INTEGRATION OF COMPUTATIONAL METHODS AND VISUAL ANALYTICS FOR LARGE-SCALE HIGH-DIMENSIONAL DATA

A Thesis
Presented to
The Academic Faculty

by

Jaegul Choo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science and Engineering

Georgia Institute of Technology
Auguest 2013

INTEGRATION OF COMPUTATIONAL METHODS AND VISUAL ANALYTICS FOR LARGE-SCALE HIGH-DIMENSIONAL DATA

Approved by:

Professor Haesun Park, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Alexander Gray
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Guy Lebanon
School of Computational Science and Engineering
Georgia Institute of Technology

Professor John Stasko
School of Interactive Computing
Georgia Institute of Technology

Dr. Pak Chung Wong
National Visualization and Analytics Center
Pacific Northwest National Laboratory

Date Approved: 06/28/2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

With the increasing amount of collected data, large-scale high-dimensional data analysis is becoming essential in many areas. These data can be analyzed either by using fully computational methods or by leveraging human capabilities via interactive visualization. However, each method has its drawbacks. While a fully computational method can deal with large amounts of data, it lacks depth in its understanding of the data, which is critical to the analysis. With the interactive visualization method, the user can give a deeper insight on the data but suffers when large amounts of data need to be analyzed.

Even with an apparent need for these two approaches to be integrated, little progress has been made. As ways to tackle this problem, computational methods have to be re-designed both theoretically and algorithmically, and the visual analytics system has to expose these computational methods to users so that they can choose the proper algorithms and settings. To achieve an appropriate integration between computational methods and visual analytics, the thesis focuses on essential computational methods for visualization, such as dimension reduction and clustering, and it presents fundamental development of computational methods as well as visual analytic systems involving newly developed methods.

The contributions of the thesis include (1) the two-stage dimension reduction framework that better handles significant information loss in visualization of high-dimensional data, (2) efficient parametric updating of computational methods for fast and smooth user interactions, and (3) an iteration-wise integration framework of computational methods in real-time visual analytics. The latter parts of the thesis

focus on the development of visual analytics systems involving the presented computational methods, such as (1) Testbed: an interactive visual testbed system for various dimension reduction and clustering methods, (2) iVisClassifier: an interactive visual classification system using supervised dimension reduction, and (3) VisIRR: an interactive visual information retrieval and recommender system for large-scale document data.

CHAPTER I

INTRODUCTION

1.1 Motivation

In these days, an increasing amount of data is being generated in various forms such as documents, images, etc. To analyze these data, the raw data are encoded as high-dimensional vectors and then computational methods are typically applied in the context of statistical machine learning and data mining. For instance, in order to perform the facial recognition given a set of facial image data, the image data are first encoded as a bag-of-feature-points scheme [91], and then a certain classification technique, such as support vector machines [118], is applied. In many cases, however, these computational methods are done in a fully automated manner, and such approaches bear many limitations as follows:

1. **Difficulty in choosing the encoding scheme and algorithm against the data at hand.** For certain types of tasks, e.g., classification, countless algorithms exist, and users may not know which one to apply. Furthermore, each method imposes its own assumptions on the data and involves a set of parameters to be carefully determined. However, these issues are by no means straightforward to solve. For instance, the characteristics of the data do not meet the underlying assumption in the algorithms. Recent manifold learning algorithms [125, 111, 17] assume the low-dimensional curvi-linear manifold structure, but there is no guarantee that the data at hand have such a structure. As another example, even though the recent nonlinear kernel-based methods sound appealing, how to determine the optimal kernel parameters, e.g., a bandwidth parameter in Gaussian kernels, is also dependent on the data.

2. **Discrepancy between the algorithm criteria/performance and humans' task objective.** Many algorithm criteria of computational modules do not necessarily reflect humans' semantics and intuition. Instead, the algorithm criteria are often driven by other aspects such as computational efficiency, tractability, closed-form solutions, etc. For example, a squared loss function employed in many computational modules is widely used due to its simple optimization processes, but it may not always give the best results in practical data analysis scenarios in that, say, the squared loss is generally prone to outliers. Even if the algorithm criteria suit humans' needs well, the performance may not be sufficient. To be specific, many carefully-designed criteria often make it hard to achieve the satisfactory criteria value due to their intensive computation and existence of multiple local minima.

3. **Ambiguity in task formulation.** Large-scale data make it hard to explore and understand them, and sometimes they even obscure what to solve and what to be able to do with our data. In this situation, people may seek for some insight about the data as to which data items may behave differently from the rest. In addition, even if people have a clear goal in mind, it is often the case that the mathematical formulation required to apply computational methods is not straightforward. For instance, suppose one wants to analyze social network data to identify which person or group has caused a certain movement. This task may not be simply interpreted as a mathematical objective function or fit to the existing formulation of computational methods.

In contrast to the fully automated computational approaches that lack deep understanding and careful treatment of the data, the area of visual analytics [127, 78], which is defined as *the science of analytical reasoning facilitated by interactive visual interfaces*, has gained increasing interest. Visual analytics has fascinating characteristics that leverage humans' ability of quick visual insight in the data analysis and decision

processes, and compared with the well-established literature of information visualization, visual analytics typically focuses on reasoning and decision-making processes rather than just understanding the data visually. Unfortunately, however, most of the state-of-the-art visual analytics techniques or systems do not properly accommodate large-scale data. One of the reasons is that although humans are good at quick visual insight, such an ability deteriorates when the number of visualized objects, either data items or features, is large. Furthermore, the limited screen space tends to create visual clutter when visualizing large-scale high-dimensional data. For instance, parallel coordinates, a widely-used visualization technique for multi-dimensional data, do not scale when the dimension exceeds several tens or hundreds.

To improve the scalability issue, computational methods can support visual analytics by transforming the raw data into more compact and meaningful information. For instance, dimension reduction and clustering can reduce the numbers of features and data items into manageable sizes for both visualization and human perception. Beyond such reduction aspects, computational methods can provide more intelligent information about the data via their formulations based on long-studied statistical or probabilistic theories in the context of machine learning and data mining. Examples of such tasks include facial/speech recognition [16], document topic modeling [20], sentiment analysis [99], and recommender systems [3].

Such appealing capabilities of computational methods motivated people towards the tight integration of them with visual analytics for large-scale data. For example, Seo et al. [117] have provided an interactive visualization system to explore the clustering results obtained by the widely-used hierarchical clustering method. A recently proposed method, latent Dirichlet allocation, has been utilized in visual analytics tools for text documents [134, 50, 38, 37]. Even though various efforts have been made for utilizing computational methods in visual analytics, there is still significant room to improve such utility. That is, even though numerous advanced

computational methods are currently being proposed and some of them claim that they can be easily adopted in visualization applications, practical visual analytics systems do not seem to currently take full advantage of these advanced methods. As a result, people still tend to only use a few of the basic computational methods, e.g., principal component analysis (PCA) [74] for dimension reduction and k-means[19] for clustering in many real-world analysis tasks. In addition, the above-mentioned intelligent information could be useful in interactive visualization approaches, but its usage in this direction is still limited. In this thesis, I address several hurdles in achieving an appropriate integration as follows:

1. **The computational module and its output are difficult to understand.** Without deep knowledge about the computational methods and the data, the output generated by the computational methods is often more difficult to understand than the original raw data. Many modern computational algorithms are complex, and often for the sake of algorithm flexibility, they involve parameters that have to be carefully determined. However, domain experts may improperly set the parameters values due to their lack of understanding the function of the parameters. Consequently, many visual analytic systems choose specific computational methods and treat them as a black box while focusing on the subsequent analysis of their output. Without a proper understanding of the algorithm and its parameters, the performance of the computational module may not be satisfactory enough to start an analysis with.

2. **Computational methods require a significant amount of time.** Most computational methods involve heavy computations. In fact, as most methods become more advanced and capable, they tend to require more intensive computations, which usually have a squared or cubic order of computational complexity in terms of the number of data items and/or features. Therefore, when dealing with large-scale data, the significant amount of computational

time required hinders real-time visualization and subsequent interaction with these computational modules.

This thesis aims to overcome these hurdles and achieve the tight integration between computational methods and visual analytics in modern data analysis scenarios. I claim that to this end, the computational methods have to be customized and even be re-invented for use in visual analytics, and at the same time, the visual analytics systems have to expose them to users out of a black box via interactive capabilities of choosing the right methods and the best parameters. Based on this claim, the thesis provides (1) several novel approaches for customizing computational methods and (2) the visual analytics systems integrating such customization in various application domains.

1.2 Objective

In summary, the thesis statement can be described as follows:

> **The theoretical and algorithmic customization of computational methods will enable their appropriate integration with visual analytics for analyses of complex large-scale high-dimensional data. Visual analytics systems equipped with interactive capabilities with various computational methods will help analysts better understand the data at hand and solve complicated analysis tasks.**

Under this statement, the thesis aims at achieving the true visual analytics where the computational analyses and the user-driven interactive visual exploration are tightly integrated. More specifically, the thesis mainly focuses on two key categories of computational methods: dimension reduction and clustering. Dimension reduction and clustering play an essential role in dealing with large-scale high-dimensional data

in visual analytics by reducing the data dimension and the number of data items. In other words, dimension reduction can reveal meaningful dimensions or features out of numerous original dimensions as well as provide a means of visualizing high-dimensional data in visual 2D/3D spaces so that analysts can obtain the insight about data relationships with respect to geometric locations of data.

On the other hand, clustering provides an overview of large-scale data in terms of a manageable number of groups based on their semantic coherences. Such cluster information can then guide analysts to a proper data group of interest on which they can further focus their analysis.

To be specific, the thesis addresses the following research questions in regards to integration of dimension reduction and clustering to visual analytics.

1. Which characteristics in terms of data, algorithms, and humans, should be exploited in order to make computational methods better support visual analytics? Based on these characteristics, how can the computational methods be re-designed in visual analytics?

2. How can visual analytics systems utilizing these improved computational methods be realized and what analytic benefits can we claim from such systems?

1.3 Contributions

I present as the main contributions of the thesis various ways to tackle each of the two addressed research questions. Basically, in response to the first question, the thesis discusses several theoretical and algorithmic improvements of computational methods when they support visual analytics, as follows:

1. Two-stage dimension reduction framework that better handles significant information loss in visualization of high-dimensional data [35].

2. Efficient parametric updating of computational methods for fast and smooth user interactions [34].

3. Iteration-wise integration framework of computational methods in real-time visual analytics [31].

On the other hand, the latter part of the thesis contribution lies mainly in the development of visual analytics systems involving the presented improvements as follows:

1. Testbed: an interactive visual testbed system providing users with an easy access to various dimension reduction and clustering methods for their own data sets [32].

2. iVisClassifier: an interactive visual classification system via a supervised dimension reduction for improving classification models in a user-driven way [36].

3. VisIRR: an interactive visual information retrieval and recommender system for large-scale document data that expands the documents of interest based on user preferences [33].

1.4 Organization

The rest of the thesis presents each of the above-listed contributions in more detail.

Chapter 2 presents a novel framework of two-stage dimension reduction for visualization of high-dimensional data. It is inevitable that significant information will be lost when reducing the original high dimension of data into 2D/3D in visualization. By using the formulations in terms of cluster-wise measures, the two-stage dimension reduction framework, which separates the criteria of the original dimension reduction methods and the further information loss, are presented. The thesis presents the detailed criteria using widely-used dimension reduction methods and their visualization examples on real-world data.

Chapter 3 focuses on improving basic interactions with compuational methods, i.e., changing their parameters in visual analytics. When dealing with real-world data, it is not trivial to determine the parameter values of used computational methods. Thus, users could naturally change the parameters and see what aspects of data the computational methods may reveal. In order to accelerate such interactions, the thesis presents efficient parametric updating algorithms and their uses.

Chapter 4 presents another novel approach called PIVE (**P**er-**I**teration **V**isualization **E**nvironment for supporting real-time interactive visualization with computational methods) to make computational methods significantly efficient in visual analytics. In this chapter, the presented approach exploits the fact that most computational methods are built upon iterative algorithms. Rather than waiting for the entire iteration to finish, the iteration-wise framework visualizes the intermediate results of computational methods and enables users to interact with them in real time during iterations. The details of the proposed framework and its applications using well-known visual analytics systems are presented.

Chapter 5 describes the fundamental visual analytics system called the Testbed system, which makes various traditional and advanced dimension reduction and clustering algorithms readily available in visual analytics scenarios. In addition to the basic but crucial capabilities of selecting different methods and their parameters, exploring raw data, and brushing-and-linking, the system features a novel capability of alignment between multiple visualization results for easy comparisons. The system details and several usage scenarios are discussed.

Chapter 6 presents iVisClassifier, another visual analytics system developed based on the Testbed system. iVisClassifier is a customized system for classification applications, which mainly utilizes a specific dimension reduction among those available in the Testbed system. iVisClassifier supports ample features in order to help users understand data, which are, for example, what classes are confusing against each other,

which data items are easy/difficult to classify, as well as enables users to intervene in the classification processes. The system details and the usage scenarios in the facial recognition context are described.

Chapter 6 presents VisIRR, an interactive visual information retrieval and recommender system based on the Testbed system. This system directly tackles the large scale of data by starting with more than 400,000 data items. Besides the basic capabilities of clustering and visualizing the retrieved documents based on users' queries, the system has the capability to provide recommendations based on users' preference information. The details of the computational methods used and visualization processes are presented along with several usage scenarios.

Finally, Chapter 8 concludes the thesis and presents interesting future research topics.

CHAPTER II

TWO-STAGE FRAMEWORK FOR VISUALIZATION OF CLUSTERED HIGH-DIMENSIONAL DATA

In this chapter, we will discuss dimension reduction methods for 2D visualization of high dimensional clustered data. We propose a two-stage framework for visualizing such data based on dimension reduction methods. In the first stage, we obtain the reduced dimensional data by applying a supervised dimension reduction method such as linear discriminant analysis which preserves the original cluster structure in terms of its criteria. The resulting optimal reduced dimension depends on the optimization criteria and is often larger than two. In the second stage, the dimension is further reduced to two for visualization purposes by another dimension reduction method such as principal component analysis. The role of the second stage is to minimize the loss of information due to reducing the dimension all the way to two. Using this framework, we propose several two-stage methods, and present their theoretical characteristics as well as experimental comparisons on both artificial and real-world text data sets.

2.1 Introduction

Within the visual analytics community, various types of information content are represented using high dimensional signatures. To make these signatures useful they often need to be transformed into a lower dimension (i.e., 2D or 3D) for a variety of visual representations such as scatter plots. Many researchers in this community have used a wide assortment of dimension reduction techniques, e.g., self-organizing map (SOM) [83], principal component analysis (PCA) [75], multidimensional scaling

(MDS) [45], etc. However, it is not always clear why a certain technique has been chosen over another, especially to the end user. Typically, the goal of dimension reduction techniques can be viewed in terms of two aspects: efficiency and accuracy. Efficiency as defined here is the time to compute the reduction, but accuracy may not be as simple to quantify. Many would amiably agree to quantify accuracy as a measure of the relationship preservation in the high dimensional space to the reduced dimensional space. Note that most techniques either directly or indirectly work on this principle.

There are other properties that are important to those interpreting the semantics of the reduced space. Specifically, we note that while local neighbor preservation is important it depends upon the analysis task. No single reduction technique will provide the complete view as various properties of the space are obscured or lost. We have mentioned that typically the primary objective is relationship preservation. However, there are at least two others: outlier and macro structure visualization. Outliers are conceptually easy (i.e., a variance beyond some threshold), but more difficult to quantify, as we do not necessarily know which set of outliers are important to accentuate to the user. Certain techniques (e.g., PCA) tend to show outliers more readily, however tend to compress the reduced space at the expense of showcasing the outliers. Other techniques (e.g., SOM) maximize space usage well, but do so at the expense of masking or even hiding those outliers. Likewise, macro structures of the high dimensional space may be masked or massively distorted during the reduction. Macro structures are those larger order groupings (e.g., clusters) that exist in the original dimensional space. We recognize they are important in dimension reduction research and to those in the visual analytics community. However, few of them focus on data representation especially for visualization of the clustered data [147, 84, 49].

We propose theoretical measures for these properties and efficient algorithms which will aid not only the researchers but ultimately the users/analysts to better

understand which balance of properties are important and for which analytic tasks.

2.2 Motivation

The focus of this chapter is the fundamental characteristics of dimension reduction techniques for visualizing high dimensional data in the form of a 2D scatter plot when the data has cluster structure. The role of dimension reduction here is to give a 2-dimensional representation of data while preserving cluster structure as much as possible. To this end, supervised dimension reduction methods that incorporate cluster information such as linear discriminant analysis (LDA) [60] or orthogonal centroid method (OCM) [71] can be naturally considered.

However, one of the issues is that with many dimension reduction methods designed to preserve the cluster structure in the data, the theoretically optimal reduced dimension, which is the smallest dimension that is acceptable with respect to the optimization criteria of the dimension reduction method, is usually larger than 2. For example, in LDA, the minimum reduced dimension that preserves the cluster structure quality measure defined as a trace maximization problem is one less than the number of clusters in the data in general [68, 67].

In this case, one may simply choose the two dimensions that contribute most to such a measure. However, with only two dimensions, such a measure may become significantly smaller than the original quantity after dimension reduction. This results in loss of information that hinders visualization in properly reflecting the true cluster relationship of the data. A similar situation may occur when using PCA for visualizing the data not having a cluster structure. Even though PCA finds the principal axes that maximally capture the variance of the data, when the resulting 2-dimensional representation of the data maintains only a small fraction of the total variance, the relationships of the data in 2 dimension are likely to be highly inconsistent with those in the original dimension.

Such loss of information is inevitable in that the dimension has to be reduced to 2. Our main motivation is to deal with such loss more carefully by separating the loss-introducing stage from the original dimension reduction methods. Based on this idea, we propose the two-stage framework of dimension reduction for visualization. In this framework, a supervised dimension reduction method is applied in the first stage so that the original dimension is reduced to the minimum dimension achievable while preserving the quality of cluster measure as defined in a dimension reduction method. The reduced dimension achieved in the first stage is often larger than 2. Thus in the second stage, we find another dimension reducing transformation that minimizes the loss introduced in further reducing the dimension all the way to 2. This two-stage framework provides us with a means to flexibly apply different types of dimension reduction techniques in each stage and to systematically analyze their effects, which provides understanding the effects of the overall dimension reduction methods for visualization of clustered data. The issues then are the design of the most appropriate dimension reduction methods, the modeling of optimization criteria, and the corresponding solution methods.

In this chapter, we present both theoretical and empirical answers to these issues. Specifically, we propose several two-stage methods utilizing linear dimension reduction methods such as LDA, orthogonal centroid method (OCM), and principal component analysis (PCA), and we present their theoretical justifications by modeling the optimization criteria for which each method provides the optimal solution. Also, we illustrate and compare the effectiveness of the proposed methods by showing empirical visualization on synthetic and real-world data sets.Although nonlinear dimension reduction methods such as MDS or other manifold learning methods such as isometric feature mapping [125] and locally linear embedding [111] may also be utilized for the effective 2D visualization of high dimensional data, our focus in this chapter is on linear methods. The linear methods are computationally more efficient

in general, and unlike most of the manifold learning methods, they also provide dimension reducing transformations that can be applied to map and visualize unseen data points in the same space where the existing data are visualized.

Our approach to successively apply two dimension reduction methods should be discerned from the previous work [144, 145, 150] in that they usually aim for improving computational efficiency, scalability, or applicability of a certain dimension reduction method, e.g., LDA.

The rest of this chapter is organized as follows. In Section 2.3, LDA, OCM, and PCA are described based on a unified framework of the scatter matrices and their trace optimization problems. In Section 2.4, we formulate two-stage dimension reduction methods, and in Section 2.5, several two-stage methods for visualization are proposed and compared along with their criteria. Experimental comparisons are given using artificial and real-world data sets in Section 2.6, and conclusions are drawn in Section 2.7.

2.3 Dimension Reduction as Trace Optimization Problem

In this section, we introduce the notions of scatter matrices used in defining cluster quality and optimization criteria for dimension reduction.

Suppose a dimension reducing linear transformation $G^T \in \mathbb{R}^{l\times m}$ maps an m-dimensional data vector x to a vector z in an l-dimensional space ($m > l$):

$$G^T : x \in \mathbb{R}^{m\times 1} \rightarrow z = G^T x \in \mathbb{R}^{l\times 1}. \tag{1}$$

Suppose also that a data matrix $A = [a_1\, a_2 \cdots a_n] \in \mathbb{R}^{m\times n}$ is given where the columns a_j, $j = 1, \ldots, n$, of A represent n data items in an m-dimensional space, and they are partitioned into k clusters. Without loss of generality, for simplicity of notations, we further assume that A is partitioned as

$$A = [A_1 \quad A_2 \quad \cdots \quad A_k], \text{ where } A_i \in \mathbb{R}^{m\times n_i} \text{ and } \sum_{i=1}^{k} n_i = n.$$

Let $\mathcal{N}_i$ denote the set of column indices that belong to cluster i, and n_i the size of $\mathcal{N}_i$. The i-th cluster centroid $c^{(i)}$ and the global centroid c are defined, respectively, as

$$c^{(i)} = \frac{1}{n_i} \sum_{j \in \mathcal{N}_i} a_j \text{ and } c = \frac{1}{n} \sum_{j=1}^{n} a_j.$$

The scatter matrix within the i-th cluster $S_w^{(i)}$, the within-cluster scatter matrix S_w, the between-cluster scatter matrix S_b, and the total (or mixture) scatter matrix S_t are defined [70, 122], respectively, as

$$\begin{aligned}
S_w^{(i)} &= \sum_{j \in \mathcal{N}_i} (a_j - c^{(i)})(a_j - c^{(i)})^T, \\
S_w &= \sum_{i=1}^{k} S_w^{(i)} = \sum_{i=1}^{k} \sum_{j \in \mathcal{N}_i} (a_j - c^{(i)})(a_j - c^{(i)})^T, && (2) \\
S_b &= \sum_{i=1}^{k} \sum_{j \in \mathcal{N}_i} (c^{(i)} - c)(c^{(i)} - c)^T = \sum_{i=1}^{k} n_i (c^{(i)} - c)(c^{(i)} - c)^T && (3) \\
&= \frac{1}{n} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} n_i n_j (c^{(i)} - c^{(j)})(c^{(i)} - c^{(j)})^T, \text{ and} && (4) \\
S_t &= \sum_{j=1}^{n} (a_j - c)(a_j - c)^T. && (5)
\end{aligned}$$

Note that the total scatter matrix S_t is related to S_w and S_b as [70]

$$S_t = S_w + S_b. \tag{6}$$

When G^T in Eq. (1) is applied to the matrix A, the scatter matrices S_w, S_b, and S_t in the original dimensional space are reduced to the $l \times l$ matrices

$$G^T S_w G,\ G^T S_b G, \text{ and } G^T S_t G,$$

respectively. By computing the trace of the scatter matrices as

$$\begin{aligned}
\mathrm{trace}(S_w) &= \sum_{i=1}^{k}\sum_{j\in\mathcal{N}_i}(a_j - c^{(i)})^T(a_j - c^{(i)}) \\
&= \sum_{i=1}^{k}\sum_{j\in\mathcal{N}_i}\|a_j - c^{(i)}\|_2^2, \qquad (7) \\
\mathrm{trace}(S_b) &= \sum_{i=1}^{k}\sum_{j\in\mathcal{N}_i}(c^{(i)} - c)^T(c^{(i)} - c) \\
&= \sum_{i=1}^{k} n_i\|c^{(i)} - c\|_2^2 \qquad (8) \\
&= \frac{1}{n}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} n_i n_j\|c^{(i)} - c^{(j)}\|_2^2, \text{ and} \qquad (9) \\
\mathrm{trace}(S_t) &= \sum_{j=1}^{n}(a_j - c)^T(a_j - c) = \sum_{j=1}^{n}\|a_j - c\|_2^2, \qquad (10)
\end{aligned}$$

we obtain values that can be used to measure the cluster quality. Note that from Eqs. (8) and (9), $\mathrm{trace}(S_b)$ can be viewed as the squared sum of the pairwise distances between cluster centroids as well as that of the distances between each centroid and the global centroid.

The cluster structure quality can be defined by analyzing how well each cluster can be discriminated from each other. High quality clusters usually have small $\mathrm{trace}(S_w)$ and large $\mathrm{trace}(S_b)$, relating to the small variance within each cluster and the large distances between clusters. Subsequently, dimension reduction methods may be intended to maximize $\mathrm{trace}(G^T S_b G)$ and minimize $\mathrm{trace}(G^T S_w G)$ in the reduced dimensional space. This simultaneous optimization can be approximated to a single criterion as

$$J_{b/w}(G) = \max \mathrm{trace}((G^T S_w G)^{-1}(G^T S_b G)), \qquad (11)$$

which is the criterion of LDA. In addition, one may focus on maximizing the distances between clusters, which can be represented as the criterion of OCM, i.e.,

$$J_b(G) = \max_{G^T G = I} \mathrm{trace}(G^T S_b G). \qquad (12)$$

On the other hand, regardless of cluster dependent terms, S_w and S_b, the trace of the total scatter matrix S_t can be maximized as

$$J_t(G) = \max_{G^T G = I} \text{trace}(G^T S_t G), \tag{13}$$

which turns out to be the criterion of PCA. In Eqs. (12) and (13), without the constraint, $G^T G = I$, $J_b(G)$ and $J_t(G)$ can become arbitrarily large.

In what follows, LDA, OCM, and PCA are discussed based on such maximization criteria, and their properties relevant to visualization are identified.

2.3.1 Linear Discriminant Analysis (LDA)

Conceptually, in LDA, we are looking for a dimension reducing transformation that keeps the between-cluster relationship as remote as possible by maximizing trace$(G^T S_b G)$ while keeping the within cluster relationship as compact as possible by minimizing trace$(G^T S_w G)$. As shown in Eq. (11), the criterion of LDA can be written as

$$J_{b/w}(G) = \max \text{trace}((G^T S_w G)^{-1}(G^T S_b G)). \tag{14}$$

It can be shown that for any $G \in \mathbb{R}^{m \times l}$ where $m > l$,

$$\text{trace}((G^T S_w G)^{-1}(G^T S_b G)) \leq \text{trace}(S_w^{-1} S_b), \tag{15}$$

meaning that the cluster structure quality measured by trace$(S_w^{-1} S_b)$ cannot be increased after dimension reduction [60]. By setting the derivative of Eq. (14) with respect to G to zero, which gives the first order optimality condition, it can be shown that the solution of LDA, where we denote it as G_{LDA}, has the columns which are the leading generalized eigenvectors u of the generalized eigenvalue problem,

$$S_b u = \lambda S_w u. \tag{16}$$

Since the rank of S_b is at most $k-1$, LDA achieves the upper bound of trace$((G^T S_w G)^{-1}$

Table 1: Comparison of dimension reduction methods. It is assumed S_b and S_t are full rank.

	Optimization criterion ($x \in \mathbb{R}^{m\times 1} \overset{G^T}{\rightarrow} y \in \mathbb{R}^{l\times 1}$)	Algorithm	Smallest dimension achieving the criterion upper bound
LDA	$J_{b/w}(G) = \max \operatorname{trace}((G^T S_w G)^{-1}(G^T S_b G))$	generalized eig. decomp.	$k-1$
OCM	$J_b(G) = \max_{G^T G=I} \operatorname{trace}(G^T S_b G)$	QR decomp.	k
PCA	$J_t(G) = \max_{G^T G=I} \operatorname{trace}(G^T S_t G)$	symmetric eig. decomp.	$\min(m,\, n)$

$(G^T S_b G))$ in Eq. (15) for any l such that $l \geq k-1$, i.e.,

$$\begin{aligned} &\operatorname{trace}((G_{LDA}^T S_w G_{LDA})^{-1}(G_{LDA}^T S_b G_{LDA})) \\ &\qquad = \operatorname{trace}(S_w^{-1} S_b) \text{ for } l \geq k-1, \end{aligned} \tag{17}$$

which indicates $\operatorname{trace}(S_w^{-1} S_b)$ is preserved between the original space and the reduced dimensional space obtained by G_{LDA}.

2.3.2 Orthogonal Centroid Method (OCM)

Orthogonal centroid method (OCM) [71] focuses only on maximizing $\operatorname{trace}(G^T S_b G)$ under the constraint of $G^T G = I$. The criterion of OCM is shown as

$$J_b(G) = \max_{G^T G=I} \operatorname{trace}(G^T S_b G). \tag{18}$$

It is known that for any $G \in \mathbb{R}^{m\times l}$ where $m > l$ such that $G^T G = I$,

$$\operatorname{trace}(G^T S_b G) \leq \operatorname{trace}(S_b), \tag{19}$$

which means the cluster structure quality measured by $\operatorname{trace}(S_b)$ cannot be increased after dimension reduction. The solution of Eq. (18) can be obtained by setting the columns of G as the leading eigenvectors of S_b. Since S_b has at most $k-1$ nonzero eigenvalues, the upper bound of $\operatorname{trace}(G^T S_b G)$ in Eq. (19) can be achieved for any l such that $l \geq k-1$, i.e.,

$$\operatorname{trace}(G^T S_b G) = \operatorname{trace}(S_b) \text{ for } l \geq k-1. \tag{20}$$

Eq. (20) indicates trace(S_b) is preserved between the original and the reduced dimensional spaces.

An advantage of OCM is that it achieves an upper bound of trace($G^T S_b G$) more efficiently by using QR decomposition, avoiding the eigendecomposition. The algorithm of OCM is as follows. First the centroid matrix C is formed so that each column of C is composed of each cluster's centroid vector, i.e., $C = \begin{bmatrix} c_1 & c_2 & \cdots & c_k \end{bmatrix}$. Then the reduced QR decomposition [61] of C is computed for $C = Q_k R$ where $Q_k \in \mathbb{R}^{m \times k}$ with $Q_k^T Q_k = I$ and $R \in \mathbb{R}^{k \times k}$ is upper triangular. The solution of OCM, G_{OCM}, is found as

$$G_{OCM} = Q_k.$$

Note that the columns of G_{OCM} are composed of the orthogonal bases for the subspace spanned by the centroids, and $l = k$ in this case. Finally, OCM achieves

$$\text{trace}(G_{OCM}^T S_b G_{OCM}) = \text{trace}(S_b), \text{ where } l = k.$$

By using the equivalence between Eqs. (3) and (4), one can prove that each pairwise distance between cluster centroids is also preserved in the reduced dimensional space obtained by OCM.

Another important property of OCM is that by projecting data into the subspace spanned by the centroids, the order of similarities between any particular point and centroids are preserved in terms of Euclidean norm and cosine similarity measure [71, 67]. In other words, for any vector $q \in \mathbb{R}^{m \times 1}$ and cluster centroids $c^{(i)}$ and $c^{(j)}$, we have

$$\begin{aligned}
&\|q - c^{(i)}\|_2 < \|q - c^{(j)}\|_2 \Rightarrow \\
&\|G_{OCM}^T q - G_{OCM}^T c^{(i)}\|_2 < \|G_{OCM}^T q - G_{OCM}^T c^{(j)}\|_2, \text{ and} \\
&\frac{q^T c^{(i)}}{\|q\|_2 \|c^{(i)}\|_2} < \frac{q^T c^{(j)}}{\|q\|_2 \|c^{(j)}\|_2} \Rightarrow \\
&\frac{\left(G_{OCM}^T q\right)^T G_{OCM}^T c^{(i)}}{\|G_{OCM}^T q\|_2 \|G_{OCM}^T c^{(i)}\|_2} < \frac{\left(G_{OCM}^T q\right)^T G_{OCM}^T c^{(j)}}{\|G_{OCM}^T q\|_2 \|G_{OCM}^T c^{(j)}\|_2}.
\end{aligned}$$

2.3.3 Principal Component Analysis (PCA)

PCA is a well-known dimension reduction method that captures the maximal variance in the data. The criterion of PCA can be written as

$$J_t(G) = \max_{G^T G = I} \text{trace}(G^T S_t G). \tag{21}$$

For any $G \in \mathbb{R}^{m \times l}$ where $m > l$ such that $G^T G = I$, we have

$$\text{trace}(G^T S_t G) \leq \text{trace}(S_t), \tag{22}$$

which means $\text{trace}(S_t)$ cannot be increased after dimension reduction. The solution of Eq. (21), where we denote it as G_{PCA}, can be obtained by setting the columns of G as the leading eigenvectors of S_t. Since the rank of S_t is at most $\min(m, n)$, PCA achieves the upper bound of $\text{trace}(G^T S_t G)$ in Eq. (22) for any l such that $l \geq \min(m, n)$, i.e.,

$$\text{trace}(G_{PCA}^T S_t G_{PCA}) = \text{trace}(S_t) \text{ for } l \geq \min(m, n).$$

In many applications of PCA, however, l is usually chosen as a fixed value less than the ranke of S_t for the purpose of dimension reduction or noise reduction. This noisy subspace corresponds to the smallest eigenvectors of S_t, and they are removed by PCA for better representation of the data.

Although S_t is related to S_b and S_w as in Eq. (6), S_t as it is does not contain any information on cluster labels. That is, unlike LDA and OCM, PCA ignores the cluster structure represented by S_b and/or S_w, which is why PCA is considered as an unsupervised dimension reduction method.

Usually, PCA assumes that the global centroid is zero by subtracting the empirical mean of the data from each data vector. The centered data can be represented as $A - ce^T$, where e is n-dimensional vector whose components are all 1's.

PCA has a unique property that, given a fixed l, it produces the best reduced dimensional representation that minimizes the difference between the centered matrix

$A - ce^T$ and its projection to the reduced dimensional space $GG^T(A - ce^T)$ where G has orthonormal columns, i.e.,

$$G_{PCA} = \arg \min_{G,\, G^T G = I_l} \|GG^T(A - ce^T) - (A - ce^T)\|,$$

where the matrix norm $\|\cdot\|$ is either a Frobenius norm or a Euclidean norm.

The three discussed methods are summarized in Table 1.

2.4 Formulation of Two-stage Framework for Visualization

Suppose we want to find a dimension reducing linear transformation $V^T \in \mathbb{R}^{2\times m}$ that maps an m-dimensional data vector x to a vector z in a 2-dimensional space ($m \gg 2$):

$$V^T : x \in \mathbb{R}^{m\times 1} \to z = V^T x \in \mathbb{R}^{2\times 1}. \tag{23}$$

Further assume that it is composed of two stages of dimension reductions as follows. In the first stage, a dimension reducing linear transformation $G^T \in \mathbb{R}^{l\times m}$ maps an m-dimensional data vector x to a vector y in the l-dimensional space ($l \ll m$):

$$G^T : x \in \mathbb{R}^{m\times 1} \to y = G^T x \in \mathbb{R}^{l\times 1}, \tag{24}$$

where l is fixed as its minimum optimal dimension by the first-stage criterion. When $l \leq 2$, we have no further dimension reduction to do after the first step. However, an optimal l in many methods and for many data sets is larger than 2, and so we assume that $l > 2$.

In the second stage, another dimension reducing linear transformation $H^T \in \mathbb{R}^{2\times l}$ maps an l-dimensional data vector y to a vector z in the 2-dimensional space($l > 2$):

$$H^T : y \in \mathbb{R}^{l\times 1} \to z = H^T y \in \mathbb{R}^{2\times 1}. \tag{25}$$

Such consecutive dimension reductions performed by G^T followed by H^T can be combined, resulting in a single dimension reducing transformation V^T as

$$V^T = H^T G^T. \tag{26}$$

In the next section, discussion will be focused on various ways for choosing the first stage dimension reducing transformation G and the second stage dimension transformation H with a purpose to construct combined dimension reducing transformation $V^T = H^T G^T$ for 2-dimensional visualization according to various optimization criteria.

2.5 Two-stage Methods for 2D Visualization

All the proposed two-stage methods start from one of the supervised dimension reduction methods such as LDA or OCM that are designed for clustered data. In the first stage (by $G^T \in \mathbb{R}^{l \times m}$ in Eq. (24)), the dimension is reduced by LDA or OCM to the smallest dimension that satisfies Eq. (17) or (20), respectively. Therefore in the first stage, the cluster structure quality measured either by trace$(S_w^{-1} S_b)$ or trace(S_b) is preserved. Then we perform the second-stage dimension reduction (by $H^T \in \mathbb{R}^{2 \times l}$ in Eq. (25)) that minimizes the loss of information either by applying the same criterion used in the first stage or by using J_t in Eq. (21), i.e., that of PCA. As seen in Section 3.3, Eq. (21) gives the best approximation of the first-stage results that minimize the difference in terms of Frobenius/Euclidean norm.

In what follows, we describe each of the two-stage methods in detail, and derive their equivalent single-stage methods (by $V^T \in \mathbb{R}^{2 \times m}$ in Eq. (23)) in case they exist.

2.5.1 Rank-2 LDA

In this method, LDA is applied in the first stage, and trace$(S_w^{-1} S_b)$ is preserved in the l-dimensional space where $l = k - 1$. In the second stage, the same criterion $J_{b/w}(H)$ is used to reduce the l-dimensional first-stage results to 2-dimensional data.

The criterion of the second-stage dimension reducing matrix H can be formulated

Table 2: Summary of the optimization criteria of the two-stage dimension reduction methods.

	Rank-2 LDA	LDA + PCA
Stage 1: Preservation $(x \in \mathbb{R}^{m\times 1} \overset{G^T}{\to} y \in \mathbb{R}^{l\times 1})$	$\text{trace}((G^T S_w G)^{-1}(G^T S_b G)) = \text{trace}(S_w^{-1} S_b)$	
Stage 2: Maximization $(y \in \mathbb{R}^{l\times 1} \overset{H^T}{\to} z \in \mathbb{R}^{2\times 1})$	$\text{trace}((H^T(G^T S_w G)H)^{-1}(H^T(G^T S_b G)H))$	$\underset{H^T H=I}{\text{trace}}(H^T(G^T S_t G)H)$
	OCM+PCA	Rank-2 PCA on S_b
Stage 1: Preservation $(x \in \mathbb{R}^{m\times 1} \overset{G^T}{\to} y \in \mathbb{R}^{l\times 1})$	$\text{trace}(G^T S_b G) = \text{trace}(S_b)$	
Stage 2: Maximization $(y \in \mathbb{R}^{l\times 1} \overset{H^T}{\to} z \in \mathbb{R}^{2\times 1})$	$\underset{H^T H=I}{\text{trace}}(H^T(G^T S_t G)H)$	$\underset{H^T H=I}{\text{trace}}(H^T(G^T S_b G)H)$

as

$$H_{b/w} = \max_{H\in\mathbb{R}^{l\times 2}} \text{trace}((H^T(G_{LDA}^T S_w G_{LDA})H)^{-1} (H^T(G_{LDA}^T S_b G_{LDA})H)). \quad (27)$$

Assuming the columns of G_{LDA}, which are generalized eigenvectors of Eq. (16), are in decreasing order of their corresponding generalized eigenvalues, i.e., $G_{LDA} = \begin{bmatrix} u_1 & u_2 & \cdots & u_{k-1} \end{bmatrix}$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{k-1}$, the solution of Eq. (27) is

$$H_{b/w} = \begin{bmatrix} e_1 & e_2 \end{bmatrix},$$

where e_1 and e_2 are the first and the second standard unit vectors, i.e., $e_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \in \mathbb{R}^{l\times 1}$ and $e_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T \in \mathbb{R}^{l\times 1}$. This solution is equivalent to choosing two dimensions with the most leading generalized eigenvalues from the first stage result, and the resulting two-stage method can be represented as a single-stage dimension reduction method by $V \in \mathbb{R}^{m\times 2}$ which directly maximize $J_{b/w}$, i.e.,

$$\begin{aligned} V_{b/w} &= \arg\max_{V\in\mathbb{R}^{m\times 2}} J_{b/w}(V) \\ &= \arg\max_{V\in\mathbb{R}^{m\times 2}} \text{trace}((V^T S_w V)^{-1}(V^T S_b V)). \end{aligned} \quad (28)$$

The solution of Eq. (28) becomes

$$V_{b/w} = G_{LDA} H_{b/w} = \begin{bmatrix} u_1 & u_2 \end{bmatrix},$$

where $u_1$and u_2 are the leading generalized eigenvectors of Eq. (16). This solution is also known as reduced-rank linear discriminant analysis [66].

2.5.2 LDA followed by PCA

In this method, LDA is applied in the first stage, and trace$(S_w^{-1}S_b)$ is preserved in the l-dimensional space where $l = k-1$. In the second stage, PCA is applied in order to obtain the best approximation of the l-dimensional first-stage results in terms of Frobenius/Euclidean norm.

The second-stage dimension reducing matrix H is obtained by solving

$$H_t = \arg \max_{H \in \mathbb{R}^{l \times 2}, H^T H = I} \text{trace}(H^T (G_{LDA}^T S_t G_{LDA}) H), \tag{29}$$

where the solution is the two leading eigenvectors of the total scatter matrix of the first-stage result, $G_{LDA}^T S_t G_{LDA}$.

From Eq. (6), we have

$$G_{LDA}^T S_t G_{LDA} = G_{LDA}^T (S_b + S_w) G_{LDA}. \tag{30}$$

Since LDA conceptually maximizes trace$(G^T S_b G)$ and minimizes trace$(G^T S_w G)$, the result is expected to satisfy

$$\text{trace}(G_{LDA}^T S_b G_{LDA}) \gg \text{trace}(G_{LDA}^T S_w G_{LDA})),$$

which means that $G_{LDA}^T S_t G_{LDA}$ is dominated by $G_{LDA}^T S_b G_{LDA}$, i.e.,

$$G_{LDA}^T (S_b + S_w) G_{LDA} \simeq G_{LDA}^T S_b G_{LDA}.$$

In this case, the principal axes that PCA gives in the second stage better reflect those of the between-cluster matrix of the first-stage result, $G_{LDA}^T S_b G_{LDA}$, and they may in turn discriminate the clusters clearly in the 2-dimensional space. In this sense, LDA followed by PCA achieves a clear cluster structure as well as a good approximation of the first-stage result.

Table 3: Description of data sets.

	GAUSSIAN	MEDLINE	NEWSGROUPS	REUTERS
Original dimension, m	1100	22095	16702	3907
Number of data items, n	1000	500	770	800
Number of clusters, k	10	5	11	10

2.5.3 OCM followed by PCA

In this method, OCM is applied in the first stage, and trace(S_b) is preserved in the l-dimensional space where $l = k$. In the second stage, PCA is applied in order to obtain the best approximation of the l-dimensional first-stage results in terms of Frobenius/Euclidean norm.

As in Section 5.2, the second-stage dimension reducing matrix H is obtained by solving

$$H_t = \arg \max_{H \in \mathbb{R}^{l \times 2}, H^T H = I} \text{trace}(H^T (G_{OCM}^T S_t G_{OCM}) H), \tag{31}$$

where the solution is the two leading eigenvectors of the total scatter matrix of the first-stage result, $G_{OCM}^T S_t G_{OCM}$.

From Eq. (6), we have

$$G_{OCM}^T S_t G_{OCM} = G_{OCM}^T (S_b + S_w) G_{OCM}. \tag{32}$$

Unlike LDA, OCM does not minimize trace($G^T S_w G$) as shown in Eq. (18). Therefore the following may not be the case:

$$\text{trace}(G_{OCM}^T S_b G_{OCM}) \gg \text{trace}(G_{OCM}^T S_w G_{OCM}),$$

which means that $G_{OCM}^T S_b G_{OCM}$ does not necessarily dominate $G_{OCM}^T S_t G_{OCM}$. Then the two principal axes of $G_{OCM}^T S_t G_{OCM}$ obtained by PCA in the second stage tend to fail to reflect those of $G_{OCM}^T S_b G_{OCM}$, which may rather scatter the data points within each cluster, eventually preventing the visualization results from showing a clear cluster structure.

2.5.4 Rank-2 PCA on S_b

In this method, OCM is applied in the first stage, and trace(S_b) is preserved in the l-dimensional space where $l = k$. In the second stage, the same criterion $J_b(H)$ is used to reduce the l-dimensional first-stage results to 2-dimensional data.

