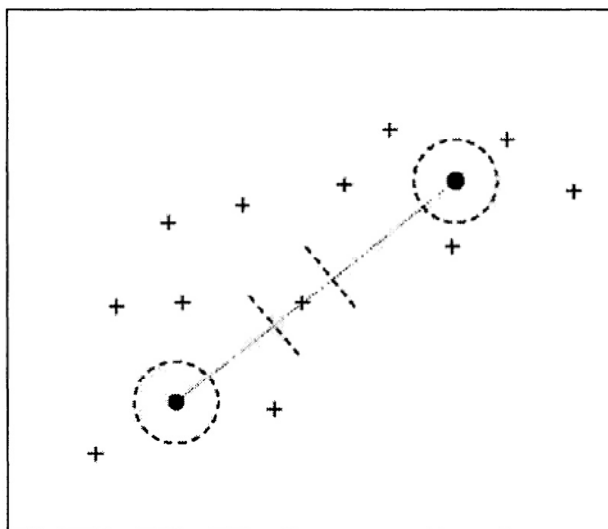
Everything I have talked about here assumes what those bounds assume. What those bounds assume is that the data are independent. In fact, the Hawking model isn't identically distributed, but it is significantly independent. What this means is that you data has to be generated by a stationary path, but then it has to be exchangeable. It can't be changing over time.

As long as that happens, in some sense, we don't need memory, because the stream itself is a memory, and that is kind of what makes our algorithms work, is that at some point you don't have to see more, because it doesn't contain any more information with respect to your model than you have already seen.

However, in a lot of applications of interest, what happens is that the data generating hypothesis is stationary. It is changing over time. So, what do we do in that case?

Example: *K*-Means Clustering

- Two sources of error in *k*-means:
 - Sampling error
 - Assignment error
- Compound these over successive iterations
- Final error is function of #examples used in each iteration
- Minimize these, subject to loss bound



The VFKM Algorithm

- Runtime quantities required for optimization
- Run *k*-means with min. #examples that could satisfy Hoeffding
- If Hoeffding satisfied, we're done
- If not, re-run with new min. #examples at each step

Time-Changing Data Streams

- Standard technique: Sliding window
- CVFDT: VFDT + Example forgetting
- Old examples subtracted from sufficient stats
- If new best split, grow alternative subtree
- When new subtree more accurate, switch it in
- Similar to windowing, but with constant cost per example, instead of $O(\text{window size})$
- Uses old information while still useful

Again, our framework handles this for very broad classes of algorithms. For an easy explanation, I am just now talking about how we do it in the context of decision trees. So, the standard technique to handle time changing data is to use a sliding window, or use some kind of weighted behavior examples.

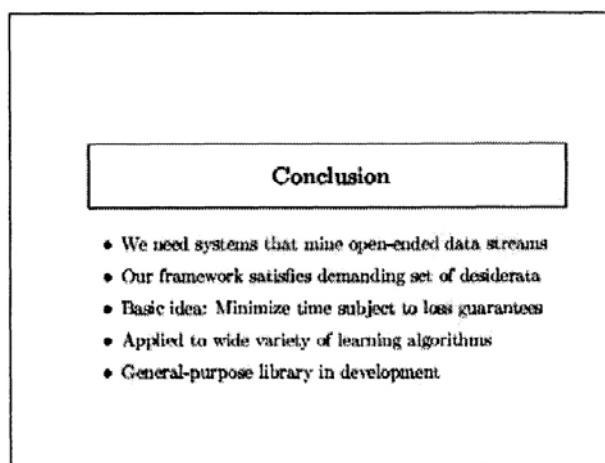
So, let's look at what is considered the sliding window case and generalizing it to the— [off microphone.] What you do is that, the model that you have at any given point is always the model run on— [off microphone.]

The problem we have is that, at the data rates that we are looking at, you don't want to have to start a sliding window. It would just be like that. Your sliding window might be a day. A day is like 100 million examples. We can't store that. Also, you don't want to relearn your model every time a new example comes up in the sliding window. However, what you can do in our framework, almost every learning algorithm under the sun basically runs by computing some sufficient statistic, the decision tree algorithm that we just saw. So, all that we need to do, in order to effectively run on the sliding window, is to be able to forget examples.

That means that, if we want to run on a sliding window, when a new example comes up, as well as adding that example to the sufficient statistics, we subtract the oldest example in the window from those statistics. If the data is stationary, then the new examples and the old ones are equivalent and nothing changes. However, if the data is not stationary, then what happens is that at some point, what looks like the best is no longer the best way. At that point, what we start doing is growing a new subtree, and we leave those two until the new one is doing better than the old one. So, we actually keep the old stuff around as long as it is still useful.

Suppose that the route suddenly becomes outdated. You don't want your performance to be bad because it is the entire tree. So, we let that tree stay there. We start growing a new tree and, when the new one becomes better than the old one, we switch it again. Again, I am glossing over a lot of details here, but this is the basic idea.

Let me just conclude. I will skip over some of the future work that we are thinking of doing. I hope I have convinced you—and you probably don't need convincing—that we need systems that mine open-ended streams of data.



What I described here is a framework that satisfies a very stringent set of algorithms that learn on massive data streams.

You know, the basic idea behind this framework is to minimize the amount of time that you need to grow a model, or you build a model on the data stream subject to guarantees that that model is going to be almost the same as you get with infinite data.

We have applied it to a wide variety of learning algorithms by now, and we have actually developed now in a library that hopefully we will make available in a few months or a year, that anybody who is interested in doing classification or clustering or regression estimation can use these primitives.

What we guarantee is that, as long as you access to data is encapsulated in the structures that we give you, it will scale to large data streams without your having to worry about it.

AUDIENCE: Could you share your thoughts on 4.5 versus—at 100,000 or 105. So, it is clear that you are ultimately going to win, and in the larger applications, that is the way to go clearly.

Is there a concern that you are not as efficient as sort of spitting out data than you would be— [off microphone.]

MR. DOMINGOS: Quite so. The reason it is doing better is because— [off microphone.]. The reason that basically C.45 does that is it is using each example multiple times, whereas we are only using each sample once.

So, you can modify algorithms by using each sample multiple times. So, what we are seeing here is the crudest version of this algorithm.

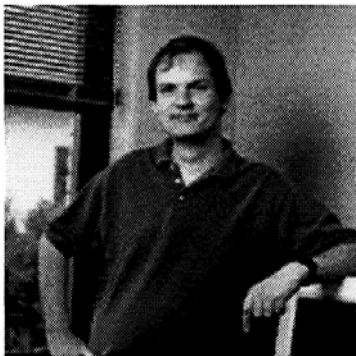
The other thing that we can do is, we can bootstrap our algorithm with C.45 running on the number of examples that we are picking at random, and then start for there.

C.45 actually makes lots of bad decisions. So, at the end of the day, we are going to offset the window. There is more stuff that I can tell you about it.

Andrew Moore

kd- R- Ball- and Ad- Trees: Scalable Massive Science Data Analysis

[Transcript of Presentation and PDF Slides](#)

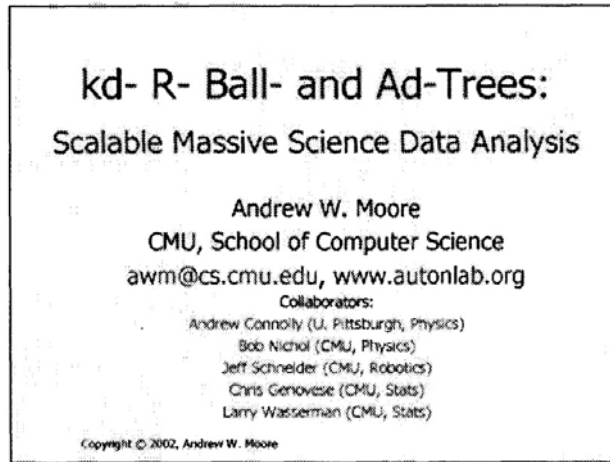


BIOSKETCH: Andrew Moore is the A. Nico Habermann Professor of Robotics and Computer Science at the School of Computer Science, Carnegie Mellon University. His main research interest is data mining, using algorithms for finding all the potentially useful and statistically meaningful patterns in massive sources of data. He is most interested in learning graphical models efficiently, probabilistic models of person-person interactions, spatio-temporal algorithms for biosurveillance, active learning, new kinds of searches for interesting interactions between variables, and any kind of spatial data structure for caching sufficient statistics.

