INFLUENCE MODELING IN BEHAVIORAL DATA

A Thesis
Presented to
The Academic Faculty

by

Liangda Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
August 2015

INFLUENCE MODELING IN BEHAVIORAL DATA

Approved by:

Professor Hongyuan Zha, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Bistra Dilkina
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Le Song
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Polo Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Doctor Ricardo Baeza-Yates

Yahoo Labs

Date Approved: 22nd April 2015

To my mom and dad

ACKNOWLEDGEMENTS

First of all, I owe an enormous debt of gratitude to my advisor, Professor Hongyuan Zha, who offered priceless help and support during the process of my research. I am grateful to Professor Haesun Park and Professor Guy Lebanon for their guidance and discussions during the process of my research. I would also like to thank my committee members, Professor Bistra Dilkina, Professor Le Song, Professor Polo Chau, and Doctor Ricardo Baeza-Yates, for all their encouragement, discussions and suggestions that have greatly improved this dissertation.

Moreover, I would like to thank my former advisers, Professor Yong Yu and Professor Gui-Rong Xue in Shanghai Jiao-Tong University, who showed me the beauty of the research world in Computer Science for the first time.

I am also thankful to all my friends for their selfless support and help through my graduate study. It is my fortunate to know all of you.

Last but not least, I am deeply in indebted to my parents, for their love and great support throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Understanding influence in behavioral data has become increasingly important in analyzing the cause and effect of human behaviors under various scenarios. Influence modeling enables us to learn not only how human behaviors drive the diffusion of memes spread in different kinds of networks, but also the chain reactions evolve in the sequential behaviors of people. In this thesis, I propose to investigate into appropriate probabilistic models for efficiently and effectively modeling influence, and the applications and extensions of the proposed models to analyze behavioral data in computational sustainability and information search.

One fundamental problem in influence modeling is the learning of the degree of influence between individuals, which we called social infectivity. In the first part of this work, we study how to efficient and effective learn social infectivity in diffusion phenomenon in social networks and other applications. We replace the pairwise infectivity in the multidimensional Hawkes processes with linear combinations of those time-varying features, and optimize the associated coefficients with lasso regularization on coefficients.

In the second part of this work, we investigate the modeling of influence between marked events in the application of energy consumption, which tracks the diffusion of mixed daily routines of household members. Specifically, we leverage temporal and energy consumption information recorded by smart meters in households for influence modeling, through a novel probabilistic model that combines marked point processes with topic models. The learned influence is supposed to reveal the sequential appliance usage pattern of household members, and thereby helps address the problem of energy disaggregation.

In the third part of this work, we investigate a complex influence modeling scenario which requires simultaneous learning of both infectivity and influence existence. Specifically, we study the modeling of influence in search behaviors, where the influence tracks the diffusion of mixed search intents of search engine users in information search. We leverage temporal and textual information in query logs for influence modeling, through a novel probabilistic model that combines point processes with topic models. The learned influence is supposed to link queries that serve for the same formation need, and thereby helps address the problem of search task identification.

The modeling of influence with the Markov property also help us to understand the chain reaction in the interaction of search engine users with query auto-completion (QAC) engine within each query session. The fourth part of this work studies how a user's present interaction with a QAC engine influences his/her interaction in the next step. We propose a novel probabilistic model based on Markov processes, which leverage such influence in the prediction of users' click choices of suggested queries of QAC engines, and accordingly improve the suggestions to better satisfy users' search intents.

In the fifth part of this work, we study the mutual influence between users' behaviors on query auto-completion (QAC) logs and normal click logs across different query sessions. We propose a probabilistic model to explore the correlation between user' behavior patterns on QAC and click logs, and expect to capture the mutual influence between users' behaviors in QAC and click sessions.

CHAPTER I

INTRODUCTION

Influences play a major role in determining how individuals behave, where their behaviors can be either subjective, such as people retweet posts, or objective, such as people get infected with some viruses. The behavior of one individual can either influence one's own future behavior or the future behaviors of other individuals in the same network. Consequently, the influenced individuals may carry on the same type of behavior, such as retweet the same post or get infected by the same virus. On the other hand, they may respond with some other type of behavior based on certain rules. For instance, the attack against one country may cause its allies to revenge; the results obtained from the current search task may trigger the user to conduct a related search task in the next. The modeling of influence enables us to track the diffusion of memes, such as posts or viruses, spread in various kinds of networks, or study the chain reactions evolve in the sequential behaviors of people. One major challenge in influence modeling is that, the specific influences under different real-world scenarios are diverse, and each scenario demands unique solutions that incorporate domain-specific knowledge to appropriately model the special influence using observed behavioral data. In particular, we consider five different problems to illustrate how to model the influence in various real-world scenarios: learning parametric models for social infectivity, energy usage behavior modeling in energy disaggregation, identifying and labeling search tasks, analyzing user's sequential behavior in query auto-completion, and exploring user behavior in QAC and click log for contextual-aware web search and query suggestion.

1.1 Thesis

In this thesis, we propose to investigate the applications and extensions of methods based on probabilistic models to solve the problems mentioned above. The key idea is to analyze the major factors that implicate the particular influence we are to model under each scenario. Specifically, we propose to using time-varying features describe instant individual property or pairwise relationship to model the current influence among individuals in diffusion networks. Moreover, we combine the idea of mutually exciting marked point processes with Latent Dirichlet Allocation (LDA) to jointly utilize both temporal and energy consumption information in modeling the influence among the energy usage of appliances across different time slots. Furthermore, we combine the idea of self-exciting point processes with Latent Dirichlet Allocation (LDA) to jointly utilize both temporal and textual information in modeling the influence that links queries to form search tasks in query logs. We also build a Markov process based probabilistic model to capture the influence between users' sequential interactions with query auto-completion (QAC) engines. Finally, we propose a probabilistic model based on LDA to explore the mutual influence between users' behaviors in QAC and click logs. In the following, we will briefly introduce the backgrounds of the above five problems, analyze existing challenges in those problems, and show how we contribute to tackle those challenges.

1.2 Backgrounds, Challenges, and Our Contributions

User behavioral data plays an increasingly important role in a wide range of applications including social networks, computational sustainability and information search. Many existing studies for those applications lack detailed analysis of the associated behavioral data, especially the temporal orders in those data. Our thesis explores the influence in the behavioral data of those applications, and analyzes how the captured influence benefits the solving of those applications.

1.2.1 Learning Parametric Models for Social Infectivity

The first and the most important issue we need to study in influence modeling is how to efficiently model the influence under various behavioral data with a general solution, especially when prior knowledge about the topologies of the network of the individuals participated in the behavioral data in not avaiable. Such a general solution provides a solid foundation for the influence modeling in the behavioral data of various applications.

For influence modeling, one popular solution in recent works [257, 270] is employing one powerful statistical tool, the multi-dimensional Hawkes process [98], to model event cascades $\{t_l\}$'s in social networks and learn the degree of pairwise influence between individuals, which we call *infectivity* in this work, by taking each individual as one dimension. The multi-dimensional Hawkes process is defined to be a M-dimensional point process with the intensity of the m-th dimension given by:

$$\lambda_m(t) = \mu_m + \sum_{t_l<t} \alpha_{m_l,m}\kappa(t-t_l)$$

Here μ_m denotes the basic intensity of the m-th dimension, $\kappa(t-t_l)$ is a time-decaying kernel, while $\alpha_{m,m'}$ denotes the infectivity from events in the m-th dimension to events in the m'-th dimension. We call $\mathbf{A} = (\alpha_{m,m'})$ the *infectivity matrix*. The Hawkes parameters need to learn include $O(M)$ μ's and $O(M^2)$ α's.

Unfortunately, although having achieved remarkable performances, existing works suffer from the following drawbacks in learning α: **Problem Complexity.** Learning one separate α for each pair of dimensions is daunting. On one hand, learning $O(M^2)$ α's can be both time-consuming and unnecessary under certain scenarios. On the other hand, the chances are very high that there are no sufficient historical events for modeling the infectivity within certain individual-pairs. **Dependency in Infectivity Matrix.** Existing works [227, 14] usually ignore the dependency among α's, while under many circumstances α's are closely related. Recent works [270] that use

a priori assumptions on the network topology limit the adaptive social networks of those approaches. The structures of different social networks vary a lot, and even contradict with each other. **Time-varying Infectivity.** The infectivity α between each pair of individuals is usually time-variable. Potential solutions for learning time-varying infectivity, such as learning separate α's for each time interval or modeling α with time-dependent function, greatly increase problem complexity.

In this work, to address above drawbacks simultaneously, we build a compact model to parameterize the infectivity between individuals. The basic idea is to design a set of K time-varying features, and substitute each α with a linear combination of those features with coefficients to learn. In this way, we 1) only need to estimate K coefficients, which is controlled by the number of features we use, instead of the square of the number of individuals in the given social network. Moreover, the estimation of each coefficient fully utilizes all historical events, thus no longer demands multiple cascades and the a priori cascade assignment of new upcoming events; 2) are free to design features capturing the dependency among the infectivity within each individual-pair, based on the pairwise direct or indirect interactions. Compared with methods that impose regularization on $\mathbf{A}$, our idea not only prevents problem complexity from increasing by calculating features ahead, but also avoids making subjective assumptions on social network topology. Our features actually incorporate various kinds of such assumptions, in complementary or in contradictory, and the coefficient estimation process validates assumptions in consistent with the specific social network we observe. For instance, in a sparse network, features recording direct interactions between individuals are more likely to weight higher than features reflecting indirect interactions, since the former ones are more rare than the latter ones; 3) our designed time-varying features are capable of describing the change of infectivity wrt. time. By calculating the value of these features for each individual-pair at each event-timestamp prior to model learning, we avoid increasing problem complexity.

We introduce a set of time-varying features that imply the instant self-properties of each individual, or the instant relationship between each pair of individuals. Replacing α's with linear combinations of time-varying features, we raise the problem of optimizing the corresponding coefficients with lasso regularization on them, and solve the problem efficiently by developing an algorithm that combines the idea of alternating direction method of multipliers (ADMM) [45] and Majorize-Minimization (MM) [111].

1.2.2 Energy Usage Behavior Modeling in Energy Disaggregation

The efficient solution introduced in the previous part generally models how the occurrence of one event influences the occurrences of future events. However, in some behavioral data, besides the event occurrence, we must also pay attention to the marks of events, which are detailed descriptions of the corresponding events other than the timestamps of their occurrences. Thus, instead of modeling the influence between event occurrences, we need to model how the occurrence and the marks of an event together influence the occurrences and the marks of future events. One typical behavioral data is the usage of electronic appliances of household members, where both the temporal information and the amount of consumed energy of appliance usage play an important role in solving energy disaggregation tasks.

Energy conservation has become a critical issue in modern society and data analysis methodology has recently been applied to the analysis of energy consumption patterns in households. Several prior studies [67, 183, 250] have shown that consumers, i.e., household members are more likely to conserve their energy usage when provided with breakdown energy consumption records. However, such fine-grained energy consumption data is not readily available, since it requires numerous additional meters installed on individual appliances. Therefore there has been much interest in

the data analysis problem of energy disaggregation — the task of taking a whole-house energy signal and separating it into its component appliances. One powerful cue for breaking down the entire household's energy consumption is user behavior in energy usage [26], which is known to be a major factor in determining the energy consumption in households. Such *energy usage behavior*s can include: how users perform their daily routines, how they share the usage of appliances, and users' habits in using certain types of appliances. Understanding such energy usage behaviors will significantly increase the accuracy of estimating the usage time of each appliance, which consequently benefits the energy disaggregation task.

Despite of the importance of energy usage behaviors, they have not received enough attention in the recent literature, especially how a user's current energy usage behavior influences his/her or other people's future usage behavior. Modeling such influence is important due to the following two reasons: 1) energy usage behaviors rarely depend on the current time slots only. One's energy usage behaviors in the previous time slots also exert a significant impact. For instance, a user's usage time of washing machine can be different from day to day, but his/her sequential behaviors in clothes washing are always similar: first using the washer, and then the dryer. 2) under many circumstances, a user's behavior is not just determined by himself/herself, but influenced by other members in the same household. For example, when parents wake up earlier than usual in the morning, they may also wake up their children earlier than usual. Another instance is that two household members are not able to use the bathroom at the same time, and consequently one member has to postpone his/her usage of the bathroom. Thus, to understand energy usage behaviors, appropriate modeling of influence among the energy usage behaviors of different users in the same household across different time slots is essential.

Unfortunately, the influence between energy usage behaviors is hard to model directly, since the state-of-the-art smart-grid data rarely records the number of household members, and the exact timestamp when a certain member uses a certain appliance. Since the energy consumption of each appliance relies on the user behavior, we turn to modeling the relationship between the energy usages of different appliances across different time slots, and expect that such relationship will be able to reveal the influence between the energy usage behaviors of different users in the same household. We want to emphasize that such relationship has so far been largely ignored by existing works on energy disaggregation [135, 136, 194]. Those works mostly focus on the distribution of energy consumption of each appliance alone. They either learned the energy usage patterns of each appliance within a certain period (for instance, a week), or studied the influence between energy usage patterns from one time slot to the next. Recent works discussed the dependency between appliances in the same time slot only [124]. More importantly, our method, while modeling the influence between energy usage patterns, also pays attention to the relationship between the energy usages of different appliances across different time slots.

One main challenge in modeling the influence among various appliances across different time slots is how to model the influence between *marked* events, which is defined to be events with marks that contain detailed information of the corresponding event. Under our scenario of energy disaggregation, an event is the usage of a certain appliance in a certain time slot, while its mark is the amount of consumed energy. Although many recent works utilize Hawkes processes, a class of self- or mutually exciting point processes, to model the influence between events, most of them are only able to model the occurred timestamps of events, not the corresponding marks. Meanwhile, many existing works utilize marked point processes to model the occurrence of marked events. However, most of them are unable to capture the self- or

mutually exciting property in event occurrence. To this end, we propose a novel probabilistic model named marked Hawkes process (M-Hawkes) based on the combination of multivariate Hawkes processes and topic models. This M-Hawkes is designed to model how the occurrence and the mark of an event *together* influence the occurrence and the mark of subsequent events in the near future. In the proposed M-Hawkes model, the topic model part models the distribution of marks of observed events, and is designed to find user behavior patterns underlying the amount of consumed energy of each appliance in each time slot, while the Hawkes process part models the occurrences of observed events, and captures the influence between different appliances under different energy usage behavior patterns across different time slots.

1.2.3 Identifying and Labeling Search Tasks

Besides infectivity learning, another challenge in influence modeling is the justification of the influence existence between events. Existing studies generally address this challenge by making assumptions on the scope of influence existence. For instance, a normal univariate Hawkes process assumes that influence exists among events from the same dimension, while a normal multivariate Hawkes process assumes that influence exists between all pairs of events. Unfortunately, for the influence in certain types of behavioral data, we find it difficult to propose appropriate assumptions on influence existence. For instance, the submissions of two queries by the same search engine user can be completely irrelevant, and our prior knowledge of information retrieval cannot foretell us whether the submission of one query is motivated by the submission of another query without analyzing query logs. Such scenarios require us to explore solutions that learn the infectivity and influence existence simultaneously. We address this challenge in the task of modeling the influence among the query submissions of search engine users. The learned influence is expected to link queries that serve the same information need, i.e., belong to the same search task.

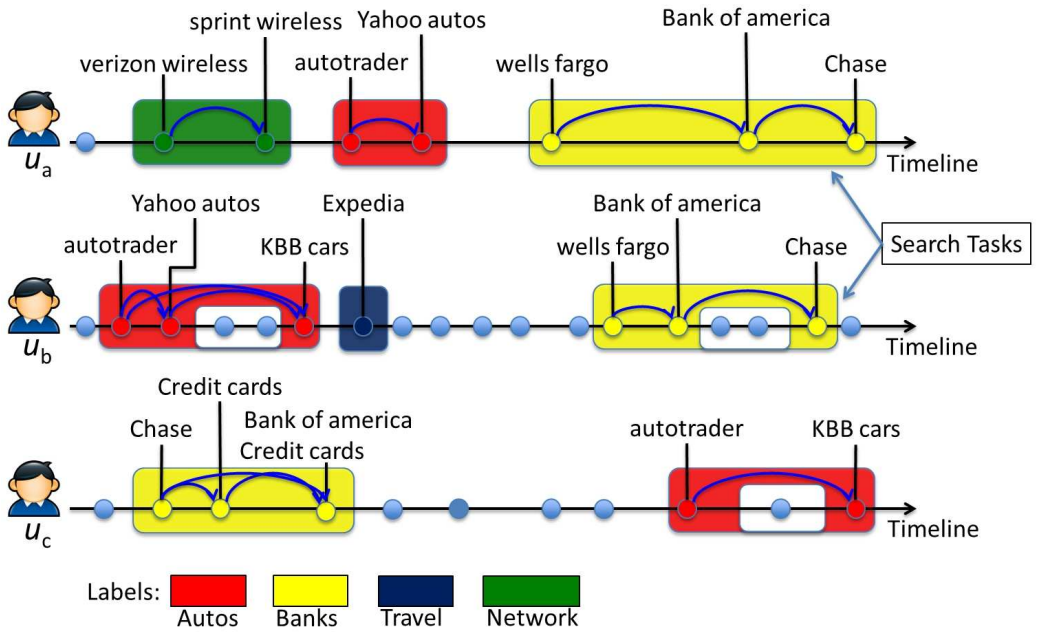


Figure 1: An Illustration of Relationship between Consecutive Queries and Search Tasks. Every circle represents a query issued by a user at time t_n. The blue arrow line indicates an influence exists between queries. A set of queries linked by blue lines denotes a search task, and some topically coherent search tasks across three users are labeled by different colors.

Nowadays, search engines have become the most important and indispensable Web portal, whereby people pursue a wide range of searches in order to satisfy a variety of *information needs*. One challenge in understanding users' information needs and search behaviors is that, the query sequence issued by a user may contain queries with multiple intents, or consist of seeking information on single or multiple topics [225]. Thus we find it important to split the query sequence into *search tasks* [164, 109, 242], which is defined as a set of queries serving for the same *information need*. Taking each *information need* as one meme, search task identification is actually the problem of detecting diffusion paths of memes under the scenario of information search.

Generally, two consecutive queries issued by a user are more likely to belong to the same search task than two non-contiguous queries, but that is not necessary always the case. It makes more sense to take into account the explicit temporal information of query sequences exhibited by many different users in the whole query logs. *The basic intuition is that if two consecutive or temporally-close queries are issued many*

times by the same user or many others users, it is more likely these two queries are semantically related to each other, i.e., belong to the same search task. Moreover, different users may engage in different search patterns, which should be treated differently based on their search activities. All in all, we choose to identify search tasks by leveraging the temporally weighted query co-occurrence — this not only guarantees sound performance by making full use of both textual and temporal information of the entire query sequences, but it also enables the labeling of the identified search tasks since semantically related queries are clustered together through query links determined by co-occurrence.