The second-stage dimension reducing matrix H is obtained by solving

$$H_b = \arg \max_{H \in \mathbb{R}^{l \times 2}, H^T H = I} \text{trace}(H^T (G_{OCM}^T S_b G_{OCM}) H), \tag{33}$$

where the solution is the two leading eigenvectors of the between-scatter matrix of the first-stage result, $G_{OCM}^T S_b G_{OCM}$. The columns of G_{OCM} form the subspace spanned by centroids, and this subspace includes the range space of S_b. Accordingly, one can easily show that the eigenvector $u_i^Y \in \mathbb{R}^{l \times 1}$ of $G_{OCM}^T S_b G_{OCM}$ is related to eigenvectors $u_i \in \mathbb{R}^{m \times 1}$ of S_b as

$$u_i^Y = G_{OCM}^T u_i$$

with their corresponding eigenvalues matched as well, i.e., $\lambda_i^Y = \lambda_i$. Hence, the solution of Eq. (33) can be written as

$$H_b = \begin{bmatrix} u_1^Y & u_2^Y \end{bmatrix} = G_{OCM}^T \begin{bmatrix} u_1 & u_2 \end{bmatrix}. \tag{34}$$

Using Eq. (34) and the relationship shown in Eq. (26), the single-stage dimension reducing transformation V_b can be built as

$$\begin{aligned} V_b^T &= H_b^T G_{OCM}^T = \begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix} G_{OCM} G_{OCM}^T \\ &= \begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix} \qquad (35) \\ &= \arg \max_{V \in \mathbb{R}^{m \times 2}} J_b(V) \\ &= \arg \max_{V \in \mathbb{R}^{m \times 2}} \text{trace}(V^T S_b V). \qquad (36) \end{aligned}$$

Eq. (35) holds since the eigenvectors of S_b, u_1 and u_2, are in the range space of G_{OCM}. The criterion of Eq. (36) has been used in one of the successful visual analytic systems, IN-SPIRE, for 2D representation of document data [138].

The discussed two-stage methods are summarized in Table 2.

2.6 Experiments

In this section, we present visualization results using the proposed methods for several data sets, especially focusing on undersampled text data visualization where the data item is represented in m-dimensional space and the number of the data items n is less than m ($m > n$).

2.6.1 Regularization on LDA for undersampled data

In undersampled cases, the LDA criterion shown in Eq. (14) cannot be applied directly because S_w is singular. In order to overcome this singularity problem, Howland et al. proposed a universal algorithmic framework of LDA using the generalized singular value decomposition (LDA/GSVD) [68, 67]. Specifically, for the case when $m \gg n \gg k$, which is usual for most undersampled problems, LDA/GSVD gives the solution for G such that $G^T S_w G = 0$ while maintaining the maximum value of trace($G^T S_b G$). This solution makes sense since LDA criterion is formulated to minimize trace($G^T S_w G$). However, it means that all of the data points belonging to a specific cluster are represented as a single point in the reduced dimensional space, which lessens the generalization ability for classification as well as for visualizing the individual data relationship within each cluster.

On the contrary, the fact that LDA makes $G^T S_w G = 0$ can be viewed as an advantage for visualization purposes since LDA has the capability to fully minimize trace($G^T S_w G$). By means of regularization on S_w one can control trace($G^T S_w G$), which determines the scatter of the data points within each cluster. In regularized LDA which was originally proposed to avoid the singularity of S_w in classification

context, S_w is replaced by a nonsingular matrix $S_w + \gamma I$ where I is an identity matrix, and γ is a control parameter. In general, as γ is increased, the within-cluster distance, trace($G^T S_w G$), also becomes larger with data points being more scattered around their corresponding centroids. As γ is decreased, the within-cluster distance becomes smaller, and the data points gather closer around their centroids. Such manipulation of γ can be exploited in a visualization context because one can choose an appropriate value of γ so that the second-stage method such as PCA, which maximizes trace($G^T S_t G$) = trace($G^T S_b G + G^T S_w G$), does not focus too much on trace($G^T S_w G$). The results that follow are based on such choices of γ.

2.6.2 Data Sets

The data sets tested are composed of one artificially-generated Gaussian-mixture dataset (GAUSSIAN) and three real-world text data sets (MEDLINE, NEWSGROUPS, and REUTERS) that are clustered based on their topics. All the text documents are encoded as term-document matrices where each dimension corresponds to a particular word, and the value of a certain dimension represents the frequency of the corresponding word shown in the document. Each data set is set to have an equal number of data per cluster, and have a mean of zero which is attained by subtracting the global mean. (See Section 6.3.)

The descriptions of data sets, which are also summarized in Table 3, are as follows.

The GAUSSIAN data set is a randomly generated Gaussian mixture with 10 clusters. Each cluster is made up of 100 data vectors, which add up to 1000 in total, and the dimension is set to 1100, which is slightly more than the number of the data items. In its visualization shown in Fig. (1), the clusters are labeled using letters as

- 'a', 'b', . . . , and 'j'.

The MEDLINE data set is a document corpus related to medical science from the

National Institutes of Health[1]. The original dimension is 22095, and the number of clusters is 5, where each cluster has 100 documents. The cluster labels that correspond to the document topics are shown as

- heart attack ('h'), colon cancer ('c'), diabetes ('d'), oral cancer ('o'), and tooth decay ('t'),

where the letters in parentheses are used in the visualization shown in Fig. (2).

The NEWSGROUPS data set [11] is a collection of newsgroup documents, and originally composed of 20 topics. However, we have chosen 11 topics for visualization, and each cluster is set to have 70 documents. The original dimension is 16702, and the cluster labels are shown as

- comp.sys.ibm.pc.hardware ('p'), comp.sys.mac.hardware ('a'), misc.forsale ('f'), rec.sport.baseball ('b'), sci_crypt ('y'), sci.electronics ('e'), sci.med ('d'), soc.religion.christian ('c'), talk.politics.guns ('g'), talk.politics.misc ('p'), and talk.religion.misc ('r'),

where the letters in parentheses are used in the visualization shown in Fig. (3).

The REUTERS data set [11] is the document collection that appeared in the Reuters newswire in 1987, and originally composed of hundreds of topics. Among them, 10 topics related to economic subjects are chosen for visualization, and each cluster has 80 documents. The original dimension is 3907, and the cluster labels are shown as

- earn ('e'), acquisitions ('a'), money-fx ('m'), grain ('g'), crude ('r'), trade ('t'), interest ('i'), ship ('s'), wheat ('w'), and corn ('c'),

where the letters in parentheses are used in the visualization shown in Fig. (4).

2.6.3 Effects of Data Centering

Fig. 5 is the example of applying OCM+PCA to the MEDLINE data set with and without data centering. Once the MEDLINE data set is encoded as a term-document

[1]http://www.cc.gatech.edu/~hpark/data.html

matrix, every component has a non-negative value, which results in the global centroid that is significantly far from the origin. Then performing PCA without data centering might give the first principal axis as the one reflecting the global centroid rather than that discriminating clusters. If we consider projecting the data onto each of the horizontal and the vertical axes in Fig. 5, the former, which corresponds to the first principal axis, does not help in showing the cluster structure clearly, and only the vertical axis, which corresponds to the second principal axis from PCA, discriminates clusters. We have found that such undesirable behavior is common in many cases without data centering, which is why we assume that data is centered throughout this chapter. Accordingly, all the results shown in Figs 1-4 are obtained after data centering.

2.6.4 Comparison of Visualization Results

The results of four two-stage methods for the tested data sets are shown in Figs.1-4[2].

In all cases, LDA-based methods show cluster structures more clearly than OCM-based methods. This proves the effectiveness of LDA that considers both within- and between-cluster measures while OCM only takes into account the latter. Due to this difference, OCM generally produces a widely-scattered data representation within each cluster. As a result, in the NEWSGROUPS dataset, such a wide within-cluster variance significantly deteriorates the cluster structure visualization even if OCM still attempts to maximize the between-cluster distance.

In the MEDLINE and the REUTERS data sets, all of the four methods produce relatively similar results. However, we have controlled the within-cluster variance in LDA-based methods using the regularization term γI. In addition, the fact that rank-2 LDA and LDA+PCA produce almost identical results indicates that $G_{LDA}^T S_t G_{LDA}$ is dominated by $G_{LDA}^T S_b G_{LDA}$ after LDA is applied in the first stage as we expected.

[2]Those figures can be arbitrarily magnified without losing the resolution in the electronic version of this chapter.

Rank-2 LDA represents each cluster most compactly by minimizing the within-cluster radii both in the first and the second stage. However, it may reduce the between-cluster distances as well because $J_{b/w}$ maximizes the conceptual ratio of two scatter measures. As can be seen in the two LDA-based methods applied to the NEWGROUPS data set, while rank-2 LDA minimizes the within-cluster radii, it also places the centroids closer to each other as compared to those in LDA+PCA. Due to this effect, which one is preferable between rank-2 LDA and LDA+PCA depends on the data set to be visualized.

Overall, OCM+PCA and Rank-2 PCA on S_b show similar results. It means $G^T S_b G \simeq G^T S_t G$ in that the difference between two methods lies in whether PCA is applied to $G^T S_b G$ or $G^T S_t G$ in the second stage. Since performing PCA on $G^T S_b G$ is computationally more efficient than PCA on $G^T S_t G$, Rank-2 PCA on S_b can be a good alternative to OCM+PCA in case efficient computation is important.

Finally, these visualization results reveal the interesting cluster relationships underlying in the data. In Fig. (2), the clusters for colon cancer ('c') and oral cancer ('o') are shown close to each other. In Fig. (3), the clusters of soc.religion.christian ('c') and talk.religion.misc ('r'), those of comp.sys.ibm.pc.hardware ('p') and comp.sys.mac.hardware ('a'), and those of sci.crypt ('y') and sci.med ('d') are closely located respectively in LDA-based methods. In addition, the two clusters, misc.forsale ('f') and rec.sport.baseball ('b'), are shown to be the most distinctive, which makes sense because those topics are quite irrelevant to the others. In Fig. (4), the clusters of grain ('g'), wheat ('w'), and corn ('c') as well as those of money-fx ('m') and interest ('i') are visualized very close.

2.7 Conclusions

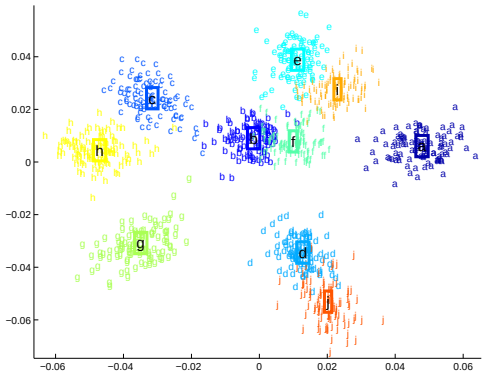
According to our results, LDA-based methods are shown to be superior to OCM-based methods since both within- and between-cluster relationships are taken into account

in LDA. Especially, combined with PCA in the second stage, LDA+PCA achieves a clear discrimination between clusters as well as the best approximation of the results of LDA when the distance between data is measured in terms of Frobenius/Euclidean norm.

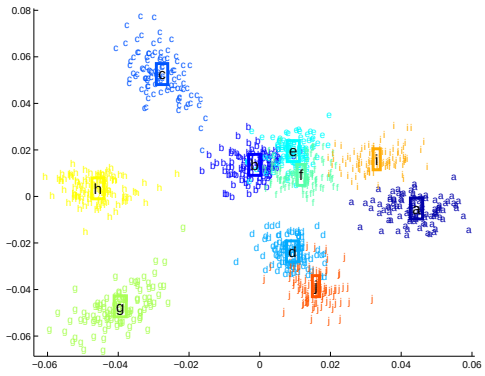
However, many classes except for few of them that are clearly unrelated tend to be overlapped especially when dealing with large numbers of data points and clusters. This is inherently due to the nature of the second-stage dimension reduction in which only the two axes are chosen so that the classes which contribute most to the second stage criteria can be well-discriminated. Such behavior can exaggerate the distances between particular clusters, and more elaboration towards new criteria that fits in visualization is required. In the MEDLINE and the REUTERS datasets, visualization results seem to have a tail-shape along specific directions. We often found this phenomenon to occur in many other data sets. It is still unclear as to what causes this and how it affects the visualization, e.g. characteristics of information loss in the second stage. Finally, in order to determine how much loss of information is introduced by each method, more rigorous analysis based on various quantitative measures such as pairwise between-cluster distance and within-cluster radii should be conducted.

Figure 1: Comparison of the two-stage methods in the GAUSS data set.

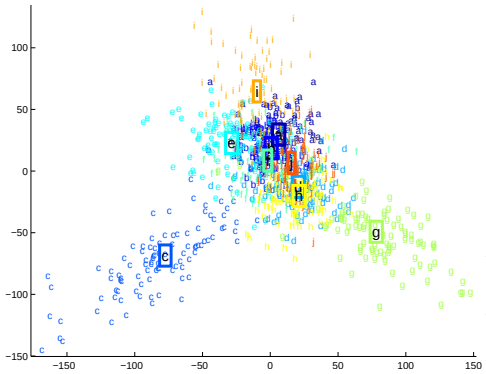
(a) Rank-2 LDA



(b) LDA+PCA



(c) OCM+PCA



(d) Rank-2 PCA on S_b

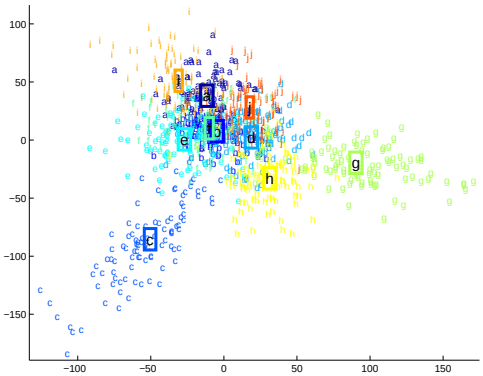
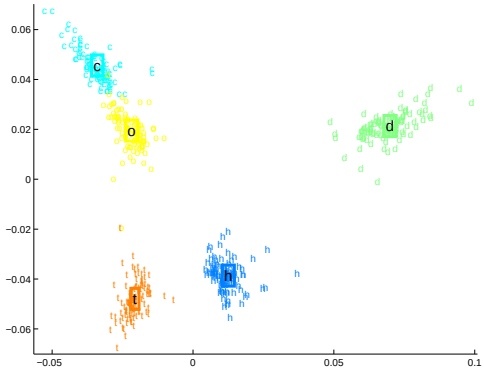
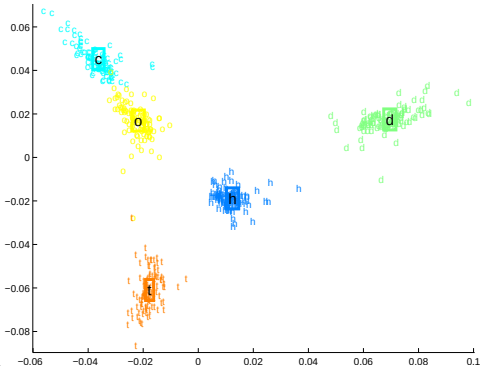


Figure 2: Comparison of the two-stage methods in the MEDLINE data set.

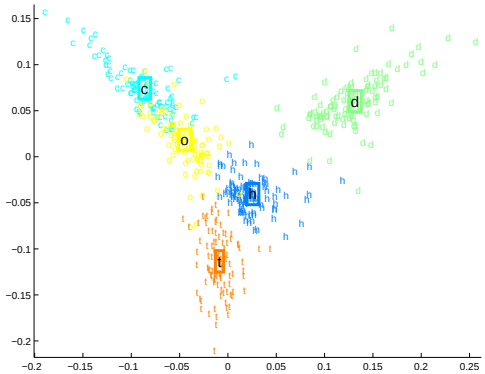
(a) Rank-2 LDA



(b) LDA+PCA



(c) OCM+PCA



(d) Rank-2 PCA on S_b

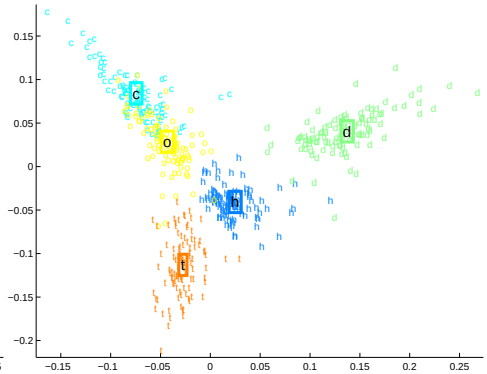


Figure 3: Comparison of the two-stage methods in the NEWSGROUPS data set.

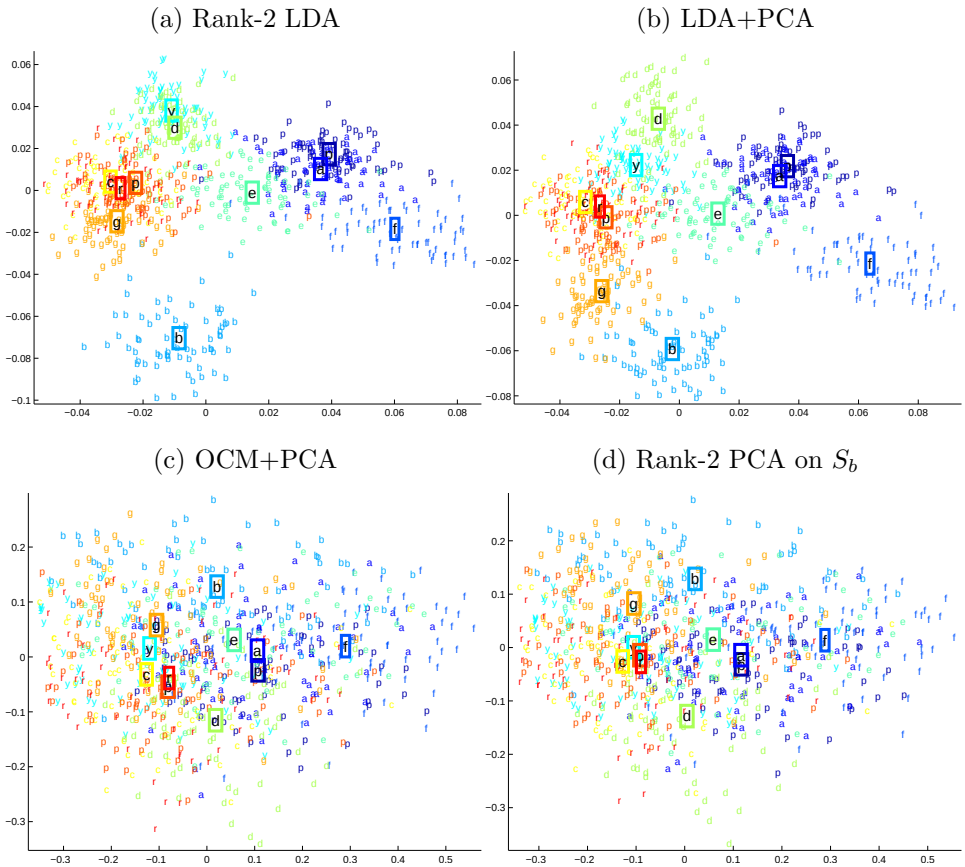


Figure 4: Comparison of the two-stage methods in the REUTERS data set.

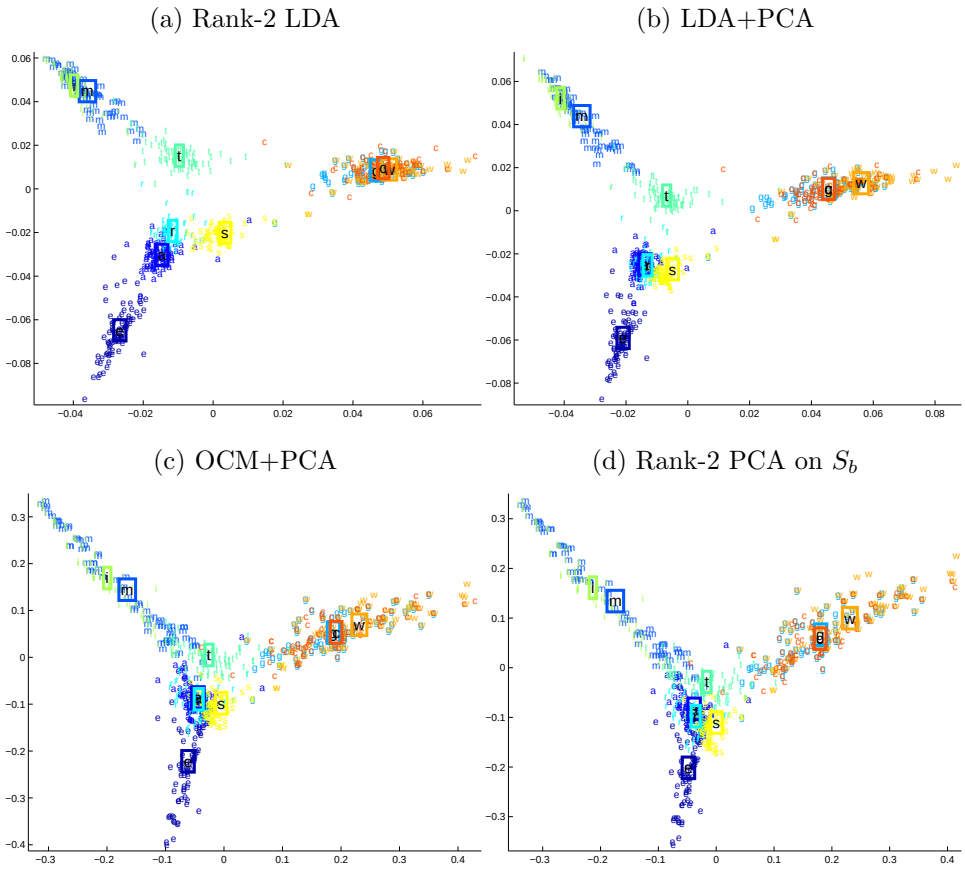
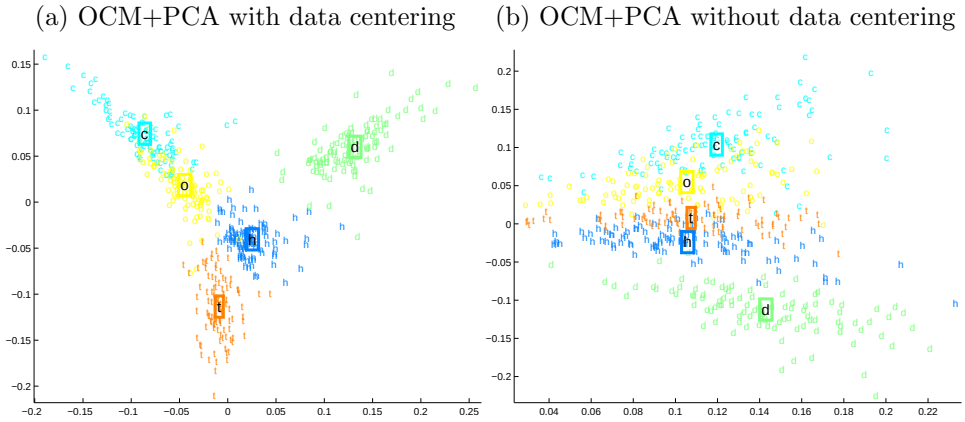


Figure 5: Example of effects of data centering in the MEDLINE data set.



CHAPTER III

EFFICIENT UPDATING OF COMPUTATIONAL METHODS DUE TO PARAMETER CHANGES

One of the most widely-used nonlinear data embedding methods is ISOMAP. Based on a manifold learning framework, ISOMAP has a parameter k or ϵ that controls how many edges a neighborhood graph has. However, a suitable parameter value is often difficult to determine because of a time-consuming optimization process based on certain criteria, which may not be clearly justified. When ISOMAP is used to visualize data, users might want to test different parameter values in order to gain various insights about data, but such interaction between humans and such visualizations requires reasonably efficient updating, even for large-scale data. To tackle these problems, we propose an efficient updating algorithm for ISOMAP with parameter changes, called p-ISOMAP. We present not only a complexity analysis but also an empirical running time comparison, which show the advantage of p-ISOMAP. We also show interesting visualization applications of p-ISOMAP and demonstrate how to discover various characteristics of data through visualizations using different parameter values.

3.1 Motivation

One of the most widely-used data mining techniques that reduce noise in data and improve efficiency in terms of computation time and memory usage is dimension reduction. Recently, nonlinear dimension reduction techniques, which have been actively investigated, revealed the underlying nonlinear structure in data. Such nonlinearity is often considered as a curvilinear manifold with a much lower dimension than that

in the original high-dimensional space. Among the most recent nonlinear dimension reduction methods, isometric feature mapping (ISOMAP) has shown its effectiveness in capturing the underlying manifold structure in the reduced dimensional space by being successfully applied to synthetic data such as "Swiss roll" data and real-world data such as facial image data [125].

ISOMAP shares the basic idea with a traditional technique, classical multidimensional scaling (MDS). Classical MDS first constructs the pairwise similarity matrix, which is usually measured by the Euclidean distance, and computes the reduced dimensional mapping that maximally preserves such a similarity matrix in a given reduced dimension. ISOMAP differs from classical MDS in that it constructs the pairwise similarity matrix based on the geodesic distance estimated by the shortest path in the neighborhood graph of data. The neighborhood graph is formed by having vertices as data points and setting each edge weight between the nodes as their Euclidean distance only if at least one node is one of the k-nearest neighbors (k-NN) of the other node (k-ISOMAP) or if their Euclidean distance is smaller than ϵ (ϵ-ISOMAP). Hence, ISOMAP has an either parameter k or ϵ to construct the neighborhood graph.

This chapter focuses on the algorithm and applications of the dynamic updating of ISOMAP when the value of k or ϵ varies. It is generally known that in ISOMAP, if k or ϵ is too small, the graph becomes sparse, resulting in infinite geodesic distances between some pairs of data points, and if k or ϵ is too large, it is prone to "short circuit" the true geometry of the manifold. However, it is not always easy to figure out which value of k or ϵ is appropriate for the data at hand. One way of optimizing these parameters is using certain quantitative measures such as residual variance [12, 125, 112] and finding the "elbow" point at which the residual variance curve stops decreasing significantly as the parameter value changes. However, running ISOMAP repeatedly using different parameter values for k or ϵ may be time-consuming since

it involves computationally intensive processes such as the all-pairs shortest path computation and the eigendecomposition, whose complexity is usually $O(n^3)$ in which n is the number of data points.[1]

In practice, there is also often no guarantee of the existence of the underlying well-defined manifold structure in data, and thus, one may not be sure if manifold learning methods such as ISOMAP are suitable for the data at hand. Even so, one may still want to try ISOMAP or another manifold learning method in order to see if it serves one's purpose. In this case, however, it may not be a good idea to rely on a particular value of k or ϵ to achieve a reasonable dimension reduction since the optimal value tends to be indistinct in terms of a certain measure. When it comes to the visualization of high-dimensional data in two- or three-dimensional space, we can acquire different insights on the data by using various dimension reduction techniques [35]. This statement also holds true even when we use just a single dimension reduction method, e.g., ISOMAP, while we test its various parameter values. In short, visualizations using ISOMAP with different parameter values for k or ϵ can provide us with various aspects of our data. In instances of the "Swiss roll" and toroidal helix data sets shown in Fig. 6, one may want to visualize them based on the unfolded version of its manifold, as shown in Figs. 6(b) and 6(f), but sometimes one may also want to see how the underlying manifold is curved in the original space, i.e., the curvature of the manifold itself, as shown in Figs. 6(d) and 6(h).[2] It is also possible that visualizations of the transition between these two cases, shown in Figs. 6(c) and 6(g), imply different insight about data. In this sense, it is worthwhile for users to test different parameter values in ISOMAP to visualize data in various ways.

[1]The complexity of the (all-pairs) shortest path computation depends on the algorithm. Floyd-Warshall algorithm requires $O(n^3)$ computations while Dijkstra's algorithm does $O(|e|n \log n)$ computations [13] in which $|e|$ is the number of edges.

[2]It may not be possible to visualize the manifold curvature perfectly without using the original dimension, but at least we can obtain some insights about it from a lower dimensional visualization.

Such visualizations, however, should provide users with smooth and prompt interaction that requires fast and efficient computations of the results. In other words, when users change the parameter value, if they have to wait for a significant amount of time, then such interaction would not be practical. Motivated by the above mentioned cases, we propose p-ISOMAP, an efficient dynamic updating algorithm for ISOMAP when the parameter value changes. Given the ISOMAP result from a particular parameter value, our proposed algorithm updates the previous result to obtain another ISOMAP result of the same data with a new parameter value instead of recomputing ISOMAP for different parameter values from scratch. We present the complexity analysis of our algorithms as well as the experimental comparison of their computation times. In addition, we demonstrate several visualization examples by varying the parameters in ISOMAP, which not only show the interesting aspects of the tested data but also help us thoroughly understand the behavior of ISOMAP in terms of parameter values.

The rest of this chapter is organized as follows. Section 3.2 briefly introduces ISOMAP and its algorithm, and Section 3.3 discusses previous work related to p-ISOMAP. Section 3.4 describes the algorithmic details and the complexity analysis of p-ISOMAP, and Section 3.5 presents not only the experimental results that compare the computation times of ISOMAP and p-ISOMAP but also interesting visualization examples of real-world data using p-ISOMAP. Finally, Section 3.6 concludes our work.

3.2 ISOMAP

Given a set of data points represented as M-dimensional vectors $x_i \in \mathbb{R}^M$ for $i = 1, \ldots, n$, ISOMAP assumes a lower dimensional manifold structure in which the data are embedded. It yields the m-dimensional representation of x_i as $y_i \in \mathbb{R}^m$ ($m \ll M$) such that the Euclidean distance between y_i and y_j approximates their geodesic distance along the underlying manifold as much as possible. Such an approximation

builds on the classical MDS framework, but unlike MDS, ISOMAP has the capability of handling nonlinearity existing in the original space since a geodesic distance reflects an arbitrary curvilinear shape of the manifold. On input, ISOMAP takes a data matrix $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{M \times n}$, a reduced dimension m, and a parameter k or ϵ. The algorithm is composed of three steps:

1. *Neighborhood graph construction.* ISOMAP first computes the pairwise Euclidean distance matrix, $D_X \in \mathbb{R}^{n \times n}$, in which $D_X(i, j)$ is the Euclidean distance between x_i and x_j. Then it determines the set of neighbors for each point either by k-nearest neighbors or by those within a fixed radius ϵ. Between a point x_i and each of its neighbors x_j, an edge $e(i, j)$ is assigned with a weight equivalent to their Euclidean distance, and in this way, ISOMAP forms a weighted undirected neighborhood graph $G = (V, E)$, where the vertices in V correspond to the data points x_i's.

2. *Geodesic distance estimation.* In the second step, ISOMAP estimates the pairwise geodesic distance based on the shortest path length for every vertex pair along the neighborhood graph G, which is represented as a matrix $D_G \in \mathbb{R}^{n \times n}$ in which $D_G(i, j)$ is the shortest path length between x_i and x_j in G.

3. *Lower dimensional embedding.* The final step performs classical MDS on D_G, producing m-dimensional data embedding, $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} \in \mathbb{R}^{m \times n}$. First, the pairwise geodesic distance matrix D_G is converted to an inner product matrix B_G as

$$B_G = -\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathbf{T}}\right) D_G.^2 \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathbf{T}}\right)/2, \tag{37}$$

in which $I \in \mathbb{R}^{n \times n}$ is an identity matrix, $\mathbf{1} \in \mathbb{R}^{n \times 1}$ is a vector whose elements are all 1's, and $D_G.^2$ is an element-wise squared D_G. Classical MDS solves Y such that it minimizes $E = \|B_G - Y^T Y\|$, where the matrix norm $\|\cdot\|$ is

Table 4: Notations used in this Chapter

Notation	Description
n	Number of data points
M	Original dimension
m	Reduced dimension
x_i	Input data vector, $i = 1, \ldots, n$
y_i	Reduced dimensional vector of x_i
D_X	Euclidean distance matrix of x_i's
$\overrightarrow{G}$	Directed neighborhood graph
G	(Undirected) neighborhood graph
q	Maximum degree of the vertices in G
D_G	Shortest path length matrix in G
B_G	Inner product matrix obtained from D_G
$\mathcal{A}$	Set of edges to be inserted in G
$\mathcal{D}$	Set of edges to be removed in G
Δe_i	Set of inserted/removed edges of x_i
F	Set of affected vertex pairs by $\mathcal{A}$ or $\mathcal{D}$
P	Predecessor matrix
H	Hop number matrix
k and k^{new}	Previous and new parameter values

either a Frobenius or Euclidean norm. Such a solution of Y is obtained by the eigendecomposition of B_G as

$$Y = \left[\begin{array}{cccc} \sqrt{\lambda_1} v_1 & \sqrt{\lambda_2} v_2 & \cdots & \sqrt{\lambda_m} v_m \end{array} \right]^T, \tag{38}$$

where $\lambda_1, \ldots, \lambda_m$ are the m largest eigenvalues of B_G, with corresponding eigenvectors $v_1, \ldots, v_m$.

3.3 Related Work

Based on the algorithmic details of ISOMAP described in Section 2, if the parameter k or ϵ varies, it would change the topology of the neighborhood graph in the first step. Such a change can be interpreted as either an insertion of new edges or a removal of existing edges in the neighborhood graph. The inserted or removed edges would in turn influence the shortest path length matrix D_G. Solving the updated D_G can be viewed as a dynamic shortest path problem in which we need to update the

existing shortest path and its corresponding length due to graph changes. Generally, a dynamic shortest path problem includes all the various situations that involve not only vertex insertion/removal but also real-valued edge weight changes rather than just edge insertion/removal, and depending on what types of changes in the graph are assumed in the algorithm, it maintains a variety of additional information such as the candidates of the future shortest paths for an efficient shortest path update [25, 47, 59]. In the context of manifold learning methods, several approaches have tried to dynamically update ISOMAP embedding for incremental data input such as a data stream [86, 87]. Similar to the parameter changes in p-ISOMAP, incremental data cause topology changes in the neighborhood graph, which can be fully expressed by edge insertion/removal, so previous studies[86, 149] have discussed the dynamic shortest path updating algorithms that can deal with such types of graph changes. However, the characteristic in terms of graph changes differs greatly between p-ISOMAP and incremental ISOMAP. First, each parameter change in p-ISOMAP involves only the form of either edge insertion or removal in p-ISOMAP while a new data point causes both at once in incremental ISOMAP. In this sense, one may regard the shortest path updating in p-ISOMAP as simpler than that in incremental ISOMAP. However, graph changes in incremental ISOMAP primarily result from a new data item, and thus, an inserted edge in incremental ISOMAP is always connected to the new data point once an edge of a certain vertex is deleted. Furthermore, when the parameter k is used, the number of inserted or removed edges in p-ISOMAP is roughly $O(n|\Delta k|)$, where n is the number of data points and $\Delta k = k^{new} - k$, whereas that in incremental ISOMAP is roughly $O(k)$, which is much smaller than that in p-ISOMAP. This difference implies that even a small change in parameter values in p-IOSMAP can lead to a significant change in the neighborhood graph and its all-pairs shortest path results. However, the graph change in incremental ISOMAP is still minor compared to the entire graph size. Considering such different behaviors, we present our own shortest

path updating algorithm that is appropriate for p-ISOMAP in Section 4. After the shortest path update, one needs to update the eigendecomposition results on a new matrix B_G, shown in Eq. (37). In general, the eigendecomposition update is done by formulating the change in B_G in a certain form, e.g., a rank-1 update [26]. In incremental ISOMAP, [86] applied an approximation technique called the Rayleigh-Ritz method [61, 48] based on a variant of the Krylov subspace in computing the eigendecomposition. However, this method is limited to the case when the reduced dimension m is fairly large and the eigendecomposition does not change significantly. However, when p-ISOMAP is used in a visual analytics system, which is one of our main motivations, it requires m to be a small value such as two or three. Furthermore, such approximation methods perform poorly in p-ISOMAP since it involves $O(n)$ graph changes and the corresponding large amount of the shortest path update. Hence, we focus on the exact solution for p-ISOMAP. In the next section, we present a novel algorithmic framework for p-ISOMAP.

3.4 p-ISOMAP

p-ISOMAP assumes the original ISOMAP result is available for a particular parameter value. Given a new parameter value, the algorithm performs three steps: the neighborhood graph update, the shortest path update, and the eigenvalue/vector update.

3.4.1 Neighborhood Graph Update

In this step, p-ISOMAP computes the set of edges to insert or remove from the previous neighborhood graph and update the neighborhood graph by applying such changes. In order to compute these edges efficiently, each data point maintains the sorted order of the other points in terms of its Euclidean distances to them. In this way, p-ISOMAP identifies which neighbor points of a particular point are to be added or deleted in $O(1)$ time. If a neighborhood graph is constructed by the parameter ϵ,

Algorithm 1 neighborhood graph update for a new k

Input: the new value of k , the directed neighborhood graph $\overrightarrow{G}$, and the undirected one G **Output:** the set of inserted edges $\mathcal{A}$ or that of removed ones $\mathcal{D}$ in G, updated $\overrightarrow{G}$ and G

```
for all data point x_i do
  for all newly added/deleted neighbor x_j do
    Assign/Remove an edge e(i, j) from x_i to x_j in G→.
  end for
end for
Initialize A := ∅ / D := ∅.
for all inserted/removed edge e(i, j) in G→ do
  if e(i, j) is an inserted edge then
    if e(i, j) is not in G then
      A ← A ∪ {e(min(i, j), max(i, j))}
      Assign the edge e(i, j) in G.
    end if
  else {e(i, j) is a removed edge}
    if e(j, i) is not in G→ then
      D ← D ∪ {e(min(i, j), max(i, j))}
      Remove the edge e(i, j) in G.
    end if
  end if
end for
```

the neighborhood relationship is symmetric, i.e., if and only if x_i is a neighbor of x_j, x_j is also a neighbor of x_i for a particular ϵ. Thus the added or deleted neighborhood pairs are equivalent to the inserted or removed edges in a neighborhood graph, and the algorithm is straightforward. On the other hand, if a neighborhood graph is constructed by k-NN, the situation becomes complex since it is possible that x_i is a neighbor of x_j, but x_j is not a neighbor of x_i, which we call a one-sided neighborhood. By considering such a one-sided relationship, a directed neighborhood graph $\overrightarrow{G}$ is initially made, and ISOMAP obtains its undirected one G by an OR operation, i.e., for x_i and x_j, if at least either one is a neighbor of the other, then ISOMAP assigns an edge with the weight equal to their Euclidean distance. In p-ISOMAP, both directed and undirected graphs are maintained and updated in an orderly manner so as to

avoid ambiguity about which changes of neighbors in a directed graph cause actual edge changes in an undirected one in which we have to actually compute the shortest paths. The detailed procedure of the neighborhood graph update are described in Algorithm 1. As an output, it produces the set of effectively inserted/removed edges, which is, in turn, used in the shortest path update stage.

3.4.1.1 Time Complexity

In ISOMAP, the time complexity in constructing a neighborhood graph is as follows. It starts with a sort operation for a given data set whose time complexity is $O(n^2 \log n)$. Then obtaining $\overrightarrow{G}$ and G requires $O(nq)$, in which q is the maximum degree of vertices in the graph G. In p-ISOMAP, the time complexity required in the neighborhood graph update is bounded by $O(n \cdot \max_i |\Delta e_i|)$, in which $|\Delta e_i|$ is the number of inserted/removed edges associated with x_i.

3.4.2 Shortest Path Update

The shortest path update stage, which is one of the most computationally intensive steps in p-ISOMAP, takes the input as either $\mathcal{A}$ or $\mathcal{D}$ and updates the shortest path length matrix D_G. In order to facilitate this process, p-ISOMAP maintains and updates the information about the shortest path itself with a minimal memory requirement in the form of a predecessor matrix $P \in \mathbb{R}^{n \times n}$, in which $P(i, j)$ stores the node index immediately preceding x_j in the shortest path from x_i to x_j.[3] For instance, if the shortest path from x_1 to x_2 is composed of $x_1 \to x_4 \to x_3 \to x_2$, then we set $P(1, 2) = 3$. For the shortest path update, p-ISOMAP performs two steps:

1. It identifies the set, F, of the "affected" vertex pairs, whose shortest paths need to be recomputed due to the inserted edges in $\mathcal{A}$ or the removed edges in $\mathcal{D}$.

[3]Here we assume the shortest path is unique for every vertex pair, which is almost always the case in ISOMAP.

Algorithm 2 Shortest path update for $\mathcal{A}$

Input: the updated neighborhood graph G, the shortest path length matrix D_G, the predecessor matrix P, and the set of inserted edges $\mathcal{A}$ **Output:** updated D_G and P

```
1: for all inserted edge e(a, b) in A do
2:   for all data point x_i do
3:     Unmark all the other nodes except for x_i.
4:     if D_G(i, b) + G(a, b) < D_G(i, a) then
5:       D_G(i, a) ← D_G(i, b) + G(a, b)
6:       P(i, a) ← b
7:       D_G(a, i) ← D_G(i, a)
8:       if b = i then
9:         P(a, i) ← a
10:      else
11:        P(a, i) ← P(b, i)
12:      end if
13:    end if{Traverse T(i, a)}
14:    Initialize an empty queue Q.
15:    Q.enqueue(a)
16:    while Q is not empty do
17:      t := Q.pop
18:      Mark x_t.
19:      for all unmarked node x_j adjacent to x_t do
20:        if D_G(i, t) + G(t, j) < D_G(i, j) then
21:          D_G(i, j) ← D_G(i, t) + G(t, j)
22:          P(i, j) ← t
23:          D_G(j, i) ← D_G(i, j)
24:          P(j, i) ← P(t, i)
25:          Q.enqueue(j)
26:        else
27:          Mark x_j.
28:        end if
29:      end for
30:    end while
31:  end for
32: end for
```

2. Then it computes their shortest paths based on the information of the rest of the vertex pairs and the newly updated neighborhood graph, which usually performs significantly faster than the original shortest path computation from scratch.

3.4.2.1 Shortest Path Update with Inserted Edges due to an Increasing Parameter

The main idea in the shortest path update due to an inserted edge $e(a,\, b)$ is that if $D_G(i,\, a) + e(a,\, b) + D_G(b,\, j)$ or $D_G(i,\, b) + e(a,\, b) + D_G(a,\, j)$ is shorter than $D_G(i,\, j)$, then $D_G(i,\, j)$ is to be replaced by the smaller one between the two new path lengths along with the corresponding update of P. Performing this comparison for all pairs of vertices would require the time complexity of $O(n^2|\mathcal{A}|)$, in which $|\mathcal{A}|$ is the number of edges in $\mathcal{A}$. Unlike incremental ISOMAP or other situations in which $|\mathcal{A}|$ is relatively small and constant, p-ISOMAP has $|\mathcal{A}| \simeq O(n)$ due to an increasing parameter, which makes the complexity of the above computation roughly equal to $O(n^3)$. Such complexity is no better than the Floyd-Warshall algorithm used in the original ISOMAP. Thus, the algorithm has to find the computational gain while identifying the subset, F, of the entire vertex pair set and applying the above comparison only in this set. For construction of F, the shortest path can conveniently be interpreted as a form of a tree in which $T(i)$ is the shortest path tree that has x_i as its root. The subtree $T(i; a)$ of $T(i)$ can then be defined as one with a root at x_a. Using a well-known property that any subpaths of the shortest path are also the shortest path, once a new shortest path from a particular vertex x_i to x_a is found using $e(a,\, b)$, one can traverse $T(i; a)$ in various ways such as a breath-first-search or a depth-first-search method and correspondingly update the shortest paths from x_i to the nodes in $T(i; a)$. In p-ISOMAP, we have used the breath-first-search, the detailed algorithm of which is summarized in Algorithm 2. In fact, such approaches using subtree traversal for inserted edges were applied in many applications [100, 149]. However, the algorithm presented here was found simpler and faster since it deals with both directional paths at once when updating D_G and P.