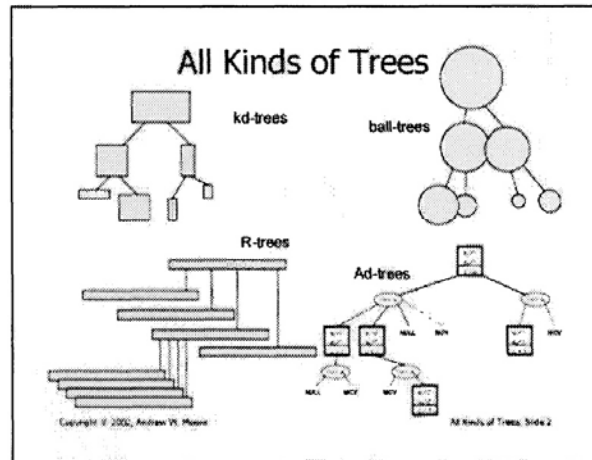
Dr. Moore began his career writing video games for an obscure British personal computer. He rapidly became a thousandaire and retired to academia, where he received a PhD from the University of Cambridge in 1991. He researched robot learning as a postdoc working with Chris Atkeson and then moved to a faculty position at Carnegie Mellon.

Dr. Moore's research group, [The Auton Lab](#), works with astrophysicists, biologists, marketing groups, bioinformaticists, manufacturers, and chemical engineers and is funded from industry and research grants from the National Science Foundation, NASA, and, more recently, the Defense Advanced Research Projects Agency. His research applications are in biosurveillance (he is part of a project led by [Mike Wagner](#) of the University of Pittsburgh) and intelligence analysis. Dr. Moore collaborates closely with [Jeff Schneider](#) and [Chris Atkeson](#), who was his postdoctoral advisor at MIT.

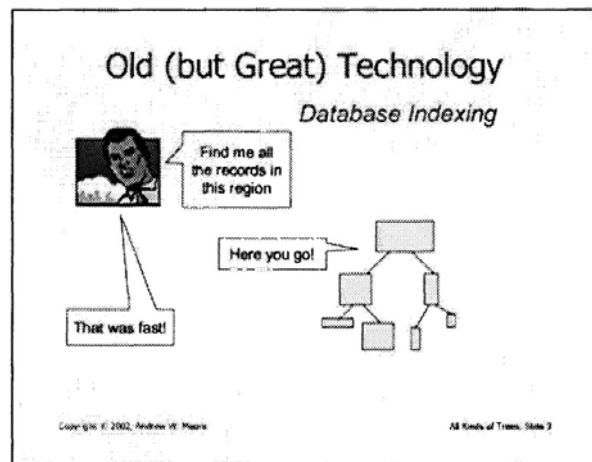
TRANSCRIPT OF PRESENTATION



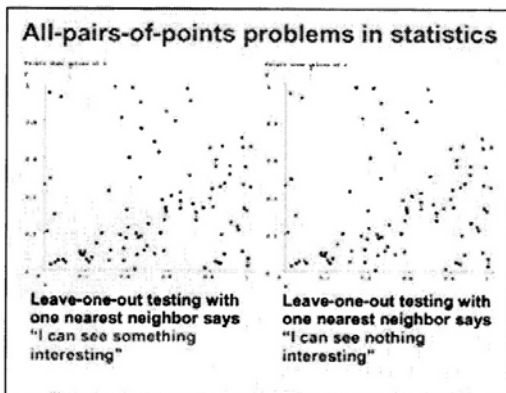
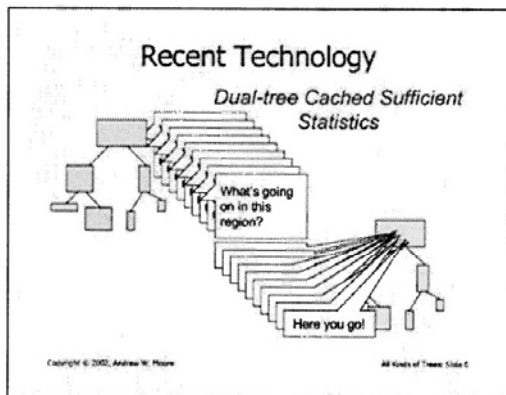
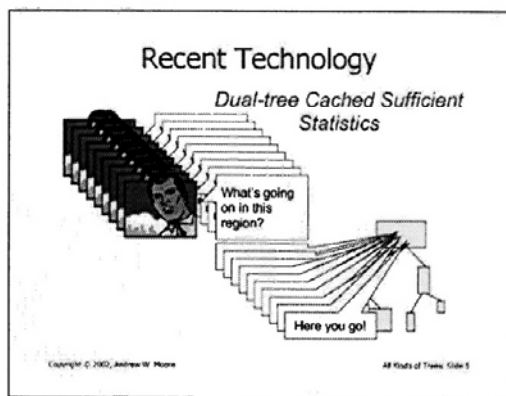
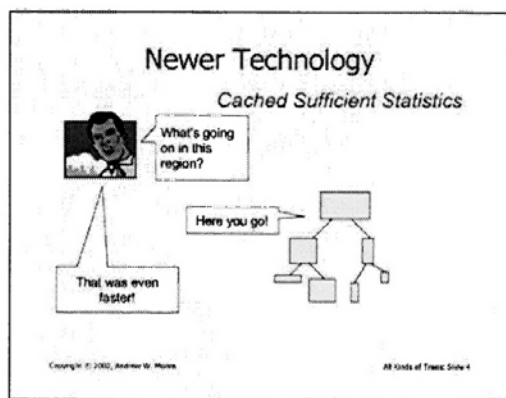
MR. MOORE: This is Joe Monk with a couple of physicists, Bob Nichol and Andy Connolly, and also with my colleague Jeff Schneider, another computer science, and two statisticians from Carnegie Mellon, Chris Genovese and Larry Wasserman.



I am going to be talking about a whole bunch of data structures which are very promising for allowing us to do apparently rather expensive statistics on large amounts of data, as opposed to trying to speed up cheap statistics, trying to do some of the apparently very expensive algorithms.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

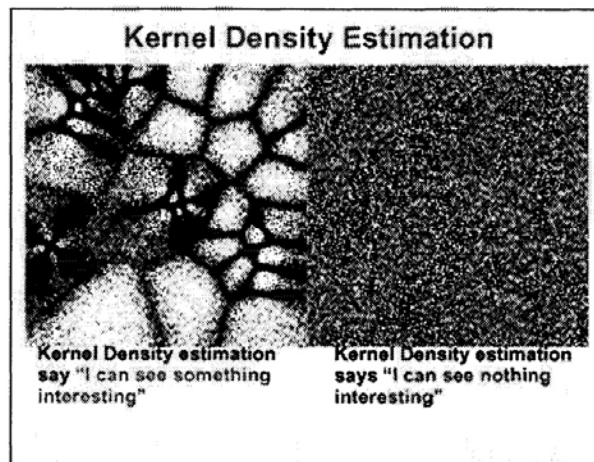
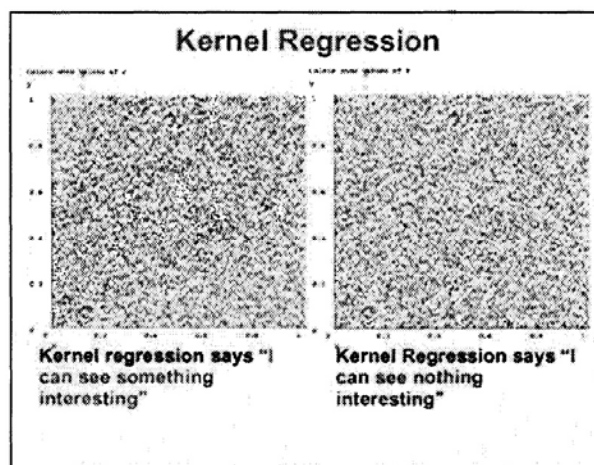
I am going to begin at this point. There are many pieces of statistics that want to do work of the sort of magnitude of looking at all pairs of records in a big database. In fact, there are plenty of them that need to look at all triples or all quadruples, and that is unpleasant, computationally.

There are also many pieces of statistics which need to look at all pairs of all variables, covariants in a database. Even building a covariance matrix requires you to do that, and in some cases, there are things that you need to look at all triples and all quadruples of covariances.

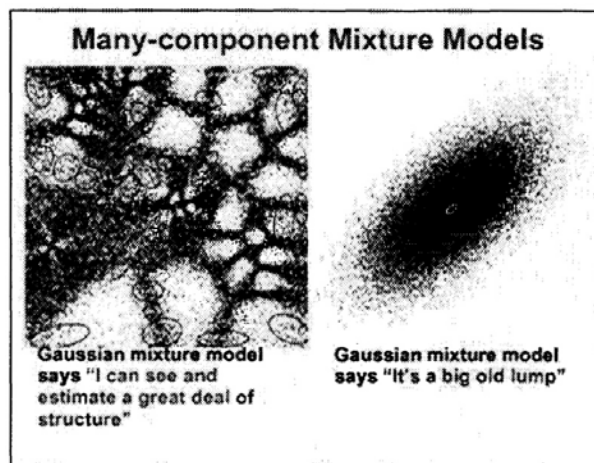
So, these things which scale quadratically or cubically with the amount of data over the number of attributes are very frightening.