To model temporally close query co-occurrences, we choose to extend Latent Dirichlet Allocation (LDA) [41], one powerful graphical model that exploits word co-occurrence to make full use of temporal information by weighing the reliability of each co-occurrence of a pair of queries based on how likely an *influence* exists between this pair of queries. Here we define query *influence* as:

- The occurrence of one query raises the probability that the other query will be issued in the near future.

Influence, rather than closeness, enables us to distinguish temporally close query co-occurrence from temporally regular query co-occurrence for each user based on one's own frequency of query submission. To model such personal frequency and *influence*, we utilize Hawkes processes [98], a special class of point processes, to fully utilize temporal information in query sequences to identify user-specific temporally close queries. However, existing Hawkes models [147, 262] find it intractable to obtain an optimal solution of *influence* existence based on temporal information only. Moreover, it is unable to directly identify search tasks by either generating topics based on query co-occurrence using LDA, or estimating all *influence* candidates by Hawkes. To address the above issues, we concentrate on the *influence* existence between semantically related queries, whose estimation can be simplified by the joint efforts of

LDA and Hawkes, and enables a direct identification of search tasks.

According to the above intuition, a search task can be viewed as a sequence of semantically related queries linked by *influence*. A query that does not satisfy user's *information need* will *self-excite* the submission of another semantically related query in the near future. Furthermore, no such semantic *influence* exists between queries from different search tasks. In reality, a query rarely excites the submission of another semantically unrelated query even their timestamps are very close. Thus we believes that those semantic *influence* are the *influence* that actually take effect, and we solve search task identification directly by identifying those *influence*. To limit the solution space of such *influence*, we cast both *influence* existence and query-topic membership into latent variables, and equalize the existence probability of pairwise *influence* with the similarity of the memberships of associated two queries. This equalization works as a bridge between LDA and Hawkes processes, as LDA assigns high *influence*-qualified co-occurred queries to the same topic, while query co-occurrence frequency narrows the solution space of *influence*. In this way, LDA and Hawkes mutually benefit each other in identifying search tasks using both temporal and textual information. We propose a probabilistic model that incorporates this equalization to combine the LDA model with Hawkes processes, and develop a mean-field variational inference algorithm to estimate the *influence* by optimizing the data likelihood.

1.2.4 Analyzing User's Sequential Behavior in Query Auto-Completion

Besides self- & mutually exciting influence explored in previous parts, the influence in behavioral data can also be of other types. For instance, the state of an individual on the current event influences his/her state choice on the next event only. One good scenario is user's sequential behavior in the procedure of query auto-completion, where users enter characters in the search box of a search engine until they click the search button or a suggested query by the search engine. For the events of typing, viewing,

and clicking in the procedure of query auto-completion, we no longer care about when the next event will occur since 1) the temporal gaps between those events are generally very small, and 2) the variance of those temporal gaps is also small. Instead, we find it critical to figure out the type of a future event, i.e., whether it is typing, viewing, or clicking, in solving the query auto-completion tasks. Thus the influence we are to model is how the type (or state) of the current event affects the types (or states) of the future events, instead of how the occurrence of the current event affects the occurrences of future events. In specific, we believe a user's interaction with search engines at the current keystroke will influence his/her interaction at the next keystroke.

Query auto-completion (QAC) has been widely used in modern search engines to reduce users' effort to submit a query by predicting the users' intended queries. The QAC engine generally offers a list of suggested queries that start with a user's input as a prefix, and the list of suggestions is changed to match the updated input after the user types each character. Let us suppose that a user is going to submit a query q to the search engine, and the user types the prefix of the query q of length i as $q[1..i]$ sequentially. The QAC engine will return the corresponding suggestion list after the user types each character, until the user clicks the suggestion q from the list or presses return, ending the interaction with the QAC engine. Usually, even for submitting the same query q, different users may have different interactions with the QAC engine, which are shown from their different sequential behaviors. For example, user u_a chooses the suggestion q at position 5 after typing 3 characters, while user u_b chooses the suggestion q at position 1 after typing 5 characters. In order to better improve users' search experience, it becomes increasingly important to analyze users' sequential behavior with the QAC engine, to understand users' real preferences and then improve the performance of QAC.

Recently, many studies have been proposed to address the QAC problem in

different perspectives, including designing more efficient indexes and algorithms [32, 248, 107], leveraging context in long term and short term query history [27], investigating the time-sensitive aspect of QAC [218, 249], learning to combine more personalized signals [217], etc. Despite of those numerous works on QAC, most of them only utilize the information of submitted queries and associated prefixes, thus lose details of how users' interact with the QAC engine, such as the suggested query lists of each prefix before query submission, users' query typing speed, and so on. Recently, a high-resolution QAC dataset was collected from PC (personal computer) and mobile phones [156], where each keystroke of users and clicks were recorded. A two-dimensional click model was trained on this high-resolution QAC dataset, revealing users' behaviors such as horizontal skipping bias and vertical position bias. However, this work assumed that users' behaviors at different keystrokes are independent in order to simplify the model estimation, which results in information loss.

Our work, on the other hand, attempts to capture three types of relationship between users' behaviors at different keystrokes that are ignored or failed to be modeled until now: 1) **State transitions between skipping and viewing.** The study on high-resolution query log data revealed that a user may choose to either view or skip the suggestion list at each keystroke in a QAC session. It already explored how users' interactions with QAC engine at the current keystroke, such as typing speed and whether the end of current prefix is at word boundary, influence users' decisions on skipping or viewing. However, besides those factors, we believe that such decisions should also be influenced by their decisions on skipping or viewing at the previous keystroke. For instance, imagine a user u has 5 sequential skipping moves in one QAC session and 2 sequential skipping moves in another QAC session, the chance becomes higher for the same user to stop and view the suggestion list at the current keystroke after 5 sequential skipping moves. On the other hands, if the same user has already viewed too many keystrokes continuously but finds no intended query, it

becomes more likely that he/she may skip the next one; 2) **Users' real preference of suggestions**. For each keystroke, the associated users' real preference is hard to be detected from the current suggested query list alone. On the other hand, we need to utilize the rankings of suggested query lists of latter keystrokes together with users' final click choices to re-rank the suggested queries in the list of the current keystroke. Intuitively, a clicked query, i.e. the user's intended query, should get a higher rank not only at the keystroke he/she makes the click, but also at previous keystrokes where this query appears, despite that it is not clicked at that time; and 3) **User-specific cost between position clicking and typing**. Some users prefer typing than viewing and clicking, while others don't. Consequently, users' click choices are not only affected by their intent, but also by the position where the intended query is shown, and their preference of clicking that position over typing the remaining keystrokes. For instance, a user that prefers clicking will probably click an intended query the first time it is shown to him/her, despite that it may be shown in a low position; while another user focuses on typing his/her intended query despite that the query already appears in the suggestion list, until it is ranked at the top position, or even worse, he/she will type the entire query manually without any intent to click the suggestions.

To model these three aspects, we propose a probabilistic model, which is a combination of three parts that address each separately. The hidden Markov model part takes the skipping and viewing choices as two different states, and assumes the transition between keystrokes is influenced by users' interactions with the QAC engine at that keystroke. The logistic regression part weighs a set of our designed user-specific relevance features that imply users' own preference on each prefix-query pair, which is expected to capture users' real preference. The Dirichlet prior part estimates the ratio between position-biased clicking and typing costs. Those three parts together determine the probability that a user clicks a certain suggested query located at a

certain position of the suggested query list of a certain keystroke in a QAC session. We develop a mean-field variational inference algorithm to learn the parameters that optimize the data likelihood.

1.2.5 Exploring QAC and Click log for Contextual-Aware Web Search and Query Suggestion

Influence not only exists in the behavioral data of a single type, but also can exist between two types of behavioral data, as song as those types of behaviors are related. Appropriate modeling of such influence can be very useful, as it enable us to utilize external data resources to benefit the solving of applications, which are typically addressed by using one type of data only. Furthermore, such influence can be more straightforward when the two types of behaviors are performed by the same user alternatively. One good example is users' interactions with search engines recorded in the query auto-completion log and users' click behaviors on returned web documents of their issued queries recorded in the click log. Users' behaviors on both query auto-completion (QAC) log and click log are important. A QAC log records the detailed procedure that users enter queries into search engines, and a click log records how users behave on the returned web documents of their issued queries. In modeling such behaviors, people find it increasingly helpful to utilize contextual data, which to a large extent influences users' current behaviors in both types of logs.

Recently several studies [215, 51, 50, 229, 27] explore contextual data to enhance web search and query suggestion from different aspects. However, existing context-aware approaches on either the application of query auto-completion or query suggestion used a single type of log alone, while a critical fact is that QAC logs and click logs are closely related as they record users' sequential behaviors in query submission in search engines. For each issued query, first QAC logs record the detailed procedure that a user enters each keystroke, and then the click logs record how user clicks the web documents returned by the search engine. In other words, we can combine QAC

log and click log according to temporal timestamps. Each query session is a combined log that starts with a QAC session and ends with a click session. Such contextual data not only come from the same type of behavior a user conducts in the recent history, but also include the other types of behaviors a user conducts in the recent history. For instance, the next query a user will submit is both determined by the manner the user entered the previous query, and the webpages the user clicked after issuing that query. Previous QAC logs only contain the query list suggested to a user after he/she finished typing, which offers little extra information than that contained in click logs. Recently, high definition QAC logs are recorded [156], which contains the suggested query lists at each keystrokes and associated users' interactions with a QAC engine. Thus those logs offer much more additional information for query prediction.

Our goal is to effectively utilize the contextual data to model user behavior. The key idea is to cluster users' behaviors on QAC logs and click logs into several patterns, separately, and investigate the correlation between users' behavior patterns on QAC logs and users' behavior patterns on click logs. We believe such correlation does exist, as users' behaviors on searching are usually consistent, which originate from users' search habits, preferences, interests, or instant circumstance. A user's QAC (or click) behavior pattern that implies a certain habit, preference, interest, or circumstance will probably be followed by a click (or QAC) behavior pattern that implies the same habit, preference, interest, or circumstance. For instance, if the QAC log records that a user types a query very fast, it is very likely that the user is very familiar with the query, then in the click log, the user may spend a large amount of time on viewing the returned web documents. If the click log records that a user clicks several returned web documents, and spends much time on it, he/she is probably very familiar with the current topic that he/she searches, then in the next query session of the QAC log, if he/she chooses to search queries under the same topic, he/she can type them very

fast. Based on the learned correlation, given the inferred user's behavior pattern on one type of log, we can more accurately infer the user's following behavior pattern on the other type of log.

To capture such correlation, we propose a novel probabilistic model based on latent Dirichlet allocation (LDA). Based on the likelihood of the co-occurrences of adjacent QAC behavior patterns and click behavior patterns, the model explores the conditional distribution of consequential behavior patterns given a certain behavior pattern of the other type. A mean-field variational inference algorithm is developed to estimate the membership of behavior patterns on two types of logs in each session.

1.3 Outline

The rest of the dissertation is organized as follows. We first introduce related work in Chapter 2. After that we propose a novel multi-dimensional Hawkes model that parameterizes pairwise infectivity using linear combinations of time-varying features in Chapter 3. In Chapter 4, we formulate the task of energy disaggregation into the modeling of marked event sequences, and introduce a probabilistic model that integrates topic models with Hawkes processes to capture the influence from the occurrence and the mark of an event to the occurrences and the marks of future events. We then propose a probabilistic model to solve the query auto-completion (QAC) task by capturing the relationship between users' behaviors at different keystrokes in high resolution QAC logs in Chapter 5. In Chapter 6, we presented a LDA-based probabilistic model to study users' behaviors on both QAC logs and click logs simultaneously by using QAC and click logs as the contextual data of each other. Finally we conclude the thesis in Chapter 7.

CHAPTER II

BACKGROUND AND RELATED WORK

In this chapter, we review the existing work related to the thesis. In Section 2.1, we introduce self- & mutually exciting point processes, one powerful tool in statistics for modeling the influence between sequential events. Then, we survey the related work on energy disaggregation in Section 2.2. The existing studies of query auto-completion, click model, and contextual search in summarized in Section 2.3, 2.4, and 2.5, respectively. Those studies are related to our work in Chapter 5 and 6.

2.1 Self- & Mutually Exciting Point Process

We first review the concepts of point processes, and describe a special class of point processes with self- & mutually exciting properties in particular, which is widely used for the modeling of influence between events.

2.1.1 Point Process

One powerful tool in statistics for modeling event sequence data is the point process, which is widely used to describe data that are localized at a finite set of time points $\{t_1, \ldots, t_N\}$. Typically, a point process is a list of times $\{t_1, \ldots, t_N\}$ at which an N sequence of events $\{E_1, \ldots, E_N\}$ occur. An event E_i can be a retweet, a query click, a query submission, the usage of an electronic appliance, and so on, where t_i records the occurrence time of that event. A point process is said to be *simple* if the times are ordered such that $t_n < t_{n+1}$ for any $n = 1, \ldots, N$.

For a point process $\{t_i\}$, the associated counting process is defined to be the right-continuous process as $N(t) = \sum 1_{t_i \leq t}$, while the associated duration process is defined by $\forall i$, $\delta t_i = t_i - t_{i-1}$. Denote $N(t)$ the number of points (i.e., occurrences of events)

in $(-\infty, t]$ and $\mathcal{H}_t = \{E|t_E < t\}$ the *history* of events up to but not including t, the *conditional intensity function* (*hazard function*) $\lambda(t|\mathcal{H}_t)$ is the most convenient way to characterize a point process:

$$\lambda(t|\mathcal{H}_t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}[N(t + \Delta t)|\mathcal{H}_t]}{\Delta t}, \tag{1}$$

which gives the expected instantaneous rate of future events at timestamp t. Let f and F be the conditional density and the corresponding cumulative distribution for t, the intensity can be also defined by: $\lambda(t|\mathcal{H}_t) = f(t|\mathcal{H}_t)/S(t|\mathcal{H}_t)$, where $S(t|\mathcal{H}_t) = 1 - F(t|\mathcal{H}_t)$ is known as the *survival function* (the probability that an event does not happen up to t). Because of the dependence on $\mathcal{H}_t$, most point processes are *not* Markovian except for a few simple cases (for instance, Poisson processes). For clarity, hereafter we use $*$ to imply the dependence on $\mathcal{H}_t$, i.e., $\lambda(t|\mathcal{H}_t)$ will be denoted $\lambda^*(t)$.

2.1.2 Hawkes Process

The Hawkes process is a class of self or mutually exciting point process models [98]. A univariate Hawkes process $\{N_t\}$ is defined by

$$\begin{aligned}\lambda^*(t) &= \mu(t) + \int_{-\infty}^{t} \kappa(t - s) dN(s), \\ &= \mu(t) + \sum_{t_i < t} \kappa(t - t_i),\end{aligned}$$

where $\mu : \mathbb{R} \to \mathbb{R}_+$ is a deterministic base intensity (i.e. how likely an event will occur when no other event triggers it), $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ is a kernel function expressing the postive influence of past events on the current value of the intensity process. One popular κ used by existing studies is the exponential kernel, i.e., $\kappa(\Delta t) = \omega e^{-\omega \Delta t}$ if $\Delta t \geq 0$ or 0 otherwise. However, the model development and inference is independent of kernel choice and extensions to other kernels such as power-law, Rayleigh, non-parametric kernels are straightforward.

The Hawkes process is well known for its *self-exciting* property , which refers to a social phenomenon that the occurrence of one event increases the probability of

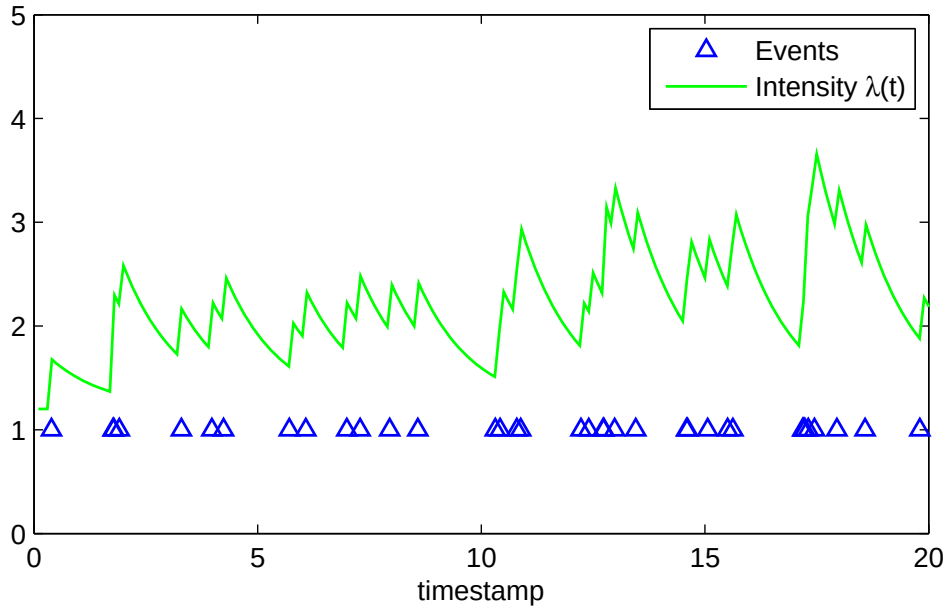


Figure 2: Illustration of a Simulation of a Univariate Hawkes Process

related events in the near future. Here related events can be events of the same type or share some common factors, for instance, participated by the same user. Such a social phenomenon is very common under many scenarios. For instance, Taliban's attacking against U.S. army can probably result in an instant retaliation. A user's submission of one query can probably increase the chance that the user issues a similar query in the near future. The usage of washing machine is likely to imply the usage of drying machine later. Figure 2 illustrates a simulation of a univariate Hawkes process.