Algorithm 3 Identification of F due to $\mathcal{D}$

Input: the removed edge set $\mathcal{D}$, the predecessor matrix P, and the hop number matrix H **Output:** the set of the affected vertex pairs F

```
1: α := max_{i,j} H_ij
2: Initialize a linked list l[h] for h = 1, ..., α
3: Unmark all vertex pairs (x_i, x_j) such that i < j
4: for all vertex pair (x_i, x_j) such that i < j do
5:    Insert (x_i, x_j) to l[H(i, j)].
6: end for
7: for h := α to 1 do
8:    for all Unmarked vertex pair (x_i, x_j) in l[h] do
9:       Set p[k] for k = 1, , ..., h as k-th node found in the shortest path from x_j
         to x_i
10:      for k := 1 to h do
11:         m[k] := max_{k≤l≤h}(l) such that a vertex pair (p[k], p[l]) is marked.
12:         Mark vertex pairs (p[k], p[v]) and (p[v], p[k]) for all v such that m[k]+1 ≤
            v ≤ h
13:      end for
14:      q := 1
15:      for k := h − 1 to 1 do
16:         if (p[k], p[k + 1]) ∈ D then
17:            Insert vertex pairs (p[u], p[v]), for all u and v such that q ≤ u ≤ k and
               max(k + 1, m[u] + 1) ≤ k ≤ h, to F.
18:            q ← k + 1
19:         end if
20:      end for
21:   end for
22: end for
```

3.4.2.2 Shortest Path Update with Deleted Edges due to a Decreasing Parameter

When edges are deleted, the vertex pairs in F are those whose shortest paths include any of these deleted edges. The set F can be identified by considering deleted edges one by one and then by performing union operation on such vertex pair sets. This approach is reasonable when $|\mathcal{D}|$ is small and thus little overlap occurs between such vertex pair sets for each deleted edge, which is the case in incremental settings [86, 149]. In contrast, p-ISOMAP has $|\mathcal{D}| = O(n)$, which possibly leads to a large amount of overlap in the affected vertex pairs among different deleted edges; therefore, the above approach results in a significantly redundant computation. For this reason, we

Algorithm 4 Recomputation of the shortest paths for F for a decreasing parameter

Input: the updated graph G, the shortest path length matrix D_G, the predecessor matrix P, the hop number matrix H, and the set of the affected vertex pairs F
Output: updated D_G, P, and H

```
1: Sort vertices in terms of how frequently they appear in F in an increasing order
2: for all x_i in the above sorted order do
3:    Initialize an empty heap Q
4:    C := {x_j|(x_i, x_j) ∈ F}
5:    for all x_j ∈ C do
6:       d := ∞
7:       for all adjacent node x_t of x_j such that x_t ∉ C do
8:          if d > D_G(i, t) + G(t, j) then
9:             d ← D_G(i, t) + G(t, j)
10:            P(i, j) ← t
11:            H(i, j) ← H(i, t) + 1; H(j, i) ← H(i, j)
12:         end if
13:      end for
14:      Insert an entry (d, j) with a key d and an index j to Q
15:   end for
16:   while Q is not empty do
17:      (d, j) :=ExtractMin from Q
18:      D_G(i, j) ← d; D_G(j, i) ← d
19:      Remove (x_i, x_j) from C and F
20:      pred := j
21:      while P(i, pred) ≠ i do
22:         pred ← P(i, pred)
23:      end while
24:      P(j, i) ← pred
25:      for all adjacent node x_t of x_j such that x_t ∈ C do
26:         d := a key of an entry with an index t in Q
27:         if d > D_G(i, j) + G(j, t) then
28:            DecreaseKey (D_G(i, j) + G(j, t), t) in Q
29:            H(i, t) ← H(i, j) + 1; H(j, i) ← H(i, t)
30:         end if
31:      end for
32:   end while
33: end for
```

propose a new algorithm for p-ISOMAP that identifies the affected vertex pairs by handling all the deleted edges at once. The key idea is that for the shortest path between a particular vertex pair, we partition it into multiple subpaths separated by any deleted edges, and then we form Cartesian products between any two such

subpaths and place them in F. For example, when the shortest path is $x_1 \rightarrow x_3 \rightarrow x_2 \rightarrow x_4$, if $e(x_3, x_2)$ is deleted, we add $\{(x_i, x_j)|i \in \{1, 3\}, j \in \{2, 4\}\}$ to F. Furthermore, we enhance the efficiency in this process in the following way. We first consider the vertex pair whose shortest path has the largest number of hops and check all of its subpaths, which are also the shortest paths between their hopping nodes. Then, the checked vertex pairs are not considered again. In other words, by first dealing with the shortest paths that cover as many other shortest paths as possible, we can handle the maximum number of vertex pairs regarding whether they are to be added to F or not. To implement this idea, we maintain a hop number matrix H in which $H(i, j)$ contains the number of hops in the shortest path from vertex i to j, which enables us to prioritize the vertex pair according to its number of hops. In addition, our algorithm takes into account overlapping subpaths between the shortest paths of different vertex pairs. That is, if any subpaths of the shortest path of a certain vertex pair are also those of another vertex pair that has been previously taken care of, the algorithm stops checking such subpaths. In this way, we completely exclude redundant computations in an efficient manner. The detailed algorithm to solve for the set F is described in Algorithm 3. Once F is obtained, the shortest paths are recomputed selectively. This process can be expedited using the available information about the unaffected vertex pairs whose shortest paths remain unchanged. We choose Dijkstra's algorithm as the main algorithm for the shortest path computation since it is suitable for a sparse graph. How to incorporate the above available information in Dijkstra's algorithm is straightforward as described in Algorithm 4. In addition, since Dijkstra's algorithm is a single-source shortest path algorithm, it needs to run n times for each source vertex. In terms of the order of the source vertices on which to run Dijkstra's algorithm, those that have the least number of destination nodes to update are processed first, and the updated vertex pairs are then removed from F. Algorithm 4, which also includes additional functionalities for updating P and H,

summarizes the shortest path update process based on F.

3.4.2.3 Time Complexity

When a parameter increases, Algorithm 2 requires the time complexity of $O(|\mathcal{A}|nq \cdot \max_{i,a}|T(i;a)|)$ in which $\max_{i,a}|T(i;a)|$ is the maximum number of nodes in subtree $T(i;a)$ over all x_i's and inserted edge $e(a, b)$'s. This complexity can be loosely bounded by $O(|\mathcal{A}|q|F|)$ where $|F|$ is the number of affected vertex pairs due to the inserted edges in $\mathcal{A}$. For a decreasing parameter, the time complexity of Algorithm 3 requires $O(n^2)$ computations since it visits every vertex pair exactly once. Now, let us partition the entire vertices into two disjoint sets $V_d(i)$ and $V_d^c(i)$ such that $V_d(i) = \{x_j|(x_i, x_j) \in F\}$ for a certain x_i. Then, the complexity of Algorithm 4 is represented as $O(n \cdot \max_i(|E_i^{'}|\log|V_d(i)| + (|E_i^{''}|))$ in which $E_i^{'} = \{e(x_a, x_b) \in G|x_a, x_b \in V_d(i)\}$ and $E_i^{''} = \{e(x_a, x_b) \in G|x_a \in V_d(i), x_b \notin V_d(i)\}$. In both cases, for small changes in the neighborhood graph, $|F|$ is expected to be much smaller than n^2, which is the maximum possible value of $|F|$.

3.4.3 Eigenvalue/vector Update

Let us denote the updated D_G after the shortest path update described in Section 4.2 as D_G^{new}. In this step, D_G^{new} is first converted into the pairwise inner product matrix B_G^{new} by Eq. (37). To get a lower dimensional embedding as shown in Eq. (38), we need to obtain m eigenvalue/vector pairs $(\lambda_1^{new}, v_1^{new}), \ldots, (\lambda_m^{new}, v_m^{new})$ for B_G^{new}. In this computation, the available information that we can exploit is the previous m eigenvalue/vector pairs $(\lambda_1, v_1), \ldots, (\lambda_m, v_m)$ of B_G. In fact, they can be good initial guesses for m eigenvalue/vector pairs for B_G^{new}, assuming the two matrices B_G and B_G^{new} are not much different in any sense. The original ISOMAP uses the Lanczos algorithm [61], which is an iterative method that is appropriate for solving the first few leading eigenvalue/vector pairs. The Lanczos algorithm iteratively refines the solution in the Krylov subspace that grows from an initial vector by multiplying it

Table 5: Computation time in seconds between ISOMAP and p-ISOMAP. In parentheses next to the data set name, the three numbers are the number of data n, the original dimension M, and the reduced dimension m, respectively. The number in the other parentheses next to k value changes indicates the ratio of vertex pairs whose shortest paths need to be updated. For each case, the average computing times of 10 trials were presented.

Synthetic data	Rand (3500, 5000 $\to$ 50)				Swiss roll (4000, 3 $\to$ 2)			
	ISOMAP		p-ISOMAP		ISOMAP		p-ISOMAP	
$k \to k^{new}$ ($\lvert F\rvert/n^2$)	28	32	$30 \to 28$ (15%)	$30 \to 32$ (13%)	14	16	$15 \to 14$ (77%)	$15 \to 16$ (73%)
Neighborhood graph	1.6	1.6	0.1	0.1	1.8	1.8	0.2	0.2
Shortest path	12.3	12.6	5.1	4.7	16.4	17.1	17.3	15.6
Eigendecomp.	7.8	7.7	6.9	6.8	1.9	1.7	1.6	1.5
Real-world data	Pendigits (3000, 16 $\to$ 5)				Medline (2500, 22095 $\to$ 200)			
	ISOMAP		p-ISOMAP		ISOMAP		p-ISOMAP	
$k \to k^{new}$ ($\lvert F\rvert/n^2$)	46	54	$50 \to 46$ (39%)	$50 \to 54$ (36%)	37	43	$40 \to 37$ (21%)	$40 \to 43$ (19%)
Neighborhood graph	1.3	1.2	0.1	0.1	1.1	1.1	0.1	0.1
Shortest path	9.3	9.8	7.1	8.3	6.8	7.0	2.8	3.3
Eigendecomp.	2.3	2.3	2.0	2.1	16.3	16.4	15.1	15.1

Table 6: Computation time in seconds required to determine the optimal k value by minimizing residual variances.

	Rand	Swiss roll	Pendigits	Medline
Range of k	[5, 50]	[5, 50]	[7, 60]	[9, 70]
ISOMAP	580	635	692	776
p-ISOMAP	142	403	305	314

with the matrix, i.e., span(b, $B_G^{new}b$, $(B_G^{new})^2b$, ...). The performance of the Lanczos algorithm largely depends on how quickly such a Krylov subspace covers that spanned by the eigenvectors. Another characteristic of the Lanczos algorithm is that the least leading eigenvalue/vector pair converges slowest within a particular tolerance. In other words, when the Krylov subspace becomes k dimensions, the first leading eigenvalue is refined k times, the second one $(k-1)$ times, the third one $(k-2)$ times, and so on. In this sense, we suggest using an initial vector from which the Krylov subspace grows as v_m, i.e., span(v_m, $B_G^{new}v_m$, $(B_G^{new})^2v_m$, ...), which possibly best recovers (λ_m^{new}, v_m^{new}). As a result, we can expect the Lanczos algorithm to terminate in less number of iterations than in any other cases.

3.5 Experiments and Applications

In this section, we present an empirical comparison between the computation times of ISOMAP and those of p-ISOMAP using both synthetic and real-world data sets. In addition, we show visualization applications of p-ISOMAP for real-world data sets. In our experiments, we used the code of ISOMAP provided by the original author.[4] However, the original code does not take advantage of sparse graphs, so we compared p-ISOMAP with an improved version of ISOMAP that runs Dijkstra's algorithm in C++ with a sparse representation of the graph. p-ISOMAP was implemented mainly in MATLAB except for the shortest path update part, which runs in C++. In both ISOMAP and p-ISOMAP, the eigendecomposition was done by MATLAB built-in function "eigs," which performs the Lanczos algorithm by using Fortran library

[4] `http://waldron.stanford.edu/~isomap/IsomapR1.tar`

ARPACK [89]. Throughout all experiments, we used the ISOMAP parameter as k, where the neighborhood graph is constructed by k-NN, since we can easily bound $|\mathcal{A}|$ or $|\mathcal{D}|$ by $O(n\Delta k)$ in which $\Delta k = k^{new} - k$. All the experiments were done using MATLAB 7.7.0 on Windows Vista 64bit with 3.0GHz CPU with a 4.0GB memory.

3.5.1 Computation Time

To compare the computation times between ISOMAP and p-ISOMAP, we tested two synthetic data sets (Rand and Swiss roll) and two real-world data sets (Pendigits and Medline). Rand data set was made by sampling a uniform distribution in a 5,000-dimensional hypercube, $[0, 1]^{5000}$, where the number of data is 3,500. "Swiss roll" data set has 4,000 data points in three-dimensional space. Pendigits data set[5] contains 10,992 handwritten digit data in a form of pen traces in 16-dimensional space [11], but we selected 3,000 data with an equal number of data per cluster because of memory constraints. Finally, Medline data set[6] is a document corpus related to medical science from the National Institutes of Health, and it has 2,500 documents encoded in 22,095-dimensional space. Table 5 compares computation times of ISOMAP with those of p-ISOMAP for each data set. In most cases, p-ISOMAP runs significantly faster than ISOMAP. However, as the number of vertex pairs whose shortest paths need to be updated increases, the computational advantage of p-ISOMAP over ISOMAP gradually vanishes. Nonetheless, except for "Swiss roll" data set, which involves a large number of the shortest path update even with a slight parameter change, most data sets require only about 10-40% the shortest path update for a reasonable parameter change, e.g., within 5. Fig. 7 shows the behaviors of p-ISOMAP depending on the number of data, Δk, and an initial k value. We selected Rand data since it was the most suitable one to clearly observe its behaviors. Fig. 7(a) shows the computation

[5] `http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits`

[6] `http://www.cc.gatech.edu/~hpark/data.html`

time in terms of the number of data. As we can see, p-ISOMAP scales well in terms of the number of data compared to ISOMAP. In Fig. 7(b), as the parameter change Δk gets bigger, the running time of p-ISOMAP increases linearly, which tells that $|\mathcal{A}|$ or $|\mathcal{D}|$, which is proportional to Δk, has a dominant influence on the performance of p-ISOMAP. Finally, Fig. 7(c) shows an increasing performance gap between two methods as an initial k value grows. This is mainly because the original Dijkstra's algorithm used in ISOMAP needs more computations as the graph gets denser while p-ISOMAP depends only on $|\mathcal{A}|$, $|\mathcal{D}|$, or correspondingly $|F|$, which probably does not increase over different initial k values. Finally, for each data set, we measured the computation times to take to determine the optimal k value that minimizes residual variances [12]. As shown in Table 6, we could significantly reduce the computation times by utilizing the dynamic update of p-ISOMAP.

3.5.2 Knowledge Discovery via Visualization using p-ISOMAP

In this section, we present interesting visualization examples of real-world data sets using p-ISOMAP. To be specific, we show how ISOMAP with different parameters can discover various knowledge about data and how the information acquired through visualization can facilitate traditional data mining problems such as a classification task. p-ISOMAP was used to efficiently update ISOMAP results throughout all the visualization experiments. To begin with, we have chosen three real-world data sets (Weizmann, Medline, and Pendigits) that have cluster structures in order to make it easy to analyze their visualization. Weizmann data set is a facial image data set[7] that has 28 persons' images with various angles, illuminations, and facial expressions. To obtain an understandable visualization, we have chosen three particular persons' images with three different viewing angles as shown in Fig. 8(a), in which each combination of a particular person and a viewing angle contains multiple images that

[7] `ftp://ftp.wisdom.weizmann.ac.il/pub/facebase`

vary based on other factors such as illuminations and facial expressions. In their visualizations shown in Figs. 8(b)-(d), each of these images is represented as a letter that corresponds to its cluster from Fig. 8(a). Medline data set, which is a document collection, has 5 topic clusters, heart attack ('h'), colon cancer ('c'), diabetes ('d'), oral cancer ('o'), and tooth decay ('t'), in which the letters in parentheses are used in its visualization in Fig. 9. Pendigits data set, which is described in Section 5.1, has 10 clusters in terms of which digit each data item corresponds to, i.e., '0', '1', ..., '9'. Several interesting visualization examples of these data based on p-ISOMAP are shown in Figs. 8-10[8] where cluster centroids and neighborhood connections are also shown in the form of letters in rectangles and grey lines in the background, respectively. Among visualization examples of Weizmann data set, Fig. 8(c), which well resembles the layout of clusters in Fig. 8(a), successfully straightens its intrinsic manifold defined by the two factors, a person and an angle. This is mainly because of the neighborhood graph constructed by a proper k value that forms its edges either within a particular person or within a particular angle, which is why we mostly see horizonal and vertical neighborhood connections as well as gaps between grid-shaped cluster centroids in Fig. 8(c). Regarding a comparison between Figs. 8(b) and 8(d), fewer neighbors in Fig. 8(b) bring connections only within images with the same angle, which in turn results in a clustered form of visualization based on angles. This indicates that even if we prefer the similarity in terms of a person to that in terms of an angle, the actual distances in the vector space into which the images are transformed are dominated by an angle. On the other hand, Fig. 8(d) connects almost all the data points between each other, which would reflect the Euclidean distances in the original space just like MDS does. In addition, we can consider the layout of cluster structure shown in Fig. 8(d) as a curved version of manifold as it appears in the original space,

[8]These figures can be arbitrarily magnified without losing the resolution in the electronic version of this thesis.

which is analogous to what we discussed in Fig. 6. Medline data shown in Fig. 9 is not visualized in a well-clustered form by ISOMAP because it is usually difficult to find a well-defined manifold structure with few meaningful dimensions for document data. However, by manipulating k values, we can at least obtain various visualization results that possibly reveal different aspects of the data. For example, when $k = 30$ in Fig. 8(b), the topic cluster, tooth decay ('t'), is shown distinct from the other clusters while so does the cluster, diabetes ('d'), in the other cases. In this situation, if one wants to focus on a certain cluster separately from the others, it would be necessary to change k values for a suitable visualization result. Visualizations of Pendigits data set shown in Fig. 10 give numerous interesting characteristics. First of all, as the parameter k increases, the overall transition from Fig. 10(a) to 10(f) is shown similar to that of "Swiss roll" data set from Fig. 6(c) to 6(d). In other words, a larger k value places more data in a curved shape, which reflects the underlying curvature in the original space, while a smaller k value does more data in a linear shape, which corresponds to a straightened manifold. To be specific, starting from Fig. 10(b), the cluster '8' gradually gets scattered and curved with an increasing k. Similarly, the cluster '0' maintains a linear shape before $k = 50$, and finally it becomes scattered in Fig. 10(f). In short, ISOMAP with a small parameter value tends to unroll the curved manifold due to geodesic paths, but that with a large parameter better shows its curvature itself. In view of clustering, Fig. 10(a) well separates the clusters '2' and '7' whereas the other visualizations gradually overlap them with increasing k values. In addition, the clusters '3' and '6' appears to overlap for a certain range of k between 9 and 11 as shown in Figs. 10(b)-(d). Now let us discuss about subcluster/outlier discovery through various visualization examples. In most examples in Fig. 10, the cluster '5' is shown to have two subclusters, one of which is near the cluster '8', and the other between the clusters '3' and '9'. Based on this observation, we examined some sample data from each cluster and found out such subclusters are due to the

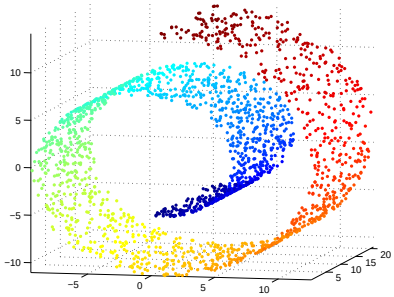
different way to write '5'.[9] From the examples in these two subclusters shown in Figs. 11(a)-(b), we can see that some people write '5' starting from the hat, which is the top horizontal line in '5', while others write the hat after finishing the bottom part. Similarly, the cluster '7' has a majority of data near the cluster '2', but it also has two minor groups of data near the cluster '1' and the cluster '6', respectively. (See, for example, the coordinates around $(-100, 50)$ and $(50, -150)$ in Fig. 10(c).) After looking at the actual data samples from these groups, we found that most people write '7' in a way shown in Fig. 11(c). However, some people first write an additional small vertical line in the top-left part but by omitting the small horizontal line in the middle part as shown in Fig. 11(d), which corresponds to the minor data near the cluster '1', but some others just reverse the direction to write the small horizontal line in the middle part of '7' as shown in Fig. 11(e), which corresponds to those near the cluster '6'. In addition, their different traces and shapes impose similarities to those of the clusters '1' and '6', respectively. Finally, in Fig. 10(d), some data in the cluster '0' seems to deviate from its major line-shaped data in Figs. 10(a)-(c). Figs. 11(f) and 11(g) represent the latter and the former data, respectively. We can see that such deviated ones shown in Fig. 11(g) start from the top-right corner rather than from the top-middle part when writing '0', which causes their connections to the cluster '5' that also starts from the top-right corner. Finally, we have incorporated the above findings in a handwritten digit recognition, which is a classification problem, using Pendigits data set. Based on the information that the cluster '5' has two clear subclusters, we modified the training data labels in the cluster '5' into two different labels and classified the test data that are assigned either label to the cluster '5.' As a classification method, we have chosen the linear discriminant analysis combined with k-nearest neighbor classification, which is a common setting in classification.

[9]Note that Pendigits data set we used here is not just static image data but the traces of the pen, which is why the order matters in the feature space.

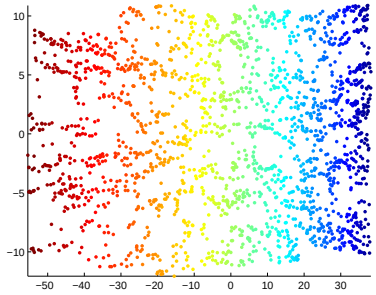
As a result, the classification accuracy increased from 89% to 93%. In fact, this is a promising example of human-aided data mining processes through visualizations with intelligent interaction. The computational efficiency of p-ISOMAP makes such processes smooth and prompt.

3.6 Conclusions

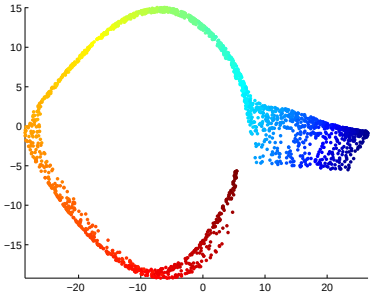
In this chapter, we proposed p-ISOMAP, an efficient algorithmic framework to dynamically update ISOMAP embedding for varying parameter values. The experiments using both synthetic and real-world data with various settings validate its efficiency. This advantage of p-ISOMAP can not only speed up the parameter optimization processes but also enable users to interact with visual analytics systems more smoothly. Such interaction provides us with deep understanding about data, which can improve even the computational data mining problems such as classification.



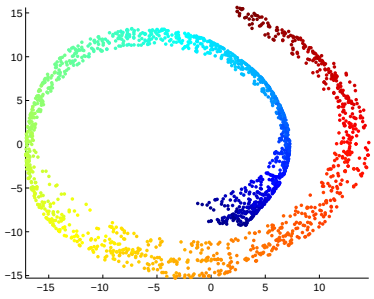
(a) The "Swiss roll" data set



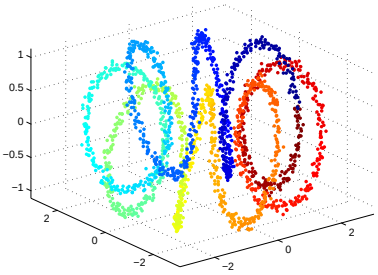
(b) ISOMAP with $k = 8$



(c) ISOMAP with $k = 40$

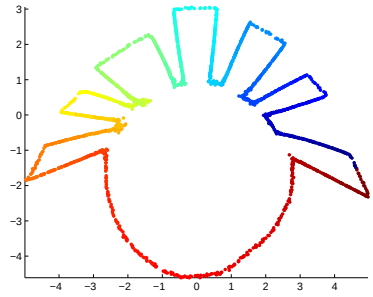


(d) ISOMAP with $k = 100$

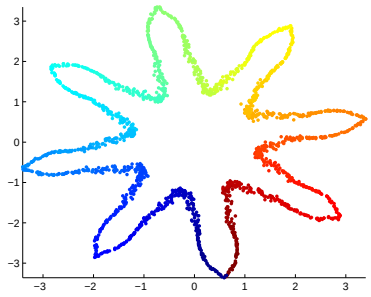


(e) The toroidal helix data set

(f) ISOMAP with $k = 8$

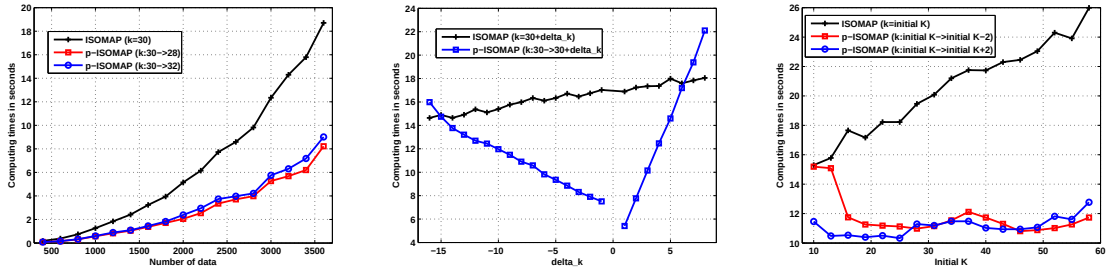


(g) ISOMAP with $k = 40$



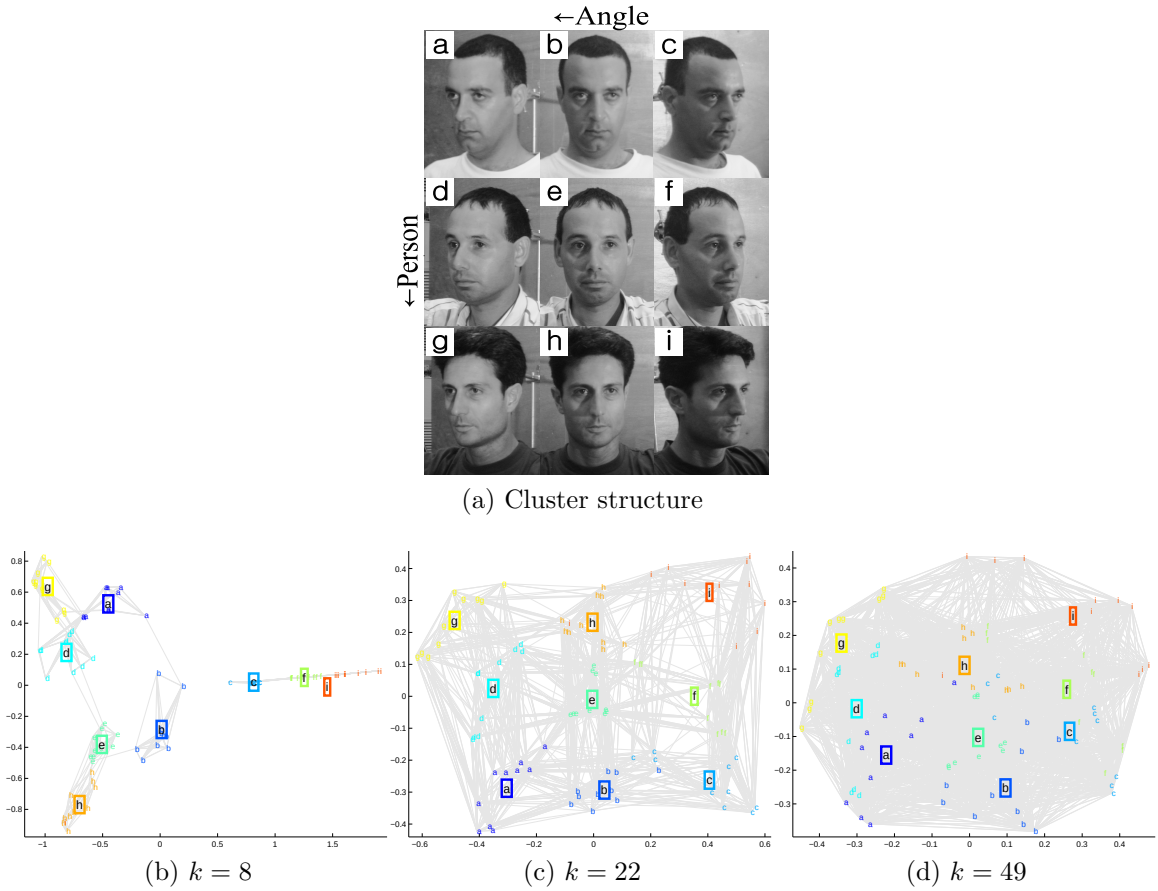
(h) ISOMAP with $k = 100$

Figure 6: ISOMAP examples with different k values. The first and second rows of figures correspond to the "Swiss roll" and the toroidal data sets, respectively.



(a) Computing times vs. number of data (b) Computing times vs. Δk (c) Computing times vs. initial k

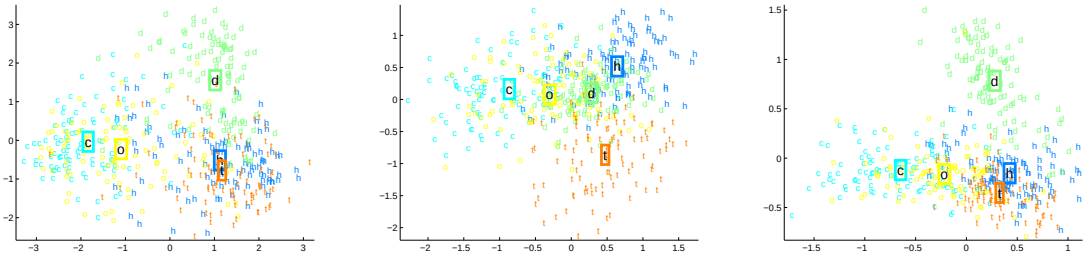
Figure 7: Behavior of p-ISOMAP depending on the number of data, Δk , and initial k on Rand data set. Other than the varied one, the rest of variables were fixed in each figure.



(a) Cluster structure

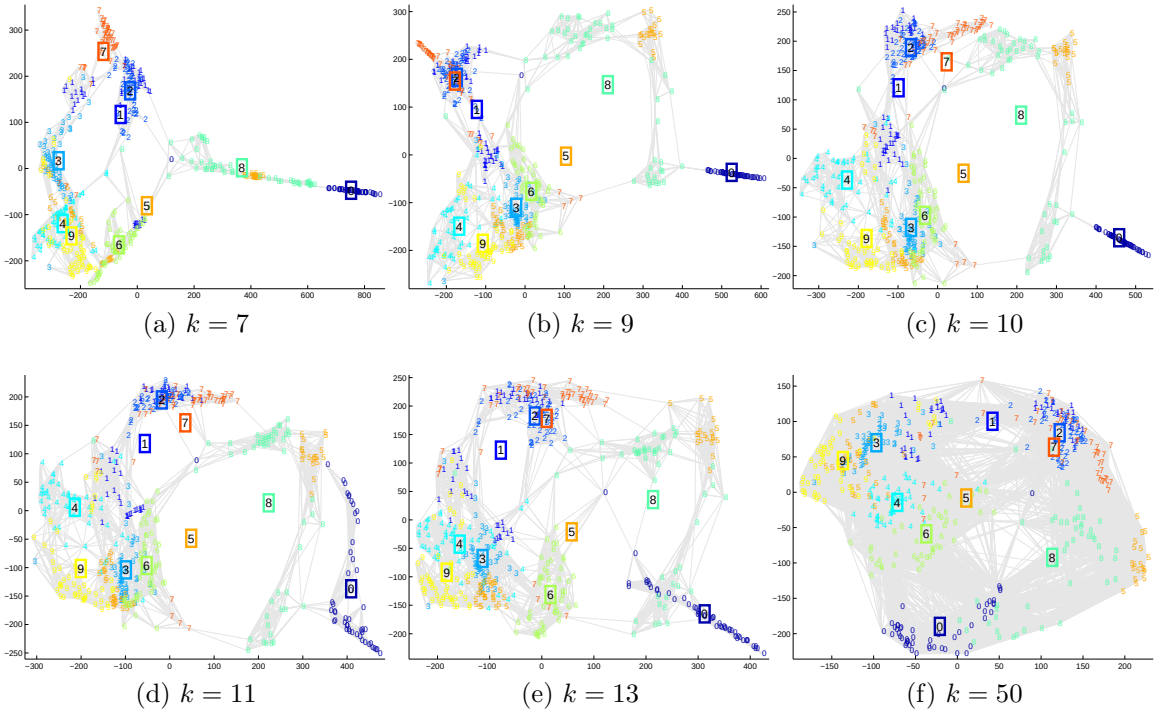
(b) $k = 8$ (c) $k = 22$ (d) $k = 49$

Figure 8: Visualization of Weizmann data set using p-ISOMAP



(a) $k = 5$ (b) $k = 30$ (c) $k = 50$

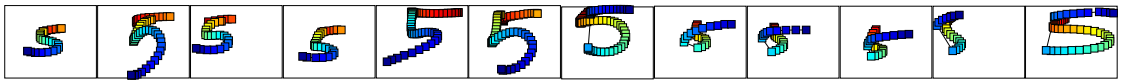
Figure 9: Visualization of Medline data set using p-ISOMAP



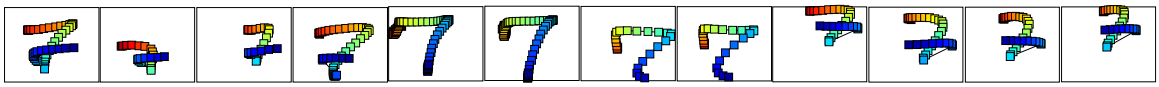
(a) $k = 7$ (b) $k = 9$ (c) $k = 10$

(d) $k = 11$ (e) $k = 13$ (f) $k = 50$

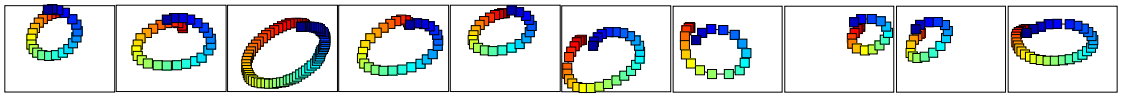
Figure 10: Visualization of Pendigits data set using p-ISOMAP



(a) A subcluster in cluster ‘5’ (b) Another subcluster in cluster ‘5’



(c) Major data in cluster ‘7’ (d) A minor group in cluster ‘7’ (e) Another minor group in cluster ‘7’



(f) Major data in cluster ‘0’ (g) Minor data in cluster ‘0’

Figure 11: Subclusters/outliers in ‘0’, ‘5’, and ‘7’. Pen traces start from red and end at blue.

CHAPTER IV

ITERATION-WISE INTEGRATION FRAMEWORK OF COMPUTATIONAL METHODS

Visual analytics has been gaining increasing interest due to its fascinating characteristic that leverages both humans' visual perception and the power of computing. Although various computational methods are being proposed, they do not properly support visual analytics. One of the biggest obstacles towards their real-time visual analytic integration is their high computational complexity. As a way to tackle this problem, this chapter presents PIVE, a Per-Iteration Visualization Environment for supporting real-time interactive visualization with computational methods. The main idea behind PIVE is that most advanced computational methods work by refining the solution iteratively. By visually delivering the result from each iteration to users, the proposed framework enables users to quickly acquire the information that the computational method provides as well as the ability to perform continuous interactions with them in real time. We show the effectiveness of PIVE in terms of real-time visualization and interaction capabilities by customizing various dimension reduction methods such as principal component analysis, multidimensional scaling, and t-distributed stochastic neighborhood embedding, and clustering methods such as k-means and latent Dirichlet allocation.

4.1 Introduction

The innate ability of humans to quickly perceive insight through visual analysis and decision processes has been a key factor in the growth of visual analytic research [78, 127]. One of the most significant efforts made by visual analytics researchers

is the integration of various computational methods from data mining and machine learning areas with visual analytics so that users can benefit from intelligent meaningful information generated by these techniques. For example, dimension reduction and clustering methods have been commonly used in high-dimensional data visual analytics [23, 117]. More recently, latent Dirichlet allocation (LDA) [20], a popular method for document topic modeling, has been adopted in a wide variety of visual analytics systems for document analysis [134, 50, 88].

However, a critical hurdle in the integration of computational methods into visual analytics is the significant amount of computational time required by these methods. As computational methods become more advanced and capable, they usually run much slower, making it almost impossible to visualize and interact with them smoothly in real-time visual analytics. Due to this significant running time, even though numerous computational methods are currently being developed and some methods such as t-distributed stochastic neighbor embedding (t-SNE) [130] even claim their suitability directly in visualization applications, the state-of-the-art in visual analytics does not seem to fully utilize the advancements in computational methods. Consequently, in many domain areas, people still resort to only a few basic computational methods such as principal component analysis (PCA) [74] and multidimensional scaling (MDS) [45] for dimension reduction, hierarchical clustering and k-means [19] for clustering, etc.

However, we believe that various important aspects have been largely ignored when integrating (advanced) computational methods into visual analytics. In a sense that such an integration essentially involves both humans and computational methods, exploiting the characteristics of each side simultaneously may bring a synergetic effect for their tight integration that would not be possible otherwise. Motivated by this general idea, this chapter focuses on the following aspects from each side: (1) *humans' perceptual precision* and (2) the *iteration-wise behavior of computational*

methods.

For *humans' perceptual precision*, we highlight that when perceiving numbers, *humans do not require a high precision* such as a double or a single precisions typically used in modern computers. For example, when perceiving the value of π, most people know its approximate value, e.g., 3.14. In practice, perceiving it as a more accurate value, e.g., 3.1415926, does not make much difference. In a more analytic context, suppose the topic modeling has given a topic-wise representation of a particular document as (55.5852%, 38.8615%, 5.533%) with respect to three topics, e.g., science, sports, and economics. People may perceive its topic contribution at a tenth value at most, which is approximately (55.6%, 38.9%, 5.5%), but it would not change their perception significantly even if more accurate numbers were considered.

This substantially low perceptual precision compared to that of computational methods opens up a variety of possibilities to reduce the intensive computational time taken in running a computational method in a visual analytics environment. As a complementary characteristics of the computational methods to achieve this goal, we focus on their *iteration-wise behavior*. These days, many modern computational methods are performed through an iterative refinement process until reaching the final converged solution. An important observation found in most methods is that throughout the iterations, *a major refinement of the solution typically occurs in early iterations while only minor changes occur in the later iterations*. It indicates that the low-precision outputs of computational methods are dominated by their major refinement made during early iterations. In this respect, humans may be able to obtain most information from the computational method outputs in a much shorter amount of time than the full iterations until convergence.

However, apart from well-principled convergence criteria studied in most computational methods, it is not straightforward to determine when to terminate the iteration at which the result is reasonably accurate from the perspective of humans'

perceptual precision. Instead, we propose an alternative approach called PIVE (**P**er-**I**teration **V**isualization **E**nvironment for supporting real-time interactive visualization with computational methods), which *visualizes the intermediate result per iteration as soon as they become available*. Unlike the previous approaches, which typically treat a particular computational method as a black box, the main novelty of PIVE lies in the idea to *break computational methods down to the iteration level and tightly integrate it with the interactive visualization so that users can check the result of computational methods without any delays and interact with them in real-time*.

Such real-time interaction capabilities based on this tight integration of computational methods at an iteration level makes significant differences in terms of the approaches for handling how we interact with a computational method. That is, from a perspective of viewing it as a black box, the turn-around time required for a particular interaction is usually equivalent to the time taken in running the entire set of iterations until its convergence. Therefore, previous efforts in adding an interaction capability to a computational method interactive have mainly focused on the sophisticated algorithmic modifications that can maximally reflect the users' intention from a single interaction. Accordingly, during a single interaction, it was generally recommended that users give the computational method a substantial amount of changes that are carefully made. Otherwise, users would be frustrated if the result due to a user interaction does not properly reflect their intention after a long time of waiting for the computational method to converge. On the contrary, in PIVE, a turn-around time for a single user interaction drastically decreases to the time taken in running a single or a small number of iterations at most instead of an entire set of iterations. In this respect, PIVE enables users to *perform multiple small interactions continuously by quickly adjusting their interactions based on the real-time response of the computational method*.

Motivated by these ideas, this chapter discusses about PIVE in detail and present

the example realizations of various well-known computational methods under PIVE. The main contributions of this chapter is summarized as follows:

- Presentation of PIVE as a general idea to tightly integrate computational methods in visual analytics at an iteration level.
- In-depth discussion about the potential issues and their solutions in PIVE
- Realizations of PIVE with various well-known computational methods (PCA, MDS, t-SNE, k-means, and LDA) in established visual analytics systems
- Customizations of the above methods for real-time user interaction capabilities under PIVE
- Use cases of the customized methods with real-time user interaction examples

The rest of this chapter is organized as follows. Section 4.2 discusses related work. Section 4.3 describes PIVE in more detail and discuss its potential issues and their solutions. Section 4.4 presents various customized computational methods with their supported interactions in PIVE. Using these customized methods, Section 4.5 describes the quantitative analyses about the iteration-wise behavior of computational methods and provide several use cases of the customized methods with their real-time interactions under PIVE in several well-known visual analytics systems. nally, Section 4.6 concludes the chapter.

4.2 Related Work

In this section, we briefly discuss various previous studies from the two main perspectives: those aiming at efficient interactive visualization and those trying to make computational method user-interactive in visualization applications.

4.2.1 Efficient Interactive Visualization

Not surprisingly, numerous studies have focus on the visualization applications of large-scale data. Among various approaches, one of the straightforward but reasonable approaches is by using a subset of data by sampling. For example, Fisher et al. [57] has proposed an efficient way of dealing with large-scale data visualization by initially using only a small portion of data and then perform an incremental update on the visualization. Ellis et al. [54] has also taken a random sampling-based visualization approach mainly for avoiding the visualization clutter due to a large number of visualized objects while considering the efficiency issues during visualization.

As another popular approach for improve the efficiency in visualizing large-scale data, numerous studies have been based on multi-threading techniques. In this context, the main role of multi-threading is to separate the data processing/computation module and the visualization/rendering modules as multi-threads, allowing their efficient concurrent running. A notable line of research is called 'in situ' visualization [92, 146]. The main idea of it is, given large-scale data, to alleviate some post-processing overheads that had to be taken care of by the visualization module and let these overheads handled in the phase of the data processing/computation in which the powerful computing resource is readily available. In this manner, even though the visualization module does not have a computing power, which is often the case, the visualization can fluidly be performed. Although similar to the 'in situ' visualization approach, Tu et at. [128] has utilized the data sharing aspects in a parallel supercomputing environment. On the other hand, there have been approaches that have utilized multi-threading mainly for the purpose of providing a efficient responsive user interactions [104] by separating an application and a visualization threads into multiple concurrent threads.

As will described in detail in Section 4.3, PIVE adopts a similar multi-threading idea in order to reduce the overhead of the visualization module that has to go through

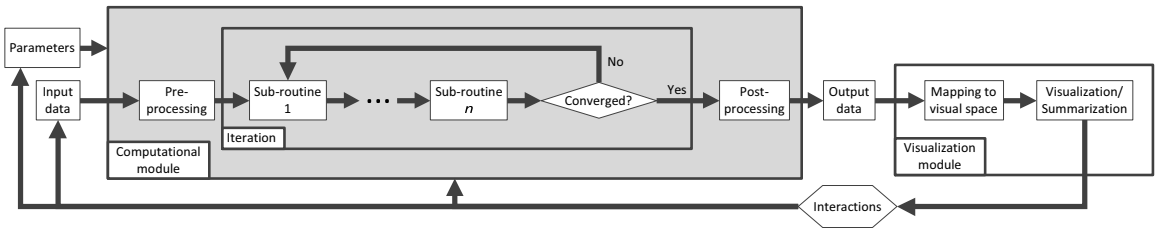
a constant updating as the iterations of the computation method go. However, *none of these multi-threading-based approaches hardly exploited the nature of the iterative refinement processes found in most computational methods, which makes a clear distinction of PIVE to the previous work.*

Furthermore, efficient interactive visualization has been a main concern in the context of dynamic/streaming data. When visualizing dynamic/streaming data, the overall theme found in various approaches is to update the visualization efficiently given incremental changes in a data set. In this context, Cottam et al. [44] has recently discussed about a taxonomy for dynamic data visualization. Although the detailed approaches may differ, several prior studies [139, 140] have started from a relatively similar idea that the visualization update is carried out only when significant changes/events have been detected. Additionally, Alsakran et al. [5] has visualized the streaming documents using a GPU-accelerated force-directed layout technique.

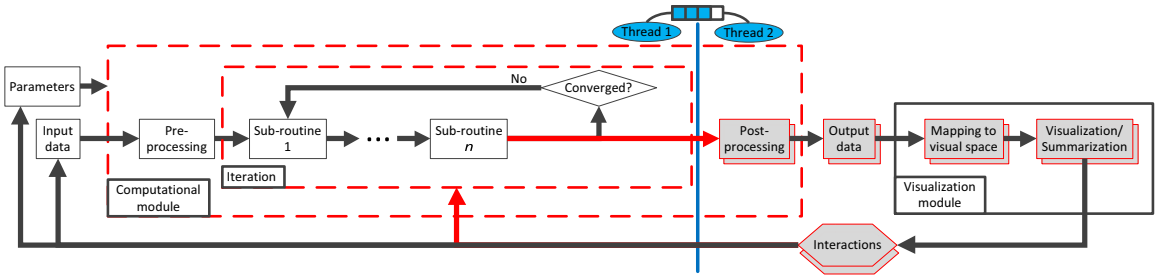
Various interesting ideas from dynamic data visualization could be applied to further improve the updating process of visualization in PIVE. Nevertheless, *the primary problem that PIVE tackles arises from the intensive amount of computations in the computational methods, and thus an efficient updating of the visualization module is not a concern in general.*

4.2.2 User Interaction with Computational Methods

There have been numerous efforts to make computational methods, which are mostly automated, user-interactive in visualization applications. One of the most representative work is based on MDS [136] that has added MDS a capability of incorporating user feedback based on a user-specified visual region. A more recent work called observation-level interaction [55] has provided a general framework in which the user interaction from a scatter plot is incorporated in a Bayesian probabilistic framework.