I am going to show you some tricks from basically borne out computational geometry which allow us to survive that. So, let's just review a few operations which people need to do involving all pairs.

Here, if you are using something like a K nearest neighbor classifiers lots of times to make lots of classifications, you would end up doing something quadratic in the size of your database.

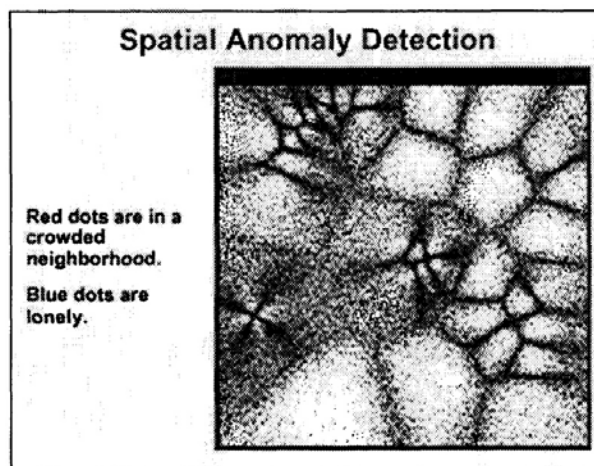


If you are trying to do kernel regression or, something more familiar, kernel density estimation, on a large amount of data, then if you implemented naively, then every one of your queries needs to ask questions of lots of other data points lying around it.

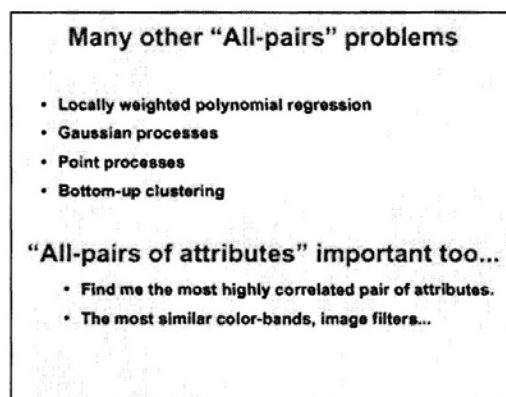


Another related example is if you are doing something like a Gaussian mixture model cluster, where you have a large number of records and a large number of clusters. Certainly, if you are using something like an AIC or a BIC type of a measure for creating your model, you would notice that the number of Gaussians that you created in your mixture model grows as the number of records grows. Again, you actually end up with a quadratic kind of problem there.

There are many other examples. I am happening to show you examples that we have needed to use in reality, but there are many examples that we haven't, ourselves, had to use so far.



Spatial anomaly detection, just finding out those, and highlighting those data points which are in dense regions of space.

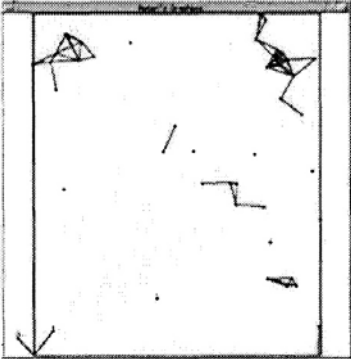


2-point correlation

...the purest form of an "all-pairs" problem.

There are 62 pairs of points that lie within 0.1 units of each other:

...important in astrophysics for characterizing matter distribution.



Now, I am going to focus here on a very, very conceptually simple statistic, which looks like it requires you to look at all pairs, and it is computing part of something called the two point spatial function of a distribution.

The computationally expensive part of it is the following. You end up wanting to ask the question, given a set of data, how many pairs of data points are within a certain distance of each other? So, how many pairs of data are close to each other? You have got a data set. All you are getting out of it in this case is one number. How many pairs are close to each other? The reason you need that is that there are many spatial statistics where knowing those kinds of numbers characterizes the clumpiness of the distribution for you.

Fast all-point-pairs: Idea One

Use an $O(n^2)$ algorithm and buy a fast computer

Problem: $O(n^2)$ is vicious.

So, let's look at how we can do this operation. You could just say, well, I am going to buy a fast computer and do this, but we just can't afford to do that. If you try to live this way, then one day you get 1,000 times as much data, your algorithm is going to run a million times more slowly.

Instead of preparing slides like a professional, I am just going to run a program to show you what we are going to do about this.

Here we have a very tiny little data set. I am actually imagining that I am part way through this process of counting all pairs of points which are within that distance—you see that little arrow up at the top—within that distance of each other, so within .4 units of each other.

I am actually showing you this part way through running the program. I am not at the beginning of the program or the end. I am partway through, whereas a subroutine, a

recursion, I happen to be in a situation where I am asking, how many pairs of data points have one data point in this left rectangle and one data point in this right rectangle, and are within distance .4 of each other.

Now, the way I am going to answer that is by breaking that single question into two smaller questions. I am going to pick one of these rectangles, and in this case I am going to pick the larger rectangle, the right one. I am going to break it into two, and I am going to ask that question about the top half of the right rectangle, working with the left one, and then later on I am going to ask it about the bottom half of the right rectangle working with the left one. So, let's see what happens.

It looks like I begin with the bottom one here, and then I recursively ask the same thing. Every time, I am going to choose the larger of the two rectangles, break it in half, do the first half first, and then later on recursively come back to the second half. So, I jump down there. Now I am asking the question, how many pairs of data points have one guy from here, one guy from here, and are within distance .4 of each other.

When I recurse again, I get to an interesting position here, and you can probably see what has happened. I can do a simple piece of geometry on these two squares and prove that it is impossible for any pair of data points to be within distance .4 of each other, because the shortest distance between these rectangles is greater than .4. So, I can save myself some work by not looking at any of those children.

AUDIENCE: If you were asked—it seems to me there is a value—if you use something different from the .4, .9, you might not be able to use that geometry; is that correct?

MR. MOORE: We are going to see an answer to that question just almost immediately. So, at this point, we back up and go somewhere else. We look at the other child. Instead of that one, we look at this one.

Now we come to the pruning. So, we just carry on with the algorithm. We jumped on to there, had to carry on. Now, here we get something that addresses your point. Here, we can do some different geometry, and we can also prove, without having to do anything expensive, that every pair of point where one point comes from here and one point comes from here, must be within distance .4 of each other.

So, now, instead of actually explicitly iterating all of those points, I can simply multiply the product of the number of points in this box with the number of points in this box, and adding that to my running count that I am accumulating. It is very simple. So, that is another kind of pruning that I can do. I can just carry on doing that now. I carry on running through the data table and you see an occasional pruning of something.

Just to give you an idea of what this means, I am going to run this problem now on a data set with 40,000 data points on it. You see it running there without the graphics, which means that it is faster. It still needs to do quite a lot of computations. Remember, the 40,000 data points is 1.6 billion pairs of data points to consider. In 11 seconds, we have got the answer. It turns out there are 296 million pairs of data points in that particular data set within distance .4 of each other.

| Comparative Results | | | | | |
|-------------------------|-----------------------|-------------------------|-----------------------|---------------------|-------------------|
| Non-approximate version | | | | | |
| Number of Points | Quadratic time (secs) | Single-tree time (secs) | Dual-tree time (secs) | Single Tree Speedup | Dual Tree Speedup |
| 10000 | 132 | 2.2 | 1.2 | 60 | 110 |
| 20000 | 528 | 4.8 | 2.8 | 110 | 189 |
| 50000 | 3300 | 11.8 | 7.0 | 280 | 471 |
| 150000 | 30899 | 37 | 20 | 835 | 1545 |
| 300000 | 123599 | 76 | 40 | 1626 | 3090 |

| Approximate version (20,000 datapoints on slower machine): | | | | | | | |
|--|-------|------|------|------|-----|-----|-----|
| ϵ | 0.001 | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 |
| secs | 37 | 30 | 30 | 18 | 10 | 10 | 0.3 |

It turns out that this takes about 10 minutes to run if you were to use the standard correcting algorithm. So, let's look back, for instance, if we have gone from 10 minutes to about 10 seconds. That is a 60-fold speed up, but I am not really excited about that. We have had been able to run this on data sets with 14 million records, and that would have taken over a year to run otherwise, and in our case, it took 3 hours.

So, that is one example of very simple geometry helping do something we wanted, and we didn't resort to any approximation there. We have got exactly the same algorithm that we wanted, the same count that we got from the naive method.