Since the log-likelihood of a simple point process $\{t_i\}$ with intensity λ can be written as:

$$\log \mathcal{L}((N_t)_{t\in[0,T]}) = \int_0^T (1-\lambda(s))ds + \int_0^T \log \lambda(s) dN(s),$$

we can calculate the log-likelihood of a Hawkes model using exponential kernel

$\kappa(\Delta t) = \alpha \exp(-\beta(\Delta t))$ through:

$$\begin{aligned}\log \mathcal{L}(\{t_i\}) &= t_n - \Lambda(0, t_n) + \sum_{i=1}^{n} \log \lambda(t_i) \\ &= t_n - \Lambda(0, t_n) + \sum_{i=1}^{n} \log \left[\mu(t_i) + \sum_{l=1}^{i-1} \alpha \exp(-\beta(t_i - t_l)) \right].\end{aligned}$$

where

$$\begin{aligned}\Lambda(0, t_n) &= \int_0^{t_n} \lambda(s) ds \\ &= \int_0^{t_n} \mu(s) ds + \int_0^{t_n} \sum_{t_l < s} \alpha \exp(-\beta(s - t_l)) ds \\ &= \int_0^{t_n} \mu(s) ds + \sum_{i=1}^{n} \frac{\alpha}{\beta} (1 - \exp(-\beta(t_n - t_i))).\end{aligned}$$

The computation of the above log-likelihood can be simplified by calculating $\sum_{l=1}^{i-1} \exp(-\beta(t_i - t_l))$ through a recursive formula as:

$$\begin{aligned}R(i) &= \sum_{l=1}^{i-1} \exp(-\beta(t_i - t_l)) \\ &= \exp(-\beta(t_i - t_{i-1})) \sum_{l=1}^{i-1} \exp(-\beta(t_{i-1} - t_l)) \\ &= \exp(-\beta(t_i - t_{i-1})) \left(1 + \sum_{l=1}^{i-2} \exp(-\beta(t_{i-1} - t_l)) \right) \\ &= \exp(-\beta(t_i - t_{i-1}))(1 + R(i-1)).\end{aligned}$$

Thus the log-likelihood function can be calculated recursively using:

$$\log \mathcal{L}(\{t_i\}) = t_n - \Lambda(0, t_n) + \sum_{i=1}^{n} \log[\mu(t_i) + \alpha R(i)].$$

According to Ogata [186], the maximum-likelihood estimator $\hat{\theta} = (\hat{\mu}, \hat{\alpha}, \hat{\beta})$ of a stationary one-dimensional Hawkes process with constant μ owns the following properties:

- consistent, i.e. converges in probability to the true values $\theta = (\mu, \alpha, \beta)$ as $T \to \infty$: $\forall \epsilon > 0$, $\lim_{T \to \infty} P[|\hat{\theta} - \theta| > \epsilon] = 0$.

- asymptotically normal, i.e. $\sqrt{T}(\hat{\theta} - \theta) \rightarrow \mathcal{N}(0, I^{-1}(\theta))$, where $I^{-1}(\theta) = \left(E[\frac{1}{\lambda}\frac{\partial\lambda}{\partial\theta_i}\frac{\partial\lambda}{\partial\theta_j}]\right)_{i,j}$.
- asymptotically efficient, i.e., asymptotically reaches the lower bound of the variance.

2.1.3 Multi-dimensional Hawkes Process

The multivariate/multi-dimensional Hawkes process $\{N_m(t)|m = 1, \ldots, M\}$, a multi-dimensional extension to the univariate case, describes the occurrences of M coupling point series [98]. The intensity function $\lambda^* = [\lambda_1^*, \ldots, \lambda_M^*]^\top$ is defined by

$$\lambda_m^*(t) = \mu_m(t) + \sum_{m'=1}^{M} \int_{-\infty}^{t} \kappa_{m'm}(t-s) dN_{m'}(s),$$

where $\kappa_{m'm}$ is a time-decaying triggering kernel between a pair of dimensions m' and m. This process is also known as linear mutually exciting process since the occurrence of an event in one dimension increases the likelihood of future events in all dimensions. While univariate Hawkes processes focus on modeling the influence among the events from one dimension, multivariate Hawkes processes are able to model the influence among the events from different dimensions, thus suit the influence model under complex real-world scenarios. Meanwhile, we must notice that the computational complexity under the case of multivariate Hawkes processes is much greater, since the number of kernels κ we need to estimate increase from M to M^2.

Assuming we use an exponential kernel $\kappa_{m'm}(t-s) = \alpha_{m'm}\exp(-\beta_{m'm}(t-s))$, the log-likelihood of a multidimensional Hawkes process can be computed as the sum of the likelihood of each dimension as:

$$\log \mathcal{L}(\{t_i\}) = \sum_{m=1}^{M} \log \mathcal{L}^m(\{t_i\}),$$

where each term is defined as:

$$\log \mathcal{L}^m(\{t_i\}) = \int_0^T (1 - \lambda^m(s))ds + \int_0^T \log \lambda^m(s) dN^m(s)$$
$$= T - \Lambda^m(0, T) + \sum_{i:m_i=m} \log \left[\mu^m(t_i) + \sum_{t_l<t_i} \alpha_{m_l m} \exp(-\beta_{m_l m}(t_i - t_l)) \right].$$

Here m_i denotes the dimension that the l-th event belongs to.

Hawkes process has been widely used in applications, such as earthquake prediction [188], sales modeling [255, 82], Asset management [254], search behavior modeling [153], crime modeling [227], and armed conflict analysis [262, 150]. To solve such model, an EM framework is proposed to estimate the maximum likelihood of Hawkes process [147]. Additionally, Marked Poisson was used to model cascades of events in [220], while events between pairs of nodes were also modeled by Hawkes process based method [43].

2.2 Energy Disaggregation

One good example of behavioral data where self- & mutually exciting influence exists is the energy consumption of household members. Energy conservation has become a critical issue in modern society and data analysis methodology has recently been applied to the analysis of energy consumption patterns in households. Several prior studies [67, 183, 250] have shown that consumers, i.e., household members are more likely to conserve their energy usage when provided with breakdown energy consumption records. However, such fine-grained energy consumption data is not readily available, since it requires numerous additional meters installed on individual appliances. Therefore there has been much interest in the data analysis problem of energy disaggregation — the task of taking a whole-house energy signal and separating it into its component appliances. One powerful cue for breaking down the entire household's energy consumption is user behavior in energy usage[26], which is known to be a major factor in determining the energy consumption in households.

Existing works on energy disaggregation mostly focus on the distribution of energy consumption of each appliance alone[135, 136, 194]. They either learned the energy usage patterns of each appliance within a certain period (for instance, a week)[135, 136], or studied the influence between energy usage patterns from one time slot to the next[194]. Recent works discussed the dependency between appliances in the same time slot only [124]. Those methods ignored the relationship between different appliances, especially how the energy consumptions of different appliances are related across different time slots. On the other hand, our work not only models the influence between energy usage patterns of different users in the same household, but also pays attention to the relationship between the energy usages of different appliances across different time slots.

2.3 Search Task Identification

Influence with the self- & mutually exciting property also exists in the query submission of search engine users. Search query logs have been extensively studied to improve the search relevance and provide better user experience. There has been a large body of work focused on the problem of identifying search tasks or sessions from sequences of queries. Many of these methods use the idea of a "timeout" cutoff between queries, where two consecutive queries are considered as two different sessions or tasks if the time interval between them exceeds a certain threshold. Often a 30-minute timeout is used to segment sessions [51, 157, 242]. In addition, other timeout thresholds have been proposed, from 1 to 120 minutes [99, 117, 164]. However, the experimental results of these methods indicate that the timeouts, whatever their lengths, are of limited utility in predicting whether two queries belong to the same task, and unsuitable for identifying session boundaries. Beyond that, Wang et al. [242] and Hua et al. [109] treated the time intervals between queries as pairwise features in their models. But no previous work has explicitly exploited the temporal

information directly in their models. In our work, we directly integrate the temporal information into our model, rather than highly relying on different timeouts, for identifying search tasks.

There have been attempts to extract in-session tasks [225, 117, 164], and cross-session tasks [117, 139, 11, 242] from query sequences based on classification and clustering methods. Jones and Klinkner [117] proposed to learn a binary classifier to detect whether two queries belong to the same task or not, which organized and segmented query sequences into hierarchical units. Moreover, Kotov et al. [139] and Agichtein et al. [11] studied the problem of cross-session task extraction via binary same-task classification, and found different types of tasks demonstrate different life spans. Another suitable mechanism for identifying sessions or tasks may rely on *unsupervised learning* approaches, i.e., query clustering algorithms, especially when no labeled training set is available. The intuition for using query clustering is based on the assumption that if two queries belong to the same cluster, then they are topically related. Cao et al. [51] proposed a clustering algorithm for summarizing queries into concepts throughout a click-through bipartite graph built from a search log. Lucchese et al. [164] and Hua et al. [109] exploited the knowledge base for detecting semantically related query pairs that are not similar from a lexical content point of view. In addition, Wang et al. [242] proposed a semi-supervised clustering method for identifying cross-session tasks. Different from these existing methods, we assume that queries belonging to the same search task are linked by influence. Moreover, instead of focusing on the query sequence of single users, we take into account the query sequences issued by different users simultaneously in a unified framework, such that our model can identify and label coherent search tasks across users.

2.4 Query Auto-Completion

Beside self- & mutually exciting, there are also many other types of influence in behavioral data, such as the influence with the Markov property, i.e., users' behaviors on the current stage only influence their behaviors on the next stage. A good instance of such behavioral data is users' sequential interactions with search engines in the procedure of query auto-completion. The main objective of QAC is to predict users' intended queries and assist them formulate a query while typing. The most popular QAC algorithm is to suggest completions according to their past popularity. Generally, a popularity score is assigned to each query based on the frequency of the query in the query log from which the query database was built. This simple QAC algorithm is called MostPopularCompletion (MPC), which can be regarded as an approximate maximum likelihood estimator [27].

Several QAC methods [27, 218, 217, 249] were proposed to extend MPC from various aspects. Bar-Yossef and Kraus [27] introduced the context-sensitive QAC method by treating users' recent queries as context and taking into account the similarity of QAC candidates with this context for ranking. But there is no consensus of how to optimally train the relevance model. Shokouhi [217] employed learning-based strategy to incorporate several global and personal features into the QAC model. However, these methods only exploit the final submitted query or simulate the prefixes of the clicked query, which do not investigate the users' interactions with the QAC engine.

In addition the above models, there are several studies addressing different aspects of QAC. For example, [218, 249] focused on the time-sensitive aspect of QAC. Other methods studied the space efficiency of index for QAC [32, 107]. Duan and Hsu [77] addressed the problem of suggesting query completions when the prefix is mis-spelled. Kharitonov et al. [122] proposed two new metrics for offline QAC evaluation, and [115] investigated user reformation behavior for QAC.

The QAC is a complex process where a user goes through a series of interactions

with the QAC engine before clicking on a suggestion. As can be seen from the related work, little attention has been paid to understand the interactions with the QAC engine. Until recently, Li et al. [156] created a two-dimensional click model to combine users' behaviors with the existing learning-based QAC model. The study assumed users' behaviors at different keystrokes, even for the consecutive two keystrokes, are independent in order to simplify the model estimation, which results in information lose. Different from those works, we attempt to directly model and leverage the relationship between users' behaviors, so as to improve the performance of QAC.

2.5 Click Models

Influence not only exists among user behaviors of the same type, but also among related user behaviors of different category, such as the QAC behavior mentioned in the previous section and users' click behaviors on the returned web documents of their issued queries. Thus, in this section, we first review existing click models that study click behaviors, and survey studies on contextual search, which is related to our strategy used in exploring the mutually influence between QAC and click behaviors. In the field of document retrieval, the main purpose for modeling users' clicks is to infer the intrinsic relevance between the query and document by explaining the positional bias. The position bias assumption was first introduced by Granka et al. [93], stating that a document on higher rank tends to attract more clicks. Richardson et al. [204] attempted to model the true relevance of documents by imposing a multiplicative factor. Later examination hypothesis is formalized in [60], with a key assumption (Cascade Assumption) that a user will click on a document if and only if that document has been examined and it is relevant to the query. In addition, several extensions were proposed, such as the User Browsing Model (UBM) [78], the Bayesian Browsing Model [160], the General Click Model [272], and the Dynamic Bayesian Network model (DBN) [54]. Despite the abundance of click models, these

existing click models cannot be directly applied to QAC without considerable modification. The click model most similar to our work is [267], which models users' clicks on a series of queries in a session. However because of the main difference between QAC and document retrieval, our model is very different from [267].

Contextual search is heavily researched in literature and is explored from different angles. A large portion of a recent comprehensive survey on contextual search is devoted to the study of personal interest from interaction, content, social, and geographical variables [171]. Traditional personalization approaches [219] usually build a profile of interests for each user from her/his search or browsing history. Contextual information is useful in identifying users' search needs. Shen et al. [215] presented context-aware language models by assuming that documents are not only similar to the current query but also similar to the previous queries and the summaries of the documents clicked on. Sun and Lou [229] focus on right-click query that is submitted to a search engine by making a text string in a Web page, and extract the contextual information from the source document to improve search results. Cao et al. [51, 50] extracted context information in Web search sessions by modeling search sessions as sequences of user queries and clicks. They learned sequential prediction models such Hidden Markov Model from search log data. Different from our study here, their models were designed for predicting search intents based on context information from one type of log only, but not leveraging both QAC and clickthrough logs.

CHAPTER III

LEARNING PARAMETRIC MODELS FOR SOCIAL INFECTIVITY IN MULTI-DIMENSIONAL HAWKES PROCESSES

The first and the most fundamental problem we consider in this thesis is how to efficiently and effectively model the influence in behavioral data. In this chapter, we come up with a general solution that efficiently models the influence under various behavioral data, without prior knowledge about the topologies of the network of the participated individuals. Such a general solution can be adapted to appropriately handle a wide range of practical problems which demand the modeling of influence between the behaviors of individuals. One fundamental problem in influence modeling is the learning of the degree of influence between individuals in social networks, which we called social infectivity. Efficient and effective learning of social infectivity is a critical challenge in modeling diffusion phenomenon in social networks and other applications. Existing methods require substantial amount of event cascades to guarantee the learning accuracy, while only time-invariant infectivity is considered.

In this chapter, we overcome those two drawbacks by constructing a more compact model and parameterizing the infectivity using time-varying features, thus dramatically reduce the data requirement, and enable the learning of time-varying infectivity which also takes into account the underlying network topology. We replace the pairwise infectivity in the multidimensional Hawkes processes with linear combinations of those time-varying features, and optimize the associated coefficients with lasso regularization on coefficients. To efficiently solve the resulting optimization problem, we employ the technique of alternating direction method of multipliers, and under that

framework update each coefficient independently, by optimizing a surrogate function which upper-bounds the original objective function. On both synthetic and real world data, the proposed method performs better than alternatives in terms of both recovering the hidden diffusion network and predicting the occurrence time of social events.

3.1 Parametric Models for Social Infectivity

3.1.1 Multi-dimensional Hawkes Process

As introduced in the previous chapter, a multi-dimensional Hawkes process estimates basic intensity μ and infectivity α by maximizing the likelihood on each observed event cascade $\{t_n, m_n\}_{n=1}^N$ as:

$$\mathcal{L} = \sum_{n=1}^{N} \log \lambda_{m_n}(t_n) - \sum_{m=1}^{M} \int_0^T \lambda_m(s) ds$$

where t_n is the timestamp of the n-th event in the cascade, and m_n indicates the dimension/individual where the n-th event occurs.

In real world social networks, M can be very large, dependency exists among α's, and α may varies with respect to time. Thus learning one separate α for each pair of dimensions (m, m') can be both inefficient and ineffective. To address those issues, instead, we decompose each α into a linear combination of K time-varying features as:

$$\alpha_{m,m'} = \beta^T \mathbf{x}_{m,m'}(t), \tag{2}$$

where β is the vector of coefficients that we are to learn instead of α. $\mathbf{x}_{m,m'}(t)$ is a time-varying dyad-dependent vector of length K, which is supposed to reflect some kinds of relationship between dimension m and m'.

Plugging Eqn (2) into the intensity function of multi-dimensional Hawkes processes, we can write the log-likelihood of model parameters μ, β as:

$$\mathcal{L}(\mu,\beta) = \sum_{n=1}^{N} \log\left(\mu_{m_n} + \beta \sum_{l=1}^{n-1} \kappa(t_n - t_l)\mathbf{x}_{m_l,m_n}(t_n)\right) - T\sum_{m=1}^{M}\mu_m$$
$$-\beta^T \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{l=1}^{n-1} \mathbf{x}_{m_l,m_n}(t_n)(K(t_n - t_l) - K(t_{n-1} - t_l))$$

where $K(t) = \int_0^t \kappa(s)ds$.

To select effective features and avoid overfitting, we enforce the sparsity of coefficients β by imposing lasso type of regularization as $\|\beta\|_1$. Under this lasso regularization, β_k will be non-zero only when its corresponding feature is highly correlated with the infectivity between two dimensions; otherwise, β_k will be enforced to be zero. In summary, we are to optimize model parameters μ, β as:

$$\min_{\mu\geq 0,\beta\geq 0} -\mathcal{L}(\mu,\beta) + \lambda\|\beta\|_1 \tag{3}$$

where λ is the regularization parameter that trades off the sparsity of the coefficients and the data likelihood.

3.2 Optimization

Optimizing β against $\mathcal{L}$ is relatively difficult, since the non-smooth regularizer on β makes the objective function non-differentiable. To optimize such an objective, we employ alternating direction method of multipliers (ADMM) [45] to reduce this ℓ_1 regularized loss minimization problem to a sequence of ℓ_2 regularized loss minimization problems, which are much easier to solve. ADMM is known as a special case of the more general Douglas-Rachford splitting method, which has good convergence properties under some fairly mild conditions [80].

3.2.1 Derivation of ADMM

In ADMM, the optimization problem in Eqn (3) can be re-written to the following equivalent form by introducing an auxiliary variable $\mathbf{z}$:

$$\begin{aligned} &\min_{\mu\geq 0,\beta\geq 0,\mathbf{z}} -\mathcal{L}(\mu,\beta)+\lambda\|\mathbf{z}\|_1, \\ &\text{subject to} \quad \beta=\mathbf{z}. \end{aligned}$$

The corresponding augmented Lagrangian of the problem is:

$$\mathcal{L}_\rho = -\mathcal{L}(\mu,\beta)+\lambda\|\mathbf{z}\|_1+\rho\mathbf{u}(\beta-\mathbf{z})+\frac{\rho}{2}\|\beta-\mathbf{z}\|_2^2,$$

where $\mathbf{u}$ is the scaled dual variables corresponding to the constraint $\beta=\mathbf{z}$, and ρ is the penalty parameter, which is usually used as the step size in updating the dual variable.

Then we solve the above augmented Lagrangian using the ADMM algorithm consisting of the following iterative steps:

$$\begin{aligned} &\mu^{i+1},\beta^{i+1} = \text{argmin}_{\mu\geq 0,\beta\geq 0} -\mathcal{L}_\rho(\mu,\beta,\mathbf{z}^i,\mathbf{u}^i), \\ &\mathbf{z}^{i+1} = S_{\lambda/\rho}(\beta^{i+1}+\mathbf{u}^i), \\ &\mathbf{u}^{i+1} = \mathbf{u}^i+\beta^{i+1}-\mathbf{z}^{i+1}. \end{aligned}$$

where S_κ is the soft thresholding operator [76]. We will derive the algorithm for optimizing μ and β in the following, which is a proximal operator evaluation.