(a) The standard workflow



(b) The PIVE workflow

Figure 12: An overall diagram of PIVE (b) in contrast to the standard (non-iteration-wise) one (a). In the standard framework (a), a computional method is treated as a black box, as depicted by a gray rectangle. On the other hand, PIVE (b) breaks down the computational method at its iteration level, allowing it to be visualized at each iteration while taking into account any user iteractions. The blue line separates the overall procedure into two separate threads with their message queues, as shown in the blue rectangle, to remove potential computational overheads.

These user interaction capabilities have long been emphasized in terms of clustering since clustering is generally a difficult problem. Seo et al. [117] has improved a traditional clustering method called hierarchical clustering so that it can have flexible interactive capability with the clustering result in a bioinformatics domain. More recently, iVisClustering [88] has tried to make a more recent method LDA interactive by supporting cluster merging/splitting, cluster keyword refinement, etc.

However, most methods have treated the computational method as a black box, and thus the interactions they support are inherently far from being real-time because the entire set of iterations for a new run of the computational method is required for each iteration. Nonetheless, a variety of work has addressed the importance of the capability for supporting a continuous set of real-time interactions with computational methods due to the highly exploratory nature of human interactions

[56, 104, 120]. In this sense, PIVE, which leverages both humans' perceptual precision and the iteration-wise behavior of computational methods, bears a potentially great impact in achieving this goal.

4.3 Per-Iteration Visualization Environment (PIVE)

First, we describe an overall flow of PIVE (Fig. 12(a)), by highlighting its differences from the standard (non-iteration-wise) approach (Fig. 12(b)).

Let us begin with a general procedure when an iterative computational method is integrated in visual analytics. As shown in Fig. 12(a), input data, which are usually represented as multidimensional vectors, are given to the computational module along with its required parameter values. The computational module pre-processes the data, if necessary, and runs through iterations, which are usually divided into multiple sub-routines, until it converges. Upon convergence, the output goes through a post-processing step.

The final output of the computational module is then passed to the visualization module, which encodes it in a visual space and finally delivers its visualization to users. For example, the output of a dimension reduction method, e.g., PCA, can map data items onto the coordinates of the screen space, and the output of clustering can be used to color-code each group of data clusters.

Users can then better explore the visually represented data with the help of the information provided by the computational method and often interact with computational methods by adjusting their input data as well as their parameters. These interactions trigger another run of the computational method. For example, given the cluster summary for a set of text documents, if a user finds an interesting cluster, the user may perform another iteration of clustering on the particular subset to obtain more details about the chosen subset. On the other hand, users might want to adjust the number of clusters, which is usually a user-specified parameter in clustering

methods, to find the best clustering result for the data.

In most of the described visualizations and interactions, the standard framework generally treats the computational module as a black box, which the visualization module has no control over, depicted by a gray rectangle in Fig. 12(a). In other words, once the computational module has been initiated, visual analytic systems must wait for it to finish its iterations before it outputs the visualization to users.

On the contrary, PIVE takes the results of intermediate iterations out of the computational module and delivers them to the visualization module whenever they are available. More specifically, as highlighted with the red horizontal line in Fig. 12(b), the result from each iteration is always passed to the post-processing step, the output of which, in turn, reaches all the way to the visualization module, regardless of whether it has converged or not. Consequently, these intermediate results are visualized to users much more quickly than having to wait for the converged solutions.

In addition, PIVE enables the above-discussed interactions to be instantly reflected by directly interacting with the process for each iteration of the computational module, as highlighted with the red vertical line in Fig. 12(b). For instance, given the result of a particular iteration, one could exclude certain data items from the following iterations, which accelerate the later iterations due to the reduced data size. Furthermore, users could change the number of clusters while a clustering method is running, which immediately affects the following iterations.

4.3.1 Issues and Solutions

4.3.1.1 Computational Overhead and Multi-threading

Computational overheads are one of the issues that can be potentially introduced by this framework. As can be seen in the red-lined stacked rectangle blocks in Fig. 12(b), visual analytics systems have to process the output for each iteration repetitively while the standard approach needs to process only the final output once. These additional computations could undermine the effectiveness of the proposed framework. Let

us suppose that a particular computational method, which requires 50 iterations to converge, converges in a minute. If the proposed framework runs only 4-5 iterations within the same amount of time, then users might prefer the standard approach instead of being able to check the intermediate results since the results from such early iterations may not be satisfactory.

However, we claim that this issue can be easily overcome by applying a multi-threaded approach to the proposed framework. As shown by the blue ellipses in Fig. 12(b), the entire process can be separated into two concurrent processes/threads. The first thread shown to the left is responsible only for the sub-routines inside the iteration while the second thread on the right handles actions from the post-processing block to the visualization block. These two threads communicate with each other via a message queue, as shown by the blue rectangle on top in Fig. 12(b), where the outputs for each iteration for post-processing are to be stored.

Modern computers are usually equipped with at least two or more cores on the CPU. These two threads can be executed virtually in parallel, which hardly slow down the computational methods compared to the standard approach. Although not included in this chapter, for the computational methods we customized, we compared the total computing time between PIVE and the standard frameworks, but with multi-threading implemented, there were essentially no differences in their running times.

Even in this multi-threading framework, the following case may still be problematic. Suppose the second thread involves more intensive computations than the first thread because, for example, the post-processing block takes more time than the processes at each iteration. As a result, the second thread would act as a bottleneck in the overall flow of the proposed framework, resulting in the message queue increasing. One way to handle this issue is to store the results of each iteration periodically rather than storing every one of them in the first thread. Alternatively, the second thread

could take the most recent iteration-level results and discard the remaining older ones from the message queue. Under this situation, the visualization of the intermediate results may be somewhat discontinuous, but users would always be given the most recent result, which should be the most accurate solution up to the current iteration.

Finally, the other overhead comes from copying results from each iteration to the message queue, which results in a memory write operation. In the standard approach, these results for each iteration are usually written to the same memory space over iterations since the results from previous iterations do not need to be maintained. However, memory write operations are generally very fast. Furthermore, the outputs from each iteration of computational methods take up a much smaller memory compared to input data. For example, even if the data is a very high-dimensional, say, in the hundreds of thousands of dimensions, such as is the case in text data, the dimension reduction outputs would only be two-dimensional representations assuming they are visualized in a 2D space. Since the amount of additional computational time and memory that is required by our approach is minimal, we do not see memory overheads being a critical issue.

4.3.1.2 Visual Inconsistency and User Control

The second issue in the proposed framework is the visual inconsistency, which occurs during visualization updates, due to dynamic results changing each iteration. The most severe case occurs when the visualization changes too frequently. Although the amount of change generally diminishes as the iterations proceed, frequently changing visualizations may prevent users from obtaining a consistent picture of the data.

To address these issues with visual inconsistencies, we've come up with several possible controls. The first most basic option would be a stop and resume control which would stop and resume updates of the visualization. Secondly, a time period control would manage the length of the visualization. Additionally, we could pair this

time controller with two choices - the option to visualize the most up-to-date result or to visualize the result of the next item in the queue, which would provide the user with smoother visual transitions. Similar to the 'stop/resume' interaction, since our approach maintains each of the intermediate results, we could simply expand the controls to also add both the 'play backwards' and 'jump to...' options. These interactions would help users understand the overall trajectory of the results through each iteration. Through the use of these controls, it is very possible that the user may uncover an interesting insight into the data at a particular iteration or a series of iterations.

4.4 Customized Methods under PIVE

In this section, following the proposed framework, we present several customized computational methods in visual analytics systems. To begin with, we have chosen three visual analytics systems, FodavaTestbed,[1] Jigsaw,[2] and iVisClustering [88], which involve computational methods.[3]

FodavaTestbed is a visual analytics system for high-dimensional data, where users can apply various dimension reduction and clustering methods for exploratory analysis. Among various methods supported, we have chosen three dimension reduction methods, 1. MDS, 2. PCA, and 3. t-SNE. Jigsaw is a well-known system for document analysis, and we have chosen 4. k-means, which is used to provide a summary in terms of a compact set of clusters. Finally, iVisClustering is an interactive document clustering system which uses 5. LDA, a popular topic modeling method.

In the following, we describe how each method is customized along with the additional interactions we implemented in the proposed framework.

[1] `http://fodava.gatech.edu/fodava-testbed-software`
[2] `http://www.cc.gatech.edu/gvu/ii/jigsaw/`
[3] We obtained the code from the original authors of the systems.

4.4.1 Principal Component Analysis (PCA)

PCA [75] is a well-known dimension reduction method that captures the maximal variance in the data via a linear projection. PCA is mainly based on the method called eigendecomposition, the algorithms of which are categorized into two different methods, the QR algorithm and the Lanczos algorithm [61].

Basically, the Lanczos algorithm approximates a given data matrix by a much smaller one in the Krylov subspace [61], the dimension of which iteratively expands, and efficiently solves the eigendecomposition on the latter matrix. Due to the nature that this matrix well-approximates the largest eigenvectors of the original one, the Lanczos algorithm performs much faster than the QR algorithm in visual analytics in which only a few dimensions are needed.

We customize the Lanczos-based PCA implementation of FodavaTestbed so that the results for each iteration are dynamically visualized.

4.4.2 Multidimensional Scaling (MDS)

MDS [45] is a traditional dimension reduction method that attempts to preserve given distances/relationships of data items in a lower-dimensional space. Given the ideal distance δ_{ij} between x_i and x_j, MDS solves

$$\min_{x_1, \ldots, x_n} \sum_{1 \le i < j \le n} \left(d_{ij} - \delta_{ij}\right)^2, \tag{39}$$

where d_{ij} is the distance between the reduced dimensional vectors x_i and x_j. A Euclidean distance $\|x_i - x_j\|_2$ is usually used for d_{ij}. Solving Eq. (39) iteratively refines x_i's based on various optimization techniques [46]. We customize MDS in FodavaTestbed by extracting the x_i's at each iteration from the MDS implementation.

4.4.2.1 User Interaction Capabilities

Additionally, while the results for each iteration of MDS are visualized in a scatter plot, we support the interaction capability that enables users to move the data points by mouse via drag-and-drop, similar to the Prefuse force-directed layout. Then, during the MDS iterations, their new positions in the screen space are translated back to the MDS output coordinates, x_i's. The changes in x_i's at a particular iteration then affect the following iterations by generating different d_{ij}'s. In terms of how MDS behaves due to these changes, we provide two different capabilities: 'soft' vs. 'hard' placement. The soft placement continues iterations without any changes in MDS behaviors. It is equivalent to restarting MDS with the intermediate result at the particular iteration as the initial values for x_i's.

The hard placement capability fixes the values of x_i's for points moved by the user. This can be easily achieved by skipping the update step of these x_i's in the following iterations. Note that, however, even though their values do not change, other data points are still influenced by these fixed points, and in this sense, our approach is a semi-supervised MDS that reflects user interventions.

When using the semi-supervised MDS, an important advantage of the proposed framework is that users can immediately check the effects of these interactions via the iteration-wise visualization. Our modifications in FodavaTestbed support both types of interactions.

4.4.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE [130] is a relatively new dimension reduction method. It interprets pairwise distances as probabilities both in high-dimensional and lower-dimensional spaces and tries to minimize their Kullback–Leibler divergence, a distance measure between probability distributions. Unlike the previous methods discussed, it focuses on preserving neighborhood relationships instead of global ones, and it has shown its outstanding

capabilities in visualizations.[4]

Although we skip the detailed formulations because of the scope of the visual analytics community, the algorithm works iteratively by refining the lower-dimensional coordinates based on a gradient descent-based framework. In practice, however, t-SNE does not provide a clear stopping criterion, and thus it typically iterates several hundred times by default for any data set, which usually takes a significant amount of time. We customize the t-SNE in FodavaTestbed in a similar manner to the way we altered MDS.

4.4.3.1 User Interaction Capabilities

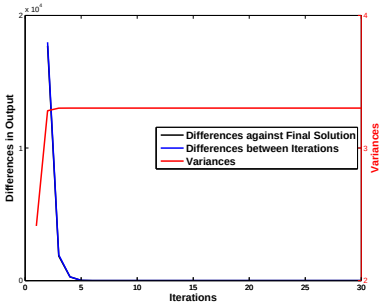
Likewise, we provide both the soft and hard placement interactions for t-SNE, as discussed in MDS. Although the algorithm details are different, the overall iterative procedure turns out to be quite similar to MDS. Thus, for the soft placement, we restart t-SNE with the intermediate results immediately during iterations. For the hard one, we skip the update step for data items moved by the user in the following iterations while they still influence other points in the t-SNE iterations. Therefore, our altered method can be viewed as a semi-supervised t-SNE.

4.4.4 *k*-means

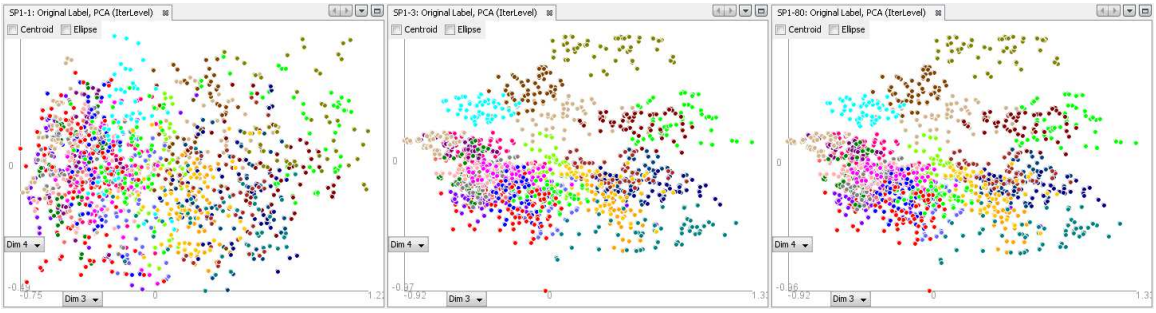
k-means, which is a widely-used clustering method, performs the following steps iteratively: 1. computing the centroid of each cluster by averaging the data vectors in the corresponding cluster and 2. updating the cluster assignment of each data item based on its closest cluster centroid. The iteration terminates when there are no cluster membership changes.

Although Jigsaw provides a cluster view based on *k*-means, it currently visualizes only the pre-computed results since *k*-means is usually very slow to converge. We customize it so that users can run *k*-means in real-time and the intermediate cluster

[4] `http://homepage.tudelft.nl/19j49/t-SNE.html`



(a) PCA criteria values and output changes



(b) The scatter plot at the first iteration

(c) The scatter plot at the third iteration

(d) The scatter plot at the 80th iteration

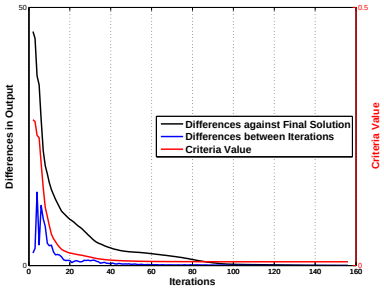
Figure 13: The behavior for each iteration of PCA and its visualization snapshots. In (a), the red lines represent the PCA criteria value, the lower-dimensional variance in PCA. The blue lines are the Euclidean distances of the lower-dimensional outputs between the current and the previous iterations, and the black lines are the Euclidean distances of the lower-dimensional outputs between the current and the final iterations. In (a), the black and the blue lines almost coincide. 1,420 facial image data representing pixel values in 2,048 dimensions have been used.

memberships are dynamically visualized.

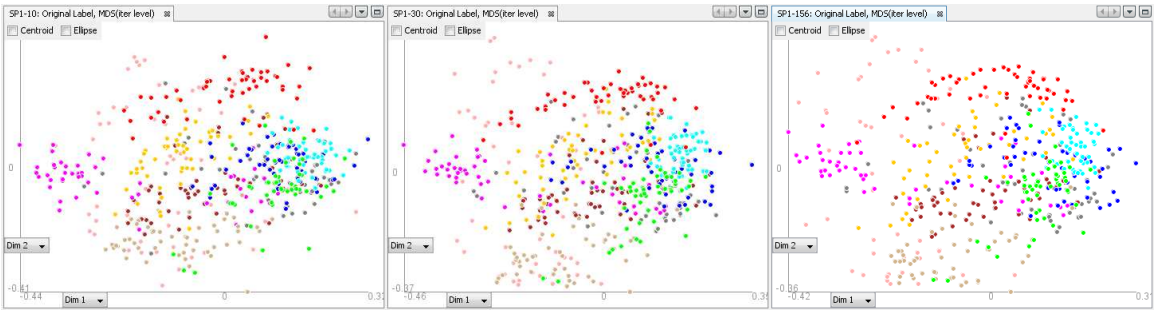
4.4.4.1 User Interaction Capabilities

Additionally, we add several interaction capabilities in the proposed framework. One is to split/merge clusters during iterations. On a split/merge interaction, similar to the soft placement in MDS and t-SNE, k-means restarts with the intermediate cluster memberships that reflect split/merged clusters, involving dynamic changes in a k-means parameter which represents the number of clusters.

Similar to the hard placement in MDS and t-SNE, another capability we provide is the option to fix the cluster assignments of the data in a particular cluster. To accomplish this, we skip the updating step of the cluster assignment for these data



(a) MDS criteria values and output changes



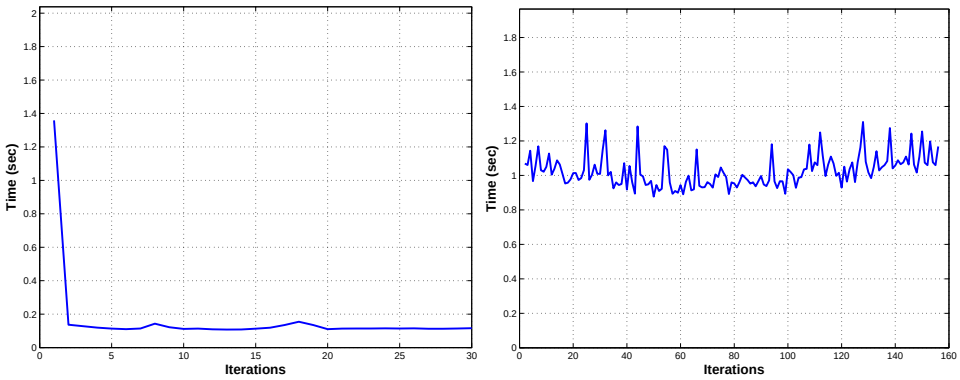
(b) The scatter plot at the 10th iteration (c) The scatter plot at the 30th iteration (d) The scatter plot at the 156th (converged) iteration

Figure 14: The behavior for each iteration of MDS and its visualization snapshots. In (a), the red lines represent the MDS criteria values, which is the stress value, i.e., Eq. (39) in MDS. The blue lines are the Euclidean distances of the lower-dimensional outputs between the current and the previous iterations, and the black lines are the Euclidean distances of the lower-dimensional outputs between the current and the final iterations. 500 handwritten digit data representing pen traces in 16 dimensions have been used.

in the following iterations. However, they still contribute to the centroid computing step. A similar semi-supervised way of k-means was previously proposed [15], but our framework significantly accelerates such interactions with k-means.

4.4.5 Latent Dirichlet Allocation (LDA)

LDA [20] is a popular topic modeling method for documents based on a generative probabilistic model. Given a number of topics, it gives two outputs: the term-wise distribution of each topic and the topic-wise distribution of each document. The iterations of LDA basically update these two outputs alternately. From a clustering viewpoint, the former corresponds to a centroid vector of each topic cluster, and the latter to a soft-clustering coefficient. By taking the topic index that has the maximum



(a) PCA (b) MDS

Figure 15: The computing times of the example in Fig. 13.

value in the latter, a document is clustered to a particular topic.

iVisClustering uses one of the fastest LDA libraries called Mallet [94], which implements LDA based on a Gibbs sampling [110]. Although this sampling-based approach does not guarantee a convergence, it is being widely used because of its simplicity and robustness against overfitting, compared to the variational approximation method proposed in [20]. Due to this convergence issue, LDA usually iterates a pre-defined number of iterations and usually requires a significant amount of time. We customize the Mallet library so that it can give the outputs from each iteration to iVisClustering, allowing iVisClustering to dynamically update its visualization.

4.4.5.1 User Interaction Capabilities

In addition to the original iVisClustering interaction capabilities being available during iterations, we also add several interactions with LDA in iVisClustering, similar to those in Jigsaw: splitting/merging clusters and fixing the cluster assignments of particular data items during iterations. The customization of LDA for such interactions is similar to k-means, and thus we skip the details due to the page limit.

4.5 Experiments

In this section, we present behaviors within each iteration for computational methods as well as their interactive aspects in the proposed framework.

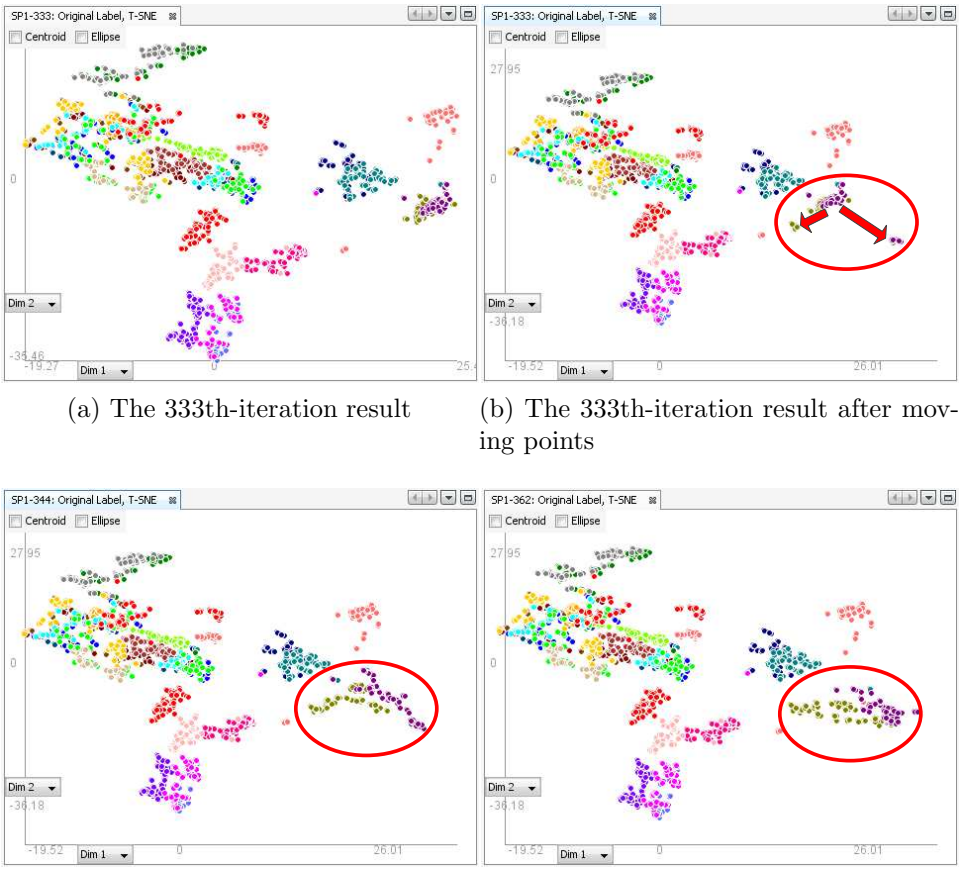
Table 7: The keyword summaries of the sampled clusters with/without fixing interactions of k-means performed in Fig. 19.

	Cluster 1	Cluster 6	Cluster 7	Cluster 8
Fig. 19(c)	process,trying,latency	turing,100,budget	quasimonte,unbalanced,choice	concern,rich,solvable
Fig. 19(d)	schur,process,trying	turing,budget,100	quasimonte,unbalanced,choice	concern,rich,solvable

Table 8: The keyword summaries of the selected clusters during splitting and merging interactions of k-means performed in Fig. 20.

	Cluster 1			
	1-a	1-b	1-c	1-d
Fig. 20(a)	analysts,cdc, earliest	weird,contents, oneyearold	fundraising, 11,7bd	chromosomes,100fold, may605375rossignol
Fig. 20(b)	enjoy,contents,weird			
Fig. 20(c)	enjoy,contents,weird			

	Cluster 2		Cluster 3	
	1-a	1-b	1-a	1-b
Fig. 20(a)	warm,rhythms, pretend	37,symptoms, said	social,cause,symptoms	
Fig. 20(b)	warm,37,rhythms		social,causes, people	social,causes, cerebellum
Fig. 20(c)	warm,37,symptoms		1000,social, causes	incredible,symptoms, cerebellum



(a) The 333th-iteration result

(b) The 333th-iteration result after moving points

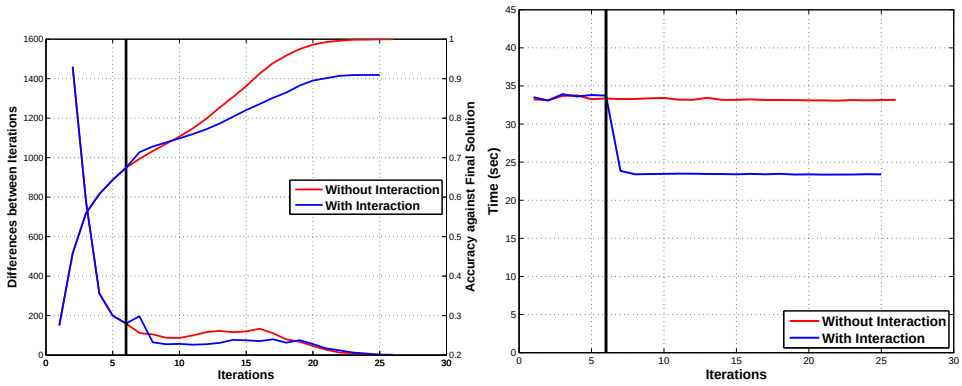
(c) The 344th-iteration result

(d) The 362th-iteration result

Figure 16: A point-moving interaction example using t-SNE. The two overlapping clusters,'l' and 'o,' are separated due to a user interaction of moving apart a few points from each cluster. 1,558 spoken letter data represented in 618 dimensions have been used.

4.5.1 Iteration-wise Behaviors and Visualization

Fig. 13 shows the behaviors of each iteration for the customized PCA and MDS along with their computing times shown in Fig. 15. In PCA, the criteria value, i.e., the lower-dimensional variance, as well as the lower-dimensional outputs (Fig. 13(a)) converge within a few iterations, indicating that only a few iterations of the Lanczos algorithm are needed in visual analytics applications (Figs. 13(b)-(d)). Nonetheless, each iteration takes roughly the same amount of computation time except for the first iteration which includes the pre-processing time. (Fig. 15(a)). Instead of having to

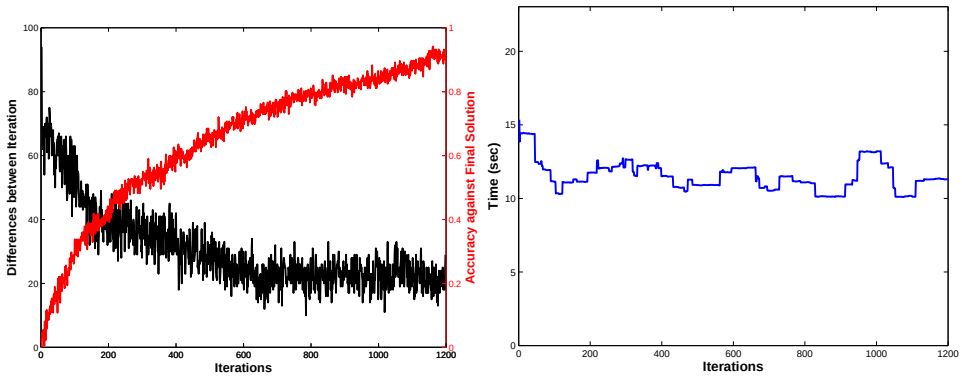


(a) The cluster membership changes (b) The computing times

Figure 17: The behaviors for each iteration of k-means with and without the interaction made in Fig. 19(b). In (a), the decreasing lines are the cluster membership changes between the current and the previous iterations while the increasing ones are the correct cluster memberships with respect to the final solutions without the interaction. The black vertical line represents the iteration point of the interaction made.

perform a fairly large number of iterations, as most PCA algorithms do, the iteration-wise visualization enables users to obtain an equivalent visualization much quickly. A similar argument applies to MDS as well. Although its convergence is relatively slow compared to PCA (Fig. 13(e)), we obtain the results similar to the final one achieved at the 10th iteration (Figs. 13(f)-(h)). We do not present the quantitative analyses for t-SNE, but we found tendencies similar to MDS, and we will focus on its interactive aspects in the following section.

In clustering, the behavior of each iteration of k-means is presented in Fig. 17 as well as their snapshots in Jigsaw in Fig. 19. In Jigsaw, in order to best assist users in easily identifying the location and the number of changes that occur while the visualization is dynamically updated, we draw arrows to represent where a particular data item has moved relative to the previous iteration, and color-code each data item to represent which cluster index it previously belonged to. As shown in Fig. 19 and in the redlines in Fig. 17, although significant change occurs in early iterations (Fig. 19(a)), they diminishes quickly, as seen in the sixth iteration (Fig. 19(b)), which is not much different from the final result (Fig. 19(d)). However, the time each



(a) The cluster membership changes (b) The computing times

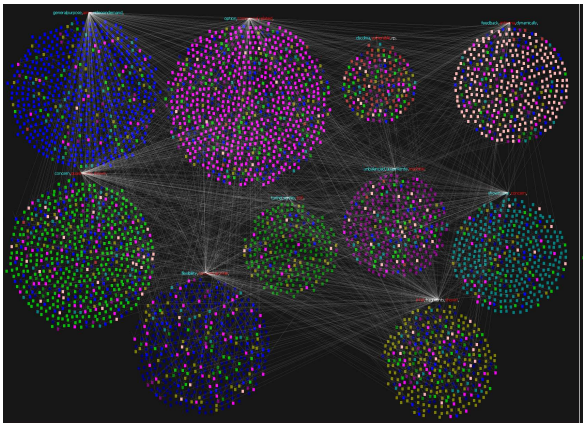
Figure 18: The iteration-wise behaviors of LDA. In (a), the black line represents cluster membership changes between the current and the previous iterations while the red line represents the correct cluster memberships with respect to the final solutions.

iteration takes is almost the same (Fig. 17(d)).

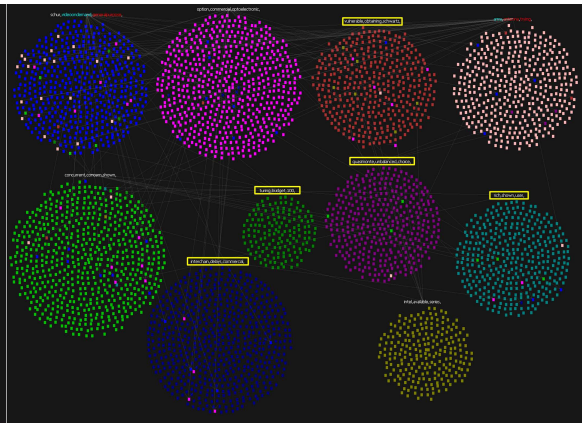
Finally, LDA, which is a sampling-based approach, shows a significantly different behavior from the previous methods (Fig. 18). Although cluster membership changes between iterations generally decrease and the solution narrows to the final solutions (Fig. 18(a)), cluster memberships change significantly even after many iterations, in this case after 1,200 iterations. In iVisClustering, we could see the top keywords of each topic become somewhat stable after several hundreds of iterations (Fig. 18(a)), but the randomness of the sampling-based algorithms might make it harder to give consistent visualizations when compared to deterministic methods in PIVE.

4.5.2 Real-time User Interactions

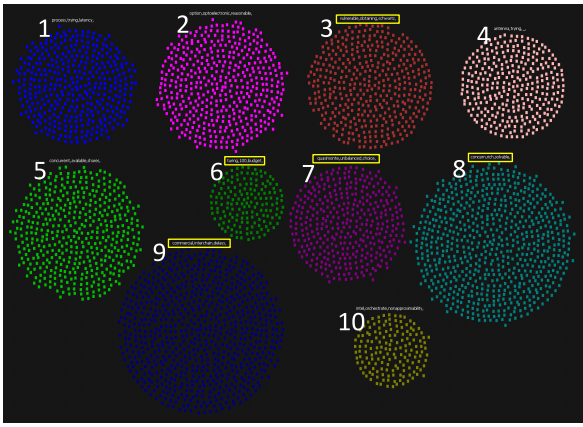
Basically, in all three systems, we provide basic interactions that control the visualization for each iteration, as discussed in Section 4.3.1. In the following, we show several use cases of the interactions discussed in Section 4.4.



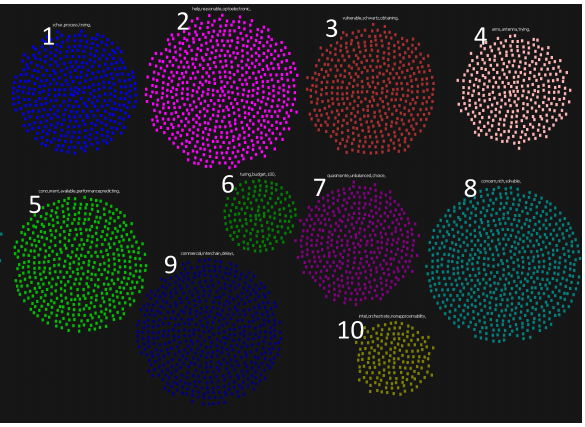
(a) The second-iteration result



(b) The sixth-iteration result



(c) The converged (25th-iteration) result after fixing clusters

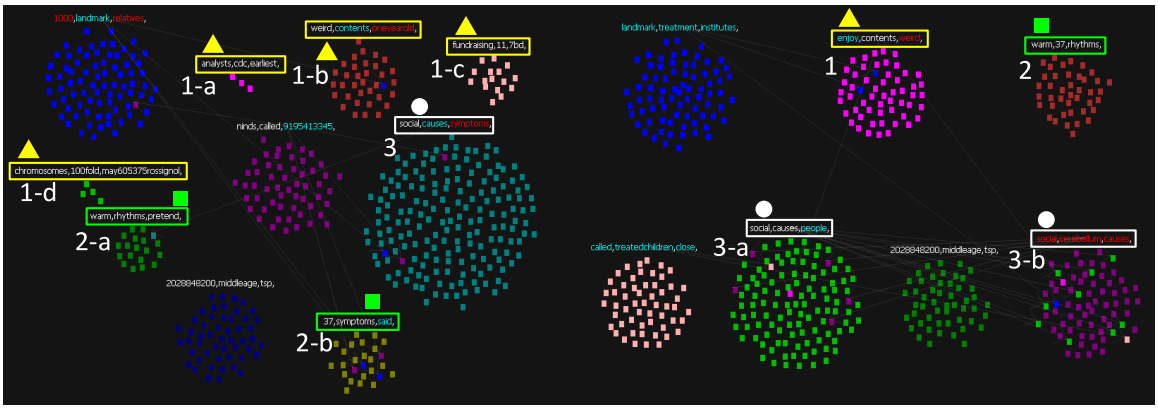


(d) The converged (26th-iteration) result without any interaction

Figure 19: The results of the PIVE integration of k-means in Jigsaw. At the sixth iteration, the interaction of fixing the yellow-colored clusters is made (b). The final result with and without this interaction is shown in (c) and (d), respectively. The NSF-awarded abstract data have been used. The detailed keyword summary is shown in Table 7

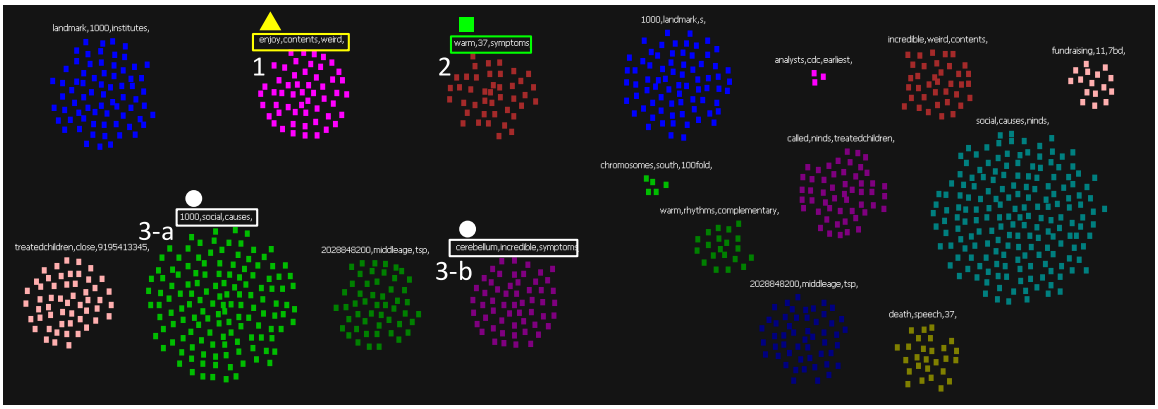
4.5.2.1 Moving data points in t-SNE

Fig. 16 shows an interesting interaction which involves moving a data point in t-SNE. Given some overlapping clusters in a particular visualization generated by t-SNE (Fig. 16(a)), users place several points from different clusters far apart (Fig. 16(b)), and then t-SNE reflects these changes in the following iterations, resulting in tje separation of most points in two clusters from each other (Figs. 16(c)(d)). This simple, yet powerful example clearly illustrates the advantage of providing users with



(a) The third-iteration result

(b) The fourth-iteration result after split/merge interactions



(c) The final (15th-iteration) result with split/merge interactions

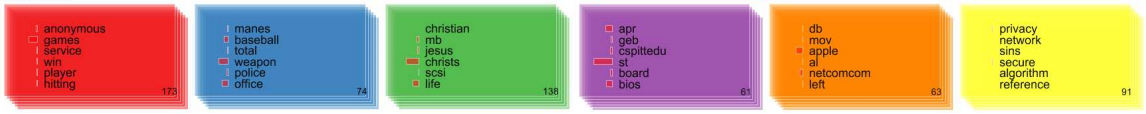
(d) The final (7th-iteration) result without split/merge interactions

Figure 20: An example of split/merge interactions. The yellow and green ones in (a) are merged to the same-colored ones, respectively, in (b), and the white one in (a) is split to the-same colored ones in (b). Webpages about autism have been used as an input data set. The detailed keyword summary is shown in Table 8

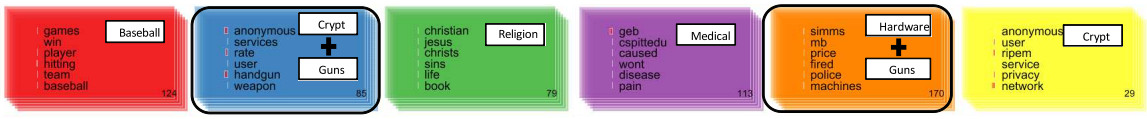
the ability to interact with computational methods in our framework in real-time visual analytics.

4.5.2.2 Fixing cluster assignments in k-means

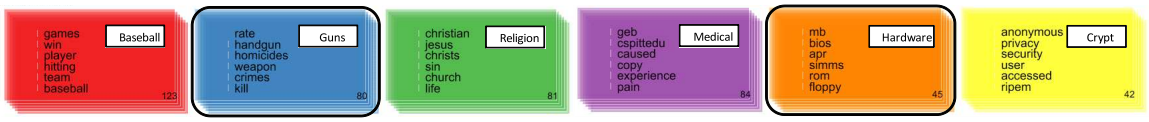
For our k-means method, we provide users with another interaction that allows them to fix cluster assignments for particular data items at a certain iteration. This interaction becomes especially useful when users feel that particular clusters are adequate and want to prevent them from changing much. In addition, fixing some clusters that



(a) The third-iteration result



(b) The 300th-iteration result



(c) The 700th-iteration result

Figure 21: An example of filtering documents whose cluster memberships are unclear. This interaction is done in the 300th iteration, and the topics become clearer in the later iterations. 20 newsgroups data have been used.

are already stable in early iterations can accelerate the later iterations by excluding them from the cluster assignment step.

Fig. 17 shows the effects of such interactions. First, we start with the same example shown in Fig. 19, but we fix the clustering assignments of the cluster highlighted in yellow rectangles, which amounts to 44% of the total data items, at the sixth iteration (Fig. 19(b)). Once this interaction is performed, the computing times for the following iterations of k-means drops significantly (Fig. 17(b)). However, only less than 10% of the final cluster memberships differ from the final results without this interaction, as shown in the increasing red line in Fig. 17(a). The final outputs of the cluster view in Jigsaw of the two cases can also be compared in Figs. 19(c) and (d), both of which are similar in terms of cluster sizes as well as keyword descriptions.

4.5.2.3 Split/merge clusters in k-means

Our customization of k-means enables users to merge multiple small or semantically related clusters or split large or unclear clusters. Fig. 20 shows its example in Jigsaw. In the third iteration, we merge yellow and green clusters and split a white cluster (Fig. 20(a)). The resulting is shown in Fig. 20(b). We obtain a much more balanced

set of clusters (Fig. 20(c)) compared to the final result in which no splitting/merging was performed (Fig. 20(d)). Furthermore, after analyzing the documents in two split clusters, we found that one of the clusters primarily contained documents about the causes of autism while the other about the symptoms, as seen in the keyword summary in Fig. 20(c). Without the interaction, one will notice in Fig. 20(d) that these clusters are not easily separated.

4.5.2.4 Filtering noisy documents to improve topics in LDA

The ability to filter noisy documents has been an appealing interaction for LDA in iVisClustering. To be specific, given parallel coordinate representations of topic-wise distributions of documents, users can interactively filter out documents that are not strongly related to a single topic, i.e., documents that have a very small maximum value in the topic-wise distribution. By removing them and re-running LDA, iVisCluster generally obtains significantly clearer topics. In PIVE, we performed this interaction near the 300th iteration (Fig. 21(b)), which is an early iteration when compared to the total number of iterations performed by LDA. However, such an interaction successfully generates clearer topics (Fig. 21(c)) over the standard approach where users have to wait for the algorithm to finish its full iterations in order to perform the same interaction.

4.6 Conclusions

We have presented PIVE (**P**er-**I**teration **V**isualization **E**nvironment for supporting real-time interactive visualization with computational methods). One of its apparent advantages is its ability to present users with the intermediate results during the interactions, which could reveal a significant amount of information immediately in visual analytics. Another important advantage is that it indeed opens up the possibility of performing small multiple interactions, which in the past have been considered to be too inefficient, and allows the real-time control over computational methods in visual

analytics. In fact, the interactions we proposed in this chapter are relatively simple, which do not involve any major algorithmic modifications, but after a sequence of interactions, the results reflects the intention of users sufficiently well in real-time. In this sense, PIVE makes them significantly useful by enabling users to perform these interactions easily and efficiently.

However, the advantage of our framework can be limited when the changes between iterations remain nontrivial, resulting in inconsistent visualizations. We have seen this kinds of limitations when using LDA under PIVE due to the random nature of the used LDA algorithm. As a future work, we plan to tackle this problem more actively by, for example, post-processing the results or even imposing additional constraints in computational methods so that the results from the following iterations do not change much from the current ones. Finally, another interesting research direction we will pursue is to extend PIVE to various parallelized computational algorithms for the large-scale data visual analytics.

CHAPTER V

TESTBED: AN INTERACTIVE VISUAL TESTBED SYSTEM FOR VARIOUS DIMENSION REDUCTION AND CLUSTERING METHODS

Many of the modern data sets such as text and image data can be represented in high-dimensional vector spaces and have benefited from computational methods that utilize advanced computational methods. Visual analytics approaches have contributed greatly to data understanding and analysis due to their capability of leveraging humans' ability for quick visual perception. However, visual analytics targeting large-scale data such as text and image data has been challenging due to the limited screen space in terms of both the numbers of data points and features to represent. Among various computational methods supporting visual analytics, dimension reduction and clustering have played essential roles by reducing these numbers in an intelligent way to visually manageable sizes. Given numerous dimension reduction and clustering methods available, however, the decision on the choice of algorithms and their parameters becomes difficult. In this chapter, we present an interactive visual testbed system for dimension reduction and clustering in a large-scale high-dimensional data analysis. The Testbed system enables users to apply various dimension reduction and clustering methods with different settings, visually compare the results from different algorithmic methods to obtain rich knowledge for the data and tasks at hand, and eventually choose the most appropriate path for a collection of algorithms and parameters. Using various data sets such as documents, images, and others that are already encoded in vectors, we demonstrate how the Testbed system can support these tasks.

5.1 Introduction

The volume of available data has been increasing at an exponential speed in recent years. Many of the modern data are generated in various forms such as documents and images of which the raw data can be represented in a high-dimensional vector space, allowing various computational methods to be applied. For instance, text documents can be encoded using a bag-of-words model, and images are represented using their feature point descriptors [91], resulting in hundreds of thousands of dimensions.