I won't show you how to do it now, but if you prefer, you can ask this kind of algorithm the following question. You can say, give me back a number that is within one-tenth of a percent of the correct number. If you tell it that you are going to allow it to make a certain error, if it will promise you that it will, at most, make that error, then this algorithm could be adapted to take advantage of that, and you usually get an extra magnitude or two of speed up.

AUDIENCE: That is not a probabilistic promise?

MR. MOORE: It is not a probabilistic promise. It is an exact hard bound promise. So, it doesn't matter how weird the distribution of data is. So, if we do that, we are getting desperate.

Sometimes it turns out we are running these things on data sets where we are doing lots and lots of randomization testing, so having to run this same operation a thousand times. That is when we get desperate and have to do those kinds of approximations.

Now, interestingly, it turns out that the reason people run two point functions is to test whether two spatial distributions have got the same flavor to them. For instance, if you have got a theory of the universe, you write a simulation of the universe, which produces a simulation distribution of galaxies, in order to ask the question, does the simulated distribution of galaxies have the same flavor, the same clumpiness, the same stringiness as the true distribution of galaxies that I have observed. That is the kind of place you actually run this algorithm.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

The n-point correlation function

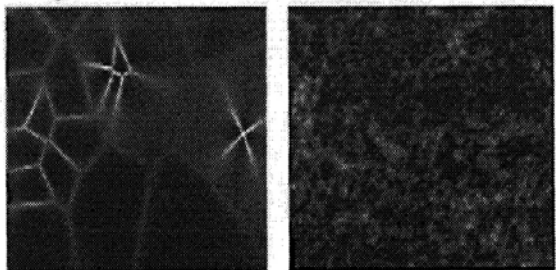
Examples:

- How many triplets of points are arranged in a nearly perfect equilateral triangle?
- How many quadruplets of points are arranged such that they consist of two pairs of close points with the point-pairs between 1 and 2 units apart?

These provide much more sensitive information than any 2-point function

Sometimes, doing it with pairs of data points isn't enough. If two distributions have got the same power in them, they will have the same two point statistics, no matter what. So, you are not able to distinguish between anything.

3-point function example (Connolly)



These two distributions have identical 2-point functions!

...but very different 3-point functions.

For that reason, people move up to three point functions where they ask questions like, how many triplets of data points were already distant .5 of each other.

That is something that previous physicists had only run out to about 1,000 data points and, even there, it would take them weeks of CPU time. They were planning on building a supercomputer to do this, until we demonstrated running it on a laptop in half an hour on a million data points.

AUDIENCE: For this particular application— [off microphone.]

MR. MOORE: It is true that sometimes, if you are comparing two point functions at two different distributions, you might just be asking the question, is one of them larger than the other one. When you do that, you get more pruning opportunities and you can go even faster. So, there is an additional opportunity for us to speed up there.

If you decide that you are not trying to answer the question, give me the counts, but instead, answer the question, find out if the counts are different, those are two slightly different questions, but it does give you the option to further speed up, and we do do that, mostly because physicists want to get the actual numbers out that we have experienced.

n-point search algorithm

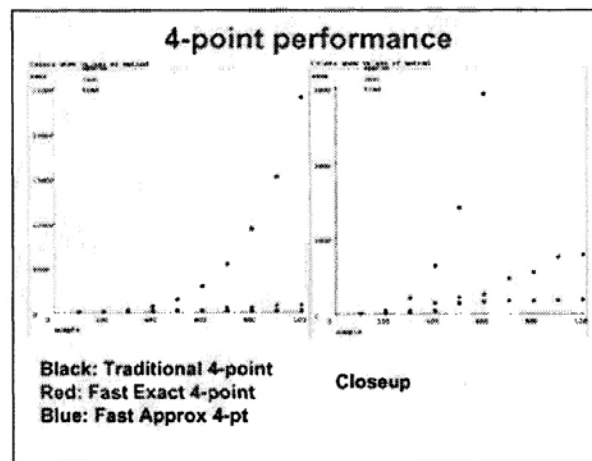
Instead of...
Searching over pairs of nodes to count how many pairs of points match a distance threshold

We now...
Search over n-tuples of nodes to count how many n-tuples of points match a geometry predicate.

Is it basically the same algorithm as for two-point code?

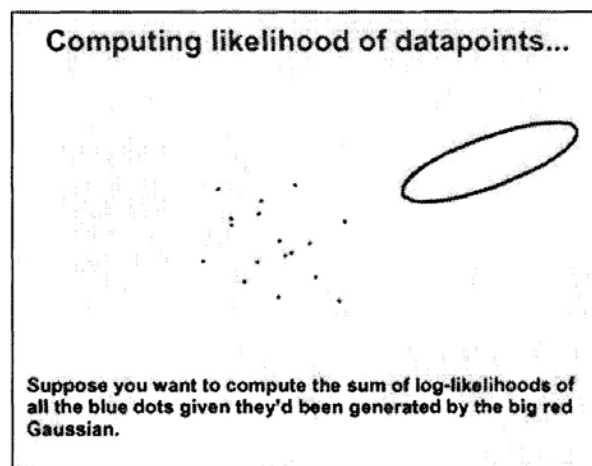
Not really...

- Anti-redundancy essential
- Predicate matching harder
- Should be very careful about how the search is expanded



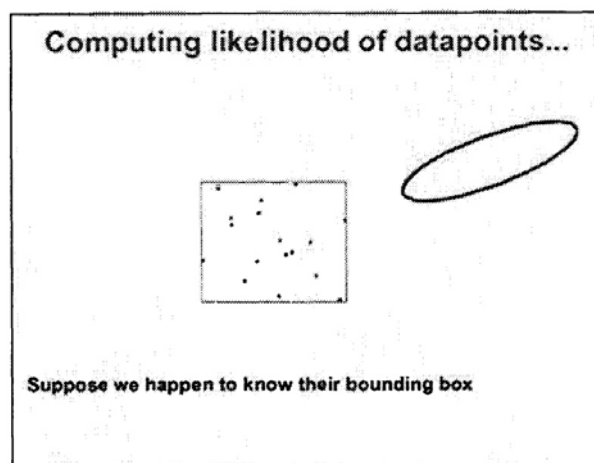
I am going to show you now how this generalizes. It seems like I was talking about a very specialized problem. There is a more general question.

Suppose you have this as a two dimensional Gaussian and I am just showing you the covariance matrix to give you an idea.

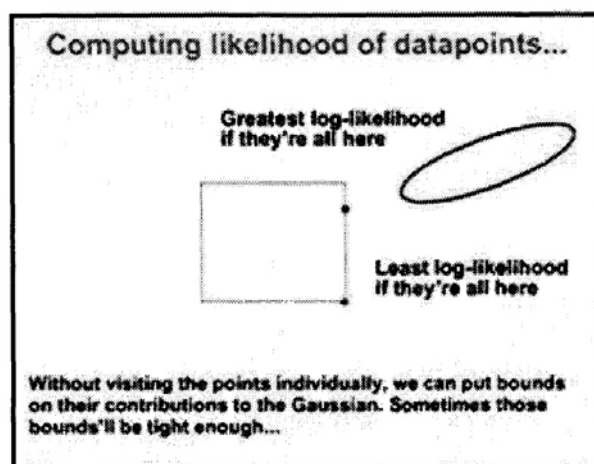


Suppose for some reason we want to compute the likelihood of all these data points. Obviously, these are rather surprising data points, if we say they are generated by that Gaussian, and I will go into the reason for that later.

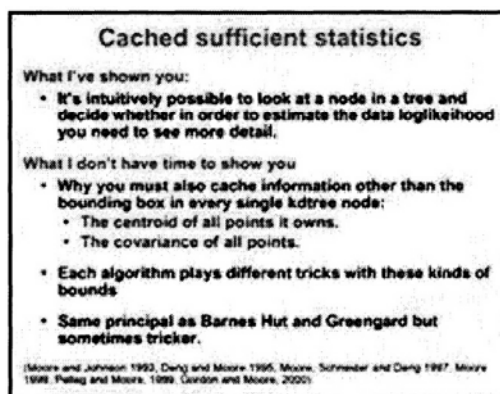
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



One thing we can do is, if we have this bounding box on our data points, then it is quite easy—it is not trivial, but it is quite easy—to compute what position in this box a data point would have to be to have the maximum possible load likelihood and where it would have to be to have the minimum load likelihood.