3.2.2 Estimation of μ and β

In order to update each μ and β independently, we choose to optimize a surrogate function which breaks down the log-sum of $\log\lambda_{m_n}(t_n)$ based on Jensen's inequality, and upper-bounds of $-\mathcal{L}_\rho(\mu,\beta,\mathbf{z}^i,\mathbf{u}^i)$. By optimizing this surrogate function in the Majorize-Minimization (MM) algorithm [111], we can reach the gloabl optimum of

$-\mathcal{L}_\rho$. We define the surrogate function as:

$$
\begin{aligned}
g(\mu, \beta | \mu^{(j)}, \beta^{(j)}) = \rho \mathbf{u}^i(\beta - \mathbf{z}^i) + \frac{\rho}{2}\|\beta - \mathbf{z}^i\|_2^2 - \sum_{n=1}^{N} \eta_{n0} \log\left(\frac{\mu_{m_n}}{\eta_{n0}}\right) \\
+ \sum_{n=1}^{N}\sum_{k=1}^{K} \eta_{nk} \log\left(\frac{\beta_k \sum_{l=1}^{n-1} x_{m_l,m,k}(t_n)\kappa(t_n - t_l)}{\eta_{nk}}\right) \\
- \beta^T \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{l=1}^{n-1} \mathbf{x}_{m_l,m_n}(t_n)(K(t_n - t_l) - K(t_{n-1} - t_l)),
\end{aligned}
$$

where η is a set of branching variables formulated by:

$$
\begin{aligned}
\eta_{n0} &= \frac{\mu_{m_n}^{(j)}}{\mu_{m_n}^{(j)} + \sum_{k=1}^{K} \beta_k^{(j)} \sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)\kappa(t_n - t_l)}, \\
\eta_{nk} &= \frac{\beta_k^{(j)} \sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)\kappa(t_n - t_l)}{\mu_{m_n}^{(j)} + \sum_{k=1}^{K} \beta_k^{(j)} \sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)\kappa(t_n - t_l)}.
\end{aligned}
$$

Notice here we can interpret η_{n0} as the infectivity of all historical events on the n-th event with regard to the k-the feature, while η_{n0} is the probability that the n-th event is sampled from the base intensity.

As proved in [270], optimizing the surrogate function g ensures that $\mathcal{L}_\rho$ decreases monotonically, thus guarantees that $\mathcal{L}_\rho$ will converge to a global optimum. Then by optimizing g, we are able to update μ and β independently with closed-form solutions, and automatically take care of the non-negativity constraints as follows:

$$
\mu_m = \frac{1}{T}\sum_{n:m_n=m} \eta_{n0}, \quad \beta_k = \frac{1}{2\rho}\left(-b_k + \sqrt{b_k^2 + 4\rho \sum_{n=1}^{N} \eta_{nk}}\right),
$$

where

$$
b_k = \sum_{n=1}^{N}\sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)(K(t_n - t_l) - K(t_{n-1} - t_l)) + \rho(u_k^i - z_k^i).
$$

3.2.3 Complexity Analysis.

The majority of our computation lies in the estimation of μ and β, where we need to calculate a vector of η for each event n. Since feature-related computations such as $\sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)\kappa(t_n - t_l)$ and $\sum_{n=1}^{N}\sum_{l=1}^{n-1} x_{m_l,m_n,k}(t_n)(K(t_n - t_l) - K(t_{n-1} - t_l))$ can

be done ahead, this estimation procedure has a computational cost of $O(N * K + M)$ only. The updates of $\mathbf{z}$ and $\mathbf{u}$ in each iteration only cost $O(K)$. Thus, our algorithm costs $O(N * K + M)$ in total, where $K \ll N$ can be ensured by controlling by the number of features we use. Thus we can view the computational cost as linear in the number of events and the number of individuals, such cost is much smaller compared with multi-dimensional Hawkes models that estimate pairwise infectivity directly, which cost at least $O(N^2 + M^2)$.

3.3 Time-varying Features

Time-varying features [230] attract ever increasing attentions in analyzing temporal data, such as email communication [196], seismic events [52], and Heart Rate Variability (HRV) signals [172]. In a given social network where memes diffuse, we collect both *individual features*, which imply the instant self-properties of each individual, and *dyadic features*, which imply the instant relationship between each pair of individuals. These features count the number of appearances of a certain pattern involving one individual or one individual-pair in a certain time range formulated as:

$$x(p)(t, \Delta t) = \#\{p \in [t - \Delta t, t)\},$$

where p represents a certain defined pattern, $[t - \Delta t, t)$ is the time interval from some ancient timestamp to the current timestamp. Table 7 shows several patterns we adopt in this work. Our feature design is inspired by the features proposed in [196]. The novelty of our design is that we propose features in more general forms, and also explore brand-new patterns in networks, thus produce far more features.

As shown in Table 7, our features generally originate from individuals' involvement in the diffusion paths of memes in networks, and reflect implicit individual property or pairwise relationship. These features can be either categorized by the number of individuals involved, or by the path length. If provided with explicit self-properties of individuals or the relationship between individuals, we can propose new features

Table 1: Patterns in Constructing Time-varying Features

Pattern p	Description
i	node i appears on one diffusion path.
dist(i)	node i appears on one diffusion path of a certain meme(the appearance on the path of the same meme will not be counted twice).
$in(i)$	node i gets infected from another node (the appearance of the same node will not be counted twice).
$out(i)$	node i infects another node (the appearance of the same node will not be counted twice).
$i \circlearrowleft^{(v)}$	there exists a length-v diffusion path from node i to itself.
pure($i \circlearrowleft^{(v)}$)	there exists a length-v diffusion path from node i to itself, and there exists one meme that diffuses on the entire path (Similar patterns are designed for all dyad-dependent patterns below).
$i \xrightarrow{(v)} j$	$v-1$ intermediate nodes exist on the diffusion path from node i to j.
$j \xleftarrow{(v)} i$	$v-1$ intermediate nodes exist on the diffusion path from node j to i.
$i \xleftrightarrow{(v,v')} j$	there exists a node h that is the ancestor of both node i and j, and the corresponding path length is v and v', respectively.
$i \xrightarrow{(v,v')}\!\!\xleftarrow{} j$	there exists a node h that is the descendant of both node i and j, and the corresponding path length is v and v', respectively.

To facilitate the description of each pattern, we take a social network as a graph, and each individual as a node, and the paths that memes diffuse as directed edges.

accordingly. Based on above collected features, we are able to form a feature vector $\mathbf{x}_{m,m'}(t)$ for each individual-pair (m, m') at any given timestamp t through:

$$\mathbf{x}_{m,m'}(t) = \{x(p)(t, \Delta t) | p \in \mathcal{P}_{m,m'}, \Delta t > 0\},$$

where $\mathcal{P}_{m,m'}$ refers to the set of patterns involving at least one individual among $\{m, m'\}$.

3.4 Experiments

We conducted experiments on both synthetic and real-world data sets , and compared the performance of our model with alternatives to demonstrate the effectiveness of our model.

Synthetic Data Set. We sample the synthetic data according to the proposed model in the following manner: Given model dimensions (M, N, K), we start by drawing the basic intensity vector μ of size M, and the coefficient vector β of size K. Each element μ_m and β_k is randomly generated in $[0.5\hat{\mu}, 1.5\hat{\mu}]$ and $[0, 2\hat{\beta}]$ respectively before simulation. Then we randomly draw a fixed feature vector $\mathbf{x}_{m,m'}$ for each pair of dimensions m and m', and finally sample event cascades from the proposed model specified by μ, β, and $\mathbf{x}$. We also generate the ground-truth infectivity matrix $\hat{\mathbf{A}}$ based on the ground-truth β and $\mathbf{x}$. Our synthetic data are simulated with two different settings:

- `Small`: M=100, N=1,200, K=10, $\hat{\mu}$=0.01, $\hat{\beta}$=0.05. Simulations were run 100 times.
- `Large`: M=1,000, N=50,000, K=100, $\hat{\mu}$=0.01, $\hat{\beta}$=0.005. Simulations were run 5 times.

We sample 100 cascades to ensure that normal multi-dimensional Hawkes models can obtain promising results, which our model doesn't necessarily need as shown in experiments. To test the how the lasso regularization works, we generate `Sparse Synthetic` data with a sparse β by randomly selecting 80% elements in the vector β to be 0.We also generate `Time-varying Synthetic` data with time-varying feature vectors. For each timestamp in a event cascade, we calculate a separate $\mathbf{x}_{m,m'}$ based on the generative process of the proposed time-varying features in the network, thus ensure $\mathbf{x}_{m,m'}$ to be time-varying.

Evaluation metrics. We consider the following evaluation metrics: 1) first, we compare the average log probability on the training data, and the average log predictive

likelihood on events falling in the final 10% of the total time of each event cascade; 2) next we compare the average relative distance between the estimated parameters and ground-truth ones by $\frac{1}{K}\sum_k |\frac{\beta_k - \hat{\beta}_k}{\beta_k}|$ and $\frac{1}{M}\sum_m |\frac{\mu_m - \hat{\mu}_m}{\mu_m}|$, and evaluate the learned infectivity α by $\frac{1}{M(M-1)}\sum_{ij:i\neq j} |\frac{\alpha_{ij} - \hat{\alpha}_{ij}}{\alpha_{ij}}|$. We classify these three metrics for parameter estimation into the class of Mean Absolute Error (MAE). 3) we also employ the metric RankCorr [270], which is defined as the averaged Kendall's rank correlation coefficient between each row of $\mathbf{A}$ and $\hat{\mathbf{A}}$. It measures whether the relative order of the estimated social infectivities is correctly recovered or not.

Baselines. To demonstrate the effectiveness of the proposed model, we compare it with the following alternatives:

`Multi-Hawkes:` This is a normal multi-dimensional Hawkes model with no regularizer on $\mathbf{A}$.

`Cox:` This is a multiplicative Cox model that parameterizes intensity. Our experiments learn this model using the same feature set as our proposed model. Note that Cox has no parameter μ [196].

`LowRankSparse:` This is a multi-dimensional Hawkes model with the infectivity matrix $\mathbf{A}$ regularized by both nuclear norm and ℓ_1 norm [270].

`NetRate:` This is a continuous time model for diffusion networks [205]. It cannot model the recurrent events, thus we only keep the first event occurrence at each individual.

`Para-Hawkes:` This is our proposed model, besides estimating μ and β, we also infer the infectivity matrix $\mathbf{A}$ accordingly for the comparison with Hawkes models which directly estimate infectivity,

3.4.1 Model Fitness on Synthetic Data.

Table 2 compares the performance of the proposed model with several alternative point process models measured by both likelihood and the accuracy of parameter

estimation. On synthetic data simulated with non-sparse β, Para-Hawkes fits the data better than Cox, while Cox performs better than Multi-Hawkes. On synthetic data simulated with sparse β, Para-Hawkes performs better than the non-sparse case, which demonstrates that the lasso regularization on coefficients β does work. The performance of Multi-Hawkes is rarely affected since the sparsity of β only influences the relationship within $\mathbf{A}$, which is ignored by Multi-Hawkes. Cox performs worse, as it imposes no regularization on coefficients. On larger synthetic data, the performances of all compared models become worse, while the advantage of Para-Hawkes over others become greater. This illustrates that Para-Hawkes is adept in modeling more complexity diffusion networks . We also notice that the lasso regularizer becomes more important on larger networks.

3.4.2 Fitness on Synthetic Data with Time-varying Infectivity.

Table 2 also shows that using time-varying features instead of invariant features slightly harms the performances of Para-Hawkes and Cox, while Multi-Hawkes performs poor, which illustrates the advantage of estimating coefficients β rather than the infectivity α directly . The degree of degeneration of the performance of Para-Hawkes is smaller than that of Cox, which proves that Para-Hawkes is more suitable for modeling networks with time-varying infectivity.

3.4.3 Model Dimension Variation.

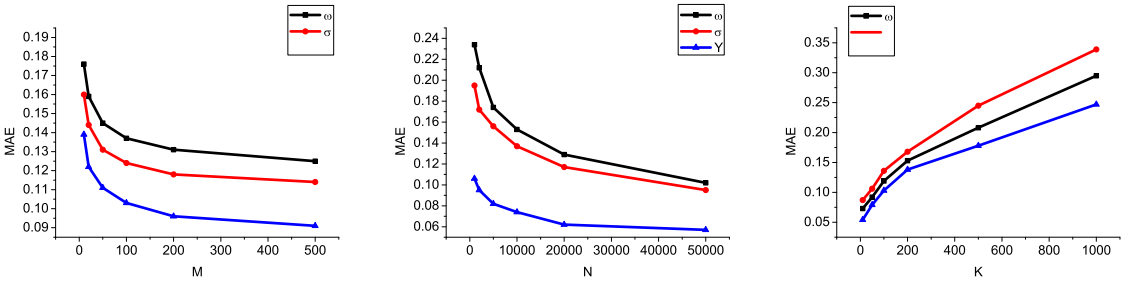
Figure 3 shows how the variation in the setting of model dimensions influences the fitness of the proposed model on the synthetic data. *When increasing the number of dimensions M and fixing all other model dimensions*, the error in both the learning of coefficients β and the estimation of Hawkes parameter μ will be significantly reduced. *Secondly, along with the increase of events N*, the proposed model fits the synthetic data better. *When the number of features K increases*, Para-Hawkes finds it more and more difficult to fit the synthetic data. The impact of model dimension variation

Table 2: Model Fitness on Synthetic Data

Data set	Metric	P-Hawkes	M-Hawkes	Cox
`S-Synthetic`	Training	-73.91	-89.13	-79.82
	Predictive	-136.23	-151.21	-143.21
	MAE(β or α)	0.103	0.257	0.148
	MAE(μ)	0.089	0.116	
`L-Synthetic`	Training	-107.89	-172.21	-135.95
	Predictive	-190.26	-310.85	-233.90
	MAE(β or α)	0.120	0.342	0.161
	MAE(μ)	0.113	0.148	
`S-Sparse`	Training	-70.73	-89.72	-80.27
	Predictive	-133.91	-151.30	-144.84
	MAE(β or α)	0.094	0.258	0.157
	MAE(μ)	0.086	0.117	
`L-Sparse`	Training	-102.46	-172.26	-137.27
	Predictive	-182.91	-310.81	-239.64
	MAE(β or α)	0.116	0.344	0.168
	MAE(μ)	0.102	0.149	
`S-T-varying`	Training	-81.62	-176.32	-97.28
	Predictive	-140.83	-418.20	-172.74
	MAE(β or α)	0.115	0.923	0.165
	MAE(μ)	0.104	0.363	
`L-T-varying`	Training	-122.43	-218.38	-160.92
	Predictive	-207.22	-693.67	-269.30
	MAE(β or α)	0.131	1.327	0.184
	MAE(μ)	0.128	0.616	

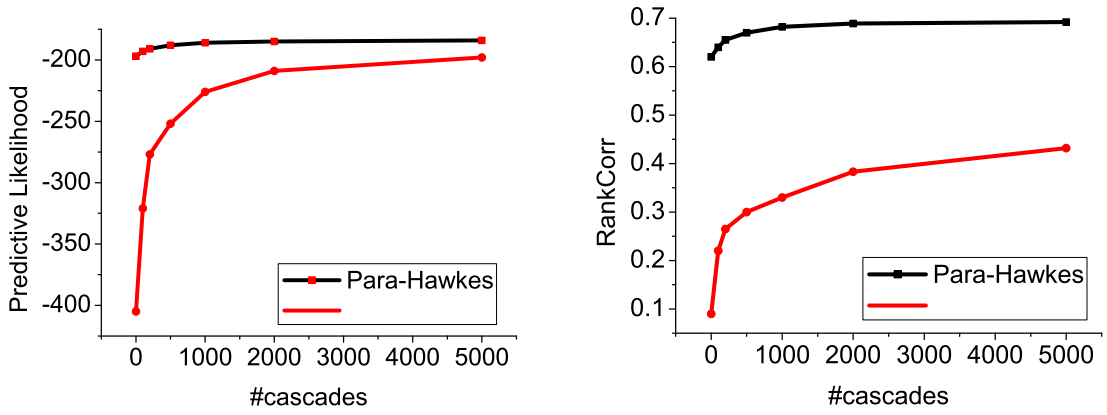
In the column of "Metric", "Training" stands for training likelihood, while "Predictive" stands for predictive likelihood. "P-Hawkes" stands for Para-Hawkes, "M-Hakwes" stands for Multi-Hawkes, "S-" stands for data setting `Small`, "L-" stands for `Large`. "T-varying" stands for `Time-varying`.

on μ is smaller than that on β. One possible explanation is that a poor performance on the estimation of coefficients does not necessarily lead to an improper estimation of infectivity α. Thus the estimation of μ can still be relatively accurate.



(a) M (b) N (c) K

Figure 3: How the variation in model dimension influences the fitness of the proposed model on the synthetic data.

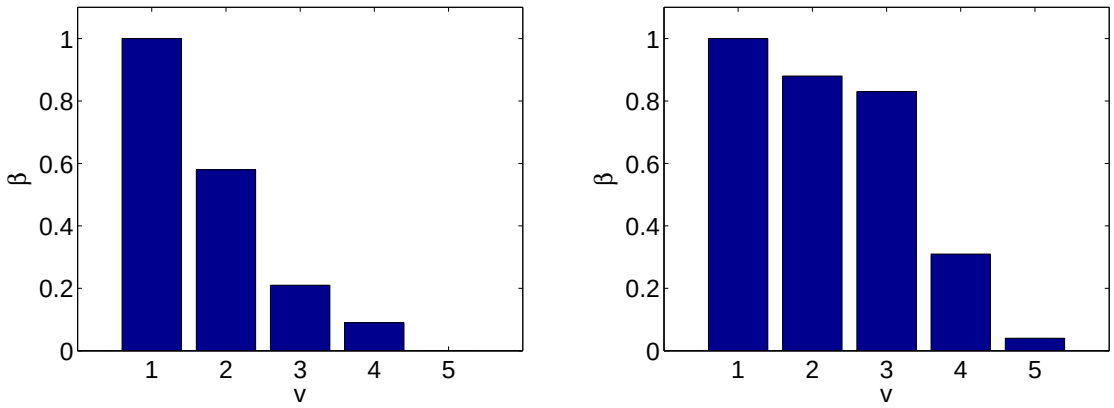


(a) Predictive Likelihood (b) RankCorr

Figure 4: Performance Comparison wrt. Cascade Number.

3.4.4 How the Number of Cascades Affects Performance.

Figure 4 shows that when the number of cascades increases, both the data fitness and the accuracy of the social infectivity estimation of Para-Hawkes are rarely affected, while Multi-Hawkes performs significantly better. However, even trained with a large number of event cascades, Para-Hawkes still performs much better than Multi-Hawkes. Such phenomenon demonstrates that the proposed model works well without multiple cascades, while a normal multi-dimensional Hawkes model requires a large number of cascades to gain a satisfactory performance.