Given high-dimensional data, understanding and analyzing these data become more challenging. Visual analytics [78, 127] has gained increasing interest due to its capability of leveraging humans' ability of quick visual insight in data analyses and decision processes. However, many state-of-the-art visual analytics techniques or systems are not equipped for high-dimensional large-scale data. One of the reasons is that although humans are good at visually grasping an overall structure, when the number of visualized objects becomes large, it is often difficult to extract meaningful information from visualization. Another factor is the limited dimension of a screen space where high-dimensional data have to be visualized. For instance, parallel coordinates, a widely-used visualization technique for multi-dimensional data, do not scale well even when the dimension reaches several tens.

To improve this scalability issue, computational methods can support visual analytics by transforming the original data into a more compact and meaningful representation. Among various methods, two main ones, dimension reduction and clustering, play an essential role in visual analytics of large-scale high-dimensional data owing to their nature to reduce the numbers of features and data items into manageable sizes, respectively. Dimension reduction methods can reveal meaningful information by allowing the visual representation of high-dimensional data in a much lower-dimensional space. In addition, it allows visualization of high-dimensional data in the form of a 2D/3D scatter plot in which one can obtain insight about data relationships with

respect to the geometric locations of data. On the other hand, clustering provides an overview of large-scale data in terms of a small number of groups based on their semantic coherences. Such cluster information can then guide us to a proper data group of interest on which we can further focus our analysis.

Given a wide variety of computational methods including dimension reduction and clustering methods, it is not easy to determine which method to choose and how to use it properly for a certain data set and a certain task. Sometimes, when a specific method is used for a certain data set, its performance may be dependent on how the data is pre-processed beforehand. In addition, many modern computational methods often require decisions on multiple parameters. Yet there is no theoretical guideline for an optimal set of parameters for a given problem, and we have to go through multiple trials only to obtain some initial understanding of parameter values. As the algorithm gets more complicated, it becomes more difficult for users to understand what these parameters mean and how to select them properly. Consequently, many visual analytics systems choose a certain computational method, which is often basic and/or generic, and treat it as a black box with fixed parameter values while focusing on the subsequent analysis after obtaining the output from it. However, without an appropriate choice of algorithms and their parameters, the performance of these methods may not be satisfactory enough to start an analysis with.

Due to these difficulties, the current state of the art in visual analytics has not taken full advantage of the recent advancements of computational methods. To tackle this problem, we claim that users have to be provided with the capability of interactively trying out various computational methods and their parameters and reviewing their results at a visual level without having to know the details of algorithms. As a cornerstone to achieve this claim, this chapter presents an *interactive visual testbed system for dimension reduction and clustering*, the two essential computational methods for the visual analytics of large-scale high-dimensional data.

The main contributions of the proposed Testbed system are as follows. First of all, given various types of input data such as text documents, images, and vector-encoded data, the testbed system provides extensive capabilities to interactively select data pre-processing options and choose a wide variety of clustering and dimension reduction methods along with their parameters. The output of these processes are then visualized in several forms, e.g., parallel coordinates and a scatter plot, equipped with various interaction capabilities, e.g., accessing the original data items and brushing and linking between multiple views. Additionally, the Testbed system facilitates easy comparisons between different dimension reduction and clustering results by computationally aligning them. Finally, the Testbed system is implemented in a highly modular way so that new data types and dimension reduction/clustering methods can be easily integrated to the current system.

Note that even though the Testbed system can be used by anyone who wants to apply various methods to their own data, some background knowledge about machine learning and data mining would be of great help in fully utilizing the Testbed system via understanding the data and the applied methods simultaneously. For example, machine learning researchers/developers, who wish to easily plug in and visually evaluate their own methods in practical data analysis scenarios, would be able to receive significant benefit from the Testbed system.

The rest of this chapter is organized as follows. In Section 5.2, we review the relevant literature in terms of dimension reduction and clustering methods as well as the visual analytics systems adopting them. Section 5.3 describes the details of the Testbed system as well as several main computational methods used in the system. Section 5.4 shows various usage scenarios of the Testbed system, and finally Section 5.5 presents conclusions along with future work.

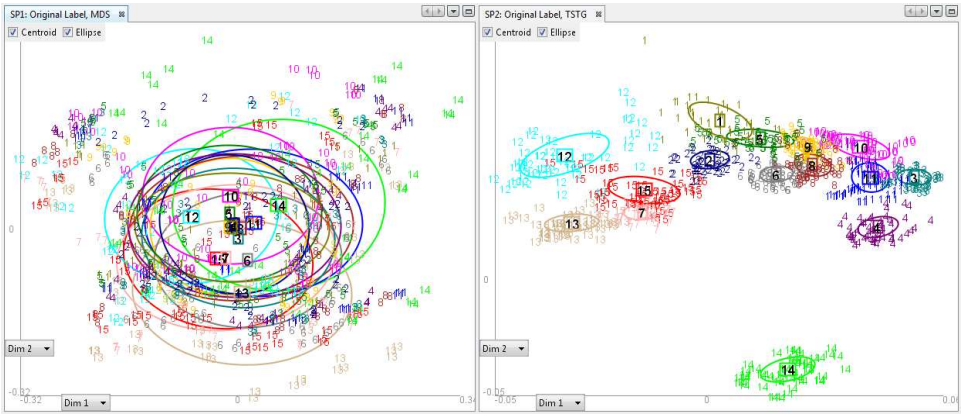


Figure 22: 2D Scatter plots obtained by two dimension reduction methods, MDS (left) and LDA (right), for a facial image data set. A different color corresponds to a different cluster.

5.2 Related Work

In this Section, various dimension reduction and clustering methods applicable to visualization are first reviewed. Afterwards, we discuss some of the currently available visual analytics systems that adopt these computational methods.

5.2.1 Dimension Reduction and Clustering for Visualization

Dimension reduction has long been one of the main research topics in data mining and statistical machine learning areas. Numerous dimension reduction methods have been proposed, among which the most commonly used dimension reduction methods include principal component analysis (PCA) [75], multidimensional scaling (MDS) [45], and linear discriminant analysis (LDA) [60, 68].

In addition to traditional data analysis problems, they have also been widely utilized in visualization due to their capability of representing high-dimensional data as a form of scatter plots in 2D/3D space. In a scatter plot, each data item is represented as a point and its 2D/3D coordinate is determined from the dimension-reduced representation. In general, the relative locations among data points reflect the pairwise relationships or proximities among data items.

Each dimension reduction method has its own optimization criteria and behaviors, which result in different visualizations. For instance, the recently proposed manifold learning algorithms, e.g., isometric feature mapping (ISOMAP) [125], locally linear embedding (LLE) [111], and Laplacian Eigenmaps (LE) [17], try to preserve the relationships between the local neighborhood rather than global relationships. These methods have been successfully applied to the data that originally have a low-dimensional manifold structure, and often they demonstrated their capability to reveal such a manifold structure in 2D/3D visualizations. However, most of these methods present just several visualization snapshots of limited data sets with no interaction abilities.

Another aspect to consider when applying dimension reduction in visualization applications is the cluster structure of data. A majority of dimension reduction methods take into account only the pairwise relationships between data items. In practice, however, it is not easy to obtain much insight from the 2D/3D scatter plot generated by them for a large number of data items. The left figure in Fig. 22 is a visualization example of such a dimension reduction method, MDS, for a facial image data set. Let us, for now, ignore the colors, which indicate the cluster labels. This visualization shows most of the data as a single chunk with a few outliers placed outside. Although these points may give some interesting insight about why they appear to be outliers, one cannot get much more information from this visualization.

Another type of dimension reduction methods incorporates additional information about the cluster structure of data in addition to individual data items. Since these dimension reduction methods require the assigned cluster label associated with each data item as an input, they are called supervised dimension reduction methods while the previous methods are called unsupervised dimension reduction methods. Some representative supervised methods include LDA [60] and orthogonal centroid method (OCM) [102]. The right figure in Fig. 22 is an example of LDA visualization. This

figure visualizes the data as groups of items computed by LDA based on the given cluster labels, and one can obtain better insight about the overall data structure at the cluster level over the individual data level.

Representing the cluster structure has been one of the main concerns in many studies on dimension reduction and 2D/3D scatter plot visualizations even when unsupervised dimension reduction methods are used. Many methods have been evaluated regarding their ability to visualize cluster structures, which are hidden at the time of computing dimension reduction. For instance, a recently proposed dimension reduction method, t-distributed stochastic neighborhood embedding (t-SNE) [130], shows its capability of grouping data and revealing the true cluster structure in 2D scatter plot visualizations.

Given the importance of the cluster structure in large-scale data visualization, clustering methods can add a significant value to visual analytics approaches by enabling visual understanding of the overview of data. Clustering partitions the entire data into groups or clusters so that the data items in the same cluster are more similar to each other than to those in different clusters. The resulting grouping information is in a form of cluster labels, which act as an additional categorical variable associated with data items. Such cluster information can be color-coded in visualization, as shown in Fig. 22, and help us understand the cluster structure in the data clearly in visualization.

Clustering, along with dimension reduction, has also been one of the well-studied research topics in data mining and machine learning areas. Widely-used methods include k-means clustering, spectral clustering [96], and Gaussian mixture models. Recently, more advanced methods such as non-negative matrix factorization (NMF) [79] and latent Dirichlet allocation [20] have shown their successful applications in image segmentation and document topic modeling, etc. These methods are usually evaluated using the data set whose cluster label information is already known and by

comparing between the true cluster labels with those obtained by the computational method. However, given the data set that may not have a clear cluster structure, clustering is typically a very challenging task, and thus it is often the case that the resulting cluster quality is unsatisfactory. From a visual analytics perspective, unsatisfactory clustering makes it difficult to understand the coherent meaning of each cluster and how one cluster contrasts with another. For instance, in recent applications of latent Dirichlet allocation for document topic modeling, while several coherent topic clusters have been successfully revealed for the document data, many other topics often seem unclear to understand.

Even with the obvious needs of computational methods such as dimension reduction and clustering in visual analytics, various issues such as data noise and improper algorithm and parameter choices, as described in Section 1, prevent their initial results from being practically useful enough to support the subsequent visual analysis. Nonetheless, among data mining and machine learning communities, which supply supposedly better computational methods, the efforts of interactively improving these initial results in real-world data analysis seem to be overlooked.

5.2.2 Visual Analytic Systems using Dimension Reduction and Clustering

In information visualization and visual analytics communities, various visual analytics systems incorporating computational methods such as dimension reduction and clustering have been proposed to deal with large-scale high-dimensional data. In this section, several systems such as IN-SPIRE [138], Jigsaw [121], GGobi [43], iPCA [72], and WEKA [65] are discussed.

IN-SPIRE [138] is one of the well-known visual analytics systems for document data in which dimension reduction and clustering play main roles. Given a set of documents, IN-SPIRE first encodes them as high-dimensional vectors using a bag-of-words model. Then it applies k-means clustering with a pre-defined number of

clusters. PCA is computed on cluster centroids and applied to the entire data, which gives 2D coordinates of document data. Based on these 2D coordinates, a galaxy view similar to a scatter plot is shown to users with a keyword summary for each cluster placed at the cluster centroid. Owing to the simple algorithms adopted such as PCA and k-means, IN-SPIRE can deal with a fairly large amount of data, but it provides only a limited number of interaction capabilities to change the algorithms and their settings.

Jigsaw [121] is another well-known visual analytics system for document analysis. The main information that Jigsaw utilizes for visualization is named entities such as person name and location and their co-occurrences between documents. Automatic named-entity extraction is one of the key computational components in the analysis in Jigsaw. The named-entity extraction can be viewed as dimension reduction that reduces the number of keywords out of the entire vocabulary. Users can modify the list of named-entities by manually adding/removing them. Jigsaw also provides a cluster view by using the k-means algorithm and visualizes the resulting clusters as groups of documents as well as their keyword summary. Jigsaw also supports basic interactions with clustering such as changing the number of clusters and providing seed documents.

GGobi [43] is an interactive visualization system for high-dimensional data that are already encoded. It mainly uses a 2D scatter plot, where the two dimensions are generated by grand tour [10]. The difference of grand tour from other dimension reduction methods is that it provides an interaction to explore the high-dimensional space by continuously changing the basis vectors that data items are projected into. However, the grand tour method is applicable only when the data dimension is not significantly high, and thus its application is limited when dealing with hundreds or thousands of dimensions, which is often the case in many data types such as text documents, images, and bio data.

Another system, iPCA [72], which also takes high-dimensional data as an input, utilizes PCA as the main visualization technique. One of the main advantages of iPCA is that beyond 2D/3D scatter plots, it visualizes the reduced-dimensional data in a higher dimension than 2D or 3D via parallel coordinates. In general, dimension reduction from the original high-dimensional space to 2D/3D space introduces significant information loss. In iPCA, it follows a useful idea to reduce the data dimension to an intermediate one that can be visualized without much clutter via parallel coordinates and then to interact with these intermediate dimensions to obtain particular scatter plots. Another aspect of iPCA is that it visualizes the PCA basis vectors in addition to the data items. In doing so, users can understand the role of the reduced dimensions in their visualizations, which leads to a better understanding about the data set as well. Even with these advantages, however, iPCA cannot handle very high-dimensional data since iPCA visualizes each of the original dimensions.

Finally, WEKA [65] is mainly a library of various machine learning algorithms for high-dimensional data with several interaction capabilities. Various algorithms can be applied to data, and their performances can be evaluated based on various measures. In addition, WEKA provides simple types of visualizations such as histograms, scatter plots, etc. Although WEKA is similar to our Testbed system in that it provides flexible algorithm choices and settings, most of its visualizations and interactions are focused on the used methods rather than data exploration. For example, WEKA does not support any interactions from its visualizations such as filtering operations and raw data access.

As discussed above, most of the current visual analytics systems do not fully utilize a wide variety of computational methods. They adopt generic traditional methods for a broad applicability to various data sets and/or treat computational methods as a black box with little options to control them, which would hamper the interactive visual analysis. In this respect, the Testbed system provides the unique capability of

bringing a variety of algorithms along with full control to practical visual analytics scenarios.

5.3 Testbed System

In this Section, we describe the Testbed system[1] in detail. First, in Section 5.3.1, we introduce the modules in the system and explain how the overall system works. Next, we describe the details of each module from both the computational and the interactive visualization points of view in Sections 5.3.2 and 5.3.3, respectively. Finally, in Section 5.3.4, we discuss implementation details of the system and how the current system can be extended to adopt new data types and clustering/dimension reduction methods.

5.3.1 Basic Workflow

As shown in Fig. 23, the Testbed system mainly has two parts: the computational and the interactive visualization parts. At the computational part, the Testbed system is composed of 1. vector encoding, 2. pre-processing, 3. clustering, and 4. dimension reduction. At the interactive visualization part, the Testbed provides the following interactive visualization modules: 1. parallel coordinates, 2. the scatter plot, 3. the cluster label view, and 4. the original data viewer.

The basic workflow of the Testbed system is as follows. Once a data set is loaded, data items are represented as high-dimensional vectors via a default encoding scheme. Then, users can interactively change the options for pre-processing, clustering, and dimension reduction methods. Each specification of these three components instantiates a particular visualization set composed of the parallel coordinates view, the scatter plot view, and the cluster label view. To generate these views, the output

[1] An introductory video can be downloaded at `http://fodava.gatech.edu/files/testbed-software/testbed.mp4`, and the executable files with the used data sets are available at `http://fodava.gatech.edu/fodava-testbed-software`.

of dimension reduction, i.e., reduced-dimensional representation, acts as the coordinates of data items in the parallel coordinates view, and two user-selected dimensions of this view are visualized in the scatter plot view. In all three views, the output of clustering, i.e., grouping information of data items, is color-coded along with the cluster name/summary provided in the cluster label view.

The Testbed system can generate as many visualization sets as needed depending on different specifications of dimension reduction and clustering, and users can explore a certain visualization set and compare between different visualization sets. To facilitate an easy comparison between different visualization results, the Testbed system offers the capability of aligning the different clustering and dimension reduction outputs. In addition, users can highlight and/or filter out certain clusters/data items and look into the details of the selected data items in the original data viewer. Users can also apply another set of clustering and/or dimension reduction to the selected data items to create new visualization sets.

5.3.2 Computational Modules

5.3.2.1 Vector Encoding

The Testbed system can take various types of data such as text documents, images, and pre-encoded vectors in a comma-separated-values (CSV) file format. For document and image data, the Testbed system provides built-in vector encoding modules. For instance, the Testbed system supports bag-of-words encoding for document data in a sparse matrix form with stop word removal and stemming. Image data are converted into vectors of rasterized gray-scale pixel values. The high-dimensional vectors obtained in this stage act as initial default vectors on which the following pre-processing is performed.

5.3.2.2 Pre-processing

Once the default vectors are generated, the system shows pre-processing options depending on the data type (Fig. 23A). The following options are provided in common for all data types: 1. normalization, which scales data vectors so that their norms equal to one, and 2. centering, which translates data vectors so that their empirical mean is zero.

In addition, for text documents, we provide options of 1. removing the terms appearing in less than a user-specified number and 2. applying the term-frequency-inverse-document-frequency (TF-IDF) weighting scheme. For images, available are the following options: 1. reducing image sizes to a user-specified ratio to enhance the computational efficiency and 2. applying contrast limited adaptive histogram equalization [107].

The Testbed system maintains multiple instances of different pre-processed vector sets, and users can interactively generate and/or choose one of them and proceed to perform its clustering and dimension reduction.

5.3.2.3 Clustering

Given the default or pre-processed set of high-dimensional vectors, the clustering module performs a user-selected clustering method with specified options (Fig. 23B), which assigns each data item a cluster label. The Testbed system currently provides the following clustering methods: 1. k-means, 2. agglomerative hierarchical clustering [66], 3. Gaussian mixture models, and 4. NMF. Once a specific method is selected in the system, user interfaces to specify the number of clusters as well as method-specific parameters are dynamically shown with their suggested default values (Fig. 23B).

Additionally, when a data set has pre-given labels, the clustering method list includes an additional item called 'Use original labels' so that users can explore data with respect to the pre-given labels.

5.3.2.4 Dimension Reduction

Given the high-dimensional vector representations of data items, the dimension reduction module reduces the data dimension from possibly hundreds of thousands to a visually manageable size, which makes it possible to visualize the data in forms of parallel coordinates and/or a scatter plot. The Testbed system provides both supervised and unsupervised dimension reduction methods, as discussed in Section 5.2.1. In cases of supervised methods, the cluster label, which is an additional required input to run dimension reduction, is taken from the output of the clustering module.

The currently available dimension reduction methods in the system include supervised ones such as 1. LDA, 2. OCM, 3. centroid method (CM) [102], 4. two-stage methods (TSTG) [35], 5. discriminative neighborhood metric learning (DNML) [133], and 6. kernel LDA [101], and unsupervised ones such as 7. PCA, 8. metric and non-metric MDS, 9. Sammon mapping [113], 10. ISOMAP, 11. LLE, 12. local tangent space alignment (LTSA) [148], 13. maximum variance unfolding (MVU) [135], 14. LE, 15. diffusion maps (DM) [42], 16. t-SNE, and 17. Kernel PCA [115]. Similar to the clustering module, once a specific method is selected in the system, user interfaces to specify the number of reduced dimensions as well as method-specific parameters are dynamically shown along with their suggested default values (Fig. 23C).

5.3.3 Interactive Visualization Modules

5.3.3.1 Parallel Coordinates View

Given an output from the computational part, i.e., lower-dimensional representations of data items and their cluster labels, the Testbed system takes a natural way to visualize the lower-dimensional data in parallel coordinates with a color coding based on the cluster labels (Fig. 23E). In this view, the Testbed system supports zoom-in/out via mouse wheel scroll and data selection via mouse drag-and-drop.

5.3.3.2 Scatter Plot View

Although parallel coordinates can fully visualize an output from the computational part, this view is often ineffective for humans to perceive the relationships between data items, and it does not scale well in terms of the number of data items and dimensions since each line representing a single data item occupies numerous pixels in a screen space. Due to these limitations, the Testbed system visualizes data in a 2D scatter plot (Fig. 23F) by selecting two of the parallel coordinates dimensions with the same color encoding as in the parallel coordinates view.

In the scatter plot view, users can interactively change these dimensions corresponding to horizontal and vertical axes via combo boxes shown in the lower left part of the view. In addition, the Testbed system shows cluster centroids and ellipses, which summarize how the data within each class are distributed, and these features can be turned on/off via check boxes shown in the upper left part of the view. Similar to the parallel coordinates view, supported are zoom-in/out via mouse wheel scroll and data selection via mouse drag-and-drop. Once a subset of data is selected, users can apply another clustering/dimension reduction only on the selected data items.

5.3.3.3 Cluster Label View

The cluster label view (Fig. 23G) shows the cluster index, color, and summary in a simple list form. Currently, the cluster summary is provided only for the text documents type, which is the most frequent keywords in each cluster. Upon clicking a certain cluster index or summary, the corresponding data items are highlighted with thicker lines/points in the parallel coordinates and the scatter plot views. Unchecking the checkboxes, which are shown in the left side of the view, hides the corresponding data items in the two views

5.3.3.4 Accessing Original Data

Both in the parallel coordinates and the scatter plot view, users can access the original form of data for user-selected data items/clusters in the original data viewer. Currently, the Testbed system provides three different original data viewers depending on the data type, e.g., text documents, images, and pre-encoded vectors (Fig. 23H).

In all the original data viewers, selected data items are shown as a list with their cluster colors on the left, and users can multi-select items in the original data viewer to see the original data items. These selected items are then highlighted with dark yellow marks in the scatter plot view (Fig. 23H). For text documents, adopting the idea in [88], we color-code a user-specified number of representative keywords per each cluster with the corresponding cluster color, which helps in understanding why a document belongs to a certain cluster. For pre-encoded vectors, if these vectors are associated with another type of data, these data can also be accessed as shown in the third viewer in Fig. 23H.

5.3.3.5 Supporting Multi-view Exploration

Once the 'visualize' button is clicked (Fig. 23C) after specifying computational methods, i.e., pre-processing, clustering, and dimension reduction, a new set of the parallel coordinates, the scatter plot, and the cluster label views are instantiated. Each of these three views is created as an individual tab in its corresponding location, and multiple views are maintained flexibly in the Testbed system, as shown in Fig. 23(a). For example, any view can be popped out as an independent window and/or split horizontally/vertically in order to make it easy to compare between different sets of views due to different computational methods. When a certain view is activated by a mouse click, all the options of pre-processing, clustering, and dimension reduction used to generate the view are shown in the left in Fig. 23(a).

Between different views with such a flexible layout, the Testbed system supports

a brushing-and-linking capability. In the current Testbed system, if certain data items/custers are selected in one view, the corresponding data items in all the other views are highlighted as well. We use different colors for highlighting depending on whether the highlighted data items are due to the same view or a different view, which helps identifying the source view in which the data selection was made.

5.3.3.6 Aligning Different Views

In addition to the above-described multi-view management and brushing-and-linking capability, the Testbed system provides a more active means to facilitate easy comparison between visualization sets composed of different clustering and dimension reduction results. To be specific, for a user-selected pair of visualization sets, users can align the clustering and/or dimension reduction outputs (Fig. 23D), which are then reflected to visualization sets.

To align the two different clustering results, the Testbed system performs the Hungarian algorithm [85]. Given two different cluster assignments of the same data items, the Hungarian algorithm finds the best pairwise matchings between their cluster indices so that the number of common data items within matching cluster pairs is maximized. Once the Hungarian algorithm finishes, the Testbed system changes the cluster indices and colors of the second visualization set according to the matching clusters of the first visualization set. As a result, users can maintain the consistent cluster indices/colors when comparing the two given visualization sets.

On the other hand, the Testbed system handles the alignment of dimension reduction results via Procrustes analysis [69, 53]. Although there exist many advanced methods to align the two sets of vectors [30], we chose Procrustes analysis due to its computational efficiency. Procrustes analysis transforms the second set of vectors via a rigid transformation, which allows only translation, rotation, and reflection, so that

their Euclidean distances to the corresponding data vectors in the first set are minimized. Currently, instead of aligning the entire dimensions, the Testbed aligns only the two dimensions selected in the scatter plot view so that the alignment between the two scatter plot views are maximized. These alignment functionalities help users understand how differently the corresponding data items/clusters are placed between the two scatter plot views.

5.3.4 Implementation and Extensibility

The current Testbed system is mainly implemented in JAVA to achieve various GUI and interaction capabilities. In order to support flexible window management, NetBeans Rich Client Platform and IDE[2] are used.

Most of the internal computational methods are, however, written in MATLAB. There are several reasons of using MATLAB codes instead of porting them to JAVA. First of all, in many cases, the source codes of advanced computational methods are readily available in MATLAB due to its simplicity for matrix computations. In this respect, it would be burdensome to re-implement each of these methods in a different programming language in order to make them visual and interactive, and it will eventually become difficult to keep up with the pace of new technologies.

Furthermore, MATLAB provides highly optimized matrix computations. For instance, MATLAB, by default, auto-identifies the parallelizable subroutines in the code and runs full CPU cores even in a single PC. There also exist many efficient mature core computational methods. For example, the k-means function in MATLAB provides various options for a distance metric to be used (Euclidean, city block, cosine, and correlation), and a seed initialization (random, uniform, pilot-clustered, and user-selected seeds). Due to these reasons, the current Testbed system interface with the computational methods via a custom JAVA library file created by MATLAB.[3]

[2]http://netbeans.org/features/platform/index.html
[3]http://www.mathworks.com/products/javabuilder/

In terms of the extensibility of the Testbed system, we designed it in a completely modular way so that it can easily accept new data types and clustering/dimension reduction methods. For instance, if one wants to use the Testbed system for a speech data type, one needs to implement only the encoding module, the possible pre-processing options specific to the speech data type, and the original data viewer that can play audio data. Otherwise, by performing vector encoding separately and putting the encoded vectors as an input to the system, one can easily utilize the full capability of the Testbed.

Adding new dimension reduction/clustering methods is also a simple process. Currently, the implementation of each computational method is composed of two source code files. One file performs the computation by taking an input and generating an output as a primitive two-dimensional double array type, and the other is for user interfaces to change the method-specific options. Therefore, whether the implementation of a new method is written in MATLAB or JAVA, as long as it deals with two-dimensional double array type as an input and an output, it can be easily integrated into the current Testbed system without having to modify the entire system.

5.4 Usage Scenarios

5.4.1 Data Sets

To show how the Testbed system can be utilized in various visual analytics scenarios, we use three different data sets: 1. Pendigits (pre-encoded vectors), 2. Weizmann (images), and 3. InfoVisVAST (text documents).

The Pendigits data set[4] is composed of 10,992 handwritten digit data items, each of which is a 16-dimensional vector representing pen trace coordinates [11]. The data set has 10 clusters in terms of which digit each data item corresponds to, i.e., '0', '1', ..., and '9.' For our experiments, 50 data items have been chosen from each

[4]http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits

cluster, resulting in 500 items in total. The Weizmann data set[5] contains 28 persons' facial images with various angles, illuminations, and facial expressions. Excluding unclear images, we have chosen 52 images from each of 15 persons, resulting in 780 data items of 15 clusters. The size of each facial image is 88×128, resulting in a 11,264 dimensional vector. The InfoVisVAST data set[6] is a document corpus of paper abstracts in IEEE Infovis (1995-2010) and VAST (2006-2010) conferences. It includes 515 documents encoded in 4,185 dimensions via a bag-of-words encoding after stemming and stop word removal.

5.4.2 Parallel Coordinates: Guiding beyond Two Leading Dimensions

When using dimension reduction in high-dimensional data visualization, the leading two dimensions of a dimension reduction method have been usually used to generate a single scatter plot while ignoring the other dimensions. Unlike these previous approaches, the Testbed system first visualizes the reduced-dimensional data in the parallel coordinates view, and then two of these dimensions are interactively selected for the scatter plot view.

Although it is difficult to visually analyze data relationships in the parallel coordinates view, it can guide users in various ways. First of all, as shown in Fig. 23(a), TSTG, e.g., LDA in this case, tends to separate different clusters into different dimensions. For instance, although the clusters '2' and '3' are not well separated in the scatter plot view with (1, 2)-dimensions selected, they seem to be separated from the other clusters in dimensions 7 and 4, respectively, based on the parallel coordinates view.

As another example, as shown in Fig. 24, given the 10-dimensional results of PCA, the scatter plot view of (1, 2)-dimensions mixes up all the clusters together. However, hinted by the parallel coordinates view showing the peaks of the cluster '12'

[5]http://www.wisdom.weizmann.ac.il/~vision/FaceBase
[6]http://www.cc.gatech.edu/gvu/ii/jigsaw/datafiles.html

and '14' at dimensions 3 and 4, respectively, the scatter plot view of these dimensions turns out to give a well-clustered view. Such an observation is surprising because PCA is an unsupervised method, which does not take into account label information. This indicates that the leading two dimensions may not give enough information for high-dimensional data in visual analytics.

5.4.3 Effects of Alignment: Helping Comparisons between Visualizations

Trying various methods/settings on a given data set and comparing between different visualizations is in the heart of the Testbed system, and the alignment functionality of the system supports this process. Fig. 25 shows the effects of the alignment for clustering and dimension reduction.

In Fig. 25(a), which shows the former, the three scatter plot views have identical coordinates of data items. With the different assignment of cluster labels, it is difficult to compare the cluster membership between the first and the third plots since the clusters have no correspondences in terms of the cluster colors and indices. After aligning the clustering, however, two different clustering results become much easier to compare between the first and the second view. For instance, compared to the original cluster labels shown in the first view, the original cluster '8' is shown to be merged to the original cluster '3.' The two subclusters of the original cluster '6,' which are shown in the bottom left and the top right in the first figure, are now split into two clusters in k-means clustering, and the former is shown to be merged to the clusters '4' and '10.'

On the other hand, Fig. 25(b) shows the example of aligning dimension reduction. In this example, the cluster labels are unchanged for all data items in the three figures, but two different dimension reduction methods, TSTG and ISOMAP, are used. When comparing between the first and the third figures, which show the different

coordinates generated by these two methods, it is difficult to recognize the correspondences between data items/clusters. Between the first and the second figures, whose dimension reduction results are aligned, one can perceive the correspondences in a much easier way. For example, the cluster '4' is shown to be close to the cluster '8' in TSTG, which is not the case in ISOMAP. Any data items in the cluster '6' are not located close to the cluster '7' in ISOMAP, but some data items between the two clusters overlap in TSTG. Such analyses cannot be easily made without the alignment.

5.4.4 Dimension Reduction: Supporting Multiple Perspectives

Different dimension reduction methods can reveal different aspects of data. To show an example, we now look into the first two figures in Fig. 25(b) from the perspective of supervised vs. unsupervised methods. Given a certain assignment of cluster labels, a supervised method, TSTG, gives a clear overview in terms of cluster relationships since most of the clusters are shown relatively compact, as shown in the first figure. On the contrary, an unsupervised method, ISOMAP, may reveal different aspects of data. For example, the second figure indicates that the cluster '6' is composed of two distinct subclusters shown at the top left and the bottom right. However, when the data do not have a clear cluster structure, e.g., most of the text document corpora, unsupervised dimension reduction methods give the results similar to the second figure in Fig. 24, which significantly reduces the utility of the scatter plot. In this case, supervised dimension reduction would be the only choice to start with in visual analytics.

Even with a single dimension reduction method with different parameter values, different aspects of data can be obtained. In Fig. 26(a), one can see that the cluster '5' (the digit '4') of the Pendigits data set moves from the top left near the cluster '10' (the digit '9') towards the cluster '7' (the digit '6') as the ISOMAP parameter

k increases. In general, ISOMAP with a smaller k value focuses more on preserving the local neighborhood relationships by making non-neighborhood distances longer. Based on the sample data of each digit shown in 26(b), it can be inferred that the digit '4' is represented much closer to the digit '9,' which forms their neighborhood relationships with small k values, than to the digit '6.' As the k value increases, the neighborhood relationship between the digits '4' and '6' starts to be formed, which is why they become closer at a bigger k value. In this way, varying the parameter values with the same method can further reveal different interesting insight about data.

5.4.5 Clustering: Combining Knowledge from Different Clustering

Clustering is a challenging task, and any single clustering method tends not to give fully satisfactory results. The Testbed system can remedy this problem by enabling users to perform different clustering methods and obtain more meaningful clusters by comparing between them. Fig. 27 shows the scatter plot views of TSTG with the cluster labels obtained by two different clustering methods, k-means and NMF. The InfoVisVAST data set are used, and the keyword summaries of clusters for each method are as follows:

k-means

1. graph, trees, node, layout, edge, draw, clusters
2. querying, interface, multiple, databases, expressive, temporal, magnification
3. document, text, collections, words, sequential, searches, information
4. multivariate, variable, data, aggregate, coordinates, multidimensional, flow
5. 3d, spatial, labelling, animation, map, coloring, display, information
6. treemaps, hierarchy, hierarchical, layout, focuscontext, spacefilling, algorithms
7. clusters, dimensions, image, visualization, measures, number, reduction
8. analytics, model, systems, video, decisions, information, framework
9. networks, traffic, arcs, diagram, social, internet, duplicate
10. collaboration, designed, histories, wikipedia, information, supports, story

NMF

1. graph, clusters, algorithms, methods, data, state, structured

2. querying, interface, databases, searches, temporal, multiple, data
3. document, text, image, content, information, collections, searches
4. dimensions, parallelize, coordinates, multivariate, multidimensional, datasets, scatterplots
5. 3d, spatial, landscapes, information, display, animation, spaces, encoded
6. treemaps, hierarchical, layout, ratio, algorithms, spacefilling, aspects
7. trees, hierarchy, node, genealogical, decisions, draw, layout
8. designed, model, information, analytics, framework, systems, data
9. networks, traffic, social, querying, analysis, data, flow
10. collaboration, analytics, wikipedia, analysts, supports, knowledge, shared

Among these clusters, the cluster '1' of the k-means clustering has a clear meaning of graph-related visualization, e.g., graph drawing, graph layout, and graph clustering. This cluster is also shown to be clearly separated from the other clusters in the left figure. As we perform brushing-and-linking on this cluster, it turns out that this cluster mainly corresponds partially to the clusters '1,' '6,' and '7' of the NMF clustering. Considering that these clusters contain the keywords, 'graph' and 'layout' and their separations from the other clusters in the right figure are not as clear as that of the cluster '1' in the left figure, one can regard the cluster '1' of k-means as a cluster with better quality.

On the other hand, in the NMF clustering, the cluster '4' seems to be clearly related to multi-variate/multi-dimensional data visualization. By brushing-and-linking on this cluster, we found it corresponds mostly to the clusters '4' and '7' of the k-means clustering, which makes sense based on their keyword summaries although they are relatively more ambiguous than the cluster '4' of the NMF clustering. This observation is also supported by their cluster separations in Fig. 27 , which indicates a clearer separation of the cluster '4' in the right figure than that of the clusters '4' and '7' in the left figure.

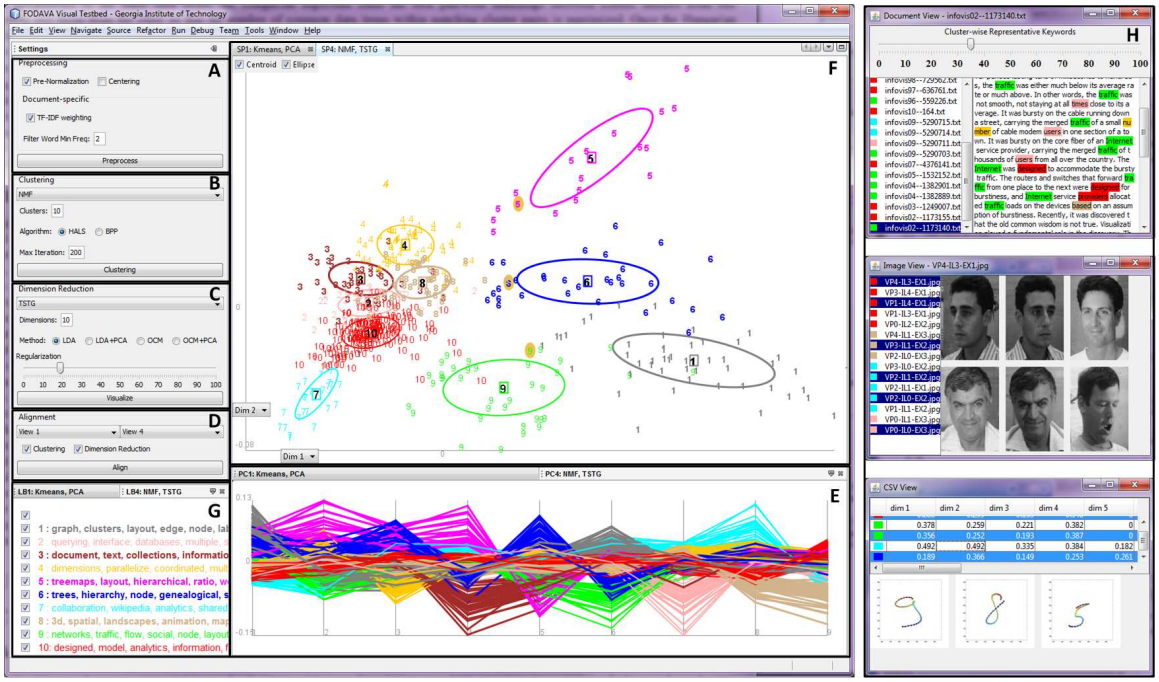
As shown in these cases, one can apply different clustering methods and take full advantage of them by visually analyzing them in the Testbed system.

5.5 Conclusions

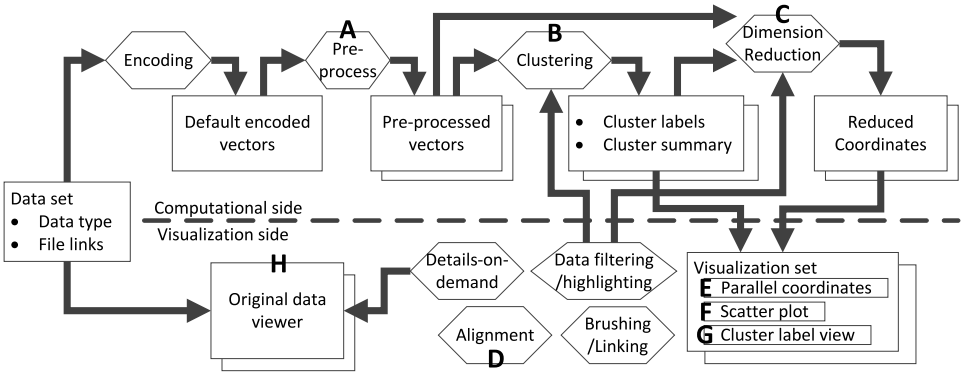
In this chapter, we have presented the visual testbed system for dimension reduction and clustering in high-dimensional data visual analytics. The main contribution of our system is to bring a wide variety of traditional and state-of-the-art dimension reduction and clustering methods to visual analytics. The Testbed system provides full control of these methods with interactive visual access to their results. In addition, our system offers a flexible extensibility for new data types and methods.

As future work, we plan to tackle a scalability issue. As the size of data gets bigger, their computational time takes even longer, which hinders real-time interactive visualizations. Another scalability problem is due to the limited amount of screen space. Even if the computational methods maintain efficiency, a large number of data items cause a clutter in visualization. These issues will be handled using various approaches, e.g., sampling, online learning algorithms, etc.

In addition, we plan to enhance the alignment capability by incorporating other advanced algorithms and user interfaces. To be specific, the currently used algorithms do not change anything in the reference view, and the Procrustes analysis does not change internal relationships within each visualization at all. This may limit the performance of alignment for easy comparison between visualizations when they are significantly different. To deal with this problem, we plan to utilize other advanced methods such as graph-embedding-based methods [30].



(a) The system overview.



(b) The general workflow. Hexagonal blocks correspond to operations/interactions, and rectangular ones to operation inputs/outputs or visualization modules. Stacked rectangles indicate their multiple instantiations, which are dynamically maintained by the system.

Figure 23: The overview and the workflow of the system. User interfaces for preprocessing (A), clustering (B), dimension reduction (C), and alignment (D) are available. Lower-dimensional data from dimension reduction are visualized as parallel coordinates (E), and the two selected dimensions are shown in the scatter plot (F). Cluster indices/summaries (G) are shown, and the original data can be accessed (H).

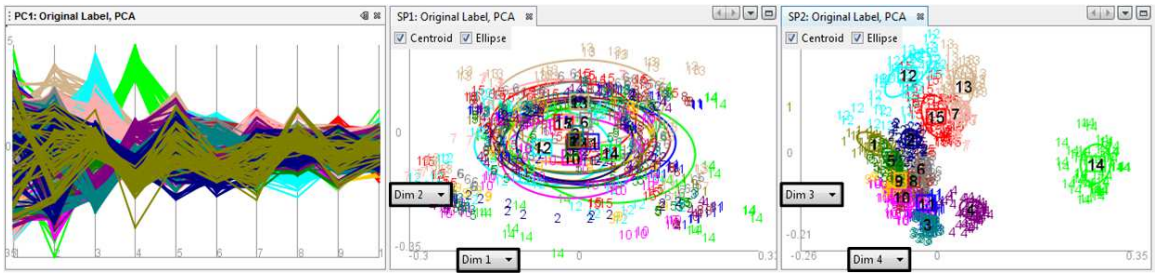
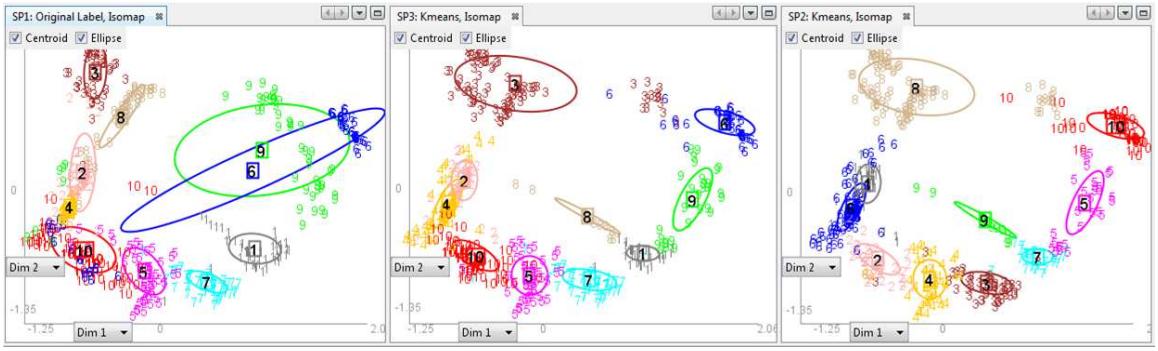
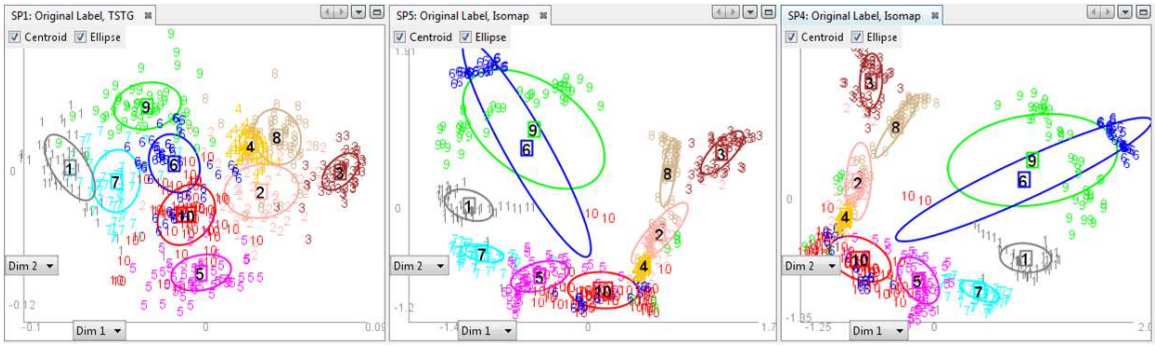


Figure 24: The 10-dimensional results of PCA for the Weizmann facial image data set. The pre-given person ID was used as a color label. The first figure is the parallel coordinates of the entire 10-dimensional representations, and the second and the third are the scatter plots of (1, 2)- and (3, 4)-dimensions, respectively.

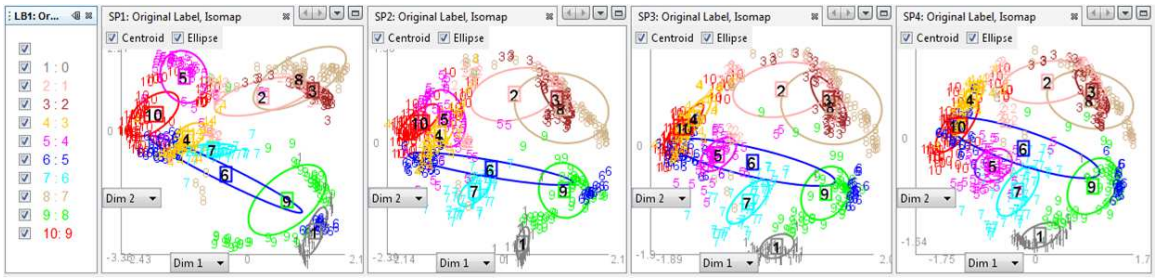


(a) The alignment of clustering. For the Pendigits data set, the first figure uses the original cluster labels, and the other two uses the same cluster labels generated by k-means. In all three figures, ISOMAP is used with the same parameter values.