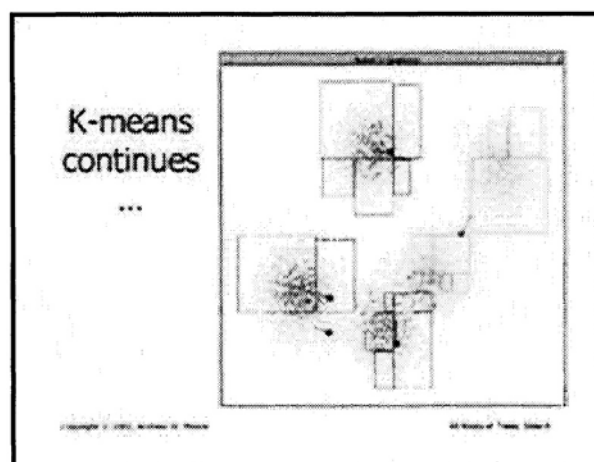


Having done that, if we know the bounding blocks, we know how many points are in the box, we can put bounds on the load likelihood of the points in that box.



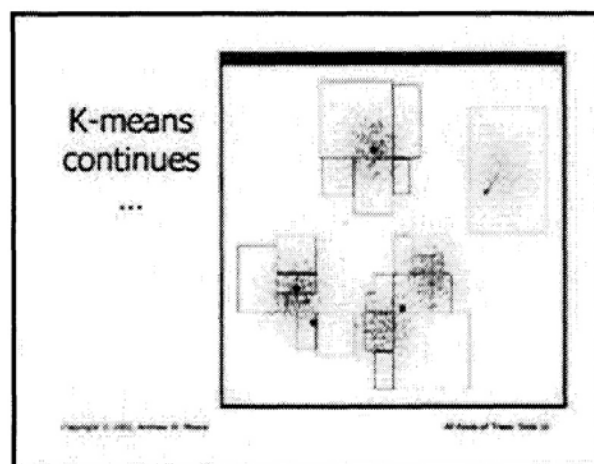
What you can then do, if you have got a whole room full of data and you want to find this load likelihood, you can play the same kind of game where you have a box that you keep splitting in half, looking at smaller and smaller regions. You stop doing that whenever you get down to a level where the box is virtually certain about what the load

likelihood is from that.



You can do a couple of extra tweaks on top of that to do some other things. So, we have now got software that will do these same kinds of tricks very quickly for kernel density estimations, LQH's, principal components. There are many algorithms in the kernel density literature which are down to the kernel sizes, and they are more complex. You can do various kinds of clustering and filament tracking and so forth.

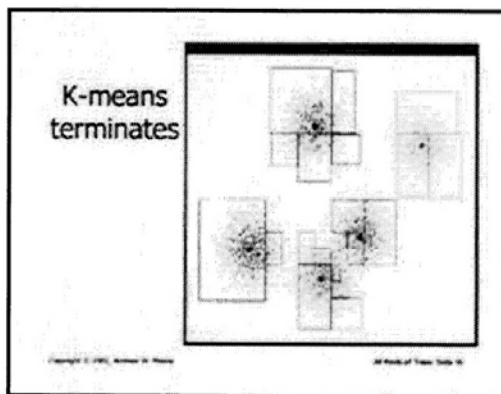
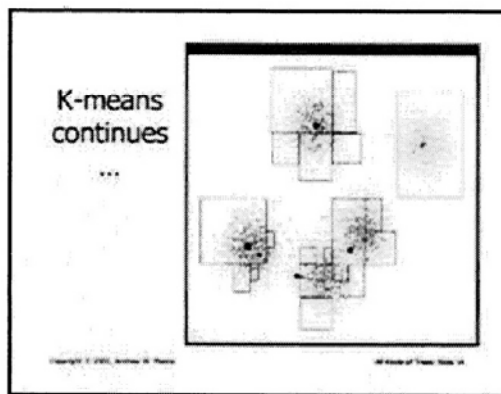
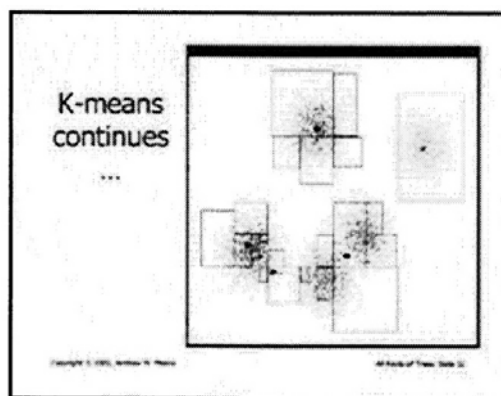
It has got to the stage now where it is boring for us to do more of these, but if anyone has any requests, we are always happy to do it.



I want to show you one of these. I have got a problem, which is I can't see on my screen what I am doing. Okay, we have a data set here. You can't see it very clearly. In fact, I am only drawing up a tiny sample of the data set. This is a data set with I think 50,000 data points in it. You can see these little blue dots lying around. These form the data set. I wish I could have shown you that more clearly, but I guess I can't.

So, 50,000 data points. We are going to run Gaussian mixture models on it with 15 centers. We see actually a tiny copy of the data point here. In fact, I think you have 20,000 data points here in the distribution. I have got a small copy of the data set just here, with the same distribution. It is very, very tiny embedded there.

I am going to run Gaussian mixture modeling and I am going to run 15 iterations. So, the thing I just wanted to show you is that, by using tricks like that, you can do things like that to your mixture models very quickly.



Fast Non-parametric Clustering

- Find areas of high density quickly
- Label areas of high density quickly
- Count areas of high density quickly

• See Weng-Keen Wong's talk on Saturday at 11:40 in Data Mining Session

Copyright © 2002, Andrew B. Rosenberg All Rights Reserved. Slide 25

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

A crazy idea

Why not precompute the answers to all possible questions in advance?
Then, when new questions come in we can immediately look up the answer.

Copyright © 2005, Andrew W. Moore. All Rights Reserved. Slide 20

AD-Tree v1.1

A mind-boggling alternate implementation is to store memory by storing NULL pointers for majority of nodes.

Copyright © 2005, Andrew W. Moore. All Rights Reserved. Slide 21

This is a crazy idea

With a tiny census dataset, would need 400 Terabytes of memory
With a medium-size pregnancy-tracking "Birth" dataset, would need... 10^{13} terabytes of memory

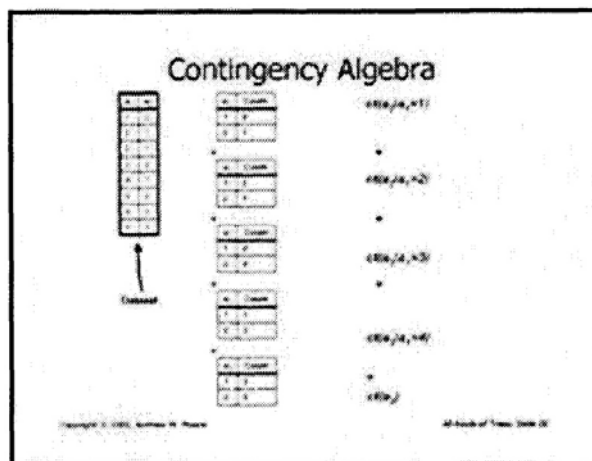
Copyright © 2005, Andrew W. Moore. All Rights Reserved. Slide 22

AD-Tree Version 2.0

Replace any address that is replaced by an MCY node with a single letter: MCY. Wonderfully, memorably/actively, you've lost nothing!!

Copyright © 2005, Andrew W. Moore. All Rights Reserved. Slide 23

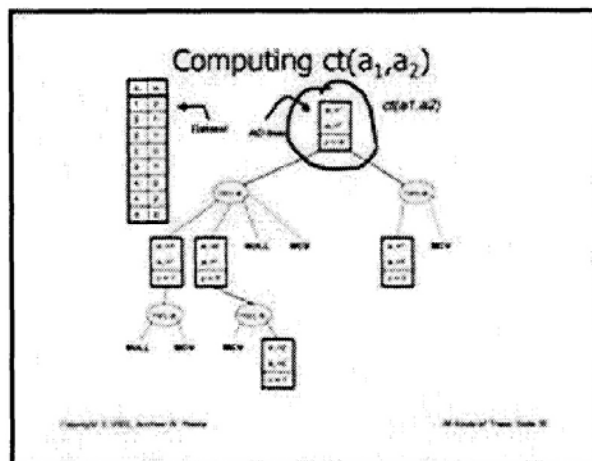
About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



Forty thousand data points, typically, with a typical implementation, you can expect each of those iterations to be somewhere between a few minutes and an hour. You can do 50 iterations in a few seconds. I will show you a slight generalization of that.

AUDIENCE: And that is the exact calculations or approximate?

MR. MOORE: That is exact.



Now I am going to show you something a little more heuristic, which is a version of the mixture models Gaussian built on this technology, which is also doing adaptive splitting and merging of Gaussians using various pieces of testing, while it is running.

It is probably going to run unnecessarily low, so I shall have to talk through this. The distribution which generates this table is actually not a mixture of Gaussian, but it was designed to make these three. That is CALD. That stands for the Center for Automated Learning and Discovery, which is a group at Carnegie Mellon University.