(a) Sparse (b) Low-rank

Figure 5: Coefficients Learned on Synthetic Networks with Different Topologies. The Y axis denotes the average value of the learned coefficients of features characterized by path length v. These average values are scaled to the range of $[0, 1]$ to clarify the comparison of relative importance of different features.

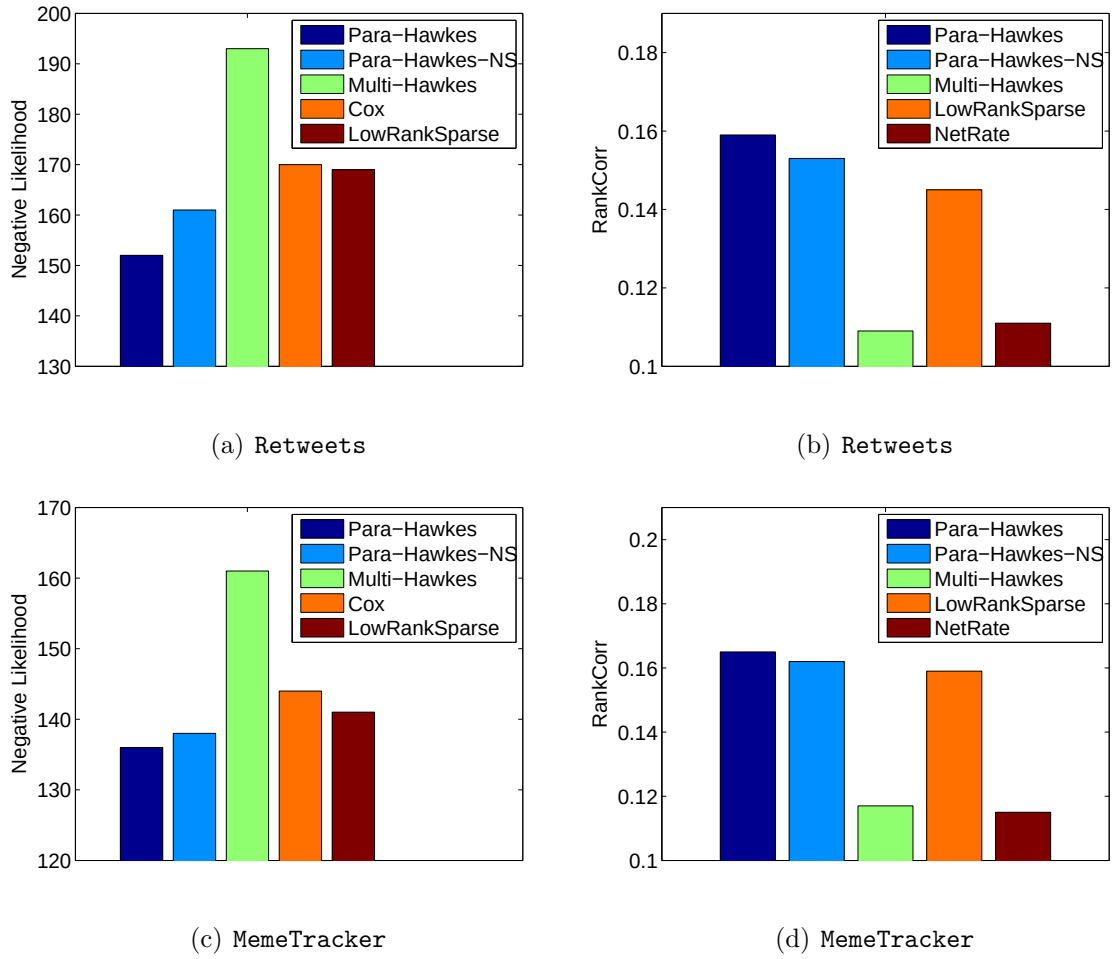
3.4.5 Coefficient Learning on Synthetic Networks with Various Topologies.

This series of experiments sample event cascades from the normal multi-dimensional Hawkes process specified by sparsity and low-rank $\mathbf{A}$'s, respectively, and estimate the coefficients in our proposed model on both data sets to explore the appropriate set of features for modeling different network topologies. Figure 5 shows that, when characterized by different path lengths v, our features weight different in model various network topologies. In a sparse social network, the weights of features characterized by a short path are larger, while in a low-rank network, the weights of features characterized by a medium-length path are relatively more significant. One explanation can be that, in a very sparse network, individuals are more unlikely to influence each other via middlemen than in a low-rank network where people form groups, and influence every other group members.

3.4.6 Real World Data

To further study how our model works in real world social networks, we apply the proposed model on `Retweets` and `MemeTracker` data sets. The `Retweets` data set contains the time-stamped information flowing among tweet users. When a new post is issued by some user, other users will retweet this post or those retweets. In this way, the content of the original post diffuses in the network, and all the time-stamped retweets concerning that post form an event cascade. From the `Retweets` data set, we extract 5000 most popular posts diffusing among around 5000 users. The `MemeTracker` data set contains the time-stamped information flows captured by hyper-links among different sites. These time-stamped hyperlinks form an event cascade for the particular piece of information flowing among numerous web sites. In particular, we extract a network consisting of top 500 sites with all hyperlinks among them.

Figure 8 compares the performance of Para-Hawkes with baselines measured by both predictive likelihood and RankCorr. In this series of experiments, we add a new model named Para-Hawkes-NS, which is our proposed model with no lasso regularizer. From Figure 8, we can see that our proposed models perform better the all compared baselines, which demonstrates the effectiveness of using time-varying features. The advantage over LowRankSparse illustrates that appropriate weighting of generic features can capture specific network topologies, such as sparsity and low-rank structure. Our advantage over LowRankSparse on `Retweets` is much larger than that on `MemeTracker`. One explanation may be that the proposed model suits various networks, while methods using topological priors only work in networks with some specific structure. Moreover, our performance advantage measured by likelihood is much greater than that measured by RankCorr, which implies that the proposed model is capable of precisely modeling observed diffusion, rather than just predicting the relative significance of pairwise infectivity. We also find that a thresholding of



(a) `Retweets` (b) `Retweets`

(c) `MemeTracker` (d) `MemeTracker`

Figure 6: Performance Comparison on Real World Data Sets.

the inferred $\mathbf{A}$ with a small constant will result in an infectivity matrix with sparsity degree similar as that learned by LowRankSparse. Meanwhile, Para-Hawkes performs better than Para-Hawkes-NS, which illustrates the importance of selecting effective features among all designed features.

3.5 Summary

In this chapter, we propose a novel multi-dimensional Hawkes model that parameterizes pairwise infectivity using linear combinations of time-varying features. Alternating direction method of multipliers (ADMM) is employed to estimate the proposed features' coefficients, which are regularized by a ℓ_1 norm to select effective features.

CHAPTER IV

ENERGY USAGE BEHAVIOR MODELING IN ENERGY DISAGGREGATION VIA MARKED HAWKES PROCESS

In the previous chapter, we consider how to raise the efficiency in the modeling of the influence in behavioral data under general settings, i.e., the occurrence of an event influences the occurrence of future events (under the same dimension). Starting from this chapter, we study the influence modeling under some special cases, which do not completely agree with the above general settings. This first special case we consider is the influence among events with marks, where the marks contain detailed descriptions of corresponding events other than the temporal information. The influence in such behavioral data is not simply the influence from the occurrence of one event to that of a future event, but the influence from both the occurrence and the marks of an event to those of a future event. Applications that desire the modeling of such behavioral data include energy disaggregation, where both the temporal information and the amount of consumed energy of the usage of an electronic appliance play an important role.

Energy disaggregation, the task of taking a whole home electricity signal and decomposing it into its component appliances, has been proved to be essential in energy conservation research. One powerful cue for breaking down the entire household's energy consumption is user's daily energy usage behavior, which has so far received little attention: existing works on energy disaggregation mostly ignored the relationship between the energy usages of various appliances across different time slots. To model such relationship, in this chapter, we combine topic models with Hawkes processes, and propose a novel probabilistic model based on marked Hawkes processes

that enable the modeling of marked event data. The proposed model seeks to capture the influence from the occurrence and the marks of one usage event to the occurrences and the marks of subsequent usage events in the future. We also develop an inference algorithm based on variational inference for model parameter estimation. Experimental results on both synthetic data and three real world data sets demonstrate the effectiveness of our model, which outperforms state-of-the-art approaches in decomposing the entire consumed energy to each appliance. Analyzing the influence captured by the proposed model provides further insights into numerous interesting energy usage behavior patterns.

4.1 Energy Usage Behavior Modeling

Let us consider a typical scenario in energy disaggregation, where M appliances are used in a sequence of N time slots $T = \{t_n, n = 1, \ldots, N\}$. Multiple appliances can be used simultaneously in one time slot, and certain appliance is not necessarily always in use. We consider the unsupervised setting, i.e., we only observe the total amount of consumed energy X_n in each time slot n, while the amount of consumed energy $x_{m,n}$ of each appliance m used in that time slot is unavailable. The target of energy disaggregation is to predict each $x_{m,n}$ based on the observed T and X.

Instead of straightforwardly predicting $x_{m,n}$ from X_n, we introduce a set of latent variables $\{Y_{m,n}\}$ to denote whether the m-th appliance is used in the n-th time slot, and turn to solving a much easier problem first: which appliances are in use in each of the time slot. The basic intuition is that the usage of one appliance raises the probability of the usage of related appliances (including itself) in the near future. For instance, people are very likely to use dryer after using washing machine. Such self- & mutually exciting nature coincides with the *self- & mutually exciting* property of the multi-dimensional Hawkes process, i.e., the occurrence of one event in the past will trigger events happening in the future.

4.1.1 Multi-dimensional Hawkes Process

As introduced in previous chapters, the multi-dimensional Hawkes process is a class of self- or mutually exciting point process models [98], which are widely used to describe data that are localized at a finite set of time points $\{t_1, \ldots, t_N\}$ [211]. Formally, the multi-dimensional Hawkes process on an event cascade $\{t_l\}_{l=1}^{N}$ is defined to be a M-dimensional point process with the intensity of the m-th dimension given by:

$$\lambda_m(t) = \mu_m + \sum_{t_l<t} \alpha_{m_l,m} \kappa(t - t_l)$$

Here μ_m denotes the base intensity of the m-th dimension, $\kappa(t-t_l)$ is a time-decaying kernel, while $\alpha_{m,m'}$ denotes the infectivity from events in the m-th dimension to events in the m'-th dimension. Hawkes process has been widely used in applications, such as earthquake prediction [188], sales modeling [255, 82], Asset management [254], search behavior modeling [153], crime modeling [227], and armed conflict analysis [262, 150].

In our tasks, building a multi-dimensional Hawkes process on Y relates the inference of m-th appliance usage state in the n-th time slot $Y_{m,n}$ with that of other appliances in different time slots, thus can be expected to sharply raise the inference accuracy.

4.1.2 Marked Hawkes Process

Although the (multi-dimensional) Hawkes process has been proved to be effective in modeling the influence between event occurrences in many applications, we find it unable to completely solve our energy disaggregation problem. For one thing, the total amount of consumed energy in each time slot has not been utilized; for another, it only predicts whether an appliance is in use rather than the energy it consumes. A better solution is modeling marked events instead of normal events, where the mark of an event refers to those additional features other than the temporal information that

describes the event.[1] In energy disaggregation, *taking the usage of an appliance in a time slot as an event, the corresponding amount of consumed energy is actually the mark of that event.* Such marked events are very common in current social networks, as the descriptions of events are usually available.

Since the marks of an event are very likely to be described by numerous features — a vector with each feature represented by continuous or categorical variables, directly modeling the relationship between marks and occurrences of different events is difficult. One widely used effective solution is the topic model, which clusters all observed marks into several topics/categories, with similar marks in the same category.

To enable the modeling of marks of events in Hawkes processes, we further introduce a new set of latent variables $\{Z_{m,n,k}\}$ to denote whether the marks of an event from the m-th dimension, whose occurrence is previously denoted by $Y_{m,n}$, belongs to the k-th category/topic, we propose the following novel multi-dimensional Hawkes process to model the entire event sequence, with the intensity of an event from the m-th dimension occurring in time slot t whose intensity can be written as: whose intensity can be written as:

$$\lambda_m(t) = \mu_m + \sum_{t_l<t} \sum_{m'} Y_{m',l} \sum_{k,k'} Z_{m,n,k} Z_{m',l,k'} \beta_{m,m',k,k'} \kappa(t - t_l). \tag{4}$$

Here the base intensity μ_m captures how often an event from m-th dimension happens spontaneously, while $\beta_{m,m',k,k'}$ models the degree of *influence* between a event from dimension m with marks of category k to a event from dimension m' with marks of category k'. Notice that the proposed new Hawkes process can handle events with multidimensional marks, while in our application, only a single dimension, the amount of consumed energy, is used.

According to the definition of $Y_{m,n}$, we have $Y_{m,n} = \text{HawkesProcess}(\lambda_m(t_n))$. Thus

[1] A mark can be the casualty of an armed conflict event, the magnitude of an earthquake event, and in our application, the consumed energy of an appliance usage event.

the proposed new Hawkes process straightforwardly models the influence between the occurrence and the mark pattern membership of past events and those of the current event. Assume each appliance has K energy consumption patterns with the k-th pattern denoted as $\theta_{m,k}$, the entire amount of consumed energy in the n-th time slot X_n can be approximated by $\sum_m Y_{m,n} \sum_k \theta_{m,k} Z_{m,n,k}$. The approximation itself does not provide much evidence for the inference of Y and Z, and the learning of θ. However, by constructing a multi-dimensional marked Hawkes process on Y and Z, we relate the inference of $Y_{m,n}$ and $Z_{m,n}$ with that of other appliances in different time slots, thus the inference/learning accuracy can be expected to be increased.

Finally, we present our generative model that produces the entire energy consumption as follows:

- Draw a vector μ of length M that denotes the base intensity of each appliance and a $MK \times MK$ infectivity matrix β that denotes the degree of influence between different appliances under different consumption patterns.
- For each appliance m,
  - draw a K dimensional vector θ_m, where each dimension indicates a single energy consumption pattern of the appliance.
  - draw a K dimensional membership vector $\pi_m \sim \text{Dirichlet}(\alpha)$.
- For the n-th time slot,
  - For the m-th appliance in the n-th time slot,
    - Draw whether it will be used by $Y_{m,n} \sim \text{HawkesProcess}(\lambda_m(\cdot))$, where the intensity λ_m is defined as in Eqn (8);
    - Draw the user energy usage pattern membership $Z_{m,n} \sim \text{Multinomial}(\pi_m)$;

* Draw the amount of consumed energy of device $x_{m,n} \sim Y_{m,n}\text{Gaussian}(\sum_k \theta_{m,k} Z_{m,n,k}, \sigma)$;[2]

– Calculate the total amount of consumed energy in the n-th time slot $X_n = \sum_m x_{m,n}$.

Note that in our M-Hawkes model, the number of appliances that can be simultaneously used in the same time slot is constrained by the total amount of consumed energy at that time. Such a constraint not only benefits the inferring of energy usage patterns of each appliance, but also enables the modeling of several events occurring in the same time slot, which existing Hawkes models hardly handled.

Under our M-Hawkes model, the joint probability of data $T = \{N(\cdot)\} = \{\{t_n\}_{n=1}^N\}$, $X = \{\{X_n\}_{n=1}^N\}$ and latent variables π, Y, Z can be written as follows:

$$p(T, X, \pi_{1:M}, Y, Z|\alpha, \theta, \mu, \beta) = P(T, Y|Z, \mu, \beta) \prod_n P(X_n|Y_n, Z_n, \theta)$$
$$\prod_n \prod_m P(Z_{m,n}|\pi_m) \prod_m \prod_n P(Y_{m,n}|\pi_m) \prod_m P(\pi_m|\alpha).$$

4.1.3 Learning

When given observations of both temporal information $T = \{N(\cdot)\} = \{\{t_n\}_{n=1}^N\}$ and consumed energy X of energy consumption event sequences, the log-likelihood for the complete data is given by $\log p(T, X|\mu, \beta, \alpha, \theta)$ under the proposed M-Hawkes model. We employ variational methods [38] to simplify the inference of true posterior $p(T, X|\mu, \beta, \alpha, \theta)$, and come up with a distribution of latent variables q shown as below:

$$q(\pi_{1:M}, Y, Z|\gamma_{1:M}, \Phi, \rho_{1:N})$$
$$= \prod_m q_1(\pi_m|\gamma_m) \prod_m \prod_n q_2(Y_{m,n}|\phi_{m,n}) q_2(Z_{m,n}|\rho_{m,n})$$

[2] In our experiments, we use a constant σ.