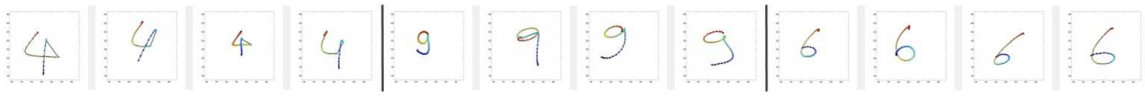


(b) The alignment of dimension reduction. For the Pendigits data set, the first figure uses TSTG and the other two use ISOMAP with the same parameter values. In all three figures, the original cluster labels are used.

Figure 25: The effects of alignment. In both (a) and (b), the first is the reference scatter plot view for alignment, and the second is the aligned plot of the third while the third is an un-aligned one.



(a) The scatter plots for the Pendigits data set generated by ISOMAP with different parameter values, k =12, 20, 30, and 50, respectively. The cluster labels represent the digits of data items, as shown on the left. The three figures on the right are aligned plots with respect to the first. As the parameter increases, the cluster '5' (the digit '4'), moves from the top left near the cluster '10' (the digit '9') towards the cluster '7' (the digit '6').



(b) The sample data for the digits '4', '9', and '6.' Note that the vector representation of the Pendigit data set encodes the pen trace coordinates, which start at the red color and end at the blue in these samples.

Figure 26: The effects of a parameter change in ISOMAP.

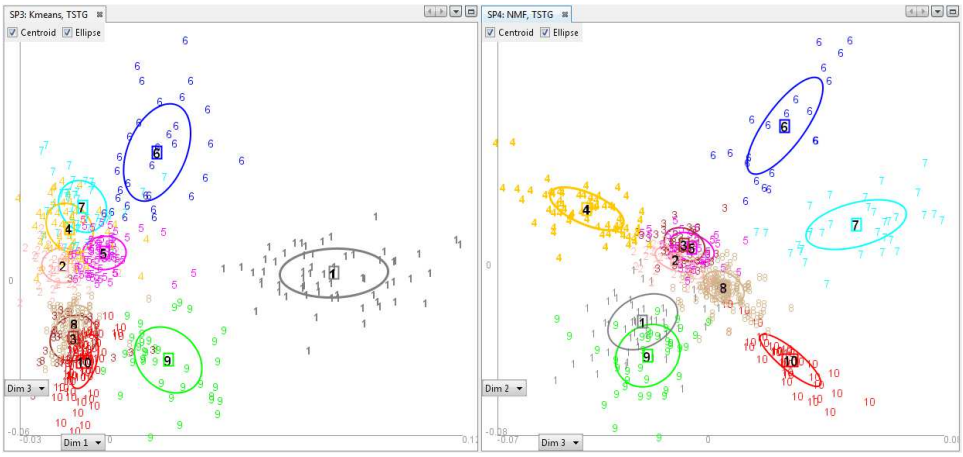


Figure 27: The scatter plot view of two different clustering, k-means and NMF, using TSTG for the InfoVisVAST data set. The right figure is aligned with respect to the left one for both clustering and dimension reduction.

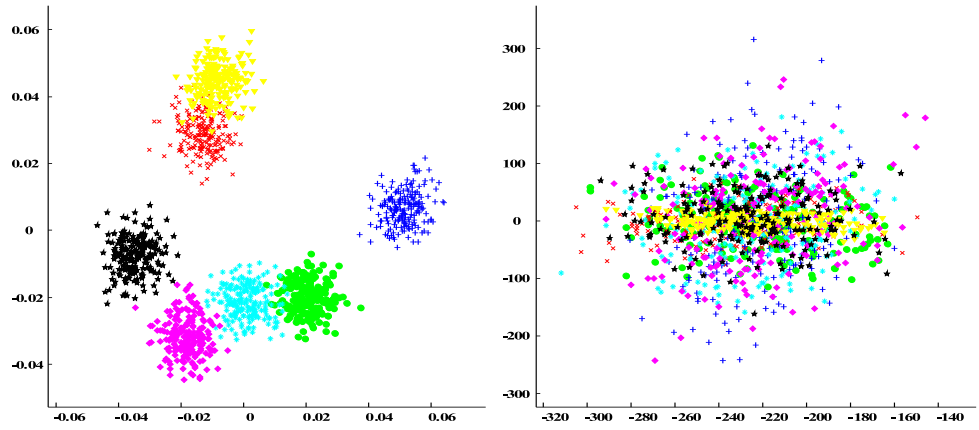
CHAPTER VI

IVISCLASSIFIER: AN INTERACTIVE VISUAL CLASSIFICATION SYSTEM USING SUPERVISED DIMENSION REDUCTION

We present an interactive visual analytics system for classification, iVisClassifier, based on a supervised dimension reduction method, linear discriminant analysis (LDA). Given high-dimensional data and associated cluster labels, LDA gives their reduced dimensional representation, which provides a good overview about the cluster structure. Instead of a single two- or three-dimensional scatter plot, iVisClassifier fully interacts with all the reduced dimensions obtained by LDA through parallel coordinates and a scatter plot. Furthermore, it significantly improves the interactivity and interpretability of LDA. LDA enables users to understand each of the reduced dimensions and how they influence the data by reconstructing the basis vector into the original data domain. By using heat maps, iVisClassifier gives an overview about the cluster relationship in terms of pairwise distances between cluster centroids both in the original space and in the reduced dimensional space. Equipped with these functionalities, iVisClassifier supports users' classification tasks in an efficient way. Using several facial image data, we show how the above analysis is performed.

6.1 Introduction

Classification is a widely-used data analysis technique across many areas such as computer vision, bioinformatics, text mining, etc. Given a set of data with known cluster labels, i.e., under a supervised setting, it builds a classifier (a training phase) to predict the label of new data (a test phase). Examples of classification tasks include



(a) LDA (b) PCA

Figure 28: 2D Scatter plots obtained by two dimension reduction methods, LDA and PCA, for artificial Gaussian mixture data with 7 clusters and 1000 original dimensions. A different color corresponds to a different cluster.

facial recognition, document categorization, spam filtering, and disease detection.

Numerous classification algorithms such as an artificial neural network, decision trees, and support vector machines have been developed so far, and each method has advantages and disadvantages making it more suitable in certain domains. Even with its broad applicability, however, most of the classification algorithms are often performed in a fully automated manner that prevents users from not only understanding how the algorithm works on their data but also reflecting their domain knowledge into the classification process. Ironically, as classification algorithms become more sophisticated and advanced, they tend to be less interpretable to users due to their complicated internal procedure. These limitations may cause unsatisfactory classification results in real-world applications such as biometrics in which the reliability of the system is critical [137]. In some cases, there may be no option other than using the manual classification process without being supported by automated techniques.

This chapter addresses how visual analytics systems support automated classification for a real-world problem. As in other analytical tasks, the first step is to understand the data. From a classification perspective, users need to gain insight in

terms of clusters such as how much the data within each cluster varies, which clusters are close to or distinct from each other, and which data are the most representative ones or outliers for each cluster. The next step is to understand both the characteristics of the chosen classifier itself and how they work on the data at hand. For instance, decision trees give a set of rules for classification, which are simple to interpret, and users can see which features in the data play an important role. In addition, analysis of misclassified data provides a better understanding of which types of clusters and/or data are difficult to classify. Such insight can then be fed back to the classification process in both the training and the test phases. In the training phase, users can refine the training data or modify the automated classification process for better performance in the long run. In the test phase, users can actively participate in determining the label of a new data by verifying each result that the automated process suggests and by performing further classification based on the interaction with a visual analytic system. The latter case ensures nearly perfect classification accuracy while maintaining much better efficiency than in the case of purely manual classification.

Not all classification algorithms are suitable for interactive visualization of how they work. Moreover, when the data is high dimensional such as image, text, and gene expression data, the problem becomes more challenging. To resolve this issue, we choose the classification method based on linear discriminant analysis (LDA) [60], one of the supervised dimension reduction methods. Unlike other unsupervised methods such as multidimensional scaling (MDS) and principal component analysis (PCA), which only use data, supervised ones also involve additional information such as cluster labels associated in the data. In case of LDA, it maximally discriminates different clusters while keeping the relationship among data within each cluster tight in the reduced dimensional space. This behavior of LDA has two advantages for interactive classification systems. The first advantage is that LDA is able to visualize

the data so that their cluster structure can be well exposed. For example, as seen in Fig. 28, LDA reveals the cluster structure better than PCA, and through LDA, users can easily find the cluster relationship and explore the data based on it. The other advantage is that the reduced dimensional representation of the data by LDA does not require a sophisticated classification algorithm in general since the data is already transformed to a well-clustered form, and such a transformation would map an unseen data item to a nearby area of its true cluster. Thus, after applying LDA, a simple classification algorithm such as k-nearest neighbors [51] can be performed, which has been successfully applied to many areas [16, 123]. Owing to this simplicity, users can get an idea about how the new data would be classified by looking at a nearby region based on visualization through LDA.

Inspired by the above ideas, we have developed a system called iVisClassifier, in which users can visually explore and classify data based on LDA. The first contribution of iVisClassifier lies in its emphasis on interpretation of and interaction with LDA for data understanding. Then, iVisClassifier features the ability to let users cooperate with the LDA visualization for the classification process. To show the usefulness of iVisClassifier, we present facial recognition examples, where LDA-based classification works well.

The rest of this chapter is organized as follows. Section 6.2 discusses previous work related to interactive data mining systems and dimension reduction methods. Section 6.3 briefly introduces LDA and its use of the regularization in visualization, and Section 6.4 describes the details of iVisClassifier. Section 6.5 shows case studies, and Section 6.6 concludes our work.

6.2 Related Work

Supporting data mining tasks with interactive systems is an active area of study. As for clustering, an interactive system for hierarchical clustering was presented in

[117], and a visualization-based clustering framework was proposed in [29], where users can analyze the clustering results and impose their domain knowledge into the next-stage clustering. In addition, various research has been conducted to make the dimension reduction process interactive. Yang et al. [143, 142] proposed a visual hierarchical dimension reduction method, which groups dimensions and visualizes data by using the subset of dimensions obtained from each group. Novel user-defined quality metrics was introduced for effective visualization of high-dimensional data in [73]. A user-driven visualization approach using MDS was proposed in [136].

However, in spite of the increasing demand from real-world applications, supporting classification tasks with an interactive visual system has not been studied extensively. Some studies [8, 9, 126] have tried to make a decision tree more interactive through visualization using circle segments [7] and star coordinates [77]. However, other classification methods have not been deeply integrated into interactive systems.

With respect to dimension reduction methods, a myriad of methods are still being proposed, and some of them claim their advantages on two or three-dimensional visualization. The recently proposed nonlinear manifold learning methods have shown the interesting ability to match the reduced dimensions to some semantic meanings such as the rotation of objects in image data [111, 125]. Another nonlinear method called t-SNE [130] has successfully revealed a hidden cluster structure in the reduced dimensional space for handwritten digit image and facial image data through computationally intensive iterations. While all the above-mentioned methods are unsupervised dimension reduction methods that do not consider cluster label information, supervised dimension reduction methods [60, 71], which explicitly utilize them in their computations, typically attempt to preserve the cluster structures by grouping the data with given labels.

Even with such technical advances, people still prefer traditional methods such as PCA, MDS, and self-organizing maps (SOM) because the state-of-the-art methods

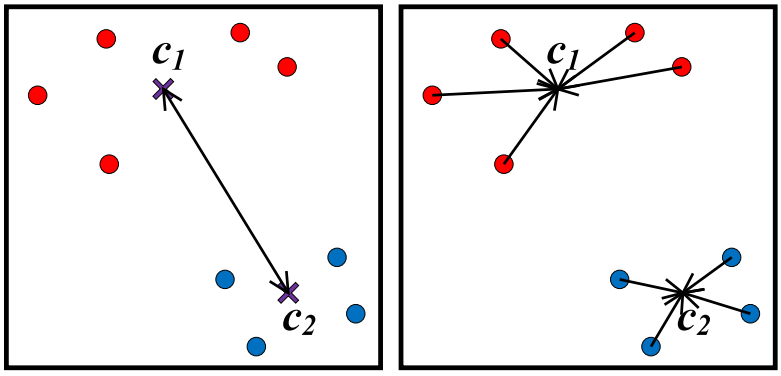
tend not to work universally for various types of data and they often lack interpretability. Motivated by this, a recently proposed system called iPCA [72] enables users to interact with PCA and its visualization results in the form of scatter plots and parallel coordinates. Our system shares a lot in common with iPCA in that users can play with LDA via scatter plots and parallel coordinates. Other than data understanding, our system aims further to support classification tasks utilizing the supervised dimension reduction.

6.3 Linear Discriminant Analysis

In this section, we briefly introduce LDA and skip rigorous mathematical derivations due to a page limit. For more technical details about LDA and its use in visualization, refer to our previous work [35].

6.3.1 Concepts

LDA is a linear dimension reduction method that represents each of the reduced dimensions as a linear combination of the original dimensions. By projecting the data onto such a linear subspace, LDA puts cluster centroids as remote to each other as possible (by maximizing the weighted sum, B, of squared distances between cluster centroids, as shown in Fig. 29(a)), while keeping each cluster as compact as possible (by minimizing the squared sum, W, of the disances between each data item in the cluster and its cluster centroid, as shown in Fig. 29(b)), in the reduced dimensional space. Due to this characteristic, LDA can highlight the cluster relationship as shown in Fig. 28(a), as opposed to other dimension reduction methods such as PCA. In LDA, this simultaneous optimization is formulated as a generalized eigenvalue problem that maximizes B while keeping its minimum value of W. Theoretically, the objective function value of LDA cannot exceed that in the original space, and such an upper bound is achieved as long as at least $k-1$ dimensions are allowed in LDA, where k is the number of clusters. Due to this characteristic, LDA usually reduces the data



(a) Maximization of distances between cluster centroids

(b) Minimization of approximate cluster radii

Figure 29: Conceptual description of LDA. A different color corresponds to a different cluster, and c_1 and c_2 are the cluster centroids.

dimension to $k-1$.

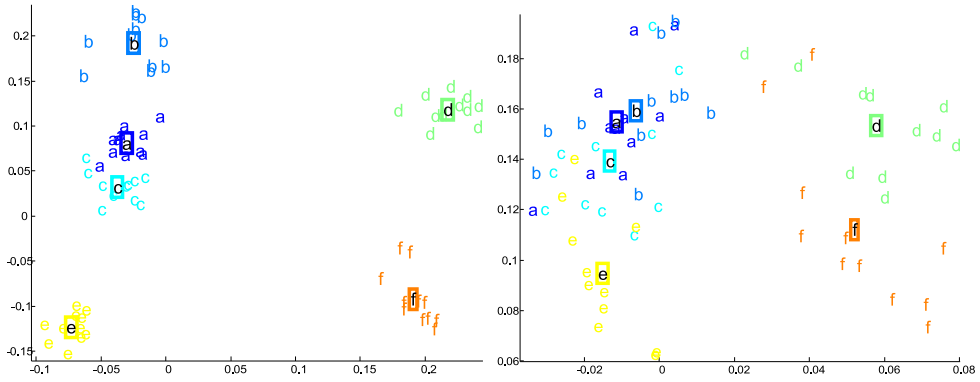
Although LDA can reduce the data dimension down to $k-1$ dimensions without compromising its maximum objective function value, it is often not enough to use for 2D or 3D visualization purposes. In this case, users can either select a few of the most significant dimensions or perform an additional dimension reduction step to further reduce the dimension to two or three [35]. In iVisClassifier, we adopt the former strategy so that we can easily interpret the dimension reduction step while interacting with all the LDA reduced dimensions.

6.3.2 Regularization to Control the Cluster Radius

In regularized LDA, a scalar multiple of an identity matrix γI is added to the within-scatter matrix S_w, the trace of which represents W.[1] It was applied to LDA [58] in order to circumvent a singularity problem when the data matrix has more dimensions than the number of data items, i.e., an undersampled case. In addition, regularization also has an advantage against overfitting in the classification context.

On the other hand, a unified algorithmic framework of LDA using the generalized

[1]Instead of W, the LDA formulation uses S_w, which is then replaced with $S_w + \gamma I$ by regularization. For more details, refer to [35].



(a) $\gamma = 10^2$ (b) $\gamma = 10^5$

Figure 30: Effects of a regularization parameter γ in $S_w + \gamma I$. It can control how scattered each cluster is in the visualization. The data is one of the facial image data called SCface, and we chose the first six persons' images.

singular value decomposition (LDA/GSVD) was proposed [68], which broadens the applicability of LDA regardless of the singularity. For undersampled data, e.g., text and image data, LDA/GSVD can fully minimize the cluster radii, making them all equal to zero. However, making the cluster radii zero results in representing all the data points in each cluster as a single point. Although it makes sense in terms of the LDA criteria, it does not keep any information to visualize at an individual data level. Thus, we utilize regularization to control the radius or scatteredness of clusters in the visualization to either focus on the data relationship or the cluster relationship, as shown in Fig. 30. In an extreme case, when we sufficiently increase the regularization parameter γ, S_w is almost ignored in the minimization term, i.e., $S_w + \gamma I \simeq \gamma I$, so that LDA focuses only on maximizing B without minimizing W. Mathematically, this case is equivalent to applying PCA on the cluster centroids [35].

6.3.3 Algorithms

To ensure real-time interactions, it is important to design an efficient algorithm for LDA. Therefore, we reduce the data matrix size by applying either QR decomposition for undersampled cases or Cholesky decomposition for the other cases before running LDA. The main idea here is to transform a rectangular data matrix of size $m \times n$

into a square matrix of size $\min(m, n) \times \min(m, n)$ without losing any information. Then, the GSVD-based LDA algorithm is performed on this reduced matrix much efficiently. For more details, refer to [103].

6.4 System Description

6.4.1 Data Encoding

Given a data set along with its labels, iVisClassifier first encodes the data into high-dimensional vectors. In its current implementation, iVisClassifier takes text documents, images, and generic numerical vectors with comma-separated values. When dealing with image data, the pixel values in each image are rasterized to form a single column vector, and text data are encoded using the bag-of-words model. Such encoding schemes determine the dimensions of image and text data as the total number of pixels in a single image and the total number of different words, respectively, which can be up to the hundreds of thousands.

Along with numerical encoding, iVisClassifier has several optional pre-processing steps such as data centering and normalization that makes the norm of every vector equal. In addition, other domain-specific pre-processing steps are also provided, such as contrast limited adaptive histogram equalization [107] for image data and stemming and stop-word removal for text data.

6.4.2 Visualization Modules

Once the data matrix whose columns represent data items is obtained, LDA is performed on this matrix with its associated labels. Users can recompute LDA with different regularization parameter values γ through a horizontal slide bar interface until the data within each cluster are adequately scattered. As described in Section 3, LDA reduces the data dimension to $k-1$ where k is the number of clusters. Just as the reduced dimensions in PCA are in an order to preserve the most variance, those in LDA are also in an order of preserving the most value of the LDA criterion. That

is, the first reduced dimension represents each cluster most compactly while keeping different clusters most distinctly. With this in mind, we visualize LDA results in four different ways: parallel coordinates (Fig. 31A), the basis view (Fig. 31B), heat maps (Fig. 31C), and 2D scatter plots (Fig. 31F).

6.4.2.1 Parallel coordinates

Parallel coordinates is a common way to visualize multi-dimensional data. In parallel coordinates, the dimension axes are placed side by side as a set of parallel lines, and the data item is represented as a polyline whose vertices on these axes indicate the values in the corresponding dimensions. The main problem of parallel coordinates is that it does not scale well in terms of both the number of data items and the number of dimensions. However, LDA can deal with both problems effectively in the following ways. First, with a manageable number of clusters, k, LDA reduces the number of dimensions to $k-1$, without losing any information on the cluster structure based on the LDA criterion. In addition, in terms of the number of data items, LDA plays the role of data reduction for undersampled cases since it can represent all the data items within each cluster as a single point by setting $\gamma = 0$, which in turn visualizes the entire data as k items. The dimension-reduced data by LDA may suffer the same scalability problem when the number of clusters and/or the regularization parameter γ increases. Nonetheless, in most cases, LDA significantly alleviates the clutter in parallel coordinates in that dealing with a large number of clusters is not practical and that users can always start their analysis with $\gamma = 0$.

Our implementation of parallel coordinates has several interactions including a basic zoom-in/out function. First, users can control the transparency of the polylines to see how densely the lines go through a particular region. To this end, users can switch all the colors indicating cluster labels to a single one, e.g. black. In addition, iVisClassifier has several shifting and scaling options. One is to align the minimum

value of each dimension at the bottom horizontal line in the view, and the other is to align both the minimum and the maximum values at the top and bottom line, respectively. iVisClassifier is also able to filter the data by selecting particular clusters and/or data points in a certain range specified by a mouse pointer, and brushing and linking is implemented between parallel coordinates and scatter plots.

6.4.2.2 Basis view

When data go through any kind of computational algorithms, it is crucial to have a better understanding of what happens in the process. For instance, even though the dimension reduction result is given by LDA, users may need to know the meaning behind each dimension and the reasons why those dimensions maximize the LDA criterion. Without such information, users cannot readily understand why certain data points look like outliers or certain clusters are prominent in the LDA result. Following this motivation, we provide users with the meaning of each reduced dimension of LDA in the following way.

First of all, LDA is a linear method where each reduced dimension is represented as a linear combination of those in the original space. Thus, we have a linear combination coefficient for each reduced dimension, which we call a basis vector, and the dimension of this basis vector is the same as the original dimension. For image data in which the original dimension is the number of pixels in the image, each coefficient value in this basis vector corresponds to each of the pixels. Based on this idea, we reconstruct the LDA basis in the original data domain, e.g., an image in our case. However, it is not always straightforward to convert the basis back to the original data domain. For example, pixel values in an image have a certain specification that they have to be all integers between 0 and 255 while the LDA basis is real-valued with positive and negative signs mixed. In the past, several heuristics to handle this issue were used in the context of PCA by mapping basis vectors to grayscale images [129, 131] by taking

either its absolute value or adding the minimum value. However, these heuristic methods lose or distort the information contained in the basis vectors. Therefore, we map positive and negative numbers in the basis vector into two color channels, red and blue, respectively. In this way, we obtain the reconstructed images of LDA basis vectors as shown in Fig. 31B.

6.4.2.3 Heat maps

With heat maps, we visualize the pairwise distances between cluster centroids, where each heat map has $k \times k$ elements. The leftmost heat map in Fig. 31C represents such information in the original high-dimensional space, and the following ones on the right side are computed within each reduced dimension of LDA. Through this visualization, we can get the information about which particular cluster is distinct from the other clusters and which cluster pairs are close or remote in each dimension. Furthermore, comparisons between heat maps of the original space and each of the reduced dimension show which cluster distances are preserved or ignored.

By clicking the (i, j)-th square in the enlarged heat map (Fig. 31D), users can compare the data items in the i-th and j-th clusters as shown in Fig. 31E. In addition, the slide bar at the bottom in Fig. 31E enables users to overlap the data image with its corresponding basis image, which tells us how the pixels in these images are weighted in its corresponding dimension and why the data of the selected two clusters are closely or remotely related in this dimension, as shown in Fig. 36.

6.4.2.4 Scatter plots

The scatter plot visualizes data points in the two user-selected reduced dimensions of LDA with a zoom-in/out functionality. In this view, a data item is represented as a point with an initial letter and a different color of its corresponding cluster label. Additionally, the first and the second order statistics per cluster, which are the mean and the covariance ellipse, give the effective information about clusters.

Our scatter plot view given by LDA allows users to interactively explore the data in view of the overall cluster structure in the following senses: 1. which data points are outliers or representative points in their corresponding clusters, 2. which data points are outliers or representative points in their corresponding clusters, 3. how widely the data points within a cluster are distributed and accordingly, which clusters have potential subclusters, and 4. which data points overlap between different clusters.

In addition, brushing and linking with parallel coordinates overcome the limitation that the scatter plot can only show two or three dimensions at a time. In this way, users can see how the selected data or clusters in the scatter plot behave in the other dimensions.

6.4.3 Classification Modules

After obtaining insight from exploring the data with known cluster labels, users can now interactively perform classification on the new data whose labels are to be determined. This process works as follows. First, a new data item is mapped onto the reduced dimensional space formed by the previous data. It is then visualized in parallel coordinates and in the scatter plot view. Such visualization significantly increases the efficiency of users' classification tasks by visually reducing the search space. Within this reduced visual search space, users can easily compare the new data item with the existing data or clusters nearby. When the new data point falls into a cluttered region where many different clusters overlap, users can select or filter out some data or clusters and recompute LDA with this subset of data including the new point, which we call a computational zoom-in process. In other words, LDA takes into account the selected clusters and/or those corresponding to the selected data, which requires a much smaller number of dimensions than $k-1$ for LDA to fully discriminate the selected clusters. Based on the new visualization generated in this way, users can better identify which clusters the new point belongs to.

On completing the visually-supported classification process, users can assign a label to the new data item and optionally include the newly labeled data in future LDA computations, which is initiated only when users want to recompute them. The reason we do not force users to include every new data in LDA computations is that users' confidence level of the assigned label may not be high enough for some reason such as noise.

6.5 Case Studies

In this section, we present an interactive analysis using two sets of facial image data, Weizmann database[2] and SCface database [64], for facial recognition. Weizmann is composed of 28 persons' frontal images in a constant background, in which each person has 52 images. The variations within each person's images exist regarding viewing angles, illuminations, and facial expressions. We resized the original 512×352 pixel images to 64×44 pixel images, resulting in 2816 dimensional vectors. SCface is an image collection taken in an uncontrolled indoor environment using multiple video surveillance cameras with various image qualities. It is composed of 4160 static images of 130 subjects, of which we randomly selected 30 persons' images for our study, where each person has 32 images. Since the images in SCface generally contain parts other than a face, such as the upper body of a person and a different background, we have cropped a facial part using an affine transformation that aligns the images based on the eye coordinates. The image samples of two data sets are shown in Fig. 32.

In the following, we present an exploratory analysis towards better understanding of both the data and the computational method we have used, i.e., LDA. Next, we describe how users interactively perform classification supported by iVisClassifier.

[2]http://www.wisdom.weizmann.ac.il/~vision/databases.html

6.5.1 Exploratory Data Analysis

In general, understanding the data at the cluster level is essential to deriving an initial idea about the overall structure in a large-scale data set. In this sense, we can begin with the heat map view of the pairwise distances in the original space to look at how the clusters are related. From the heat maps shown in Fig. 33(a) and 34(a), we can see that pairwise cluster distances vary more in Weizmann than in SCface. This view also reveals the clusters that look distinct from the other clusters, e.g., person 14 in Weizmann and person 7 in SCface. Element-wise comparisons reveal that persons 11 and 14 look quite distinct, which makes sense due to baldness and shirt colors, but persons 2 and 10 look similar in Fig. 33(a). Similarly, persons 1 and 7 look different while persons 2 and 26 are indistinguishable in Fig. 34(a).

Next, let us look at the heat maps of the LDA dimensions shown in Figs. 33-34. The first dimension turns out to reflect the most distinct clusters in the original space. In addition, the heat maps in the LDA dimensions have mostly blue-colored elements, i.e., almost zero, except for a few rows and columns, which indicates that each of the LDA dimensions tends to discriminate only a few clusters.

Next, Fig. 35 shows the image reconstruction of the first six LDA bases for both data sets. It is interesting to see that in both cases, the forehead part is heavily weighted in the first dimension,[3] and then in the second dimension, the forehead part is differentiated into upper and lower parts. This indicates that the forehead part is the most prominent factor for facial recognition based on LDA in our data.

Basis images can be overlapped with the original images to highlight the region in the images that is heavily weighted in a specific reduced dimension. The example shown in Fig. 36 was obtained by selecting one of the most remote cluster pairs (red-colored one in Fig. 33(b)) in the first dimension. In the region covered by a blue color,

[3]Negative weighting coefficients represented as blue colors are equivalent to positive ones by negating the basis and the corresponding coordinate values of the data.

we can see that the pixel values are quite different, i.e., light in the first cluster and dark in the second cluster, which puts them far apart in the corresponding reduced dimension.

6.5.2 Interactive Classification

As described in Section 4.3, the main benefit of iVisClassifier for classification is that it visually guides users to the correct clusters for unseen data while allowing users to have control over the classification process. In general, most of the new data would be closely placed to their corresponding clusters in the scatter plot. If only a few clusters are found nearby, e.g., when a point to classify is placed near the cluster 7, which is almost isolated from the other clusters at the leftmost part in Fig. 37(a), then by checking some of the nearby data in the cluster 7, users can quickly classify them into their corresponding clusters. However, a problem arises when the new point is visualized near a cluttered region as shown in Fig. 37(a). With this visualization, we have a less clear idea as to which clusters to look at because numerous clusters exist near the point of interest. In this case, we can select a subset of data points around it and then recompute the dimension reduction only with this subset. Fig. 37 shows that this process guides the new point to its true cluster.

Another scenario for interactive classification in iVisClassifier is cooperative filtering between parallel coordinates and the scatter plot. Fig. 38(a) shows a case where the new point is placed in an ambiguous region to classify. As we find that the new point (shown in a gray color in parallel coordinates) goes through the top region in dimension 7, we can filter the data in this dimension, and accordingly, the selected data are also highlighted in the scatter plot with a black circle, as shown in Fig. 38(b). Additional filtering in the scatter plot by selecting either nearby clusters or data items ends up with only one possible cluster, as shown in Fig. 38(c).

Once some of the new data are assigned their labels, users can recompute LDA

by taking into account the newly labeled data. Fig. 39 shows the distributions of the new data whose label is '0' before and after LDA recomputation with a newly labeled data item. As we can see, the rest of the unseen data in cluster 0 becomes closer to its centroid after LDA recomputation, which indicates that the updated LDA dimensions potentially better discriminates the unseen data.

6.6 Conclusions

In this study, we have presented iVisClassifier, a visual analytics system for clustered data and classification. Our system enables users to explore high-dimensional data through LDA, which is a supervised dimension reduction method. We interpret the effect of regularization in visualization and provide an effective user-interface in which users can control the cluster radii depending on whether they focus on the cluster- or the data-level relationships. In addition, iVisClassifier facilitates the interpretability of the computational model applied to their data. Various views such as parallel coordinates, the scatter plot, and heat maps interactively show rich aspects of the data. Finally, we showed that iVisClassifier can efficiently support a user-driven classification process by reducing humans' search space, e.g., recomputing LDA with a user-selected subset of data and mutual filtering in parallel coordinates and the scatter plot.

As our future work, we plan to improve our system to better handle other types of high-dimensional data and their classification tasks. Although our system can currently load and visualize other types of high-dimensional data such as text data, how we accommodate the basis view and blend the data item with the basis in the original data domain, as shown in Fig. 36, would be the main issues.

In addition, although our tool works well when there is a reasonable number of clusters, it may not scale well when we have many clusters, e.g., hundreds of people in facial recognition. To handle this problem, we are considering the hierarchical

approaches that group the clusters based on their relative similarities to keep the number of clusters manageable in an initial analysis.

Finally, the computation of LDA can be burdensome for user interactions when we have a large-scale data. Novel interactions with LDA provided by iVisClassifier motivate the new types of dynamic updating algorithms based on the previous LDA results in various situations. For instance, updating the LDA results when changing the regularization parameter value has not been studied before. Thus, we are currently exploring for various situations and their corresponding updating algorithms when computational algorithms are integrated into user-interactive systems.

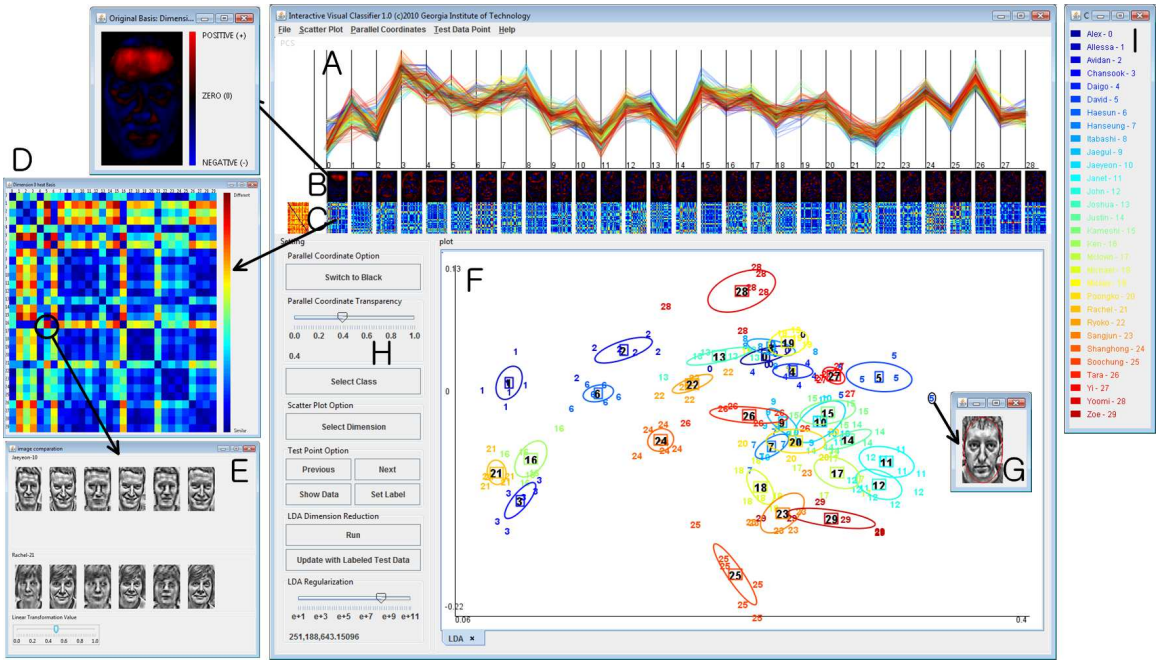
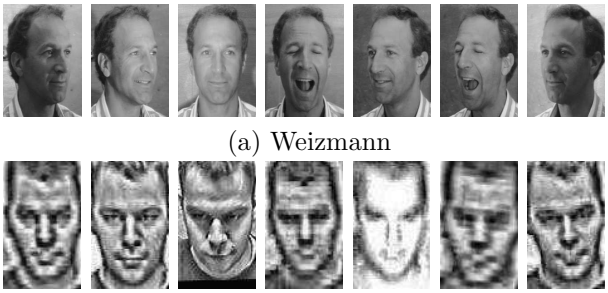


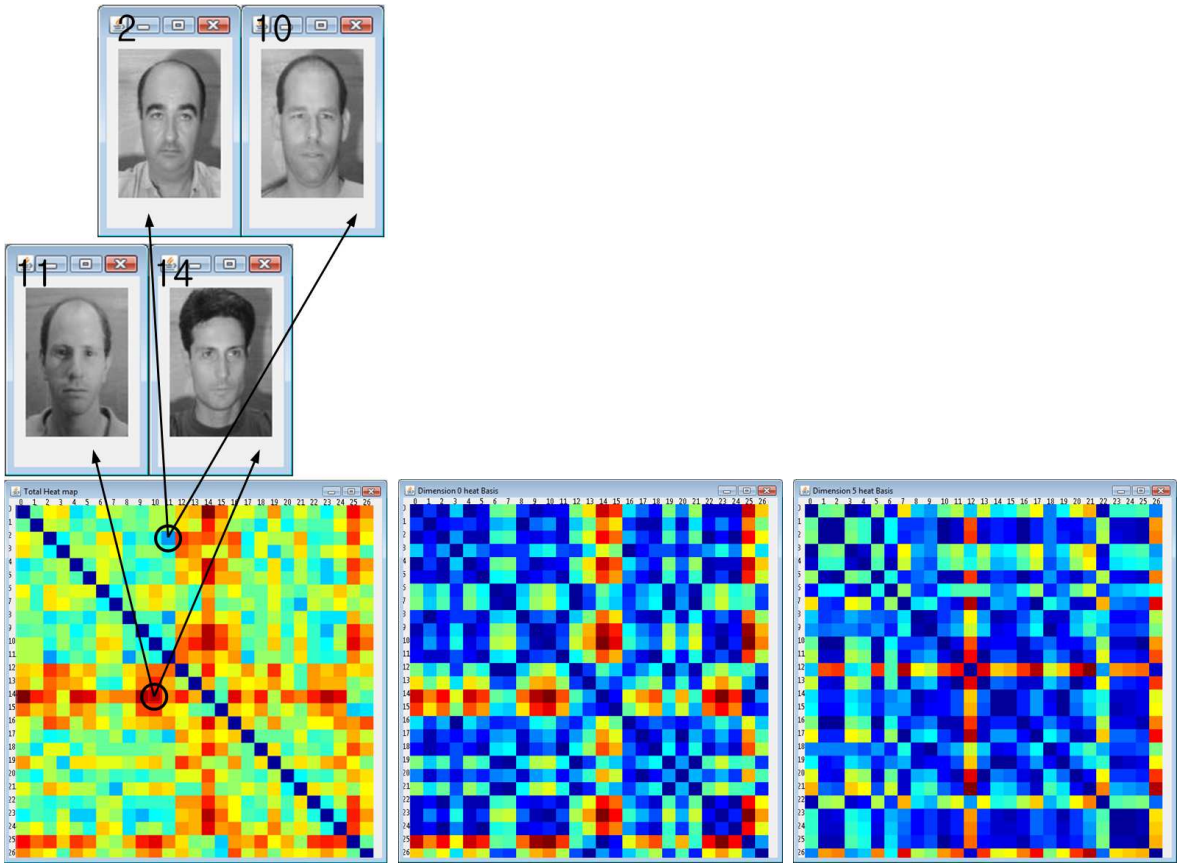
Figure 31: The overview of the system. SCface data with 30 randomly chosen persons' images were used, and different colors correspond to different clusters, e.g., persons. The arrow indicates a clicking operation. (A) Parallel coordinates view. The LDA results are represented in 29 dimensions. (B) Basis view. The LDA basis vectors are reconstructed in the original data domain, which in this case is an image. (C) Heat map view. The pairwise distances between cluster centroids are visualized. The leftmost one is computed from the original space, and the rest from each of the LDA dimensions. Upon clicking, the full-size of a heat map is shown (D), and clicking each square shows the existing data in the corresponding pair of clusters (E). (F) Scatter plot view. A 2D scatter plot is visualized using two user-selected dimensions. When clicking a particular data point, its original data item is shown (G). (H) Control interfaces. Users can change the transparency and the colors in parallel coordinates. Data can be filtered at the data level as well as at the cluster level. The interfaces for unseen data visualize them one by one, interactively classify them, and finally update the LDA model. A horizontal slide bar for the regularization parameter value in LDA controls the scattering of each cluster. (I) shows the legend about cluster labels in terms of their assigned colors and enumerations.



(a) Weizmann

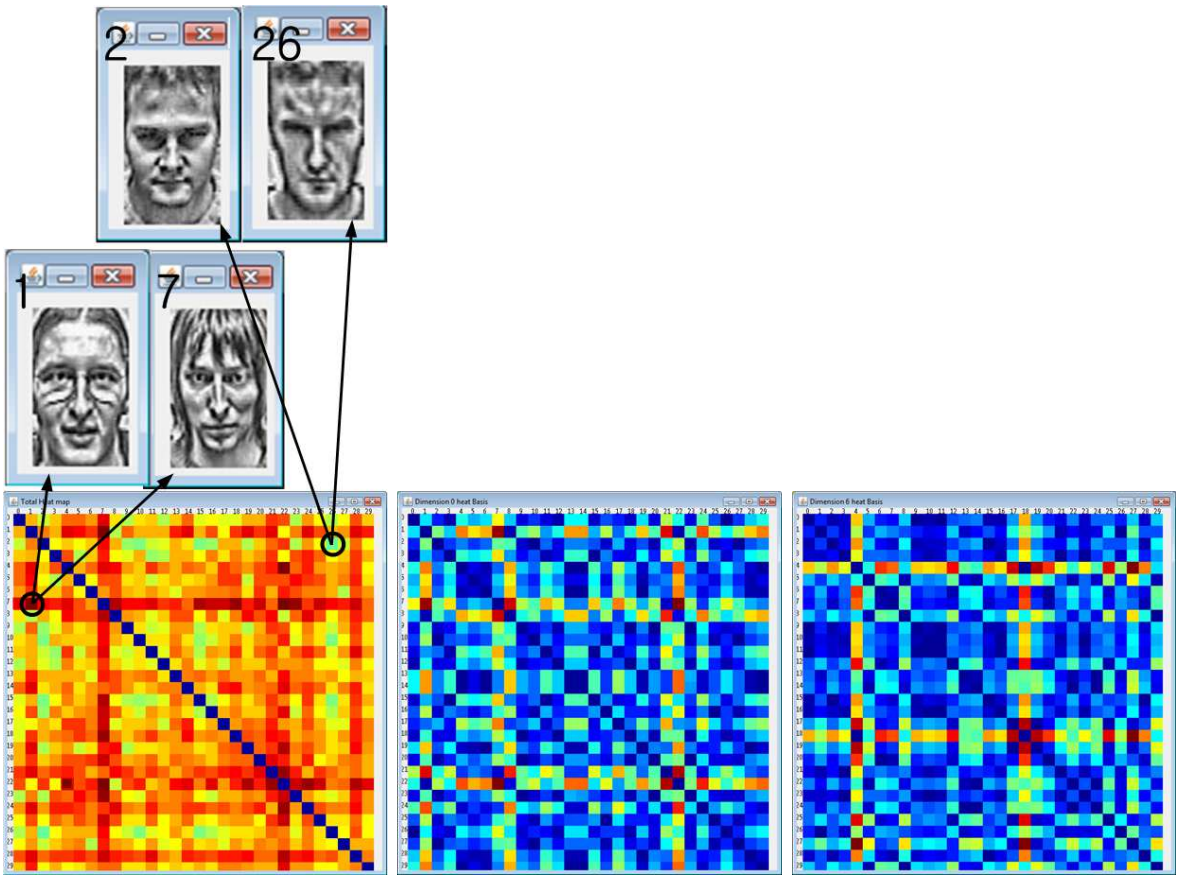
(b) SCface

Figure 32: A single person's image samples in two data sets.



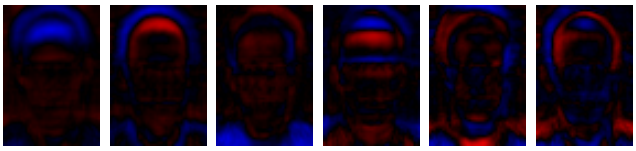
(a) The original space (b) The first dimension (c) The fifth dimension

Figure 33: Heat map view of the pairwise cluster distances of the Weizmann data set.

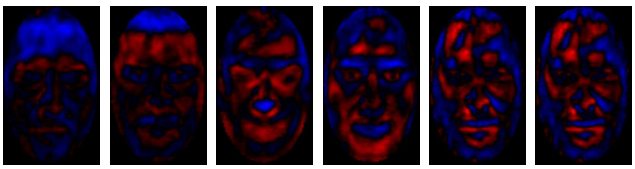


(a) The original space (b) The first dimension (c) The sixth dimension

Figure 34: Heat map view of the pairwise cluster distances of the SCface data set.



(a) Weizmann



(b) SCface

Figure 35: Reconstructed images of the first six LDA bases.