All it does, it doesn't do anything very clever to choose whether to split or merge Gaussian. All it does is, it temporarily splits them. Watch what happens to them. It does a BAC test to find out whether it was worth splitting the Gaussian, and then undoes the split.

It is slowing down a little bit now. It is up to about 50 Gaussians. Notice when I stopped it ended up with far fewer Gaussians that we saw it running with most of the times. That is because this was the best model according to the scoring.

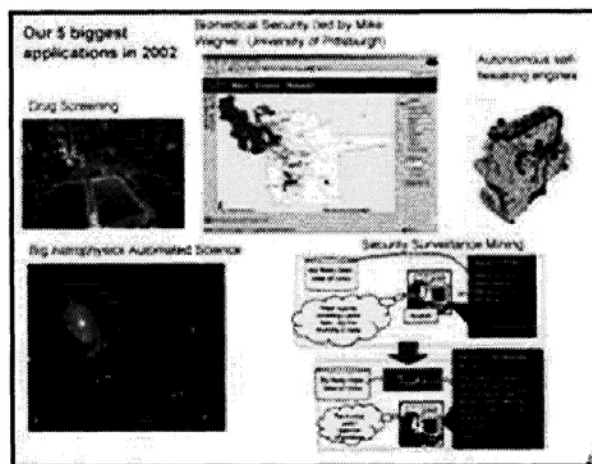
Now we are at the stage for least pass. This is an operation that had previously taken days and we can quickly do this now on the fly.

I should mention that we can go to higher dimensional data for this. We are not

you can answer by going into a database, getting a hold of appropriate records, getting some sufficient statistics from them, and then doing a computation.

Suppose that we know that we are not just about to be asked to do one of them, but we are going to be asked to do lots and lots of them. For example, you end up wanting to do this if you are building a Bayesian network. You end up wanting to do it if you are, for instance, using a very nice algorithm by DuMichelle and Pregibon, that they introduced a couple of years ago based on doing lots and lots of empirical Bayes tests on subsets of the data.

Those are just two examples. We have had about 10 examples of situations where you want to have a big database and ask billions of those kinds of questions. You want to ask them quickly.



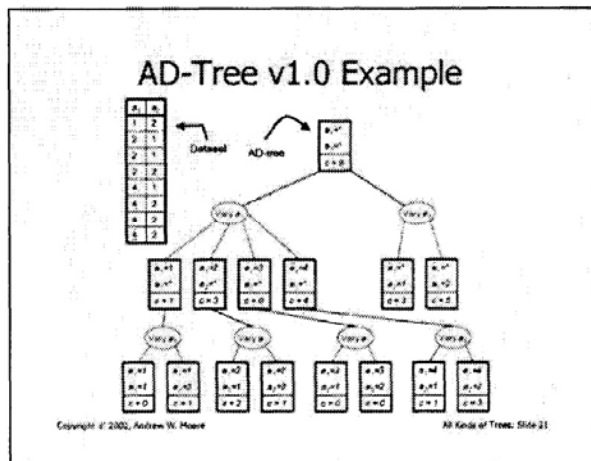
This is a crazy idea. In order to make this answer a question like that, why don't we build a secondary database where we pre-computed the complete sets of possible questions you could ask, and we pre-computed the answers for all of them.

The reason that is a good thing is, if we had that database, then I could say, I have a constant time algorithm, independent of the amount of data, for doing those kinds of statistical questions. The only problem with it is, a, it would take a huge amount of memory. In fact, on a typical database that we have been working with, we computed it would take about 10^{40} bytes of memory to contain this secondary database.

The other problem is, it would take a very long time to build the secondary database. The universe would have ended by the time we built it.

Although it is a crazy idea to do this, sometimes we can just push our luck and get away with it if we can very intensively compress the secondary database while we are building it.

AD is a way to compress these databases. I am going to try to give you a sketch in about three minutes as to how we did it.

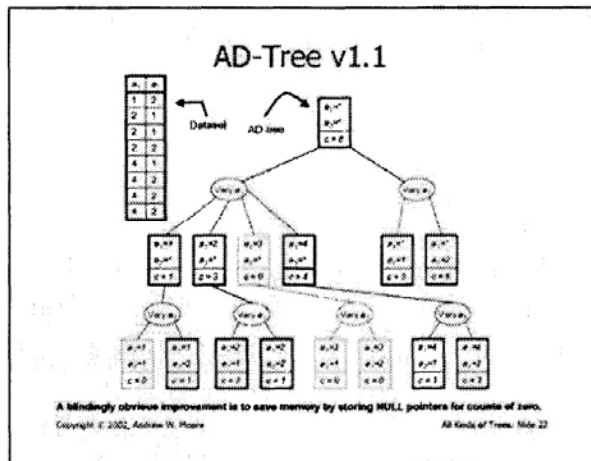


Copyright © 2002, Andrew W. Moore All Kinds of Trees, Slide 21

Here is a data structure which actually implements this impossibly expensive database I mentioned. It is a tree structure.

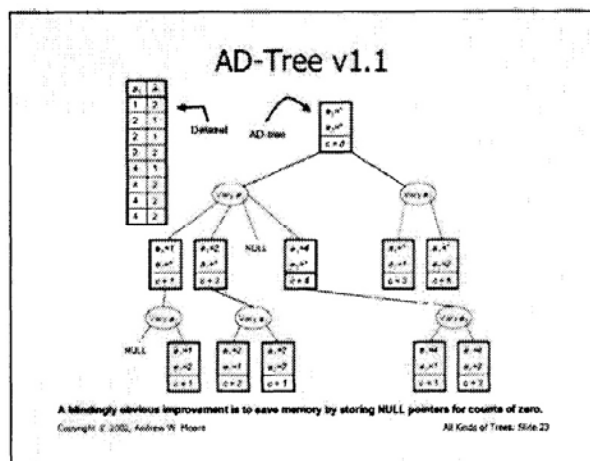
Every node in it contains a count—for instance, this node contains a count of four. What does that mean? This node is answering the question, how many records have actually been run in sector four and attribute two we don't care. So, it turns out there are four records of that form. How many records have four two, we end up looking at this class of data sets. a_1 equals four and the children of that are actually a_2 , because you come down here and you see that there are three such records.

So, that is a database in which we can access any counts we like. If you like, it is an implicit representation of a big hypercube combinatory index by all combinations of attributes. We can do it with two attributes, but even with just a mere 20 attributes, this thing would take up too much memory to store on your disk drive, let alone your memory. So, you can try to save memory while you build this data structure.



Copyright © 2002, Andrew W. Moore All Kinds of Trees, Slide 22

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



The first thing you can do, whenever you get a count of zero, you don't need to store that record and you don't need to store any of the children in that record. Any node here with a count of zero, any specializations of those queries, also count as zeroes.

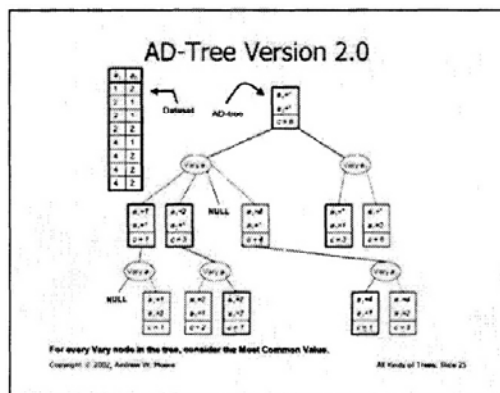
This is a crazy idea

With a tiny census dataset, would need 400 Terabytes of memory

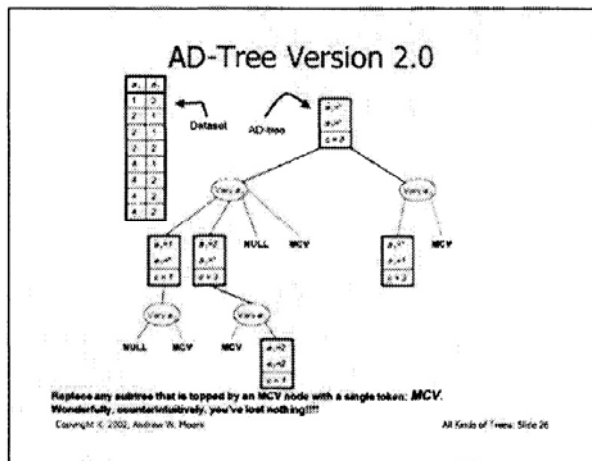
With a medium-size pregnancy-tracking "Birth" dataset, would need... 10^{13} terabytes of memory

Copyright © 2002, Andrew W. Moore All Kinds of Trees, Slide 21

So, that saved us a little bit of data, but not much. The example I described before, it went down from 10^{40} down to 10^{30} bytes of memory. So, although we have decreased our memory requirements 10 billion-fold, it doesn't help us much.