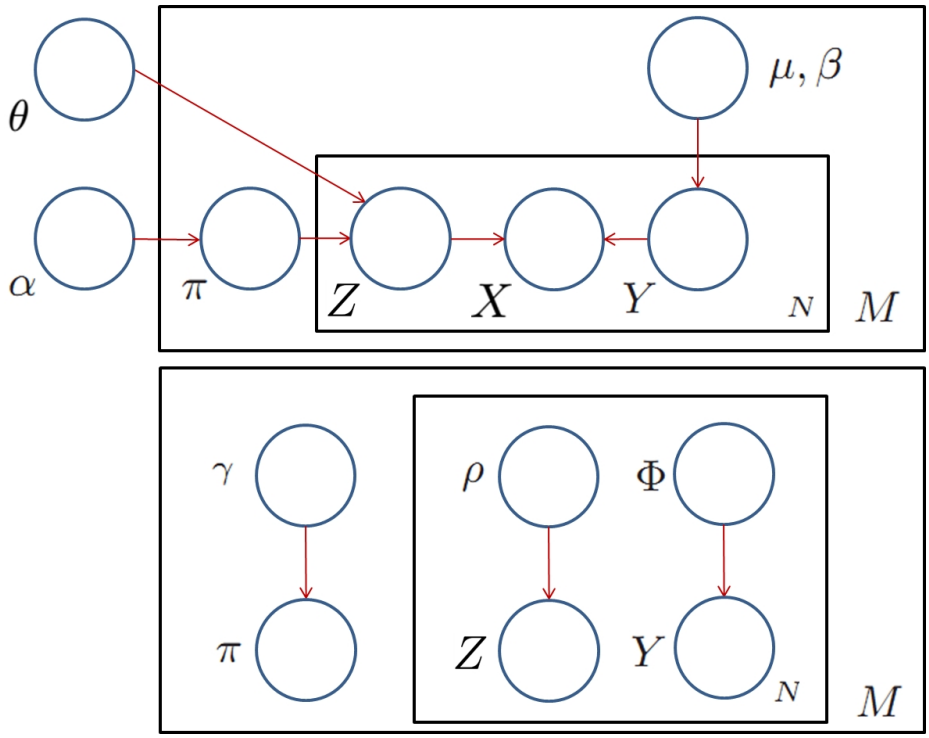


Figure 7: Graphical model representation of M-Hawkes and the variational distribution that approximates the likelihood. The upper figure shows the graphical model representation of M-Hawkes, while the lower figure shows the variational distribution that approximates the likelihood.

where q_1 is a Dirichlet, q_2 is a multinomial, and $\{\gamma_{1:M}, \Phi, \rho\}$ are the set of variational parameters. We optimize those free parameters to tighten the following lower bound $\mathcal{L}'$ for our likelihood:

$$\begin{aligned}\log p(T, X|\mu, \beta, \alpha, \theta) \geq & E_q[\log p(T, X, \pi_{1:M}, Y, Z|\alpha, \theta, \mu, \beta)] \\ & - E_q[\log q(\pi_{1:M}, Y, Z)].\end{aligned} \tag{5}$$

Isolating terms containing λ in Eqn (15), we have

$$\mathcal{L}_h = \sum_{m=1}^{M} \sum_{n} E_q(\log \lambda(Y_{m,n})) - \sum_{m=1}^{M} \int_0^T E_q(\lambda(s))ds, \tag{6}$$

as the partial likelihood on temporal data assuming consumption pattern distribution is known. On one hand, we have $\sum_{m=1}^{M} \int_0^T E_q(\lambda(s))ds = \sum_{m=1}^{M} b_m \beta_{m,m'k,k'} +$

$T\sum_{m=1}^{M}\mu_m$. Here

$$b_{m,m',k,k'} = \sum_{n=1}^{N}\sum_{l=1}^{n-1} r_{m,m',lnkk'}(K(t_n - t_l) - K(t_{n-1} - t_l)),$$

where $K(t) = \int_0^t \kappa(s)ds$, and we define function $r_{m,m',lnkk'} = \phi_{m',l}\rho_{m,n,k}\rho_{m',l,k'}$. On the other hand, in order to update each Hawkes hyper-parameter μ and β independently, we adopt the strategy in [257], and break down the log sum $E_q(\log\lambda(t_n))$ based on Jensen's inequality as:

$$\begin{aligned}
&\mathbb{E}_q(\log(\lambda_m(t_n))) \geq \eta_{m,nn}\log(\mu_m) - \eta_{m,nn}\log(\eta_{m,nn}) \\
&+\sum_{l=1}^{n-1}\sum_{m',k,k'} \eta_{m,m',lnkk'}\log(r_{m,m',lnkk'}\beta_{m,m',k,k'}\kappa(t_n - t_l)) \\
&-\sum_{l=1}^{n-1} r_{m,m',lnkk'}\eta_{m,m',lnkk'}\log(\eta_{m,m',lnkk'}),
\end{aligned}$$

where $\{\eta\}$ is a set of branching variables constrained by:

$$\eta_{m,m',lnkk'} \geq 0,\ \eta_{m,nn} + \sum_{l=1}^{n-1}\sum_{m',k,k'} r_{m,m',lnkk'}\eta_{m,m',lnkk'} = 1.$$

Under a coordinate descent framework, we optimize the lower bound as in Eqn (15) against each variational latent variable[3] and the model hyper-parameter. For variational latent variables, we have the following process

- update rules for ρ's as:

$$\begin{aligned}
\rho_{m,n,k} \propto \exp\Bigg(&\sum_m\left(\Psi(\gamma_{m,k}) - \Psi\left(\sum_k \gamma_{m,k}\right)\right) \\
&+\log\left(\left[X_n - \sum_{m'\neq m,k'\neq k}\phi_{m',n}\rho_{m',n,k'}\theta_{m',k'}\right]_+\right) \\
&-\log(\phi_{m,n}\theta_{m,k}) + \sum_{l=1}^{n-1} f_{l,n} + \sum_{l'=n+1}^{N_m} f_{n,l'}\Bigg),
\end{aligned}$$

[3]Here we categorize branching variables η as variational latent variables.

where we define

$$f_{l,n} = \sum_{m',k'} (\eta_{m,m',lnkk'} \phi_{m',l} \log \left(\frac{\beta_{m,m',k,k'} \kappa(t_n - t_l)}{\eta_{m,m',lnkk'}} \right)$$
$$- \phi_{m',l}(K(t_n - t_l) - K(t_{n-1} - t_l)))\rho_{m',l,k'}$$

- update rules for γ's as:

$$\gamma_{m,k} = \alpha_k + \sum_n \rho_{m,n,k};$$

- update rules for ϕ's as:

$$\phi_{m,n} \propto \exp \left(\eta_{m,nn} \log(\mu_m) - \log \left(\sum_k \rho_{m,n,k} \theta_{m,k} \right) \right.$$
$$+ \log \left(\left[X_n - \sum_{m' \neq m,k} \phi_{m',n} \rho_{m',n,k} \theta_{m,k} \right]_+ \right)$$
$$+ \sum_{l=1}^{n-1} \sum_{m',k,k'} \eta_{m,m',lnkk'} \log(b_{mm',lnkk'})$$
$$\left. + \sum_{l=n+1}^{N} \sum_{m',k,k'} \eta_{m',m,nlkk'} \log(b_{mm',nlkk'}) \right).$$

where $b_{mm',lnkk'} = r_{m,m',lnkk'} \beta_{m,m',k,k'} \kappa(t_n - t_l)$.

- and update rules for η as:

$$\eta_{m,nn} = \frac{\mu_m}{\mu_m + \sum_{l=1}^{n-1} \sum_{m',k,k'} b_{mm',lnkk'}},$$
$$\eta_{m,m',lnkk'} = \frac{\beta_{m,m',k,k'} \kappa(t_n - t_l)}{\mu_m + \sum_{l=1}^{n-1} \sum_{m',k,k'} b_{mm',lnkk'}}.$$

In updating α, we use a Newton-Raphson method, as no closed form solution exists for the approximate maximum likelihood estimate of α. The Newton-Raphson method is conducted with a gradient and Hessian through:

$$\frac{\partial \mathcal{L}'}{\partial \alpha_k} = N \left(\Psi \left(\sum_k \alpha_k \right) - \Psi(\alpha_k) \right) + \sum_m \left(\Psi(\gamma_{m,k}) - \Psi \left(\sum_k \gamma_{m,k} \right) \right),$$
$$\frac{\partial \mathcal{L}'}{\partial \alpha_{k_1} \alpha_{k_2}} = N \left(\mathbb{I}_{(k_1 = k_2)} \Psi'(\alpha_{k_1}) - \Psi' \left(\sum_k \alpha_k \right) \right).$$

The maximum likelihood estimation of *energy usage pattern* θ can be derived through calculating the first derivative of lower-bound $\mathcal{L}'$ against corresponding parameters. We obtain the update formulas given as follows:

$$\theta_{m,k} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{x}.$$

where $\mathbf{A} = [\phi_{m,n}\rho_{m,n,k}]_{n,mk}$ is a matrix of size $n \times mk$, and $\mathbf{x} = [X_n]_n$ is a vector of length n.

To obtain the approximate maximum likelihood estimation of Hawkes hyper-parameters, we optimize the lower bound as in Eqn (15) against each hyper-parameter, and update μ and β independently with closed-form solutions as:

$$\beta_{m,m',k,k'} = \frac{1}{b_m}\sum_{n,l<n} r_{m,m',lnkk'}\eta_{m,m',lnkk'},\ \mu_m = \frac{1}{T}\sum_{n=1}^{N}\eta_{m,nn}$$

In real world scenario β is usually a sparse matrix, as influence only exist in limited pairs of appliances and patterns. Thus to select effective influence and avoid overfitting, we enforce the sparsity of β by imposing lasso type of regularization as $\|\beta\|_1$, and employ the widely used alternating direction method of multipliers (ADMM) [45, 151] to address the constraint optimization problem.

Our variation inference algorithm, named Marked-Hawkes (M-Hawkes), can be interpreted intuitively in the following way. The mark pattern distribution γ of each appliance is determined by both the topic/pattern prior and the pattern assignment of each appliance at each time slot. The probability ϕ of an appliance m used in the n-th time slot is jointly determined by: (a) other appliances used in the current time slot; (b) how likely an appliance was used spontaneously; (c) the influence from the occurrence and the mark pattern of past events to the current occurrence; and (d) the influence from the occurrence and the mark pattern of future events to the current occurrence. The energy consumption pattern ρ of an appliance m used in the n-th time slot is jointly determined by: (a) the pattern prior of this appliance; (b)

the mark patterns of other appliances; (c) past/future influence to the current mark pattern.

In our mean-field variation inference algorithm, the computational cost of inferring variational variables is $O(NM^2K^2)$. The computational cost of the estimation of topic hyper-parameters is $O(NM^2K^2 + M^3K^3)$. The computational cost of the estimation of Hawkes hyper-parameters is $O(N^2M^2K^2)$, which can be reduced to $O(NM^2K^2)$ by only considering the influence in temporally-close time slots. Thus the total computational cost of our algorithm is $O(NM^2K^2 + M^3K^3)$. Since in real-world scenarios, influence exists only among limited pairs of appliances and patterns, M^2K^2 can be reduced to some much smaller constant, thus the above cost can be viewed as linear in the number of events or time slots.

4.2 Experiments

We evaluated our M-Hawkes model on both synthetic and real-world data sets, and compared the performance with the following baselines:

Hawkes: This is a normal multi-dimensional Hawkes process that models the occurrence of events only and no marks of events;

AFAMAP: This method proposed an approximation inference algorithm, named Additive Factorial Approximate MAP, to efficiently solve the additive factorial hidden Markov model by looking at the observed difference in consumed energy, and incorporating a robust mixture component that can account for unmodeled observation [136].

NIALM: This method, named non-intrusive load monitoring, iteratively separated individual appliances from an aggregate energy consumption record, and updated prior models of general appliance types for each specific appliance instance [194].

4.2.1 Synthetic Data

Data Generation. Given parameters $(M, N, K, \alpha, \theta, \mu, \beta)$, the synthetic data is sampled according to the proposed generative model. Here each element μ_m and $\beta_{m,m,k,k'}$ are randomly generated in $[0.5\hat{\mu}, 1.5\hat{\mu}]$ and $[0.5\hat{\beta}, 1.5\hat{\beta}]$ respectively before the simulation. In addition, α is a vector of size K, where the element α_k is generated in $[0.5\hat{\alpha}, 1.5\hat{\alpha}]$ before the simulation. Our synthetic data are simulated with two different settings:

- `Small`: $M = 10$, $N = 120$, $K = 3$, $\hat{\mu} = 0.01$, $\hat{\beta} = 0.5$, $\hat{\alpha} = 0.1$, $\hat{\theta} = 10$. Simulations were run 1,000 times using the pre-generated parameters μ, β;
- `Large`: $M = 50$, $N = 10{,}000$, $K = 5$, $\hat{\mu} = 0.01$, $\hat{\beta} = 0.5$, $\hat{\alpha} = 0.1$, $\hat{\theta} = 10$. Simulations were run 10 times.

To test the robustness of our method, we add two types of noise to the original synthetic data:

`Event Noisy`: We generate additional 10% of total number of events randomly in the time window of each already sampled event sequence, and add them to the sequence;

`Mark Noisy`: Instead of using the simulated X_n as the consumed energy at the n-th time slot, we use a noisy value X'_n which is obtained by adding Gaussian noise on X_n:

$$X'_n = \max(0.1e + 1, 0)X_n, \ e \sim \mathcal{N}(0, \sigma'). \tag{7}$$

The default value of σ' is set to be 1.

Evaluation metrics. We consider the following evaluation metrics: 1) first, we compare the average log predictive likelihood on events falling in the final 10% of the total time of each event cascade; 2) next we compare the average relative distance between the estimated parameters and ground-truth ones by Mean Average Error (MAE). For instance, the MAE of parameter β and $\frac{1}{M}\sum_m |\frac{\mu_m - \hat{\mu}_m}{\mu_m}|$, which we denote as MAE(β). 3) finally, we measure the performance of energy disaggregation by the

Table 3: Inference and Estimation of M-Hawkes on Synthetic data

Data set	MAE(μ)	MAE(β)	MAE(Y)	MAE(Z)
S-Synthetic	0.065	0.197	0.9251	0.9432
S-E-Noisy	0.077	0.281	0.9042	0.9229
S-M-Noisy	0.092	0.313	0.8847	0.9085
L-Synthetic	0.148	0.346	0.8718	0.8942
L-E-Noisy	0.163	0.353	0.8503	0.8642
L-M-Noisy	0.187	0.386	0.8284	0.8372

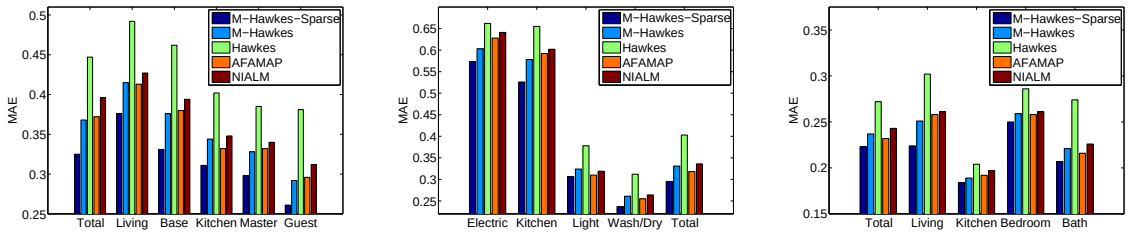
"S-" stands for data setting `Small`, "L-" stands for `Large`, "E-" stands for `Event Noisy`, and "M-" stands for `Mark Noisy`.

Table 4: Log Predictive Likelihood on Both Synthetic and Real-world Data

Data set	M-Hawkes	Hawkes	AFAMAP	NIALM
S-Synthetic	-96.23	-136.26	-104.28	-108.63
S-E-Noisy	-109.21	-148.32	-120.94	-125.27
S-M-Noisy	-116.93	-161.24	-134.27	-140.05
L-Synthetic	-152.39	-194.38	-168.03	-173.26
L-E-Noisy	-165.82	-208.43	-181.46	-186.85
L-M-Noisy	-171.47	-224.06	-186.94	-191.27
Smart*	-145.39	-182.55	-157.83	-160.35
Pecan	-192.17	-234.88	-209.12	-216.43
REDD	-171.37	-210.26	-182.37	-187.51

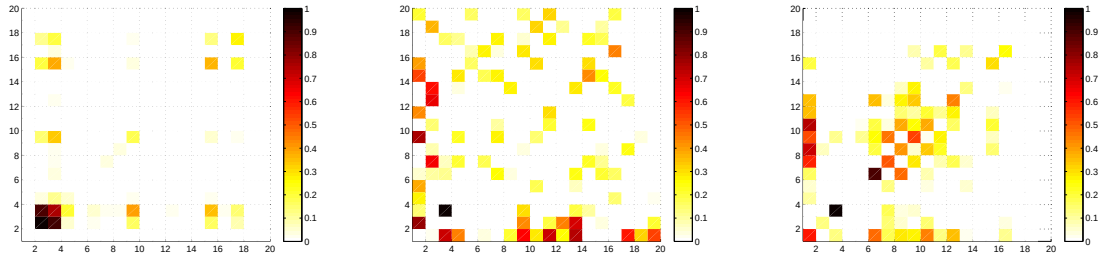
MAE between the ground-truth consumed energy of each appliance $x_{m,n}$ and the estimated consumed energy $\hat{x}_{m,n}$, which is calculated based on the inferred $\rho_{m,n}$ and the estimated $\hat{\theta}_m$.

Inference and Estimation. Table 5 evaluates both the accuracy of our proposed variational inference algorithm in parameter estimation and latent variable inference on the synthetic data. We find that, on the small synthetic data, M-Hawkes can recover the Hawkes parameters μ and β very well, and accurately estimate the model's hyper-parameters. On the large synthetic data, M-Hawkes's performance on parameter estimation becomes worse. The shapely increased number of appliances makes the event occurrence prediction more difficult, and further affects the learning of users' energy usage behavior patterns. On both noisy data sets, M-Hawkes's performances in both inference and estimation become worse. We also find that the performance



(a) `Smart*` (b) `REDD` (c) `Pecan`

Figure 8: Performance Comparison of Energy Disaggregation on Real World Data Sets.



(a) `Smart*` (b) `REDD` (c) `Pecan`

Figure 9: Energy Usage Pattern on Real World Data Sets.
Indices of significant appliances: Smart*: 2-lamp, 3-ac, 4-fan, 9-toaster, 15-refrigerator, 17-microwave. REDD: 1-main, 6-dishwaser, 15-kitchen_outlets, 17-light, 19-washer-dryer. Pecan: 1-ac, 2-dishwasher, 13-microwave, 16-refrigerator.

of energy disaggregation become worse with respect to the increase of the number of appliances, which shapely increases the complexity of the problem.

4.2.2 Performance on Energy Disaggregation

Real-world Smart Meter Data. We also conducted extensive experiments on two real-world data sets. The first data set is Smart* [29], which is a high-resolution data set from three homes including over 50 appliances. The second data set is Reference Energy Disaggregation Dataset (REDD) [137]. This data set comprises six houses including around 20 appliances. The third data set is Pecan Street [4]. This data set collects one-minute resolution disaggregated data for 450+ homes including around

[4]http://www.pecanstreet.org/

20 appliances, dating from late 2012 to early 2014.

Model Fitness. Table 11 shows the log predictive likelihood on energy consumption falling in the final 10% of the total time of data. According to Table 11, M-Hawkes fits both synthetic and real-world data better than alternative probabilistic models. The comparison on synthetic data is meaningful since we add noise into it. AFAMAP performs better than the normal multi-dimensional Hawkes process, which shows the importance of modeling marks of events besides the occurrences. On both noisy data sets, the performances of all models become worse. However, the decrease of the performance of M-Hawkes is smaller than baselines, which demonstrates the robustness of our proposed model. Thus when the usage timestamps and the amounts of consumed energy of some appliances are misrecorded, M-Hawkes performs better in energy disaggregation, and learns energy usage behaviors better.

4.2.3 Performance on Energy Disaggregation.

To illustrate the effectiveness of the proposed model in energy disaggregation, we compare it with all baselines measured by MAE(X). Here we use M-Hawkes-NS to denote the M-Hawkes model with no sparsity constraint on Hawkes hyper-parameter β. According to Figure 8, M-Hawkes performs at least 5% better than all compared methods with comparable time costs. Also, M-Hawkes outperforms compared methods on all categorized appliances. Such results demonstrate the importance of modeling the relationship between the consumed energy of different appliances across different time slots. M-Hawkes's advantage over M-Hawkes-NS illustrates that only a limited number of dependencies exist between appliances in real world energy consumption.