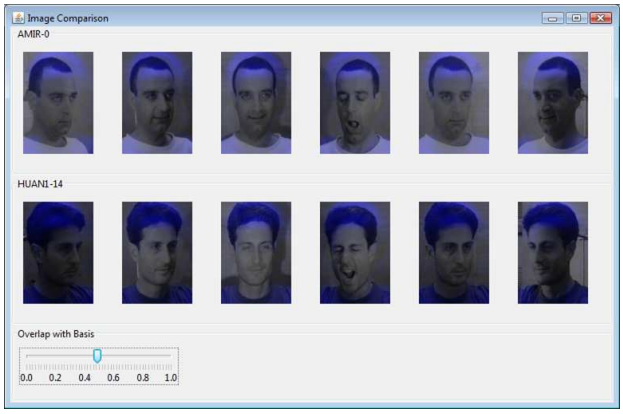
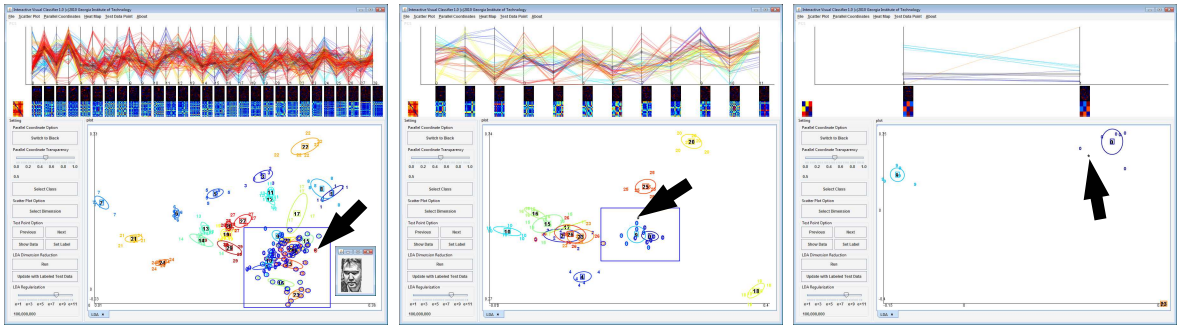
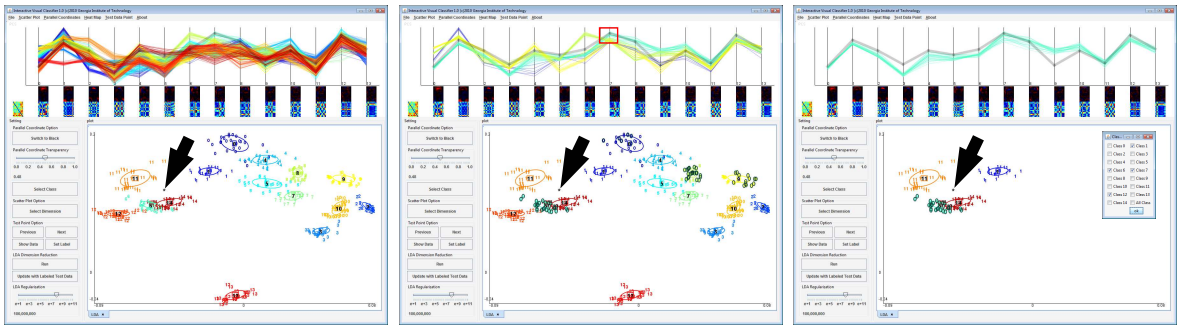


Figure 36: The effect of overlapping a basis image over the original data. Users can see which part of images are weighted by a basis vector.



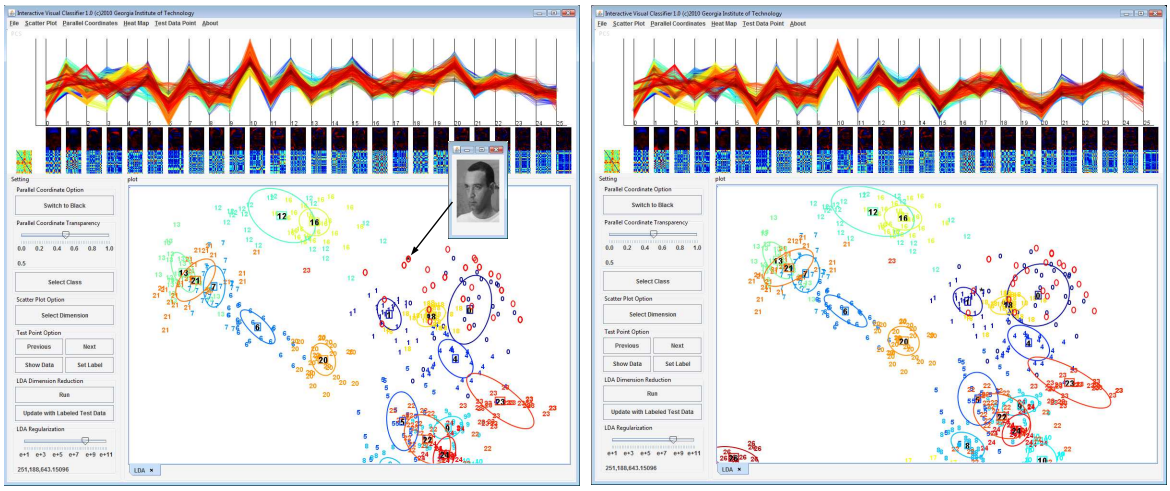
(a) The initial filtering (b) The second filtering (c) The final visualization result

Figure 37: Interactive classification by computational zoom-in. Recursive visualization by recomputing LDA for interactively selected subsets of data guides a new point into its corresponding cluster. The thick arrow indicates the new point position.



(a) The initial visualization (b) The filtering in parallel coordinates (c) The filtering in the scatter plot

Figure 38: Interactive classification by mutual filtering. Filtering both in parallel coordinates and the scatter plot leads to a single cluster. The thick arrow indicates the new point position.



(a) Before labelling the test point

(b) After labelling the new point

Figure 39: Effects of LDA recomputation when including a newly labeled point in the existing data. The arrow indicates the newly labeled point, and the red circles represent the distribution of the remaining unseen data in cluster 0.

CHAPTER VII

VISIRR: AN INTERACTIVE VISUAL INFORMATION RETRIEVAL AND RECOMMENDER SYSTEM FOR LARGE-SCALE DOCUMENT DATA

We present a visual analytics system called VisIRR, which is an interactive visual information retrieval and recommendation system for document discovery. VisIRR effectively combines both the paradigms of *passive pull* through a query processes for retrieval and *active push* that recommends the items of potential interest based on the user preferences. Equipped with efficient dynamic query interfaces for a large corpus of document data, VisIRR visualizes the retrieved documents in a scatter plot form with their overall topic clusters. At the same time, based on interactive personalized preference feedback on documents, VisIRR provides recommended documents reaching out to the entire corpus beyond the retrieved sets. Such recommended documents are represented in the same scatter space of the retrieved documents so that users can perform integrated analyses of both retrieved and recommended documents seamlessly. We describe the state-of-the-art computational methods that make these integrated and informative representations as well as real time interaction possible. We illustrate the way the system works by using detailed usage scenarios. In addition, we present a preliminary user study that evaluates the effectiveness of the system.

7.1 Introduction

These days, researchers are faced with a deluge of new papers appearing each day, any of which might potentially contain a new development which could be critical to one of the questions he or she is investigating. The challenge is similar to that of

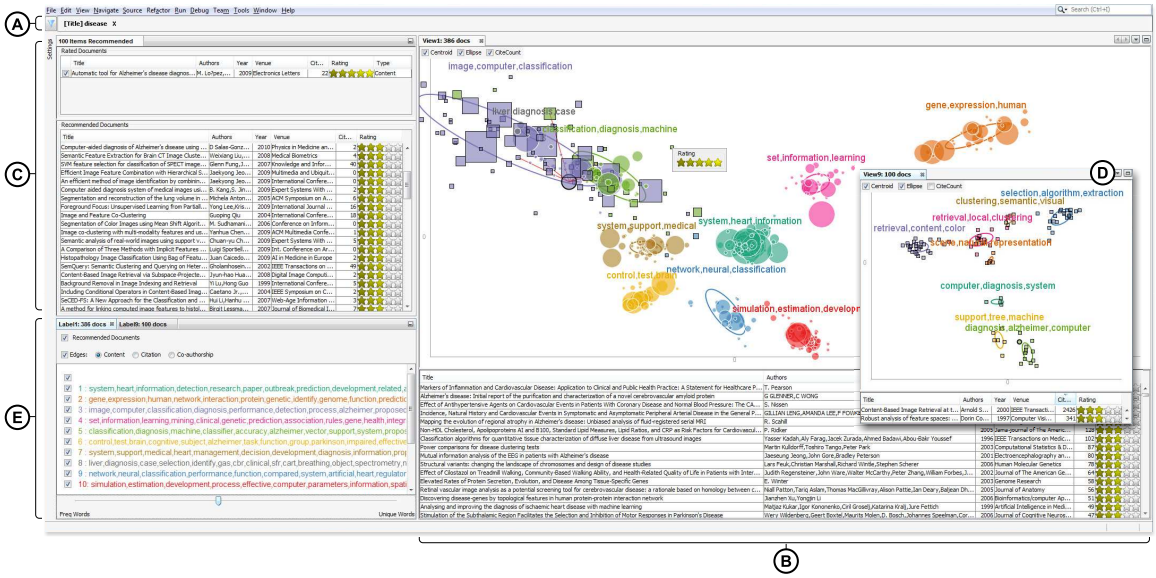


Figure 40: An overview of the VisIRR system. Given about half a million academic papers in the system, the user can start by issuing a query (A), which in this case is a keyword 'disease'. By performing clustering and dimension reduction, VisIRR visualizes the retrieved documents in a scatter plot and a table view (B) along with a topic cluster summary (B)(E). In the scatter plot view, a circular node represents a query-retrieved item, and a rectangular one denotes a recommended item. Their node size encodes the number of citations. After identifying a few documents of interest, the user can assign them his/her preference in a 5-star rating scale both in a scatter plot and in a table view. Based on this preference feedback, the system now provides a list of recommended items in another table view (C), and furthermore they are projected back to the existing scatter plot view (B) so that the consistent topical perspective can be maintained. To better understand the recommended items, the user can apply 'computational zoom-in' on this set, which gives a clearer scatter plot with a more semantically meaningful summary (D). Finally, the system provides the option to choose different recommendation schemes based on contents, a citation network, and a co-authorship network.

finding an available needle in a haystack each day, with limited attention and time resources.

This problem regime is highly under-explored, compared to the billions that have been invested in the related paradigm of web search. Instead, the researcher or analyst is solving a subtle investigative problem for which each of several documents provides clues. By seeing this as an information retrieval (IR) problem, the focus in this chapter is on the long tail, or **recall** (making sure that few relevant documents are missed), while in web search the focus is generally on the quicker gratification of **precision** (making sure the first page of hits or so contain very relevant documents).

In general, search is a form of "**pull**" technology, in which the user takes actions by forming and issuing queries. However, in the former case where a high recall is concerned, what queries to issue, e.g., proper keywords, becomes crucial in order for users to obtain the documents of their interest. As a way to compensate this issue, a **recommendation**, or a "**push**" technology, with which the system finds things of interest to suggest to the individual user, has recently been popular in various domains. Whereas a search engine is more or less stateless and the same for all users, a recommendation system involves personalization, remembering aspects of the state of the user's interests and investigations so far.

In the context of visual analytics, document analyses have long been one of the main areas studied. Visual analytics systems for document data, such as IN-SPIRE [138] and JIGSAW [121], can help to give an overall understanding about a set of documents as well as revealing their intra-set relationships that would have been difficult and time-consuming without the help of interactive visualization. However, despite the fact that personalized recommendations seem to be a natural fit with interactive visualization in that it directly utilizes the history of user interactions, there are few instances of such work in the visual analytic community.

As one of the milestones to fill this gap, we present a novel document visual analytics system called VisIRR, an interactive "Vis"ual "I"nformation "R"etrieval and "R"ecommendation for document data, which effectively combines traditional query-based information retrieval and personalized recommendation. Basically, as seen in Fig. 40, VisIRR adopts a scatter plot as a main visualization form similar to IN-SPIRE. In other words, the documents to be visualized are first clustered into several groups via a clustering algorithm and then projected to a 2D space via a dimension reduction algorithm. However, VisIRR features various novel aspects compared to existing systems, as follows.

- *Efficient large scale data processing*: VisIRR currently handles about half a million documents and scales linearly with respect to newly added documents in terms of the amount of the required computation and memory size.

- *Advanced clustering and dimension reduction techniques*: As core computational modules, VisIRR adopts state-of-the-art techniques such as nonnegative matrix factorization (NMF) for clustering and linear discriminant analysis (LDA) for dimension reduction. These techniques give the results with a much better quality as well as with a faster computational time than traditional methods including k-means, principal component analysis (PCA), and multidimensional scaling. Additionally, VisIRR provides an alignment capability for both clustering and dimension reduction to facilitate easy comparisons between different visualization snapshots.

- *Preference-based personalized recommendation*: In addition to exploratory analysis of query-retrieved results, VisIRR supports recommendation of potentially interesting documents to users based on the preferences users assign to documents. This recommendation enables users to discover those documents users' query processes cannot reveal easily. The back-end recommendation module,

which is based on PageRank-style graph diffusion algorithm [98], performs efficiently with large-scale data.

To integrate all these capabilities into a mature visual analytics system, we incorporate various building blocks for front-end GUI's and back-end computational algorithms. This chapter mainly presents these building blocks in more detail with detailed usage scenarios. The rest of this chapter is organized as follows. Section 7.2 discusses related work. Section 7.3 explains the front-end GUI modules and comprehensive usage scenarios that highlight the key capabilities of the system. Afterwards, Section 7.4 mainly discusses how we efficiently handle all the necessary information from a large-scale data corpus with a scalable expansion, and Section 7.5 describes computational methods used in the back-end of the system. Section 7.6 briefly presents the user study we conducted to evaluate the system. Finally, Section 7.7 concludes the chapter and discusses about the future work.

7.2 Related Work

Information seeking behavior is a complex human activity, and one that varies dramatically with system capabilities and user's model of those capabilities [93]. Ill-defined document search tasks such as literature searches are often termed 'exploratory search' tasks, in contrast with well-defined tasks such as finding a known, specific item from among a set. In the past, traditional information retrieval has focused much more on the latter than the former.

In the context of exploratory interfaces, information foraging [106] and scent theory [105] suggest making clusters of related data clear and facilitating the process of finding new clusters of interest. To that end, many search result visualization systems also work in concert with automated clustering algorithms, especially when the information space is extremely large or unstructured. The Pacific Northwest National Lab's SPIRE system (and IN-SPIRE follow-on) uses clustering to extract common

themes, and includes several visualization components [138]. Its Themescape component is an abstract 3D landscape depiction of a document space, with arrangements of hills and valleys representing the relatively strength of various themes in the document corpus and how those themes interrelate. Other systems have used this general clusters-in-landscapes (both 2D and 3D) as well [116, 21, 6]. iVisClustering [88] is an interactive document clustering system focused on the user interactions to improve cluster quality based on an advanced technique called latent Dirichlet allocation [20]. On the other hand, rather than providing user interactions customized to a particular clustering technique, the Testbed system [32] offers a wide variety of clustering algorithms and easy comparisons between them via an alignment process VisIRR has adopted.

Using visualization for exploring text data is an active research area within and among many fields. Here, we highlight only a sample of relevant work from different areas and refer the reader to a recent survey of visual text analytics [4] for a more comprehensive treatment.

Unsurprisingly, visualization of document collections has been explored for some time in library science. A relatively early example is the Envision digital library, which includes a visualization system that places documents in a 2D grid according to user-selectable attributes [97]. Systems have used various information visualization techniques such as hyperbolic trees [76, 119] and treemaps [62, 41] to visualize results. Curated collections such as those found in digital libraries more often have pre-formed hierarchies to leverage in visual analytics applications, but simple clustering methods have been implemented as well [119].

When document categories and groupings are not already extant, automated methods of clustering and classifying collections are key to exploratory tools, including those supporting visual analysis. A recent survey [4, 63] distinguishes between the visualization of a single document (e.g., tag clouds) and a document collection and

between time- (e.g., TIARA [134]) and network-oriented collection systems. Because VisIRR's clustering system implicitly creates relationships among members (and its graph diffusion-based recommendation system explicitly uses such data), examples of the last category are most relevant. Jigsaw [121] visualizes network relationships between documents and various entities, e.g., actors, events, etc., automatically extracted from them.

A recently proposed Apolo system [27] uses a mixed-initiative approach that bootstraps initial user-specified categories and classifications into more comprehensive system-suggested categorization of new documents. However, Apolo is exemplar-based method where the user is assumed to clearly have a few of documents of their interest. In this sense, Apolo mainly supports a bottom-up style of analyses. On the contrary, VisIRR initially takes a top-down approach in that it initially starts from an overview visualization of a potentially fairly large amount of documents retrieved by user queries. Once the documents of the user's interest is identified, however, VisIRR also supports a bottom-up style approach via recommendation processes based on the user preferences on particular documents, thereby gradually expanding the user's scope beyond the query-retrieved set.

There has been significant commercial and academic interest in the topic of exploratory search for scientific literature for some time. Several commercial tools are targeted to this problem, with a variety of automated and visual features. Google Scholar [1] automatically extracts research works and their citation networks, but has few visual or recommendation features. The Microsoft Academic Search system from Microsoft Research [2] is a similar offering that also includes more advanced network-style visualization of authorship connections as well as various ways of examining topical, institutional, and venue trends and rankings.

Direct introspection of the academic research process has been a common topic in academia as well. One variation is automated recommender/matching systems,

often applied to the problem of matching individual papers from a corpus to individuals from a slate of candidate reviewers [14, 132]. More relevant to VisIRR are those systems that are more exploratory or analytical in nature. The Action Science Explorer (ASE) [52] focuses on co-citation network visualization, with document clusters created manually or by heuristics [95]. It also includes full-text citation context features not available on VisIRR. The FacetAtlas system [24] automatically clusters document collections using a Kernel density estimation algorithm and provides multi-faceted links between document nodes (rather than just keyword or author searches as in VisIRR). CiteSpace II [28] is a visual tool for identifying new or old research trends in a given set of documents (assumed to be a relatively coherent set produced by a keyword query on a large corpus).

However, none of these systems include one of VisIRR's key contributions: *a user-driven recommendation system that explicitly includes relevant documents from the larger search space vs. a dramatically reduced one from an initial search query.*

7.3 VisIRR Design and Function

In this section, we briefly introduce the user interfaces of VisIRR and describe example analysis scenarios to demonstrate how VisIRR works in detail.[1]

7.3.1 User Interface

The user interface of VisIRR is mainly composed of four parts. The *Query Bar* at the top (Fig. 40(A)) enables users to issue queries dynamically using various fields such as a keyword, an author name, a publication year, and a citation count. The *Scatter Plot view* (with document details shown in the lower table) (Fig. 40(B)) visualizes the retrieved documents (as well as any recommended documents) with their cluster summary labels. The color and the size of each node in a scatter plot represent the

[1] A high-quality video introducing VisIRR is available at `http://www.cc.gatech.edu/~joyfull/vast13/visirr/visirr_final.html`

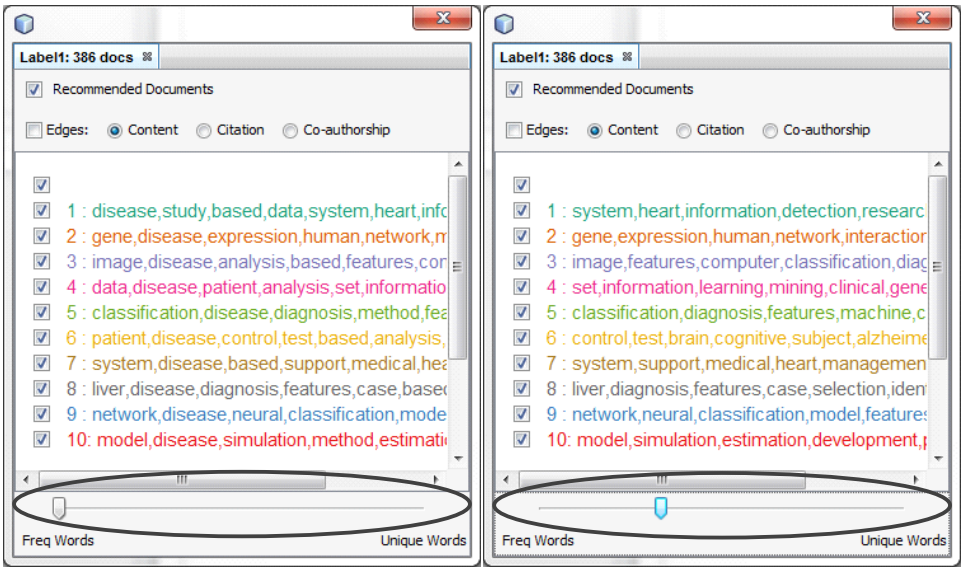
cluster it belongs to and its citation count, respectively. Such a view can also be generated from any user-selected subset of data (Fig. 40(D)). The *Recommendation view* on the top left (Fig. 40(C)) provides tabular representations of the documents whose ratings have been assigned by users (Fig. 40(C) upper table) as well as the resulting recommended documents (Fig. 40(C) lower table). These recommended documents are also visualized in the *Scatter Plot view* as rectangles while the query-retrieved documents are shown as circles. Finally, the *Label panel* provides additional controls such as highlighting and/or hiding particular clusters, changing how cluster summary labels are chosen, and showing direct edge relationships from rated documents to their system-derived recommended documents (Fig. 40(E)).

7.3.2 Usage Scenarios

VisIRR has been implemented using a modified version of the ArnetMiner dataset, which contains approximately 430,000 academic research articles from a variety of disciplines and venues (primarily conferences, journals and books), as will be described in detail in Section 7.4. The following scenarios illustrate the utility of VisIRR for tasks related to this dataset.

7.3.2.1 A Visual Overview of Query-Retrieved Documents

The user starts by issuing queries from the *Query Toolbar*. Suppose the user issues a query of keyword "disease" from a title field. Once documents are retrieved due to this query, the clustering and dimension reduction steps are performed to generate the *Scatter plot view* (Fig. 40(A)). Since most clusters contain the keyword "disease", the user can adjust a slider in the *Label panel* in order to obtain more distinctive cluster summaries, as shown in Fig. 41. From the *Scatter plot view*, the user can drill down to a cluster of interest, e.g., the clusters about gene expression data (the top right), and image analysis (the top left). By moving a mouse pointer to a data point, the user can check the document details via a tooltip text and also skim through



(a) Default cluster summary (b) Distinct cluster summary

Figure 41: A Comparison between default and distinct cluster summaries. Since all the documents include the query word "disease", most clusters contain this word as one of the most frequent keywords (a). By adjusting the slider of *common-vs-unique words* in the *Label panel*, the cluster summary shows much clearer meanings (b).

the document list in the lower table, which is by default sorted by the number of citations. The user can also pan and zoom to enlarge a particular cluster or area of interest.

7.3.2.2 Drilling Down via Computational Zoom-in

Now, the user can drill down a particular cluster via an interaction we call *computational zoom-in*. The computational zoom-in enables the user to select an arbitrary subset of documents by visualizing them as a separate view with their own clustering and dimension reduction results. These subsets can be, for example, particular clusters when their semantic meanings are not clear involving multiple topics. On the other hand, the user can select a cluttered region where many points are mixed together.

Fig. 42 shows an example of the computational zoom-in interaction. After performing computational zoom-in on a highly cluttered area in an original view (black

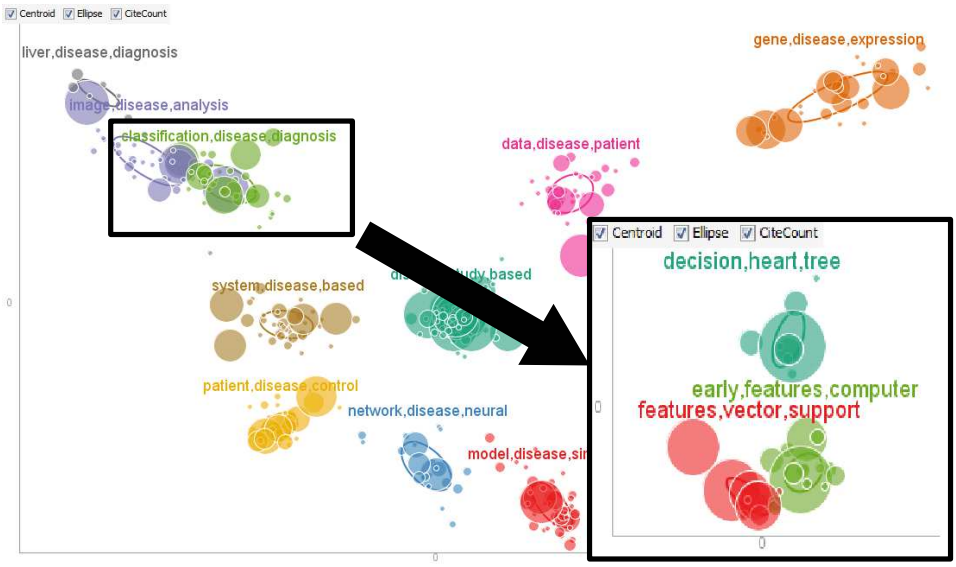
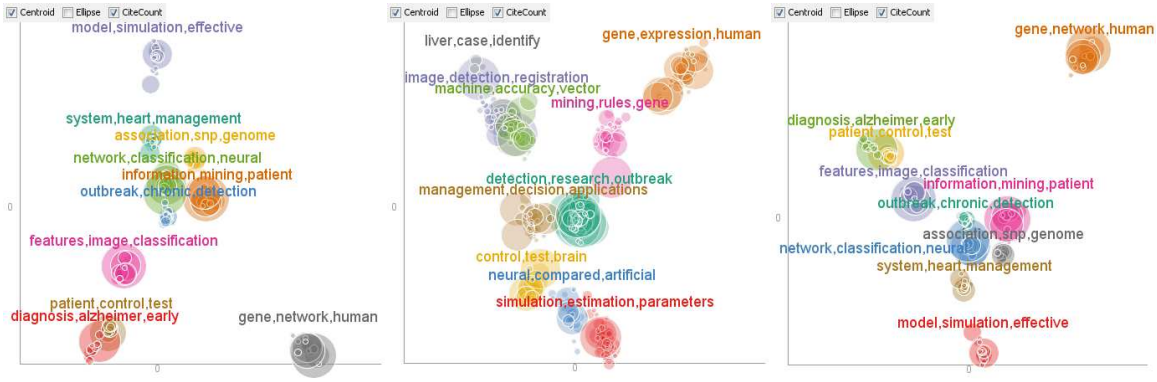


Figure 42: An example of computational zoom-in interaction. For a user-selected region (black rectangle on the top left), this interaction provides a separate view by involving only these points to compute their own cluster summary and dimension reduction coordinates. The resulting view now shows a clear overview about these cluttered data, revealing detailed clusters about 'support vector machines' and 'decision trees' that are typically applied in medical image analyses (black rectangle on the bottom right).

rectangle on the top left), the resulting view successfully reveals several clear clusters e.g., the one about 'support vector machines' and another about 'decision trees' typically applied in medical image analyses (black rectangle on the bottom right).

7.3.2.3 Dynamic Queries and Multi-view Alignment

In addition to exploring visualized clusters, the user can apply additional queries to further narrow down the retrieved document set. Suppose the user wanted to focus on those recently published in 2008 or later and thus created another filter from the *Query Toolbar* in conjunction with the previous keyword query "disease." Given such a new set of documents, VisIRR creates another visualization with its own clustering and dimension reduction. The user could then compare between the new and the previous visualization results, as shown in Figs. 43(a) and (b), respectively, by brushing-and-linking in order to identify, for example, which topic clusters were more/less popular



(a) An unaligned view (b) A reference view (c) An aligned view

Figure 43: Effects of clustering and dimension reduction alignments. A reference view (b) shows the documents with a query word "disease" while the other two views (a)(c) contain the subset of them published from year 2008 with their own clustering and dimension reduction steps applied. For an unaligned view (a), it is difficult to compare against the reference view since there is no correspondence in terms of the coordinates of data points and clusters. However, in an aligned view (c), the clusters match those in the reference, and their spatial correspondences in the scatter plot are maintained.

from 2008. However, since the cluster colors and the dimension reduction results have been computed independently, it is not straightforward to easily compare these differences based on the visualization results.

To solve this problem, once a new visualization is created, VisIRR performs an alignment step on the new clustering and dimension reduction results with respect to the previous visualization result so that the visual coherences in terms of the cluster colors and the spatial coordinates of data points can be maintained. The algorithm details are discussed in Section 7.5.3. For instance, as opposed to an unaligned visualization in Fig 43(a), an aligned one in Fig 43(c) is shown to be much easier to compare against the previous visualization shown in (Fig 43(b). From the aligned visualization, the user can easily see that the cluster about *outbreak detection*, shown as a green cluster in the middle of Figs. 43(b)(c), was not actively studied from 2008.

7.3.2.4 Content-based Recommendation

Throughout analyses, the user can assign ratings to the documents he/she likes or dislikes. Among the retrieved documents, suppose the user found a document "Automatic tool for Alzheimer's disease diagnosis using PCA and Bayesian classification rules" interesting and assigned the document a 5-star rating(highly-like) by right-clicking the corresponding data point in the *Scatter Plot View*. Based on this user preference information, VisIRR identifies the recommended documents based on the content similarity. These rated and the recommended documents are displayed in a tabular form in the *Recommendation view* (Fig. 40(C)).

From the list of recommended documents shown in the lower table, the user could obtain an idea that the research about Alzheimer's disease mainly involves an image analysis, clustering, classification, etc. Notice that without such a recommendation capability of VisIRR, the user would not be able to obtain these documents since these documents were not included in the retrieved set by user queries. In the *Scatter Plot view*, the user can see these recommended documents at the upper left corner around the rated document and its nearby clusters. To obtain a better idea about the recommended documents, the user can create another visualization by only using this subset with a new clustering and dimension reduction (Fig. 40(D)). From its own cluster summary and visualization, the user could see that the documents directly related to Alzheimer's disease are mainly shown in the bottom half while the upper half in the *Scatter Plot view*, shows documents mainly related to image analysis such as content-based image retrieval, clustering, etc.

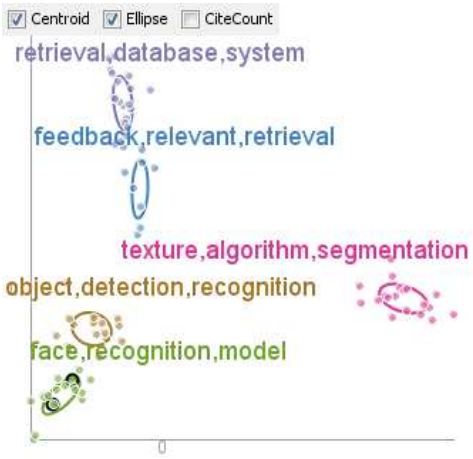
7.3.2.5 Citation- and Co-authorship-based Recommendation

Now, among the recommended documents, the user chose another document "Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations" and assigned it a 5-star rating. This time, the user changes

Recommended Documents

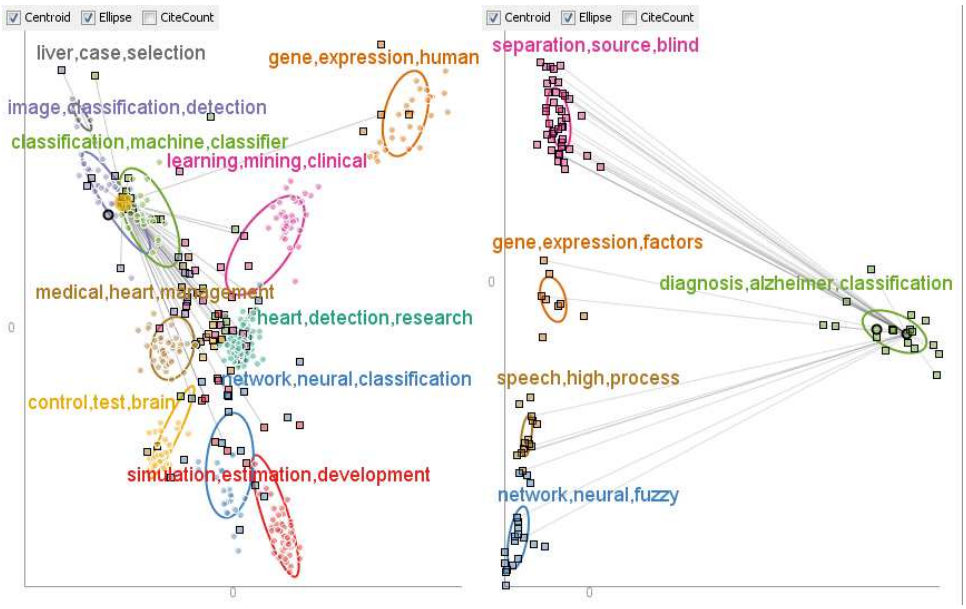
Title	Authors	Year	Venue	Cit...	Rating
Asymmetric Bagging and Random Subspace for Support...	Dacheng Tao,...	2006	IEEE Transactions on ...	168	
Face Recognition Using Component-Based SVM Classific...	Jennifer Huan...	2002	Support Vector Machines	24	
Face Recognition: Features Versus Templates	Roberto Brun...	1993	IEEE Transactions on ...	1208	
Example-Based Object Detection in Images by Compon...	Anuj Mohan,...	2001	IEEE Transactions on ...	424	
Face Recognition with Support Vector Machines: Global ...	Bernd Heisele...	2001	International Confere...	168	
Component-based Face Detection	Bernd Heisele...	2001	Computer Vision and P...	85	
Texture Features for Browsing and Retrieval of Image ...	B. Manjunath...	1996	IEEE Transactions on ...	1498	
Texture analysis and classification with tree-structured ...	Tianhorng Ch...	1993	IEEE Transactions on ...	648	
SIMPLIcity: Semantics-Sensitive Integrated Matching fo...	James Wang,...	2001	IEEE Transactions on ...	876	
Content-Based Image Retrieval at the End of the Early ...	Arnold Smeul...	2000	IEEE Transactions on ...	2426	
Image retrieval using color and shape	Anil Jain,Adit...	1996	Pattern Recognition	455	
Boosting Image Retrieval	Kinh Tieu,Paul...	2000	Computer Vision and P...	340	
Support vector machine active learning for image retrieval	Simon Tong,E...	2001	ACM Multimedia Confe...	543	
Multi-class relevance feedback content-based image re...	Jing Peng	2003	Computer Vision and I...	21	
Hierarchical Mixtures of Experts and the EM Algorithm	Michael Jorda...	1994	Neural Computation	1269	
Incorporate Support Vector Machines to Content-Base...	Pengyu Hong...	2000	International Confere...	86	
On Combining Classifiers	Josef Kittler,...	1998	IEEE Transactions on ...	2176	

(a) The top-ranked recommended document list



(b) A visualization of recommended documents

Figure 44: Citation-based recommendation results obtained by assigning a 5-star rating to the paper, "Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations." VisIRR recommends various papers mostly with high citation counts, which are relevant to the rated paper.



(a) A visualization of retrieved and recommended documents

(b) A visualization of only the recommended documents

Figure 45: Co-authorship-based recommendation results based on the paper, "Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations." Edges show direct co-authorship relations from the rated document.

its recommendation type to a citation-based one from the *Recommendation panel* in order to obtain highly-cited documents relevant to this document. As expected, VisIRR's top-ranked recommended documents are relatively highly cited papers, as shown in Fig. 44(a). After generating another visualization only using these recommended items, the user can obtain a summary about them, the clusters of which are composed of image retrieval, object detection/recognition, face recognition, and texture analyses (Fig. 44(b)). Notice that these types of recommendation results would not be easily obtained by a simple keyword search since these recommended documents do not contain specific keywords in common. Instead, they are only implicitly related with each other via a citation network on which VisIRR can perform a recommendation based.

In addition, the user also wanted to know what other topics or areas the authors of this paper are involved in. To this end, the user changed the recommendation type to a co-authorship-based one from the *Recommendation view*. In addition, to better show the direct co-authorship relationships from the rated paper, the user turned on the "Edges" checkbox by selecting the edge type as "Co-authorship" in the *Label panel*. The existing visualization of the retrieved documents now includes the recommended documents as well as the direct co-authorship relations from the rated document, as shown in Fig. 45(a). Similar to the previous case, the user can generate another visualization of only the recommended items to have a better idea about the recommended documents. After varying the number of clusters, the user obtains a new visualization as shown in Fig. 45(b). From this visualization, the user could gain an insight that the authors of the rated paper have written the papers, other than Alzheimer's disease-related papers (the green cluster on the right), in the four areas corresponding to blind source separation, gene expression, speech processing, and neural networks. This potentially indicates that the user, who was originally interested in Alzheimer's disease diagnosis, could expand his/her research by following

the ways the authors of the rated paper have published in other domains.

7.4 Data Collection / Ingestion

7.4.1 Initial Data Collection

VisIRR is intended to efficiently handle a large-scale document corpus with a rich set of features. To this end, VisIRR begins with the ArnetMiner data set, which is composed of about half a million academic papers, books, etc. [124].[2] Although the data set is mainly used in citation network analyses, it includes a variety of both structured and unstructured information such as a title, keywords, an abstract, authors, a publication year, a venue, a document type such as a book, a paper, etc., papers in the reference list, papers citing this document, the number of references, and the number of citations.

However, the original data set has numerous missing values and inconsistencies such as different expressions of an author's name, a publication venue, etc. To clean up the data, we utilized the Microsoft Academic Search API's.[3] Specifically, we used a title of each document as a query in order to obtain the full information about the document from the Microsoft Academic Search API, which fills the missing values and rectifies the inconsistencies. Finally, VisIRR builds upon 432,605 documents spanning from year 1825 to 2011.

7.4.2 Data Ingestion

Now we describe how we make these large-scale data readily available for real-time interactive analyses in VisIRR. Basically, VisIRR maintains the information about data in three different forms, (1) original fields of data, (2) a vector representation, and (3) graph representations, in an efficient and scalable way. In order to efficiently

[2]The used data is available as 'DBLP-Citation-network V5' at `http://arnetminer.org/citation`.

[3]`http://academic.research.microsoft.com/About/Help.htm`.

manage the large-scale data in all these various forms, we carefully optimized various data processing/storage techniques via database construction, pre-computation of frequently used information, and balanced storage between disk and memory. Eventually, the system is easily and widely deployable in typical commodity PC's instead of requiring high-performance parallel machines.

7.4.2.1 Original Field of Data

For efficient and flexible query support, we have encoded the original data as a SQL database including full-text search capabilities on a title, keywords, an abstract, and a venue fields. For clustering and dimension reduction steps, we have pre-computed the sparse vector representations of individual documents based on a title, keywords, and an abstract fields together via a bag-of-words encoding scheme. Each vector representation is stored as a single file in a disk, the file name of which is the document ID. In this way, VisIRR can retrieve the vector representations of documents using their document ID's in the time complexity of $O(1)$.

7.4.2.2 Vector Representation

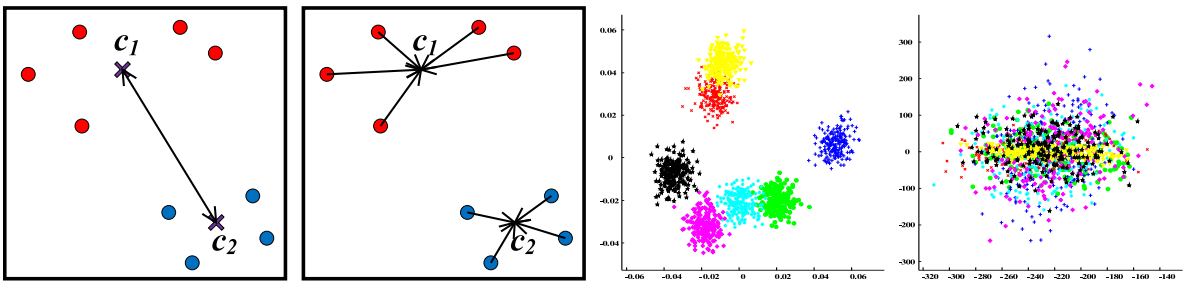
Once the vector representations of documents are loaded into a memory, VisIRR manages them in a similar way to cache replacement algorithms. That is, the vector representations already loaded into the memory is referenced from the memory whenever needed. When the total memory-loaded vectors exceed a pre-defined maximum memory size, the least recently used vectors are removed from the memory. When needed later, they are loaded from a disk once again. This way, VisIRR does not need to load the vector representations of all the documents from the beginning, which will take significant time and memory at the system startup. At the same time, VisIRR prevents the required memory size from blowing up due to a long-term usage of the system.

7.4.2.3 Graph Representation

The recommendation module, which will be described in Section 7.5, requires an input graph where the nodes correspond to documents and the edges represent their pairwise similarities/relationships. We have pre-computed three such graphs for the entire data set using contents, a citation network, and co-authorship, respectively, in order to support various recommendation capabilities. For content-based graph, we initially computed the pairwise cosine similarities between all the pairs of documents using their vector representations. Since maintaining all the pairwise information requires $O(n^2)$ storage where n is the total number of documents, we identified the fixed number (10 in our case) of the most similar documents for each document and kept only the edges between them. For the citation graph, we formed edges between a pair of documents if either cites the other. For the co-authorship graph, edges are created if two documents share the common author(s). Since citation and co-authorship graphs are typically sparse, we stored all these edge information. For each graph, VisIRR maintains the mappings from an individual document to a list of edges in terms of the destination document and its edge value so that it can retrieve the edge information for particular documents in the time complexity of $O(1)$.

7.4.3 Scalable Update for New Data

Even though VisIRR already contains a large-scale data of about half a million documents, it is crucial to have a capability to efficiently update the above-described information including newly added documents. An updating process is composed of two parts: updating the information about existing documents and obtaining the representations of new documents. First, in the case of the original fields of data, the information about new documents can be easily added to the database without affecting the existing data. Second, In the case of updating bag-of-words vector representations, new documents generally causes newly appearing keywords to be



(a) Maximization of distances between cluster centroids

(b) Minimization of approximate cluster radii

(c) LDA

(d) PCA

Figure 46: A high-level idea of LDA and a comparison example betwee LDA and PCA. A different color corresponds to a different cluster, and c_1 and c_2 are the cluster centroids. LDA tries to find a reduced-dimensional representation of data by putting different clusters as far as possible (a) and representing each cluster as compact as possible (b). (c) and (d) show an example 2D scatter plots obtained by PCA and LDA, respectively, for artificial Gaussian mixture data with 7 clusters and 1,000 original dimensions. From a comparison between them, LDA is shown to reveal a much clearer cluster structure than PCA in a 2D space.

indexed as additional dimensions. However, sparse vector representations of existing documents would still remain the same, and thus we only need to compute the representation of new documents, which can also be easily done.

Finally, in the case of updating graph representations, the only tricky part is to update the content similarity graph, where the top 10 most similar documents and their cosine similarity values are maintained. Specifically, we have to compute the pairwise similarity between all the existing documents and all the new documents, and then compare these similarity values against the current top 10 similarity values. If any of the former similarity values are greater than the latter similarity values, the corresponding edges are replaced with those to the new documents. The computational complexity of this process is $O(n \times n_{new})$ where n and n_{new} are the numbers of the existing and the new documents, respectively.

7.5 Computational Methods

The key computational methods in VisIRR are clustering, dimension reduction, alignment, and graph-based recommendation. In this section, we describe each module in detail.

7.5.1 Clustering

Clustering plays a crucial role in providing a summary of a given set of documents as a manageable number of groups based on their semantic meanings. The resulting cluster indices are used to color-code documents in a scatter plot with their cluster summaries in terms of the most frequently shown keywords (Fig. 40(B)(E)). VisIRR adopts a state-of-the-art technique called nonnegative matrix factorization (NMF) [80], which have shown superior performances in document clustering over traditional methods such as k-means [81, 141].

Given a nonnegative matrix $X \in \mathbb{R}^{m \times n}$, and an integer $k \ll \min(m, n)$, NMF finds a lower-rank approximation given by

$$X \approx WH, \tag{40}$$

where $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ are nonnegative factors. NMF can be formulated using the Frobenius norm as

$$\min_{W,\, H \geq 0} \|X - WH\|_F^2 . \tag{41}$$

In the context of document clustering, each column vector $x_i \in \mathbb{R}^{m \times 1}$ of X represents each document as an m-dimensional vector via a bag-of-words encoding, with some additional pre-processing steps such as inverse-document frequency weighting and vector norm normalization. The value of k represents the number of clusters. For clustering, one can utilize H as a soft clustering vector representation of documents. That is, the column vector $h_i \in \mathbb{R}^{k \times 1}$ of H represents such a soft clustering vector

for the i-th document, and by taking the index the value of which is the largest, the cluster index of the document can be obtained.

The specific NMF algorithm we have used is based on a recently proposed block principal pivoting algorithm [82],[4] which is found to be one of the fastest and reliable algorithms. Although not reported, we have conducted an extensive amount of comparison of NMF against traditional clustering techniques such as k-means, and we found that NMF mostly gives semantically more meaningful clusters than any other methods while requiring a significantly faster computational time.

7.5.2 Dimension Reduction

Given high-dimensional vector representations of documents, dimension reduction computes their 2D representations so that they can be visualized in a scatter plot (Fig. 40(B)). From the scatter plot, users can get an idea about how clusters/documents are related with each other. VisIRR adopts an advanced dimension reduction method called linear discriminant analysis (LDA) [68].

Unlike traditional methods such as principal component analysis and multidimensional scaling, LDA explicitly utilizes additional cluster label information, which are taken from the clustering module, associated with the input high-dimensional vectors. Using this information, LDA tries to preserve the cluster structure in the low-dimensional space such that the dimension-reduced result can clearly reveal the underlying cluster structure in the input data. In this manner, as shown in Fig. 46, LDA has an advantage over most traditional methods such as PCA and MDS in that it can provide a clear cluster structure in the data when the cluster label information is given.