Now, here is something else at this thing. Any node on this thing puts a tail on considering all possible values of the attribute a_1 . I am going to mark in green the most common value of a_1 . Here, a_1 contains the value four, and that has the highest count. That happened four times. The others happened less often.



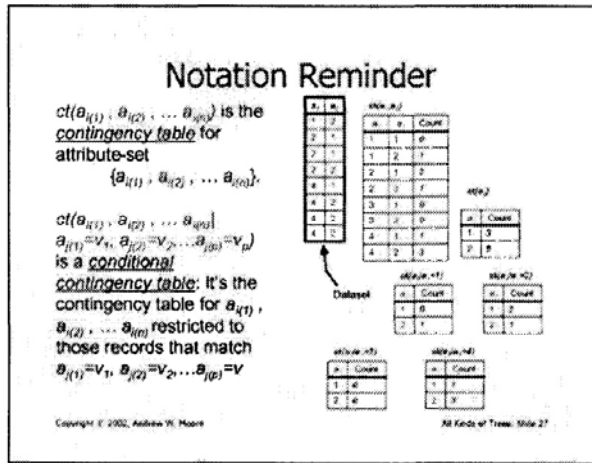
Everywhere on the tree, whenever I am instantiating a variable, I take the most common value of it and delineate that in green, and delete all of those from the database. All I am going to do is leave behind this little thing saying most common value. This was the most common value before I deleted it.

Now, there are two consequences of that. It turns out that you save a very, very large amount of data here. The reason is because it is always the most common value on all levels that you are getting rid of. So, you might get rid of, if you are lucky, 40 percent of all the nodes on this level. Then, of the remaining 60 percent, you get rid of 40 percent of those on this level. Of the remaining whatever it is, 20 percent, you get rid of 40 percent of those on the next level.

We can prove the amount of memory you need really goes down. In the example I was describing we then end up taking 10^6 bytes in this thing. It is much, much smaller. So, that helps us a lot.

The other good piece of news is that you haven't lost any information you really needed. You can reconstruct any counts that you had been able to get before through this prudent data structure.

I will give you an intuition of why that is true. I am going to try to do this in four minutes.



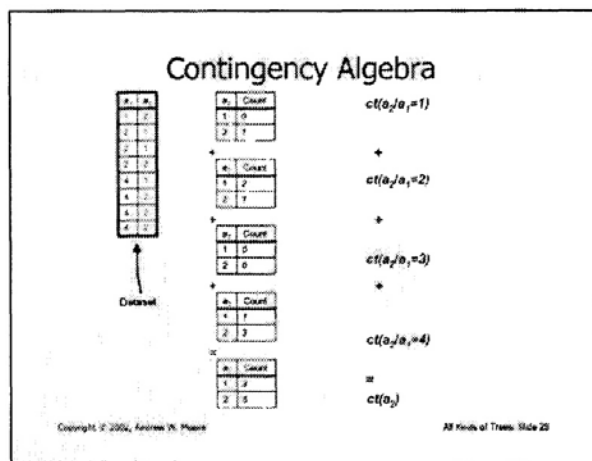
I will use this notation of a contingency table of a set of attributes. It is just this— well, you know, you are probably all familiar with what a contingency table is. In this data set, a contingency table at a_1 and a_2 is very similar to the distribution over a_1 and a_2 .

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

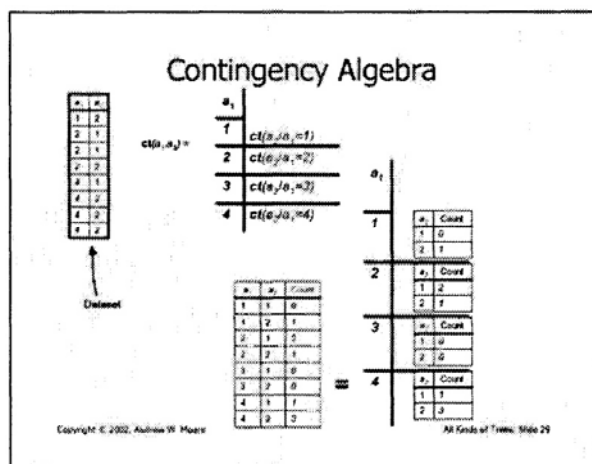
It is a table recording the fact, for instance, that 2, 2 occurs once in this database. There it is.

Here is the marginal contingency table over a_2 , just as there is a histogram over the values of a_2 .

This is a conditional contingency table. This is the conditional contingency table of a_2 among those records in which a_1 has the value two. So, if you can get your head around this, among records in which a_1 has the value two, a_2 takes the value of one twice there. Those are the two records in which a_1 has a value two, a_2 takes the value of one twice. These are just objects you need to know about.

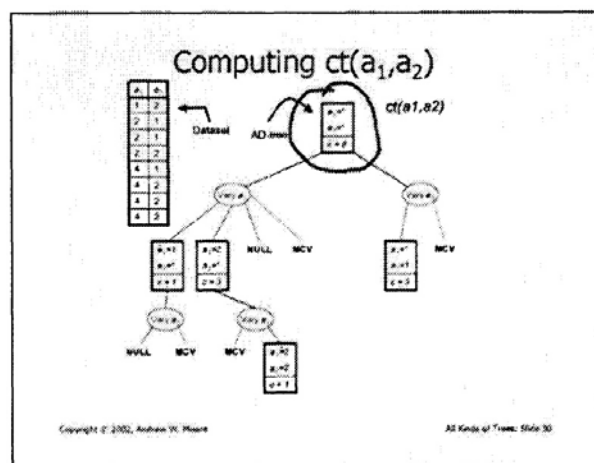


There are two properties that contingency tables have, and they are very similar to the properties that you get from the axioms of probability, only this one is based on counts. One of them, the marginal contingency table over a_2 , I can just row-wise add together the conditional contingency tables over each of the values of a_1 .



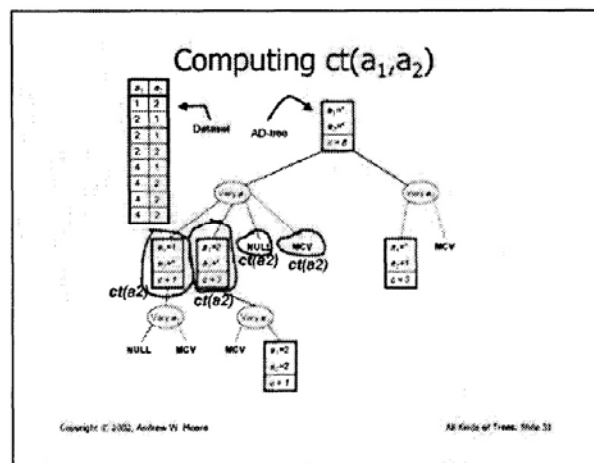
Second fact, if I want to make the joint contingency table over a_1 and a_2 , I can do that by gluing together contingency tables of a_2 , conditioned on each of the values of a_1 .

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.



So, let's hold those two facts in our memory, and then I will show you how you can do reconstruction. So, here is our mutilated data structure. I am going to show you we can recreate the joint contingency table over a_1 and a_2 , from this mutilated data structure.

I am going to have a recursive call, a program which progressively runs over this tree. It seems to take a node, in this case, the root node, and a list of attributes—in this case a_1 and a_2 .



To build the joint contingency table of a_1 and a_2 , it is going to be just what I said before. It is going to compute the conditional contingency tables, conditioned on each value of a_1 , from each of these nodes. To do this one, it just does a recursive call to the same algorithm. To do this one, it just realizes it has to create a table full of zeroes. This is the place where it looks like we are getting stuck, because this is the place where we don't have any data structures to run down to compute this conditional contingency table.

We can use the other facts I told you about contingency tables to save us. I know that I do a row-wise addition of these four contingency tables, I should get the marginal contingency table on a_2 .