4.2.4 Energy Usage Behavior Pattern Analysis

Based on the parameters learned by the proposed M-Hawkes model, we analyze the energy usage behavior patterns detected in real world energy consumption. According

to Figure 9, influences exist in only limited pairs of appliances. Moreover, the degrees of those influences are very different. In the Smart* data, the influence between lamp and ac is greater than those between all other pairs of appliances. The influence between refrigerator and microwave is greater than that between refrigerator and toaster, which implies that people are more likely to cook food using microwave than toaster. Notice that Smart* data only recorded significant energy consumptions of refrigerator, which makes its usages easily detectable. In addition, the self-influence on some appliances, such as ac, are also very significant. The interpretation is that those appliances are often used for a long time continuously. The results on REDD also show that rarely used appliances, such as dishwasher and washer-dryer influence much less other appliances than those frequently used appliances, such as light and kitchen outlets. Moreover, the influence between a certain pair of appliances is not always symmetric. In Pecan, the influence from refrigerator to microwave is greater than the influence from microwave to refrigerator. One explanation is that people are used to open refrigerator to fetch food before turn on the microwave to cook them. We also find such phenomenon in Smart* data.

4.3 Summary

In this chapter, we formulated the task of energy disaggregation into the modeling of marked event sequences. We presented a probabilistic model that integrates topic models with Hawkes processes to capture the influence from the occurrence and the mark of an event to the occurrences and the marks of future events.

CHAPTER V

IDENTIFYING AND LABELING SEARCH TASKS VIA QUERY-BASED HAWKES PROCESSES

Besides the learning of infectivity, another important issue that deserves our attention in influence modeling is the scope of influence existence. For many behavioral data, the influence can be modeled appropriately based on assumptions that influence exists among events from the same dimension, or between all pairs of events. However, for some special behavioral data, there exist no reasonable assumptions on the scope of influence existence based on prior knowledge only. One good instance is the query submission of search engine users, where we find it difficult to judge whether the submission of one query motivates the submission of a future one, even if they are temporally close or performed by the same user. In this chapter, we study how to simultaneously learn the infectivity and the influence existence in influence modeling under the scenario of information search, and consequently address the problem of search task identification. Specifically, the influence we are to model is what excites users to issue queries serving the same information need in the near future. Since a search task is defined as a set of queries that serve the same information need of search engine users, in information search, each diffusion path of one meme (i.e. information need), which can be also interpreted as a sequence of queries linked by influence, is actually one search task. Thus, analyzing search tasks from user query streams is equivalent to the identification of influence between queries.

We propose a probabilistic method for identifying and labeling search tasks that rely on the following intuitive observations: influences exist between queries which are both temporally close and semantically related. To capture the above intuitions,

we directly model query temporal patterns using a special class of point process called Hawkes processes, and combine topic model with Hawkes processes for simultaneously identifying and labeling search tasks. Essentially, Hawkes processes utilize their self-exciting properties to identify search tasks if influence exists among a sequence of queries for individual users, while the topic model exploits query co-occurrence across different users to discover the latent information needed for labeling search tasks. More importantly, there is mutual reinforcement between Hawkes processes and the topic model in the unified model that enhances the performance of both. We evaluate our method based on both synthetic data and real-world query log data. In addition, we also consider application to query clustering and search task identification. By comparing with state-of-the-art methods, the results demonstrate that the improvement in our proposed approach is consistent and promising.

5.1 Problem Definition

Let us consider a typical scenario that M users issue M corresponding query sequences, and we mark the query sequence of user m as $T_m = \{t_{m,n}, n = 1, \ldots, N_m\}$. We denote the word set of the n-th query by user m as $W_{m,n} = \{w_{m,n,1}, \ldots, w_{m,n,c_{m,n}}\}$. Existing works generally identify search task by sequentially solve two subproblems: 1) using query's textual information to cluster queries in observed query sequences, and 2) using obtained clusters together with temporal information to partition query sequences into search tasks. In this section, we show how these two subproblems can be simultaneously addressed by combining Hawkes processes with the LDA model, and how temporal and textual information can be collaboratively combined to address the above two subproblems. We also show how our model can be used to automatically label search tasks along with search task identification.

5.1.1 Query Co-occurrence and LDA

We choose to address the query clustering problem using graphical models like LDA [41], which has been proven to be effective in topic discovery by clustering words that co-occur in the same document into topics. Let us first introduce how to use LDA to cluster queries based on their textual information only. One straightforward idea is to treat each user's query sequence as a document, and cluster queries that co-occur in the same query sequence into topics, since queries issued by the same user are generally more likely to share the same *information need* than queries issued by different users. Since we focus on query co-occurrence instead of word co-occurrence, we enforce that words in one query belong to the same topic. Our LDA model assumes K topics lie in the given query sequences, and each user m is associated with a randomly drawn vector π_m, where $\pi_{m,k}$ denotes the probability that a query issued by user m belongs to topic k. For the n-th query in the query sequence of user m, a K-dimensional binary vector $Y_{m,n} = [y_{m,n,1}, \ldots, y_{m,n,K}]^T$ is used to denote the query's topic membership. One challenge we encounter in the inference of topic membership Y is that, without temporal information of queries, it is difficult to judge whether two non-contiguous co-occurred queries should belong to the same topic or not. A pair of queries that co-occurs a lot may be completely unrelated if the temporal gap between them is always large.

Since the co-occurrence of queries with large temporal gap is useless or harmful, we make use of temporal information to decide which query co-occurrence should be taken into account by LDA, i.e., how a document in LDA model is defined/constituted. One simple way of utilizing temporal information is to define a document as consecutive queries in a fixed time window (or time session), thus enable us to focus on temporally close query co-occurrence. Temporally close queries that issued many times by the same user or many other users are more likely to be semantically related to each other, i.e., belong to the same search task. However, a time window based LDA

model may suffer from the following drawbacks: 1) Usually no optimal solution exists for cutting the entire query sequence into different time-sessions. If we allow different time-sessions to overlap, redundant query co-occurrence will be taken into account; otherwise, pairs of queries with very small temporal gap can be partitioned into different tasks, which may cause information loss. 2) Using time windows will ignore or misunderstand users' own temporal patterns in searching.

To address the above drawbacks, we can weigh each query co-occurrence based on how likely an *influence* exists between this pair of queries, i.e., the occurrence of one query raises the probability that the other query will be issued in the near future. That is to say, one document is a subsequence of queries linked through *influence*. This *influence*, rather than time window, enables us to distinguish temporally close query co-occurrence from temporally regular query co-occurrence for each user based on his/her own frequency of query submission. To model such personal frequency and *influence*, we utilize Hawkes processes, to capture the temporal information in different query sequences.

5.1.2 Hawkes Process

As introduced in previous chapters, one powerful tool in statistics for modeling event sequence data is the point process, which is widely used to describe data that are localized at a finite set of time points $\{t_1, \ldots, t_N\}$. Typically, in a point process, $N(t)$ counts the number of points (i.e., occurrences of events) in $(-\infty, t]$, and the conditional intensity function $\lambda(t|\mathcal{H}_t)$ denotes the expected instantaneous rate of future events at timestamp t depending on $\mathcal{H}_t$, the history of events preceding t. For clarity, hereafter we use $*$ to imply the dependence on $\mathcal{H}_t$, i.e., $\lambda(t|\mathcal{H}_t)$ will be denoted $\lambda^*(t)$.

The Hawkes process is a class of self- or mutually exciting point process models [98]. A univariate Hawkes process $\{N(t)\}$ is defined by its intensity function

$$\lambda^*(t) = \mu(t) + \int_{-\infty}^{t} \kappa(t-s) dN(s),$$

where $\mu : \mathbb{R} \to \mathbb{R}_+$ is a deterministic base intensity, $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ is a kernel function expressing the postive influence of past events on the current value of the intensity process. The process is well known for its *self-exciting* property, which refers to the phenomenon that the occurrence of one event in the past increases the probability of events happening in the future. Such *self-exciting* property can either exists between every pair of events as assumed in a normal univariate Hawkes process, or only exists between limited pair of events. For instance, any query but the last in a search task can imply an increased probability of future queries issued in the same search task, since the user's *information need* in this search task hasn't been satisfied. Meanwhile, queries from different search tasks may rarely affect each other.

Since our definition of *influence* coincides with the *self-exciting* property of Hawkes process, we propose to identify the *influence* among queries by building one separate Hawkes process on each user's query sequence. In the query sequence of user m, we use $R_{m,n,n'}$ to denote whether *influence* exists between the n-th and n'-th query. If *influence* exists, we believe that the occurrence of n-th query has a time-decay effect on increasing the intensity at the timestamp of the occurrence of the n'-th query. Thus based on *influence* R_m, we model the query sequence issued by user m with a univariate Hawkes process, whose intensity can be written as:

$$\lambda_m(t) = \mu_m + \sum_{t_{m,l}<t} R_{m,l,n}\beta_m\kappa(t - t_{m,l}) \tag{8}$$

The baseline intensity μ_m captures how often user m issues a query spontaneously[1] (i.e., not triggered by any other queries), while β_m models the degree of *influence* between sequential queries issued by user m, and $\kappa(t - t_{m,l})$[2] captures the time-decay effect only.

[1]For simplicity, we assume this cascade-birth process is a homogeneous Possion process with $\mu_m(t) = \mu_m$.

[2]Our work uses the exponential kernel in experiments, i.e., $\kappa(\Delta t) = \omega e^{-\omega\Delta t}$ if $\Delta t \geq 0$ or 0 otherwise. However, the model development and inference is independent of kernel choice and extensions to other kernels such as power-law, Rayleigh, non-parametric kernels are straightforward.

Influence R can be estimated together with μ and β by maximizing the likelihood of the proposed Hawkes model on observed query sequence $\{T_m = \{t_{m,n}\}\}$. The estimation of R is actually to identify query-pairs that the occurrence of the later query most likely violates the normal query-submission frequency, and gets triggered by the earlier one. In other words, if *influence* exists between two queries, the corresponding temporal gap can be significantly less than the average temporal gap of pairs of queries in the same query sequence(issued by the same user). Since the definition of *influence* suggests that queries linked by significant *influence* naturally form search tasks, a thresholding of $R_{m,l,n}\beta_m\kappa(t-t_{m,l})$ with a small constant automatically results in search task partition. The estimation of R consequently partitions observed query sequences into search tasks.

5.1.3 LDA-Hawkes

Estimated by Hawkes processes, *influence* R captures the unique temporal pattern of each user's query sequence. We use R to weight the query co-occurrence, which bridges the LDA model and Hawkes process through:

$$R_{m,n,n'} = Y_{m,n}^T * Y_{m,n'}, \tag{9}$$

that is to say, *influence* exists between these two queries if and only if the two queries share the same topic. Since queries in the same search task are linked by *influence*, all queries in the same search task share the same topic, which labels this search task as well.

Though our defined bridge between *influence* R and query-topic membership Y, the Hawkes process and the LDA model mutually benefit each other in identifying and labeling search tasks. On one hand, provided *influence* among queries, we obtain 0-1 weighted query co-occurrence of each candidate query-pair in observed query sequences, and generate topics accordingly. For instance, in Figure 10, although 8 pairs of queries (9 possible combinations with 8 unique query-pairs) co-occur in query

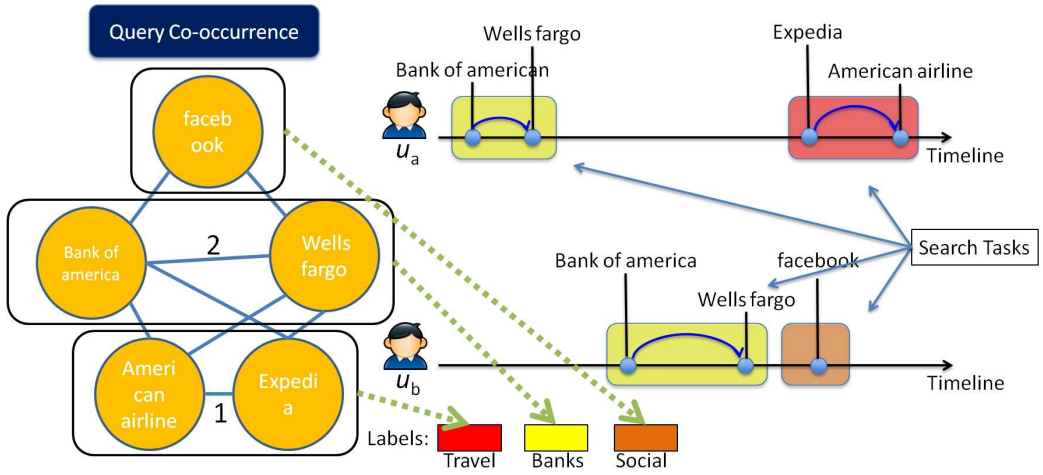


Figure 10: A Toy Example of our LDA-Hawkes model. Blue line denotes the influence among queries. Green dash line shows the label each query belongs to.

sequences, only the co-occurrences of query-pairs "bank of america"–"wells fargo" and "Expedia"–"american airline" have positive weights. These weighted query co-occurrences embed personal temporal information, thus are expect to lead to improved topics compared with existing LDA-based methods [103, 245, 244] that used no weight scheme or only uniform standard weight scheme.

On the other hand, the estimation of *influence* R based on temporal data $\{T_m\}$ only can be intractable, since the exploration of the whole space of R is known to very costly ($2^{\sum_m N_m}$ possible solutions). LDA-Hawkes further makes use of textual data to limit the output space of R to the most probable subspace, since topics learned by the LDA part in turn justify the *influence* existence between each pair of queries. Two queries rarely co-occur can be clustered into different topics by the LDA part, based on such query-topic membership no *influence* exists between these two queries. For example, in the query sequence of user u_b shown in Figure 10, the temporal gap between query-pair "bank of america"–"wells fargo" is larger than the temporal gap between query-pair "wells fargo"–"facebook". However, the pair of queries "bank of america"–"wells fargo" also co-occurs in the query sequence of user u_a, while "wells fargo"–"facebook" does not, which in turn emphasizes that *influence*

should exist between "bank of america" and "wells fargo" rather than between "wells fargo" and "facebook". To sum up, combined through *influence*, Hawkes process and LDA reciprocally contribute to the search task identification and labeling.

Finally, we present our generative model that combines Hawkes process and LDA as follows:

- For each topic k, draw a V dimensional membership vector $\sigma_k \sim \text{Dirichlet}(\alpha')$.
- For each user m, draw a K dimensional membership vector $\pi_m \sim \text{Dirichlet}(\alpha)$.
- For the content of the n-th query issued by user m,
  - $Y_{m,n} \sim \text{Multinomial}(\pi_m)$;
  - For the i-th word in the n-th query issued by user m,
    - $w_{m,n,i} \sim \text{Multinomial}(Y_{m,n}, \sigma)$;
- For the timestamp of the sequence of queries issued by user m,
  - draw personal base intensity μ_m and degree of *influence* β_m;
  - derive R_m from $\{Y_{m,n}\}$ through Eqn (9);
  - $N_m(\cdot) \sim \text{HawkesProcess}(\lambda_m(\cdot))$, where the intensity λ_m is defined as in Eqn (8).

Here V is the size of vocabulary. Note that in our LDA-Hawkes model, queries issued by one user share the same topic distribution, while words in one query belong to the same topic. The topic membership of the n-th query of user m, $Y_{m,n}$, determines not only the words the query owns, but also the timestamp of its occurrence through Hawkes process $\lambda_m(\cdot)$.

Under our LDA-Hawkes model, the joint probability of data $T = \{N_m(\cdot)\} = \{\{t_{m,n}\}_{n=1}^{N_m}\}$, $W = \{\{W_{m,n}\}_{n=1}^{N_m}\}$ and latent variables $\{\pi_{1:M}, Y\}$ can be written as

follows:

$$
\begin{aligned}
&p(T, W, \pi_{1:M}, Y, \sigma | \alpha, \alpha', \mu, \beta) \\
=&\prod_m P(\{t_{m,n}\}_{n=1}^{N_m} | Y_{m,1:N_m}, \mu_m, \beta_m) \prod_m \prod_n \prod_i P(w_{m,n,i} | Y_{m,n}, \sigma) \\
&\prod_m \prod_n P(Y_{m,n} | \pi_m) \prod_m P(\pi_m | \alpha) \prod_k P(\sigma_k | \alpha')
\end{aligned}
$$

5.2 Efficient Optimization

Statistical inference of non-Markovian point process has attracted increasingly interest recently. On the other hand, despite that a tremendous amount of works on inference of topic models have been published, very few of them are proposed to solving topic model combined with point processes. In this section, we derive a mean-field variational Bayesian inference algorithm for our proposed LDA-Hawkes model.

5.2.1 Learning of Influence Existence

We start our optimization process by the derivation of updating rules for latent variables $\pi_{1:M}$, Y, σ. The most important latent variable is Y, based on which we directly compute the R, which indicates the existence of influence among query submissions.