Furthermore, VisIRR provides a slider interface for controlling how compactly each cluster is represented by using regularization on LDA, which enables users to

[4]The source code is available at `http://www.cc.gatech.edu/~hpark/nmfsoftware.php`.

focus their analyses at either a cluster level or an individual document level. For more details, refer to [35, 36].

7.5.3 Alignment

In VisIRR, users can create multiple scatter plots for (1) new parameter values, e.g., the number of clusters in NMF, a regularization value in LDA, and (2) a new set of data from different queries or arbitrary selection by users. In order to maintain consistency between different scatter plots and facilitate their easy comparisons, VisIRR provides alignment capabilities on different clustering and dimension reduction results. By aligning clustering results, users can expect that the same cluster index and color indicate semantically similar meanings. On the other hand, by aligning dimension reduction results, users can expect that the same data point is located in a similar position in the 2D space between different scatter plots.

To align different clustering results, VisIRR utilizes the Hungarian algorithm [85]. Given two sets of cluster assignments for the same set of documents, the Hungarian algorithm finds the optimal pairwise matching of cluster indices between the two sets so that the number of common data items within matching cluster pairs can be maximized. Based on the resulting matching, VisIRR changes the cluster indices and the colors of the newly created scatter plot with respect to those of the used reference scatter plot. In this manner, VisIRR maintains the cluster indices/colors with their consistent semantic meanings throughout multiple visualization results.

The alignment of different dimension reduction results is based on Procrustes analysis [69, 53], which best maps one result to the other with only a rotation matrix. In addition, VisIRR extends the original Procrustes analysis by incorporating translation and isotropic scaling factors as well. That is, given two reduced-dimensional matrices $X,\ \mathrm{Y} \in \mathbb{R}^{m \times n}$, where m is the number of dimensions and n is the number of

data points, VisIRR solves

$$\min_{Q,\mu_X,\mu_Y,k} \left\| \left(X - \mu_X \mathbf{1}_n^T\right) - kQ\left(Y - \mu_Y \mathbf{1}_n^T\right) \right\|_F, \tag{42}$$

where $Q \in \mathbb{R}^{m \times m}$ is an orthogonal matrix (for rotation), μ_X and μ_Y are m-dimensional column vectors (for translation), k is a scalar (for isotropic scaling), and $\mathbf{1}_n$ is an n-dimensional column vector whose elements are all 1's. Eq. (42) is efficiently solved by using eigendecomposition. These alignment functionalities help users understand how similarly/differently the corresponding data items/clusters are placed between different views.

7.5.4 Recommendation

The main input to the recommendation algorithm is the personalized preference to particular documents, which are interactively assigned by users in a 5-star rating scale, as shown in the bottom-right in Fig. 40(B). By default, all the documents are assumed to have a 3-star rating, which is converted to a zero preference value, but users can interactively assign ratings to particular documents, where a 1-star corresponds to a preference value of -2, and 5-star to +2, etc.

Given these user preference information, VisIRR identifies the recommended documents by performing a PageRank-style graph diffusion algorithm on a weighted graph of the entire document set. As briefly discussed in Section 7.4, such a graph can be based on either contents, a citation network, or co-authorship depending on users' choice. Particularly, VisIRR has adopted a heat-kernel-based algorithm [40], which gives a much faster convergence than the other traditional algorithms. In detail, given an input graph $W \in \mathbb{R}^{N \times N}$ between N documents, where each column of W is normalized such that its sum is equal to one, and a user preference vector $p \in \mathbb{R}^{N \times 1}$, where the i-th component p_i is the preference value, VisIRR computes the

recommendation score vector $r \in \mathbb{R}^{N\times 1}$ of N documents

$$r = \alpha \sum_{k=0}^{n} (1-\alpha)^k W^k p, \tag{43}$$

where α and n are user-specified parameters, e.g., by default, $\alpha = 0.7$ and $n = 3$. An intuitive explanation of this formulation is that the preference value $p_i$of node i is propagated to its neighbor nodes with the corresponding weights specified in the graph W at the first iteration, and then the resulting values are then propagated again with the same graph W with the scale factor $(1-\alpha)$ at the next iteration, and so on. Finally, those values computed from each iteration is added up, forming a final recommendation score vector r. Once the computation is done, VisIRR presents the documents with the biggest scores in r as the recommended ones.

One may think that Eq. (43) is computationally intensive because our input graph W is very large-scale. However, all the computations, which are basically matrix-vector multiplications, are performed based on sparse representations. Therefore, as long as W and p have few non-zero entries, the computation is typically done fast. Furthermore, VisIRR supports the capabilities of interactively adding/removing the rated documents as well as changing the ratings of the existing documents. Such computations are performed dynamically per their individual interactions, which essentially makes p have only one non-zero entry. In this way, VisIRR maintains the real-time efficiency of computations during users' frequent interactions.

7.5.5 Implementation

The system is mainly implemented in JAVA for front-end UI and rendering modules, which are partly based on the FODAVA Testbed system [32]. NetBeans Rich Client Platform and IDE[5] have been used for flexible window management. The back-end computational modules NMF and LDA are originally written in MATLAB but we have wrapped them into a JAVA library by using a Matlab built-in functionality

[5] `http://netbeans.org/features/platform/index.html`

called 'Javabuilder.'[6] Since the library made in this manner is self-contained, VisIRR does not require an actual Matlab to be installed. For querying and accessing with the database, we have used H2 library.[7]

7.6 Confirmatory User Study

The evaluation of information visualization and visual analytic systems has been an acknowledged challenge [108]. Insight-based evaluation [114, 109] has gained popularity recently as an alternative to traditional time-and-accuracy measures. As a preliminary gauge of how well our usage scenarios match real user behaviors, we have conducted an evaluation of VisIRR with end users, which consisted of an informal, non-experimental insight-based protocol.

The design of this study is evidence-by-existence. That is, our goal is to provide some support of our implicit VisIRR design claims. For example, we seek to show that recommendations outside the initial query set are useful to some people and they can find useful documents with VisIRR. It is not an experimental design as it includes no control condition, so we cannot and do not make any relative claims about VisIRR's effectiveness compared to other research or commercial alternatives (e.g., Google Scholar). Instead, our purpose is modest: demonstrate VisIRR *can* meet its intended purpose for real users (providing evidence that our imagined user scenarios above are valid), and provide direction for a future, comprehensive experimental or quasi-experimental design.

7.6.1 Method and Limitations

Participants in the study used VisIRR implemented with the same ArnetMiner-based set of academic articles described in the usage scenarios above. After completing a consent form and a brief demographics questionnaire, they were provided a live demo

[6] `http://www.mathworks.com/products/javabuilder/`
[7] `http://www.h2database.com/html/main.html`

Table 9: The study UI action counts across all participants and tasks.

Action	Description	Count
Tooltip	A tooltip showing document details triggered by hovering over a table row or scatter plot node	38897
Rating	The user picks a non-default 1-5 star rating from table entires or scatterplot nodes	80
Details	The user shows the details dialog box for one or more documents	146
Copy	The user copies document information to the clipboard	35
Filter	The user performs a filter (by keyword, year, citation count or author's name) on the current results	24

of the system usage scenario (lasting 5-10 minutes, depending on questions). Participants then used the system to conduct searches of their own choosing and to complete a set of pre-defined tasks concerning either *ubiquitous computing* or *information visualization* (e.g., "Describe any apparent subfields or application areas of information visualization."). Finally, we deployed a version of the IBM Computer System Usability Questionnaire (CSUQ) [90] along with a few other subjective assessment questions specific to VisIRR.

The system was installed on a workstation with dual 2.5GHz Intel Xeon processors and 128GB RAM running 64-bit Windows 7, though the Java VM memory limit was set to only 8 GB. It was connected to both a 30" monitor (1920x1200) and a 19" monitor (1280x1024); users were free to arrange windows on either monitor, but most chose to use the majority of the 30" screen for the VisIRR windows and dialogs with the task response window on the 19" screen.

We recruited 7 male Ph.D. students between the ages of 24-40 enrolled in various technical degree programs (engineering, computer science, robotics). As such, they all had experience doing academic literature searches using online resources such as Google, Google Scholar, the IEEE/ACM digital libraries, etc. We asked participants to self-rate their familiarity with information visualization and ubiquitous computing literature; all self-rated 4 or less on a 7-point Likert scale for information visualization

and 6 of the 7 did so for ubiquitous computing. Participants completed tasks for the area with which they were less familiar. The VisIRR system was instrumented to log the UI actions shown in Table 9. We non-intrusively observed users while they completed the tasks.

We present only a few quantitative measures in our results and no mean values as the limited sample and non-experimental nature of the study would render them specious. The tooltip counts in Table 9 are somewhat exaggerated because the VisIRR tooltips have a very short timeout triggering their appearance, meaning many tooltips could be triggered just from panning over one of the document lists or through the scatterplot.

7.6.2 Results and Discussion

Table 9 shows the raw action counts across all 7 users and all tasks. Those counts match our subjective impressions of watching users complete tasks: they consistently made use of the major VisIRR features (visualization, ratings and recommendations and details-on-demand). Since one of our most basic questions was whether users would actually make use of the more novel features like ratings and recommendations, this preliminary result was encouraging.

All users made at least 9 distinct document ratings (again, across all tasks), and interestingly did so relatively evenly from different portions of the UI (the recommended, rating and query lists, and the scatterplot). Document details were disproportionately triggered from the visualization (112/146), indicating both that participants interacted with the visualization and drilled down into document details from there. This matches both our subjective observations and post-test user comments like "*It's good to have that first clustering result ... It's easy to go deeper down from one or two clusters.*" Unfortunately, the logging does not distinguish between regular and recommended document nodes in the scatter plot.

On the subjective CSUQ, scores were generally 5 or higher, with the lowest rated scores coming on the questions "*The system has all the functions and capabilities I expect it to have*"; "*The system gives error messages that clearly tell me how to fix problems*"; and *"Whenever I make a mistake using the system, I recover easily and quickly.*" We suspect these ratings reflect occasional software bugs and crashes that occurred during some of the participant sessions.

Our results also suggest a potential interesting contrast in user behavior with more traditional keyword-based search algorithms: one might expect in exploratory tasks with keyword engines to see multiple iterations of keyword refinement and result inspection for a given task or user. However, our users performed relatively few filter actions (all keyword refinements rather than by author, time or citation). However, because VisIRR recommendations expand the search query outside its original bounds (and highlight those nodes which are outside those bounds), iterating keyword terms is less necessary, though future work is necessary to confirm this idea, or to gauge whether this approach is more or less effective than keyword refinement.

Of course, we would hypothesize that rating-based refinement is more productive since it does require less user expertise at generating useful keyword sequences; at least one user agreed, saying that VisIRR "*... is definitely much better than blindly searching Google Scholar or basic search engines using just a few keywords.*"

7.7 Conclusions

In this chapter, we have presented a visual analytics system called VisIRR, an interactive visual information retrieval and recommendation system for document discovery. One of the primary contributions of VisIRR is that it has effectively combined both paradigms of passive query process and active recommendation by reflecting the user preference feedback. In addition, VisIRR directly tackles a large-scale document corpus via efficient data management and new data updating as well as a suite

of state-of-the-art computational methods such as NMF, LDA, and graph diffusion-based recommendation.

Our future work includes the following.

- *Collaborative filtering-based recommendation*: In addition to the preference-based recommendation we have taken, it would be more effective if VisIRR could support collaborative filtering-based approach [22] by using multiple other users' preference information. However, collecting this preference information from various users is not easy. In this respect, VisIRR could conversely be used as an easy visual interactive tool to collect these preference information after deployed to many users, just as we have collected various information about the user interaction history in Section (7.6).

- *Fast interactive clustering and layout*: We found that many users often complained about visualization not coming up immediately, which is due to high computation time. When hundreds or thousands of documents are involved, the clustering and the dimension reduction computation typically takes from a few seconds to a minute. In addition, the user sometimes wanted to move documents/clusters to see what other documents/clusters move correspondingly. The fast and interactive clustering and layout algorithms incorporating this user feedback would help VisIRR substantially.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

8.1 Summary of Contributions

In this thesis, we have discussed how to tightly integrate automated computational approaches and interactive visualization approaches for large-scale high-dimensional data analysis such as images and text documents. Even with a clear motivation of the integration between them, there exist several hurdles such as significant computational time and interpretation difficulties. To handle these problems, the thesis presents several ways to customize the computational techniques in terms of the theoretical re-formulation and algorithmic re-design. Such improved algorithms make it possible for complicated and computationally intensive techniques to be easily integrated into visual analytics scenarios. Based on the redesigned techniques, the thesis includes development of an actual visual analytics system that tightly integrates the advanced computational techniques and enables users to take advantage of them in practical data analysis scenarios.

In summary, the contributions of the thesis are summarized as follows:

1. A theoretical framework of visualizing clustered high-dimensional data via dimension reduction. The proposed two-stage framework enables various combinations and their interpretations of several well-known supervised and unsupervised dimension reduction methods to obtain appropriate 2D/3D representations of high-dimensional data. [35].

2. An algorithmic redesign of computational modules to enable real-time visualization and interaction with computationally intensive algorithms and large-scale data. The presented parametric updating algorithms will support one of the

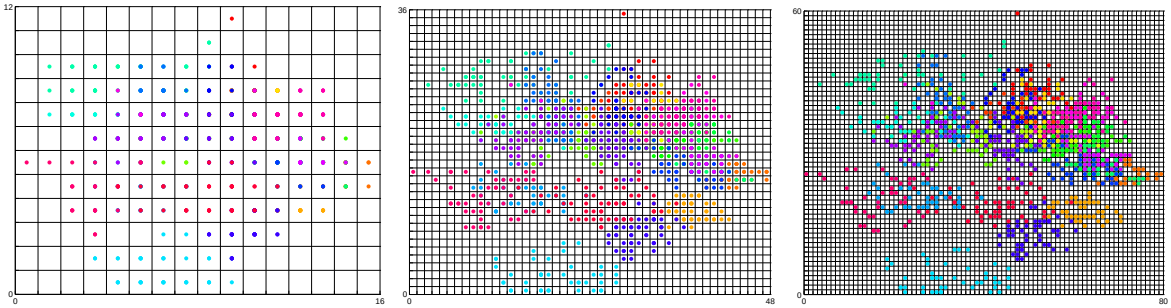
most essential interactions, changing the parameters, and the iteration-level integration of computational modules with interactive visualization will help users quickly explore the results visually [34].

3. Iteration-wise integration framework of computational methods for real-time visualization and interaction. The presented framework and several applications of this idea in existing visual analytics systems, such as Jigsaw, iVisClustering, and the Testbed system, shows the effectiveness of the proposed framework using widely-used computational methods such as PCA, MDS, t-SNE, k-means, and latent Dirichlet allocation [31].

In terms of the developed visual analytics systems, the thesis has presented

1. Testbed: an interactive visual testbed system for various dimension reduction and clustering methods. The Testbed system brings a wide variety of traditional and state-of-the-art dimension reduction and clustering methods to visual analytics. The Testbed system provides full control of these methods with interactive visual access to their results. In addition, our system offers a flexible extensibility for new data types and methods [32].

2. iVisClassifier: an interactive visual classification system that uses supervised dimension reduction. iVisClassifier enables users to explore high-dimensional data through a supervised dimension reduction method, LDA. We interpret the effect of regularization in visualization and provide an effective user-interface in which users can control the cluster radii depending on whether they focus on the cluster- or the data-level relationships. In addition, iVisClassifier facilitates the interpretability of the computational model applied to their data. Various views such as parallel coordinates, scatter plots, and heat maps interactively show rich aspects of the data. Finally, we showed that iVisClassifier can efficiently support a user-driven classification process by reducing humans' search space,



(a) The scatter plot in the screen of a resolution 16×12.

(b) The scatter plot in the screen of a resolution 48×36.

(c) The scatter plot in the screen of a resolution 80×60.

Figure 47: Hierarchical precision refinement of PCA computational results. 1,420 facial image data represented as 11,264-dimensional vectors have been visualized with their person ID color-coded.

e.g., recomputing LDA with a user-selected subset of data and mutual filtering in parallel coordinates and the scatter plot[36].

3. VisIRR: an interactive visual information retrieval and recommender system for large-scale document data. VisIRR integrates two main notions of information retrieval and personalized recommendation into a single visual analytics system. VisIRR is well-engineered to handle large-scale data and streaming data and utilizes the state-of-the-art clustering and dimension reduction methods such as NMF and LDA. The recommendation module works on an efficient graph diffusion algorithm on large-scale sparse graphs based on various criteria such as content, co-authorhship, and citation network. [33].

8.2 Future Directions

This thesis opens up various future research directions when truly integrating computational methods with human-in-the-loop visual analytics approaches. In order for computational methods to be fully utilized in visual analytics, computational methods have to provide users with *real-time interactivity* and *output trustworthiness* for

users' own tasks as detailed in the following subsections. Based on such improvements in various ways, users should be able to focus more on their own tasks and goals instead of worrying much about computational methods themselves. In this manner, the visual analytics with computational methods would be able to find more real-world values in practical application domains.

8.2.1 Real-time Interactivity

For the former, this thesis already presented one approach called PIVE, described in Chapter 4, where we have exploited the existing characteristics of most modern algorithms that they are mainly based on algorithmic iterations. However, rather revolutionary paradigm changes of computational methods could be considered when designing the algorithms of computational methods for visual analytics applications. One such approach would be to re-design the algorithms by hierarchically refining the precision of the solutions of computational methods. Fig. 47 shows an example in which the precision of the computational results are iteratively refined with respect to the increasing screen resolution. During this precision refinement process, the next step of refinement could utilize the results of the previous step, which makes each refinement step efficient. More specifically, to find out the new position of a specific data item in a higher resolution, the refinement process may only need to examine the nearby areas from the previous position. Although we do not provide detailed algorithms on how to realize this approach, one could find relevant literature from other domains. Such literature includes adaptive mesh refinement [18][1] in numerical analysis and wavelet transform [39][2] in image coding/compression. Applying these ideas to the context of integrating computational methods in visual analytics would be a promising research direction.

Another potential idea to achieve real-time interactivity is to confine the data

[1] http://en.wikipedia.org/wiki/Adaptive_mesh_refinement
[2] http://en.wikipedia.org/wiki/Wavelet_transform

scale. As clearly seen in Fig. 47(a), the finite resolution in the screen space introduces the limitation in the number of data items that can be visualized. Suppose there are much more data items than the total number of pixels available. In this case, it does not make sense to compute algorithms on the entire set of data items even though there is no possible way to visualize all of them. This approach is particularly useful when it comes to the computational complexity of algorithms. In principle, as the number of data items increases, the algorithm complexity cannot be more efficient than $O(n)$, which assumes that every data item is processed at least once. Even with such an ideal complexity, a computational bottleneck can exist in real-time visual analytics. The notion of a fixed number of available pixels can turn the algorithm complexity into $O(1)$ in the sense that we can visualize only a specific number of data items at most. One of the easiest ways to select this subset of data is random sampling, although one could adopt other more carefully designed sampling methods that better represent the entire data set.

However, some user interactions such as zoom-in/out may require the computational results on the rest of the data items whose results have yet to be computed. However in this case, one can handle the situation via different efficient computations. For example, suppose one wants to perform clustering on a large-scale data set, and the computations have been performed only on a certain subset of data. Then, to obtain the cluster labels of the other data items, one could apply a simple classification method based on already computed clusters. In addition, in the case of dimension reduction, suppose PCA has been computed on a subset of data. Then, the rest of the data can be projected onto the same space via a linear transformation matrix given by PCA, which is a much more efficient process than computing PCA on the entire data set. Although these approximated approaches cannot give the exact same results as the ones generated by using the entire data from the beginning, it is a viable approach to ensure real-time visual analytics for large-scale data.

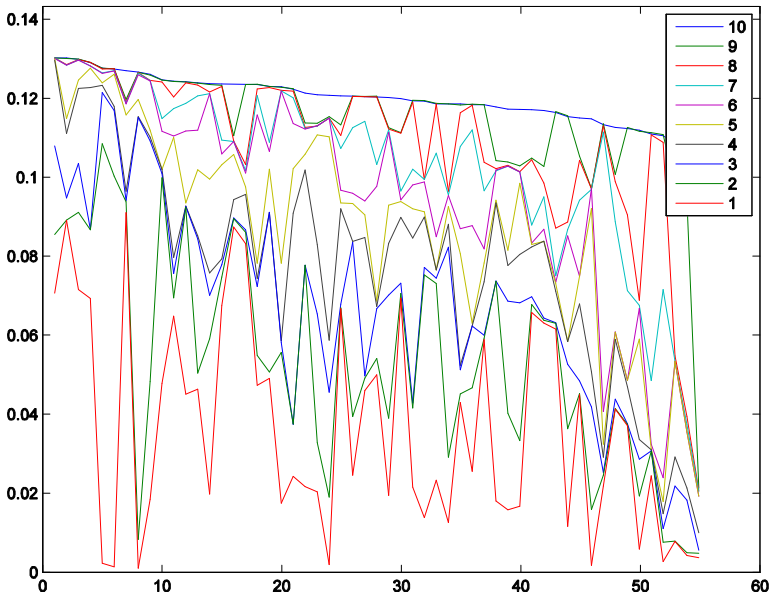


Figure 48: The degradation of pairwise distances of classical MDS depending on the target dimension. The original data are 500 synthetically generated data items in a 10-dimensional space. All their pairwise distance values computed in the 10-dimensional space are sorted in a decreasing order and are depicted as a blue line on the top. After computing MDS results with a particular target dimension, their corresponding distances are aligned along a vertical axis and depicted as a separate line.

8.2.2 Output Trustworthiness

Other than the real-time interactive capability, the overall quality of computational results should be reasonably trustworthy enough in practice for users' own tasks. Even though a particular computational method gives the best result from the perspective of its own criteria, it may not be a faithfully good quality of results to users.

For example in dimension reduction, MDS gives the reduced-dimensional result that best preserves the original pairwise dimension reduction under the low-dimensional space within a given dimension. Fig. 48 shows how these pairwise distances in the lower-dimensional space are degraded compared to those in the original high-dimensional space as the target dimension decreases. Each depicted

line in Fig. 48 represents the pairwise distances as their original values in the high-dimensional space decrease along with a horizontal axis. Fig. 48 indicates that MDS significantly distorts the original data relationships by severely decreasing particular pairwise distances while some others are almost preserved. Even though MDS supposedly gives the best result in preserving the pairwise distances, one would not be able to trust the result considering such a significant distortion.

There are two ways to tackle these problems. The first one would be to design a new computational method based on improved criteria that perceptually make more sense. For instance, one could come up with a new criterion for an alternative method to MDS so that distance losses can be evenly distributed throughout all the pairwise distances. However, such perception-friendly criteria may cause additional computational complexities. Therefore, as another way to tackle the trustworthiness problem, visual analytics could at least provide users with the information about how trustworthy the computational results are by showing the perceptual quality measures. Studying these new computational methods as well as corresponding perceptual quality measures would be another promising research direction.

REFERENCES

[1] "Google scholar." `http://scholar.google.com`. Accessed: Mar. 2013.

[2] "Microsoft academic search." `http://academic.research.microsoft.com/`. Accessed: Mar. 2013.

[3] Adomavicius, G. and Tuzhilin, A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 734 – 749, june 2005.

[4] Alencar, A. B., de Oliveira, M. C. F., and Paulovich, F. V., "Seeing beyond reading: a survey on visual text analytics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 476–492, 2012.

[5] Alsakran, J., Chen, Y., Zhao, Y., Yang, J., and Luo, D., "Streamit: Dynamic visualization and interactive exploration of text streams," in *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pp. 131–138, 2011.

[6] Andrews, K., Gutl, C., Moser, J., Sabol, V., and Lackner, W., "Search result visualisation with xfind," in *User Interfaces to Data Intensive Systems, 2001. UIDIS 2001. Proceedings. Second International Workshop on*, pp. 50–58, IEEE, 2001.

[7] Ankerst, M., Keim, D., and Kriegel, H., "'Circle Segments': A Technique for Visually Exploring Large Multidimensional Data Sets," in *Proc. Visualization*, (Hot Topic Session,), 1996.

[8] Ankerst, M., Elsen, C., Ester, M., and Kriegel, H.-P., "Visual classification: an interactive approach to decision tree construction," in *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 392–396, ACM, 1999.

[9] Ankerst, M., Ester, M., and Kriegel, H.-P., "Towards an effective cooperation of the user and the computer for classification," in *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 179–188, ACM, 2000.

[10] Asimov, D., "The grand tour," *SIAM Journal of Scientific and Statistical Computing*, vol. 6, no. 1, pp. 128–143, 1985.

[11] Asuncion, A. and Newman, D., "UCI machine learning repository." University of California, Irvine, School of Information and Computer Sciences, 2007.

[12] Balasubramanian, M., Schwartz, E. L., Tenenbaum, J. B., de Silva, V., and Langford, J. C., "The Isomap Algorithm and Topological Stability," *Science*, vol. 295, no. 5552, p. 7a, 2002.

[13] Barbehenn, M., "A note on the complexity of dijkstra's algorithm for graphs with weighted vertices," *Computers, IEEE Transactions on*, vol. 47, p. 263, Feb 1998.

[14] Basu, C., Cohen, W., Hirsh, H., and Nevill-Manning, C., "Technical paper recommendation: A study in combining multiple information sources," *arXiv preprint arXiv:1106.0248*, 2011.

[15] Basu, S., Banerjee, A., and Mooney, R. J., "Semi-supervised clustering by seeding," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, (San Francisco, CA, USA), pp. 27–34, Morgan Kaufmann Publishers Inc., 2002.

[16] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J., "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[17] Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[18] Berger, M. and Colella, P., "Local adaptive mesh refinement for shock hydrodynamics," *Journal of computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.

[19] Bishop, C., *Pattern recognition and machine learning*. Springer, 2006.

[20] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[21] Borner, K., Dillon, A., and Dolinsky, M., "Lvis-digital library visualizer," in *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pp. 77–81, IEEE, 2000.

[22] Breese, J. S., Heckerman, D., and Kadie, C., "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, (San Francisco, CA, USA), pp. 43–52, Morgan Kaufmann Publishers Inc., 1998.

[23] Buja, A., Cook, D., and Swayne, D., "Interactive High-Dimensional Data Visualization," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 78–99, 1996.

[24] Cao, N., Sun, J., Lin, Y.-R., Gotz, D., Liu, S., and Qu, H., "Facetatlas: Multifaceted visualization for rich text corpora," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1172–1181, 2010.

[25] Chabini, I., "Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1645, no. -1, pp. 170–175, 1998.

[26] Champagne, B., "Adaptive eigendecomposition of data covariance matrices based on first-order perturbations," *Signal Processing, IEEE Transactions on*, vol. 42, pp. 2758–2770, Oct 1994.

[27] Chau, D. H., Kittur, A., Hong, J. I., and Faloutsos, C., "Apolo: making sense of large network data by combining rich user interaction and machine learning," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, pp. 167–176, ACM, 2011.

[28] Chen, C., "Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 359–377, 2006.

[29] Chen, K. and Liu, L., "ivibrate: Interactive visualization-based framework for clustering large datasets," *ACM Trans. Inf. Syst.*, vol. 24, no. 2, pp. 245–294, 2006.

[30] Choo, J., Bohn, S., Nakamura, G., White, A., and Park, H., "Heterogeneous data fusion via space alignment using nonmetric multidimensional scaling," in *Proceedings of the 2012 SIAM International Conference on Data Mining (SDM12)*, pp. 177–188, 2012.

[31] Choo, J., Lee, C., Lee, H., and Park, H., "Visualize it-wise! an iteration-wise computational framework for real-time visual analytics," *Computer Graphics Forum*, 2013.

[32] Choo, J., Lee, H., Liu, Z., Stasko, J., and Park, H., "An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, To appear, 2013.

[33] Choo, J., Liu, Z., Lee, C., Li, F., Lee, H., Kannan, R., Stolper, C., Inouye, D., Mehta, N., Ouyang, H., Som, S., Gray, A., Stasko, J., and Park, H., "Vizirr: An interactive visual system for information retrieval and recommendation for large-scale document data."

[34] Choo, J., Reddy, C. K., Lee, H., and Park, H., "p-isomap: An efficient parametric update for isomap for visual analytics," in *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM10)*, pp. 502–513, 2010.

[35] Choo, J., Bohn, S., and Park, H., "Two-stage framework for visualization of clustered high dimensional data," in *IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009.*, pp. 67 –74, oct. 2009.

[36] Choo, J., Lee, H., Kihm, J., and Park, H., "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Visual Analytics Science and Technology (VAST), 2010 IEEE Conference on*, pp. 27 –34, oct. 2010.

[37] Chuang, J., Manning, C. D., and Heer, J., "Termite: Visualization techniques for assessing textual topic models," in *Advanced Visual Interfaces*, 2012.

[38] Chuang, J., Ramage, D., Manning, C. D., and Heer, J., "Interpretation and trust: Designing model-driven visualizations for text analysis," in *ACM Human Factors in Computing Systems (CHI)*, 2012.

[39] Chui, C. K., *An introduction to wavelets*. San Diego, CA, USA: Academic Press Professional, Inc., 1992.

[40] Chung, F., "The heat kernel as the pagerank of a graph," *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 19735–19740, 2007.

[41] Clarkson, E., Desai, K., and Foley, J., "Resultmaps: Visualization for search interfaces," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1057–1064, 2009.

[42] Coifman, R. R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006.

[43] Cook, D. and Swayne, D., *Interactive and Dynamic Graphics for Data Analysis: with R and GGobi*. Springer, 2007.

[44] Cottam, J., Lumsdaine, A., and Weaver, C., "Watch this: A taxonomy for dynamic data visualization," in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 193–202, 2012.

[45] Cox, T. F. and Cox, M. A. A., *Multidimensional Scaling*. London: Chapman & Hall/CRC, 2000.

[46] De Leeuw, J., "Applications of convex analysis to multidimensional scaling," *Recent developments in statistics*, pp. 133–146, 1977.

[47] Demetrescu, C. and Italiano, G. F., "A new approach to dynamic all pairs shortest paths," *J. ACM*, vol. 51, no. 6, pp. 968–992, 2004.

[48] Demmel, J. W., *Applied numerical linear algebra*. SIAM, 1997.

[49] Dhillon, I. S., Modha, D. S., and Spangler, W. S., "Class visualization of high-dimensional data with applications," *Computational Statistics and Data Analysis*, vol. 41, no. 1, pp. 59 – 90, 2002.

[50] Dou, W., Wang, X., Chang, R., and Ribarsky, W., "Paralleltopics: A probabilistic approach to exploring document collections," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 231 –240, oct. 2011.

[51] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*. New York: Wiley-interscience, 2001.

[52] Dunne, C., Shneiderman, B., Gove, R., Klavans, J., and Dorr, B., "Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 12, pp. 2351–2369, 2012.

[53] Eldén, L. and Park, H., "A procrustes problem on the stiefel manifold," *Numerische Mathematik*, vol. 82, pp. 599–619, 1999.

[54] Ellis, G. and Dix, A., "Enabling automatic clutter reduction in parallel coordinate plots," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 717–724, 2006.

[55] Endert, A., Han, C., Maiti, D., House, L., Leman, S., and North, C., "Observation-level interaction with statistical models for visual analytics," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 121–130, 2011.

[56] Faconti, G. and Massink, M., "Continuous interaction with computers: Issues and requirements," *Vol. 3 of the proc. of HCI International 2001*, pp. 301–305, 2001.

[57] Fisher, D., Popov, I., Drucker, S., and schraefel, m., "Trust me, i'm partially right: incremental visualization lets analysts explore large datasets faster," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 1673–1682, ACM, 2012.

[58] Friedman, J. H., "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.

[59] Frigioni, D., Marchetti-Spaccamela, A., and Nanni, U., "Fully dynamic algorithms for maintaining shortest paths trees," *Journal of Algorithms*, vol. 34, no. 2, pp. 251 – 281, 2000.

[60] Fukunaga, K., *Introduction to Statistical Pattern Recognition, second edition.* Boston: Academic Press, 1990.

[61] Golub, G. H. and van Loan, C. F., *Matrix Computations, third edition.* Johns Hopkins University Press, Baltimore, 1996.

[62] Good, L., Popat, A., Janssen, W., and Bier, E., "A fluid interface for personal digital libraries," *Research and Advanced Technology for Digital Libraries*, pp. 162–173, 2005.

[63] Görg, C., Liu, Z., Kihm, J., Choo, J., Park, H., and Stasko, J., "Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw," 2013.

[64] Grgic, M., Delac, K., and Grgic, S., "SCface - surveillance cameras face database," *Multimedia Tools and Applications Journal*, 2009.

[65] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, November 2009.

[66] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[67] Howland, P., Jeon, M., and Park, H., "Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 25, no. 1, pp. 165–179, 2003.

[68] Howland, P. and Park, H., "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 995 –1006, aug. 2004.

[69] Hurley, J. R. and Cattell, R. B., "The Procrustes program: Producing direct rotation to test a hypothesized factor structure," *Behavioral Science*, vol. 7, no. 2, pp. 258–262, 1962.

[70] Jain, A. and Dubes, R., *Algorithms for clustering data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[71] Jeon, M., Park, H., and Rosen, J. B., "Dimensional reduction based on centroids and least squares for efficient processing of text data," in *Proceedings of the First SIAM International Workshop on Text Mining*, Chiago, IL, 2001.

[72] Jeong, D., Ziemkiewicz, C., Fisher, B., Ribarsky, W., and Chang, R., "iPCA: An Interactive System for PCA-based Visual Analytics ," *Computer Graphics Forum*, vol. 28, no. 3, pp. 767–774, 2009.

[73] Johansson, S. and Johansson, J., "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 993–1000, 2009.

[74] Jolliffe, I. T., *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[75] Jolliffe, I. T., *Principal component analysis*. Springer, 2002.

[76] Kampanya, N., Shen, R., Kim, S., North, C., and Fox, E. A., "Citiviz: A visual user interface to the citidel system," *Research and Advanced Technology for Digital Libraries*, pp. 122–133, 2004.

[77] Kandogan, E., "Visualizing multi-dimensional clusters, trends, and outliers using star coordinates," in *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 107–116, ACM, 2001.

[78] Keim, D., "Information visualization and visual data mining," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, pp. 1 –8, jan/mar 2002.

[79] Kim, H. and Park, H., "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[80] Kim, H., Park, H., and Elden, L., "Non-negative tensor factorization based on alternating large-scale non-negativity-constrained least squares," *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, pp. 1147–1151, Oct. 2007.

[81] Kim, J. and Park, H., "Sparse nonnegative matrix factorization for clustering," 2008.

[82] Kim, J. and Park, H., "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[83] Kohonen, T., *Self-organizing maps*. Springer, 2001.

[84] Koren, Y. and Carmel, L., "Visualization of labeled data using linear transformations," in *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pp. 23–30, Oct. 2003.

[85] Kuhn, H. W., "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[86] Law, M. and Jain, A., "Incremental nonlinear dimensionality reduction by manifold learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 377–391, March 2006.

[87] Law, M., Zhang, N., and Jain, A., "Nonlinear Manifold Learning For Data Stream," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, pp. 33–44, Society for Industrial Mathematics, 2004.

[88] Lee, H., Kihm, J., Choo, J., Stasko, J., and Park, H., "iVisClustering: An interactive visual document clustering via topic modeling," *Computer Graphics Forum*, vol. 31, no. 3pt3, pp. 1155–1164, 2012.

[89] Lehoucq, R., Sorensen, D., and Yang, C., *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Society for Industrial and Applied Mathematics, 1998.

[90] Lewis, J. R., "Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use," *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.

[91] Lowe, D., "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150 –1157 vol.2, 1999.

[92] Ma, K.-L., "In situ visualization at extreme scale: Challenges and opportunities," *Computer Graphics and Applications, IEEE*, vol. 29, no. 6, pp. 14–19, 2009.

[93] Marchionini, G. and Shneiderman, B., "Finding facts vs. browsing knowledge in hypertext systems," *Computer*, vol. 21, no. 1, pp. 70–80, 1988.

[94] McCallum, A. K., "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu, 2002.

[95] Newman, M. E., "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[96] Ng, A., Jordan, M., and Weiss, Y., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[97] Nowell, L. T., France, R. K., Hix, D., Heath, L. S., and Fox, E. A., "Visualizing search results: some alternatives to query-document similarity," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 67–75, ACM, 1996.

[98] Page, L., Brin, S., Motwani, R., and Winograd, T., "The pagerank citation ranking: Bringing order to the web.," Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[99] Pang, B. and Lee, L., "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[100] Pang, C., Dong, G., and Ramamohanarao, K., "Incremental maintenance of shortest distance and transitive closure in first-order logic and sql," *ACM Trans. Database Syst.*, vol. 30, no. 3, pp. 698–721, 2005.

[101] Park, C. and Park, H., "Nonlinear Discriminant Analysis Using Kernel Functions and the Generalized Singular Value Decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 1, pp. 87–102, 2005.

[102] Park, C. H. and Park, H., "Nonlinear feature extraction based on centroids and kernel functions," *Pattern Recognition*, vol. 37, no. 4, pp. 801 – 810, 2004. ¡ce:title¿Agent Based Computer Vision¡/ce:title¿.

[103] Park, H., Drake, B., Lee, S., and Park, C., "Fast Linear Discriminant Analysis using QR Decomposition and Regularization," *Technical Report GT-CSE-07-21*, 2007.

[104] Piringer, H., Tominski, C., Muigg, P., and Berger, W., "A multi-threading architecture to support interactive visual exploration," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1113–1120, 2009.

[105] Pirolli, P., "Computational models of information scent-following in a very large browsable text collection," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 3–10, ACM, 1997.

[106] Pirolli, P. and Card, S., "Information foraging," *Psychological review*, vol. 106, no. 4, pp. 643–775, 1999.

[107] Pisano, E., Zong, S., Hemminger, B., DeLuca, M., Johnston, R., Muller, K., Braeuning, M., and Pizer, S., "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated

spiculations in dense mammograms," *Journal of Digital Imaging*, vol. 11, no. 4, pp. 193–200, 1998.

[108] Plaisant, C., "The challenge of information visualization evaluation," in *Proceedings of the working conference on Advanced visual interfaces*, pp. 109–116, ACM, 2004.

[109] Plaisant, C., Fekete, J.-D., and Grinstein, G., "Promoting insight-based evaluation of visualizations: From contest to benchmark repository," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 1, pp. 120–134, 2008.

[110] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M., "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, (New York, NY, USA), pp. 569–577, ACM, 2008.

[111] Roweis, S. T. and Saul, L. K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[112] Samko, O., Marshall, A., and Rosin, P., "Selection of the optimal parameter value for the isomap algorithm," *Pattern Recognition Letters*, vol. 27, no. 9, pp. 968 – 979, 2006.

[113] Sammon, John W., J., "A nonlinear mapping for data structure analysis," *Computers, IEEE Transactions on*, vol. C-18, pp. 401 – 409, may. 1969.

[114] Saraiya, P., North, C., and Duca, K., "An insight-based methodology for evaluating bioinformatics visualizations," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 11, no. 4, pp. 443–456, 2005.

[115] Schlkopf, B., Smola, A., and Mller, K.-R., "Kernel principal component analysis," in *Artificial Neural Networks - ICANN'97* (Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., eds.), vol. 1327 of *Lecture Notes in Computer Science*, pp. 583–588, Springer Berlin / Heidelberg, 1997. 10.1007/BFb0020217.

[116] Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., and Miller, M. S., "Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–10, ACM, 1999.

[117] Seo, J. and Shneiderman, B., "Interactively exploring hierarchical clustering results [gene identification]," *Computer*, vol. 35, pp. 80 –86, jul 2002.

[118] Shawe-Taylor, J. and Cristianini, N., "An introduction to support vector machines and other kernel-based learning methods," *Cambridge University Press, UK*, 2000.

[119] Shen, R., Vemuri, N. S., Fan, W., da S Torres, R., and Fox, E. A., "Exploring digital libraries: integrating browsing, searching, and visualization," in *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pp. 1–10, ACM, 2006.

[120] Spence, R. and Press, A., "Information visualization," 2000.

[121] Stasko, J., Görg, C., and Liu, Z., "Jigsaw: supporting investigative analysis through interactive visualization," *Information Visualization*, vol. 7, no. 2, pp. 118–132, 2008.

[122] Swets, D. L. and Weng, J. J., "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.

[123] Tam, K. Y. and Kiang, M. Y., "Managerial Applications of Neural Networks: The Case of Bank Failure Predictions," *MANAGEMENT SCIENCE*, vol. 38, no. 7, pp. 926–947, 1992.

[124] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z., "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, (New York, NY, USA), pp. 990–998, ACM, 2008.

[125] Tenenbaum, J. B., Silva, V. d., and Langford, J. C., "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[126] Teoh, S. and Ma, K., "StarClass: Interactive visual classification using star coordinates," in *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM03)*, 2003.

[127] Thomas, J. and Cook, K., *Illuminating the path: The research and development agenda for visual analytics*, vol. 54. IEEE, 2005.

[128] Tu, T., Yu, H., Ramirez-Guzman, L., Bielak, J., Ghattas, O., Ma, K.-L., and O'Hallaron, D. R., "From mesh generation to scientific visualization: an end-to-end approach to parallel supercomputing," in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, SC '06, (New York, NY, USA), ACM, 2006.

[129] Turk, M. and Pentland, A., "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[130] van der Maaten, L. and Hinton, G., "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[131] Vasilescu, M. and Terzopoulos, D., "Multilinear Analysis of Image Ensembles: TensorFaces," *Necture Notes in Computer Science*, pp. 447–460, 2002.

[132] Wang, C. and Blei, D. M., "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 448–456, ACM, 2011.

[133] Wang, F., Sun, J., Li, T., and Anerousis, N., "Two heads better than one: Metric+active learning and its applications for it service classification," in *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*, pp. 1022 –1027, dec. 2009.

[134] Wei, F., Liu, S., Song, Y., Pan, S., Zhou, M. X., Qian, W., Shi, L., Tan, L., and Zhang, Q., "Tiara: a visual exploratory text analytic system," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, (New York, NY, USA), pp. 153–162, ACM, 2010.

[135] Weinberger, K. and Saul, L., "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, pp. 77–90, 2006.

[136] Williams, M. and Munzner, T., "Steerable, progressive multidimensional scaling," in *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pp. 57 –64, 0-0 2004.

[137] Willing, R., "Airport anti-terror systems flub tests; Face-recognition technology fails to flag "suspects"," *USA TODAY, September 2, 2003. Available at http://www.usatoday.com/travel/news/2003/09/02-air-secur.htm.*

[138] Wise, J., Thomas, J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V., "Visualizing the non-visual: spatial analysis and interaction with information from text documents," in *Information Visualization, 1995. Proceedings.*, pp. 51–58, 1995.

[139] Wong, P. C., Foote, H., Adams, D., Cowley, W., and Thomas, J., "Dynamic visualization of transient data streams," in *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pp. 97–104, 2003.

[140] Xie, Z., Ward, M., and Rundensteiner, E., "Visual exploration of stream pattern changes using a data-driven framework," in *Advances in Visual Computing* (Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammound, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., and Avila, L., eds.), vol. 6454 of *Lecture Notes in Computer Science*, pp. 522–532, Springer Berlin Heidelberg, 2010.

[141] Xu, W., Liu, X., and Gong, Y., "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, (New York, NY, USA), pp. 267–273, ACM, 2003.

[142] Yang, J., Ward, M. O., Rundensteiner, E. A., and Huang, S., "Visual hierarchical dimension reduction for exploration of high dimensional datasets," in *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, (Aire-la-Ville, Switzerland, Switzerland), pp. 19–28, Eurographics Association, 2003.

[143] Yang, J., Peng, W., Ward, M., and Rundensteiner, E., "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," in *Information Visualization, 2003. INFOVIS 2003. IEEE*

Symposium on, pp. 105 –112, 21-21 2003.

[144] Ye, J. and Li, Q., "A two-stage linear discriminant analysis via qr-decomposition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 929–941, June 2005.

[145] Yu, H. and Yang, J., "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[146] Yu, H., Wang, C., Grout, R., Chen, J., and Ma, K.-L., "In situ visualization for large-scale combustion simulations," *Computer Graphics and Applications, IEEE*, vol. 30, no. 3, pp. 45–57, 2010.

[147] Zhang, X., Myers, C., and Kung, S., "Cross-weighted fisher discriminant analysis for visualization of dna microarray data," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*, vol. 5, pp. V–589–92 vol.5, May 2004.

[148] Zhang, Z. and Zha, H., "Principal manifolds and nonlinear dimension reduction via tangent space alignment," *SIAM Journal of Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[149] Zhao, D. and Yang, L., "Incremental isometric embedding of high-dimensional data using connected neighborhood graphs," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 86–98, Jan. 2009.

[150] Zhao, W., Chellappa, R., and Krishnaswamy, A., "Discriminant analysis of principal components for face recognition," *Automatic Face and Gesture Recognition, IEEE International Conference on*, vol. 0, p. 336, 1998.