SDSS Galaxies using ADtrees

1,580,000 Galaxies, 27 binary attributes per galaxy.
 Time to build ADtree: 4 minutes. Tree Memory: 2 Megs

| Operation | Efficient Non-adtree implementation | AD-tree (secs) |
|---|-------------------------------------|----------------|
| All 1-cubes | 7 mins | 0.1 seconds |
| All 2-cubes | 1 hour, 20 mins | 1 second |
| All 3-cubes | 10 hours | 11 seconds |
| All 4-cubes | 5 days | 1.3 minutes |
| All 5-cubes | 1.2 months | 21 minutes |
| 500 iterations of Bayes Net stochastic search | 14 minutes | 0.2 seconds |
| 50000 iterations of Bayes Net stochastic search | 13 days | 1.9 minutes |
| Size 2 Association rules | 5 minutes | 0.05 seconds |
| Size 4 Association rules | 8 hours | 8 seconds |

Copyright © 2002, Andrew W. Moore All Kinds of Trees, Slide 34

So, when we actually do use this for various things, we have to search, in this case, over a relatively small database, just 1.5 million records, and 27 attributes, so 27 covariates. We also search over all five-dimensional tables. There are 27, choose five of those. Previously, we would have done a pass through data set for each of those 27 things. We don't have to do that any more. Instead of taking 1.2 months of CPU time, it takes 21 minutes of CPU time to do it.

The final closing comments, what happens as we have been developing these algorithms? I apologize because I have been showing you algorithms that we actually developed a couple of years ago.

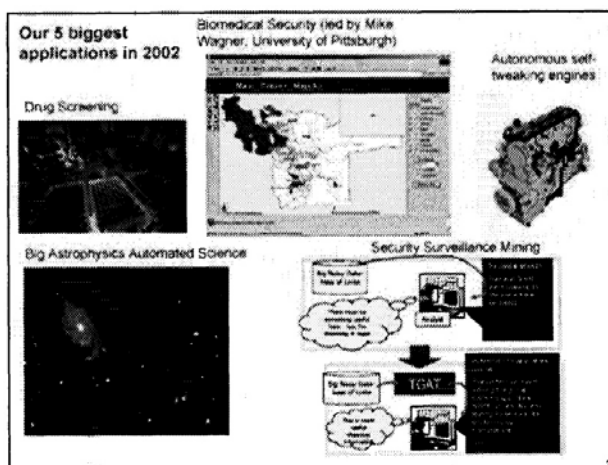
| 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 |
|--|---|---|--|---|---|---|
| M&M Mars Line Control Administration (NDC) (m... malization) | Aviation (Damage sta- tization) Digital Equipment (priority monitoring) | M&M Mars (multicac- turing) NASA/JPL (Astrophysics meeting) in TeDe mission control | Calculator (Space parts) US Army (airforce detection) M&M Mars (scheduling with uncer- tainty in (Adaptive design) | Digital ABC (Music Sales) California (insurance) Qualcomm (equipment) Straker (Brand Management) in multiple Deployment (workforce optimization) Cellphones (increased anomaly detection) | Biometrics (company (health monitor) Sales) (Mining) (insurance) (new product development) California (epi- surveillance screen) CHARM (national disease monitor) ADIS (Contract insurer supply chain) Biosensor (Flight delays) SD (Security) Washington Public Hospital System (ER delays) Unleaver (targeted marketing) Carnegie Mellon (airway protection) | NASA (National Virtual Observatory) NSF (astroinformatics software) Mitsubishi (Bio chemistry) Plexus (high-throughput screen) California (Bio- informatics (Engineering) State of PA (National Disease Monitor (with Miss Wagner of U. Pitt) State of PA (Anti-Cancer (collaboration with CMU Biology) CGI (detecting patterns in sites) Other Government Departments (identifying dangerous regions, potential collaborators, and aliases) Message Company (ingredient manufacturing pharmaceuticals (Bayes Net) Transform Pharms (make autonomous experiment design) Psychogenics Inc. (Effects of psychotropic drugs on rats) |

Auton/SPR
Deployments

I didn't show you our recent work because I didn't have time to get to that in this talk. What happened after we reduced those algorithms was a bunch of basically government places and also some commercial places who had run into the problem they had a piece of statistics they really wanted to run, but they really could run it using off the shelf tools, came to us. So, we had a consulting company which now routinely turns these things out, in special cases, that we do things. We have found it is a very popular thing. We are turning away business rather than having to seek it out.

So, I strongly recommend that this is a good business to be in, the computer science of making statistics go fast.

About this PDF file: This new digital representation of the original work has been recomposed from XML files created from the original paper book, not from the original typesetting files. Page breaks are true to the original; line lengths, word breaks, heading styles, and other typesetting-specific formatting, however, cannot be retained, and some typographic errors may have been accidentally inserted. Please use the print version of this publication as the authoritative version for attribution.

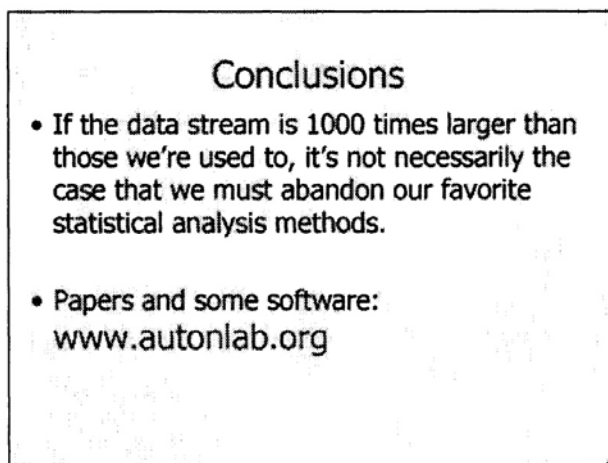


Just to ground it a little bit, some of the main academic things we are working on at the moment, is applying spatial statistics for biosurveillance of possible bioterrorist attacks.

We have a system running monitoring western Pennsylvania, another one monitoring Utah. It was in operation during the Olympic Games, looking for hot spots in the sales of over-the-counter pharmaceuticals, emergency department registrations. We are currently trying to take that national to the stage where we are still going to be, by your standards, quite small. We will be getting in 10 million records a day, so not the billions of records that some of you have been talking about, and hoping to apply that in those cases.

I have shown you some of the physics applications. This has also been very useful in some large, high-throughput screening work with Pfizer.

The coolest algorithm is, we are putting something on Caterpillar diesel engines and, this is a long shot, but if it could become a product in 2008, when new emissions will come into place, which will monitor everything going on in a diesel engine real-time, to be doing real-time monitoring of emissions as a function of the timing of the diesel engine.



We are also following this up using this for a number of intelligence applications. So, that is it. With good computational geometry, you can sometimes get even apparently impractical problems to a practical size, and papers and software at that Web address. That is it.

MR. WILKINSON: Time for one or two questions.

MR. DOMINGOS: Going back to the pairwise distances problem, they were all guaranteed to be within .4 or farther than .4. What if that never happens? Do you just go down to single problems?

You could always have two boxes where the smallest distance is less than .4 and the largest distance is more than .4.

MR. MOORE: In the exact version of the algorithm there will be some cases where you have to go down to individual points in the database. You can show that the number of times that you will have to do that is the square root of the number of pairs in the database. So, the square root of the number of pairs in the database is linear, so the whole thing ends up being linear in the database size.

If you tell the system that you are prepared to accept, say, a 1 percent error in your final count, usually it will never go down.

AUDIENCE: What about the geometric points are scaled to higher dimensions?

MR. MOORE: Good point, I should have mentioned that. These are based on a data structure called a KD tree, for which Jerry Friedman was one of the inventors. KD trees typically don't work very well above about 10 or 20 dimensions. So, some of these algorithms, like the mixture of Gaussian ones, we get into computational problems with about 20 dimensions.

Some of the other things, like the kernel methods or the two point counts we have done, we actually these days run them in a different data structure called a metric tree, where everything is stored in balls instead of hyper-rectangles. Under some circumstances, we have been able to run those in thousands of dimensions. In one particular case, we ran it in a million dimensions. That is not generally possible. If our distribution of the data is uniform, you could prove that you should not be able to do this efficiently.

In empirical data, there are correlations among the covariates. Even if they are non-linear correlations, then you expect it to be practical, which is why we have been able to do this in thousands of dimensions.

Concluding Comments

Among the research topics presented at this meeting were remote sensing for climate modeling, computer or network intrusion detection monitoring, records from communication networks or Web logs, e-commerce recommendation systems, and real-time imaging problems arising in robotic vision. By design, none of the presentations is so broad that it touches all five major areas of research. However, common threads did emerge, such as statistical modeling, visualization, and the need to store, move, and manipulate massive data streams, which are nontrivial challenges.

The workshop explored both theoretical aspects of dealing with massive data streams as well as practical means of effective analysis of such streams. For many massive data streams, there is no model of the data and little a priori knowledge, and the workshop presenters demonstrated the vital role that statistical modeling, analysis, and visualization play in enabling the essence of otherwise hidden information to be distilled from these data streams.

This is an exciting new realm for statisticians, motivated by a rapid increase in streaming data. The Committee on Applied and Theoretical Statistics (CATS) has already made plans for a workshop on visualization of uncertain information, to be held in the winter of 2005, and the Committee will explore other dimensions of these challenges in the coming years. Please check out the CATS Web site for future events.