Under LDA-Hawkes model, given observations of both temporal information $T = \{N_m(\cdot)\} = \{\{t_{m,n}\}_{n=1}^{N_m}\}$ and textual information $W = \{\{W_{m,n}\}_{n=1}^{N_m}\}$ of query sequences, the log-likelihood for the complete data is given by $\log P(T, W | \mu, \beta, \alpha, \alpha')$. Since this true posterior is hard to infer directly, we turn to variational methods [38], whose main idea is to posit a distribution of the latent variables with free parameters, and then fit those parameters such that the distribution is close to the true posterior in Kullback-Leibler (KL) divergence. The variational distribution is supposed to be simpler than the true posterior, thus enable us to approximately solve the original optimization problem. In Figure 19, the lower part shows the variational distribution that approximates the data likelihood. We choose to introduce a distribution of latent

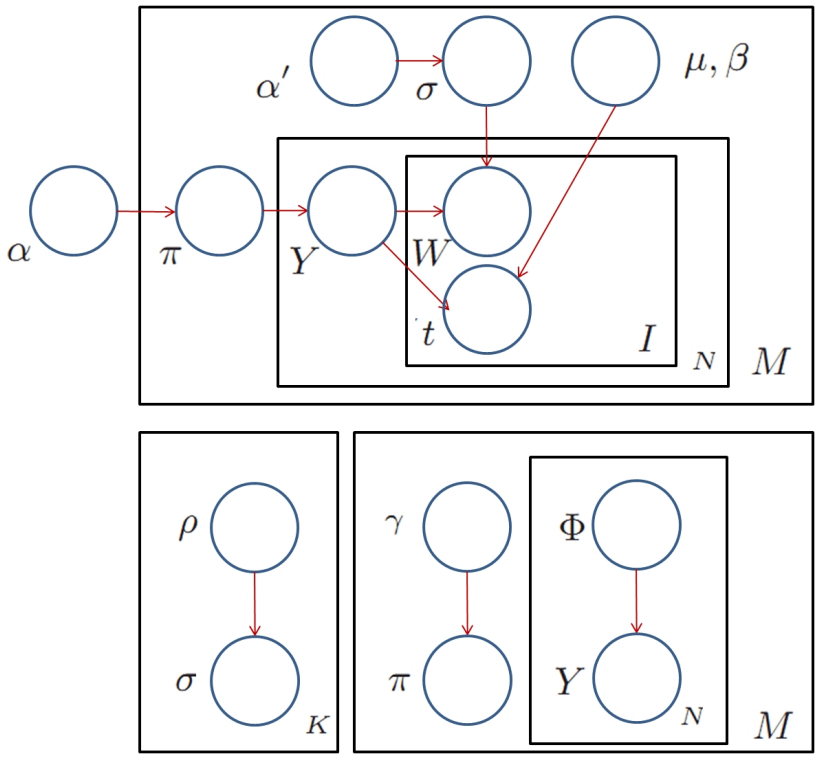


Figure 11: Graphical model representation of LDA-Hawkes and the variational distribution that approximates the likelihood. The upper figure shows the graphical model representation of LDA-Hawkes, while the lower figure shows the variational distribution that approximates the likelihood.

variables q that depend on a set of free parameters, and specify q as the mean-field fully factorized family as follows:

$$\begin{aligned} &q(\pi_{1:M}, Y, \sigma_{1:K} | \gamma_{1:M}, \Phi, \rho_{1:K}) \\ =&\prod_m q_1(\pi_m|\gamma_m) \prod_m \prod_n q_2(y_{m,n}|\phi_{m,n}) \prod_k q_1(\sigma_k|\rho_k) \end{aligned}$$

where q_1 is a Dirichlet, q_2 is a multinomial, and $\{\gamma_{1:M}, \Phi, \rho_{1:K}\}$ are the set of free variational parameters that are optimized to tight the following lower bound $\mathcal{L}'$ for our likelihood:

$$\begin{aligned} \log p(T, W|\mu, \beta, \alpha, \alpha') \geq& E_q[\log p(T, W, \pi_{1:M}, Y, \sigma|\alpha, \alpha', \mu, \beta)] \\ &- E_q[\log q(\pi_{1:M}, Y, \sigma_{1:K})]. \end{aligned} \tag{10}$$

Isolating terms containing λ in Eqn (15), we have

$$\mathcal{L}_h = \sum_{m=1}^{M} \sum_{n} E_q(\log \lambda(t_{m,n})) - \sum_{m=1}^{M} \int_0^T E_q(\lambda(s))ds, \tag{11}$$

as the partial likelihood on temporal data assuming query-topic distribution is known, where the second term reduces to $\sum_{m=1}^{M} b_m + T \sum_{m=1}^{M} \mu_m$. Here

$$b_m = \sum_{n=1}^{N_m} \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n})(K(t_{m,n} - t_{m,l}) - K(t_{m,n-1} - t_{m,l})).$$

where $K(t) = \int_0^t \kappa(s)ds$, and we define function $r(\phi_{m,l}, \phi_{m,n}) = \sum_k \phi_{m,l,k}\phi_{m,n,k}$, which can be viewed as the latent variable that approximates *influence* R. On the other hand, to break down the expectation of the log-intensity $E_q(\log \lambda(t_{m,n}))$ involved in the first term in Eqn (11), we apply Jensen's inequality

$$\mathbb{E}_q[\log(\lambda(t_{m,n}))] \geq \eta_{m,nn} \log(\mu_m) + \sum_{l=1}^{n-1} \eta_{m,ln} \log(\beta_m \kappa(t_{m,n} - t_{m,l}))$$
$$- \eta_{m,nn} \log(\eta_{m,nn}) - \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \eta_{m,ln} \log(\eta_{m,ln}),$$

where we introduce a set of branching variables $\{\eta_m\}_{m=1}^{M}$. Note that each η_m is a $n \times n$ lower-triangular matrix with n-th row $\eta_{m,n} = [\eta_{m,1n}, \ldots, \eta_{m,nn}]^T$. The branching η also satisfies the following conditions:

$$\eta_{m,ln} \geq 0,\ \eta_{m,nn} + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \eta_{m,ln} = 1.$$

Under a coordinate descent framework, we optimizing the lower bound as in Eqn (15) against each variational latent variables[3] and the model hyper-parameters, including both LDA hyper-parameters and Hawkes hyper-parameters. For variational latent variables, we have

[3]Here we categorize branching variables η as variational latent variables.

- update rules for ϕ's as:

$$\begin{aligned}\phi_{m,n,k} \propto \exp &\left(\sum_m \left(\Psi(\gamma_{m,k}) - \Psi\left(\sum_k \gamma_{m,k}\right)\right)\right. \text{ : Topic distribution} \\ &\times \sum_i \sum_v w_{m,n,i,v} \left[\Psi(\rho_{k,v}) - \Psi\left(\sum_v \rho_{k,v}\right)\right] \text{ : Word content} \\ &\times \sum_{l=1}^{n-1} f_{l,n} \text{ : influences from past} \\ &\left.\times \sum_{l'=n+1}^{N_m} f_{n,l'}\right), \text{ : influences to future} \end{aligned} \tag{12}$$

where we define

$$\begin{aligned} f_{l,n} =& \phi_{m,l,k} \left(\eta_{m,ln} \log\left(\frac{\beta_m \kappa(t_{m,n} - t_{m,l})}{\eta_{m,ln}}\right)\right. \\ &\left. -\beta_m (K(t_{m,n} - t_{m,l}) - K(t_{m,n-1} - t_{m,l}))\right); \end{aligned}$$

- update rules for γ's as:

$$\gamma_{m,k} = \alpha_k + \sum_n \phi_{m,n,k};$$

- update rules for ρ's as:

$$\rho_{k,v} \propto \alpha'_v + \sum_m \sum_n \sum_i \phi_{m,n,k} w_{n,i,v};$$

- and update rules for η as:

$$\begin{aligned} \eta_{m,nn} &= \frac{\mu_m}{\mu_m + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \beta_m \kappa(t_{m,n} - t_{m,l})}, \\ \eta_{m,ln} &= \frac{\beta_m \kappa(t_{m,n} - t_{m,l})}{\mu_m + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \beta_m \kappa(t_{m,n} - t_{m,l})}. \end{aligned}$$

5.2.2 Learning of Infectivity

In the following, we come up with solutions for the learning of both LDA hyper-parameters, which implies users' general preference on topics, and Hawkes hyper-parameters, which denotes the degree of infectivity of our learned influence. We use a

variational expectation-maximization (EM) algorithm [69] to compute the empirical Bayes estimates of the LDA hyper-parameters α and α' in our LDA-Hawkes model. This variational EM algorithm optimize the lower bound as in Eqn (15) instead of the real likelihood, it iteratively fits the variational distribution q to approximate the posterior and maximizes the corresponding bound with respect to the parameters. The latter M-step is equivalent to finding the MLE using expected sufficient statistics under the variational distribution. We consider the maximization step for each parameter in turn.

Notice that a closed form solution for the approximate maximum likelihood estimate of α does not exist, we use a linear-time Newton-Raphson method, where the gradient and Hessian are

$$\frac{\partial \mathcal{L}'}{\partial \alpha_k} = N\left(\Psi\left(\sum_k \alpha_k\right) - \Psi(\alpha_k)\right) + \sum_m \left(\Psi(\gamma_{m,k}) - \Psi\left(\sum_k \gamma_{m,k}\right)\right),$$
$$\frac{\partial \mathcal{L}'}{\partial \alpha_{k_1}\alpha_{k_2}} = N\left(\mathbb{I}_{(k_1=k_2)}\Psi'(\alpha_{k_1}) - \Psi'\left(\sum_k \alpha_k\right)\right).$$

Similar update rules can be derived for α'.

In the following, we derive the maximum likelihood estimation of infectivity, i.e., Hawkes hyper-parameters of our LDA-Hawkes model. The Hawkes hyper-parameters include the base intensity $\mu \in \mathbb{R}_+^M$ and the degree of *influence* $\beta \in \mathbb{R}_+^M$, where $\mathbb{R}_+$ denotes the nonnegative real domain. Similar as the case of the LDA hyper-parameters, the MLE for the Hawkes hyper-parameters are obtained by optimizing the lower bound as in Eqn (15) upon the convergence of the variational inference, which leads to the following update formulas:

$$\beta_m = \frac{1}{b_m}\sum_{n=1}^{N_m}\sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n})\eta_{m,ln}, \qquad \mu_m = \frac{1}{T}\sum_{n=1}^{N_m}\eta_{m,nn}.$$

Our variation inference algorithm, named LDA-Hawkes, is intuitively interpretable. This algorithm has two loops. The inner loop 1) infers the label/topic distribution γ in each user's query sequence, based on both the topic prior and the

topic assignment of each query; 2) infers the word distribution ρ in each topic, based on both the word prior and the topic assignment of each word; 3) and clusters queries accordingly. The query clustering process as in Eqn (12) assigns queries to different topics based on not only the textual information of queries, but also the temporal pattern underlying the corresponding query sequence. As marked in the formula, the query-topic assignment of each query n is inferred by integrating four types of evidences: (a) Users' label/topic distribution; (b) Semantic clustering of queries; (c) Past influence: how labels of queries in the past affect the label of the current query; (d) Future influence: how the label of the current query affect labels of queries in the future. The outer loop estimates our model's hyper-parameters, both LDA parameters interpreted as topic/word priors and Hawkes parameters that capture the temporal pattern of the query submission of each user.

In our mean-field variation inference algorithm, the computational cost of inferring variational latent variables is $O((\sum_m N_m) * K * \bar{C})$, where $\bar{C}$ is the average number of words in a query. The computational cost of the estimation of LDA hyper-parameters is $O(K+V)$. The computational cost of the estimation of Hawkes hyper-parameters is $O(\sum_m N_m^2)$, which can be further reduced to $O(\sum_m N_m)$ by controlling the number of *influence* candidate for each query. Most queries has only limited number of *influence* associated, since for each query, most of the rest queries are far from it, and there exist many other queries in between. Thus the total computational cost of our algorithm is $O((\sum_m N_m) * K * \bar{C} + V)$.

5.3 Experiments

We evaluated our LDA-Hawkes model on both synthetic and real-world data sets, and compared the performance with the following:

- two alternative LDA-based probabilistic models:

  Time-Window(TW): This model assumes queries belong to the same search

task only if they lie in a fixed or flexible time window, and uses LDA to cluster queries into topics based on the query co-occurrences within the same time window. We tested time windows of various sizes.

Word-Related: This model assumes queries belongs to the same search task only if they share at least one word, and uses LDA to cluster queries into topics based on the co-occurrences of queries that sharing at least one word.

- two state-of-the-art query clustering approaches:

  Session-Similarity[268]: This method evaluated query similarity based on both query sessions and query content, and used those similarity scores for query clustering.

  GATE[12]: This is a Greedy Agglomerative Topic Extraction algorithm. It extracted topics based on a pre-defined topic similarity function, which considered both semantic similarity and mission similarity. Here mission similarity refers to the likelihood that two queries appear in the same mission, while missions are sequences of queries extracted from users' query logs through a mission detector.

- and three state-of-the-art search task identification approaches: **Bestlink-SVM [242]:** This method identified search task using a semi-supervised clustering model based on the latent structural SVM framework. A set of effective automatic annotation rules were proposed as weak supervision to release the burden of manual annotation.

  QC-HTC/QC-WCC [164]: This series of methods viewed search task identification as the problem of best approximating the manually annotated tasks, and proposed both clustering and heuristic algorithms to solve the problem. QC-WCC conducted clustering by dropping query-pairs with low weights, while

QC-HTC considered the similarity between the first and last queries of two clusters in agglomerative clustering.

Reg-Classifier[117]: This method designed a diverse set of syntactic, temporal, query log and web search features, and used them in a logistic regression model to detect search tasks.

5.3.1 Synthetic data

Data Generation. Given parameters $(M, N, K, \alpha, \alpha', \mu, \beta)$, the synthetic data is sampled according to the proposed generative model. We record the sampled values of Y, and calculate the ground-truth *influence* R for evaluating the accuracy of our prediction of *influence* among queries. Notice μ and β are both vectors of size M, where each element μ_m and β_m is randomly generated in $[0.5\hat{\mu}, 1.5\hat{\mu}]$ and $[0.5\hat{\beta}, 1.5\hat{\beta}]$ respectively before simulation. Vectors α and α' are of size K and V respectively, where each element α_k and α'_v is generated in $[0.5\hat{\alpha}, 1.5\hat{\alpha}]$ and $[0.5\hat{\alpha'}, 1.5\hat{\alpha'}]$ respectively before simulation.

Our synthetic data are simulated with two different settings:

- `Small`: $M = 100$, $N = 120$, $K = 10$, $\hat{\mu} = 0.01$, $\hat{\beta} = 0.5$, $\hat{\alpha} = 0.1$, $\hat{\alpha'} = 0.1$. Simulations were run 1,000 times using the pre-generated parameters μ, β;
- `Large`: $M = 10{,}000$, $N = 10{,}000$, $K = 50$, $\hat{\mu} = 0.01$, $\hat{\beta} = 0.5$, $\hat{\alpha} = 0.1$, $\hat{\alpha'} = 0.1$. Simulations were run 10 times.

To test the robustness of our method, we add two types of noise to the original synthetic data:

`Event Noisy`: We generate additional 10% of total number of queries randomly in the time window of each already sampled query sequence, and add them to the sequence;

`Intensity Noisy`: Instead of using $\lambda(t)$ to simulate the query occurrence at time t,

we use a noisy value $\lambda'(t)$, which is obtained by adding Guassian noise on $\lambda(t)$:

$$\lambda'(t) = \max(0.1 * e + 1, 0) * \lambda(t),\ e \sim \mathcal{N}(0, \sigma). \tag{13}$$

The default value of σ is set to be 1.

Inference and Estimation. Table 5 evaluates both training likelihood, and the accuracy of our proposed variational inference algorithm in parameter estimation and latent variable inference on the synthetic data. We can find that, on the small synthetic data, LDA-Hawkes can recover the Hawkes parameters μ and β very well, which represent users' personal temporal patterns of query submission. Meanwhile, based on the inferred query-topic membership $\hat{Y}$, we predict the *influence* $\hat{R}$ among queries, and compare with the ground-truth *influence* R to evaluate the accuracy of our *influence* prediction through:

$$\text{Proc}_R = \sum_m \frac{1}{N_m(N_m-1)/2} \sum_{n=1}^{N_m} \sum_{n'=n+1}^{N_m} I(R_{n,n'} = \hat{R}_{n,n'}).$$

Results in Table 5 show that LDA-Hawkes can accurately predicts *influence*. We also find an interesting phenomenon that the accuracy of our estimated Hawkes parameters and the accuracy of our predicted *influence* are highly correlated, since given different predicted *influence* $\hat{R}$, the optimal parameters μ and β that maximize the likelihood of Hawkes processes on a query sequence can be very different. On the large synthetic data, LDA-Hawkes's performance on parameter estimation becomes worse, while the accuracy of *influence* prediction also decreases. Due to the shapely increased data size, the combination of textual and temporal information becomes more complicated, which makes *influence* prediction more difficult, and further affects the learning of users' personal temporal patterns. On both noisy data set, LDA-Hawkes's performances in both inference and estimation become worse.

Table 5: Inference and Estimation of LDA-Hawkes on Synthetic data

Data set	$\frac{1}{M}\sum_m \lvert\frac{\bar{\mu}_m-\mu_m}{\mu_m}\rvert$	$\frac{1}{M}\sum_m \lvert\frac{\bar{\beta}_m-\beta_m}{\beta_m}\rvert$	Prec_R	log likelihood
Small Synthetic	0.058	0.204	0.9175	-92.38
Small Event Noisy	0.083	0.317	0.8847	-95.02
Small Intensity Noisy	0.101	0.362	0.8675	-96.80
Large Synthetic	0.174	0.381	0.8573	-115.29
Large Event Noisy	0.202	0.413	0.8291	-119.38
Large Intensity Noisy	0.219	0.436	0.8107	-122.25

5.3.2 Real-world Data

We also conducted extensive experiments on two real-world data sets. The first data set is adapted from the query log of `AOL` search engine [17]. The entire collection consists of 19.4 million search queries from about 650,000 users over a 3-month period. We cleaned the data by removing the duplicated queries which were submitted consecutively within 1 minute. We randomly selected a subset of users who submitted over 1,000 queries during this period, and collected their corresponding search activities, including the anonymized user ID, query string, timestamp, the clicked URL. As a result, we collected 1,786 users with 2.2 million queries, and their activities span from 18 days to 3 months. The second data set is collected from Yahoo search engine, from Jan 2013 to September 2013. Similarly, we cleaned the data and randomly selected a subset of users who submitted over 3,000 queries during this period. As a result, we collected 1,475 users with 1.9 million queries, and their activities span from 54 days to 9 months.

Model Fitness. Table 11 shows the log predictive likelihood on events falling in the final 10% of the total time of query data. According to Table 11, LDA-Hawkes fits both synthetic and real-world data better than TW and Word-Related. This illustrates that a Hawkes process better utilizes the temporal information in benefiting LDA's learning of textual data than simply considering the co-occurrence of queries within a time session or queries sharing at least one same word. The larger

Table 6: Log Predictive Likelihood on Both Synthetic and Real-world Data

Model/Data set	LDA-Hawkes	TW(5 min)	TW(1 week)	Word-Related
Small Synthetic	-110.32	-121.87	-168.40	-504.83
Small Event Noisy	-122.83	-135.23	-184.50	-536.21
Small Intensity Noisy	-127.36	-139.21	-192.23	-543.19
Large Synthetic	-163.84	-177.48	-239.04	-846.14
Large Event Noisy	-179.34	-193.05	-263.91	-880.04
Large Intensity Noisy	-184.27	-198.30	-270.92	-889.36
AOL	-153.12	-165.03	-221.32	-815.42
Yahoo	-192.36	-217.32	-275.74	-896.17

time-window TW use, the worse its performance will be. Time-window based LDA models generally perform better than Word-Related. Word-Related performs the worst, which illustrates using lexicon-similarity only is far from enough for grouping semantically related queries. On both noisy data sets, the performance of all models become worse. However, the decrease of the performance of LDA-Hawkes is smaller than that of TW and Word-Related, which demonstrates the robustness of our proposed model.

In addition, another experiment is conducted to study how well the proposed model can fit the temporal data of query logs. Figure 12 shows the Q-Q plot of the predictive query sequences based on Hawkes parameters inferred from `AOL` versus the real query sequence in `AOL`. If the distribution of the timestamps of the predictive query sequences and that of the real query sequence are similar, the points in the Q-Q plot will approximately lie around the diagonal. If these two distributions are linearly related, the points in the Q-Q plot will approximately lie on a line, but not necessarily on the diagonal. From Figure 12, we can find that LDA-Hawkes fits the temporal data of real-world query logs very well.

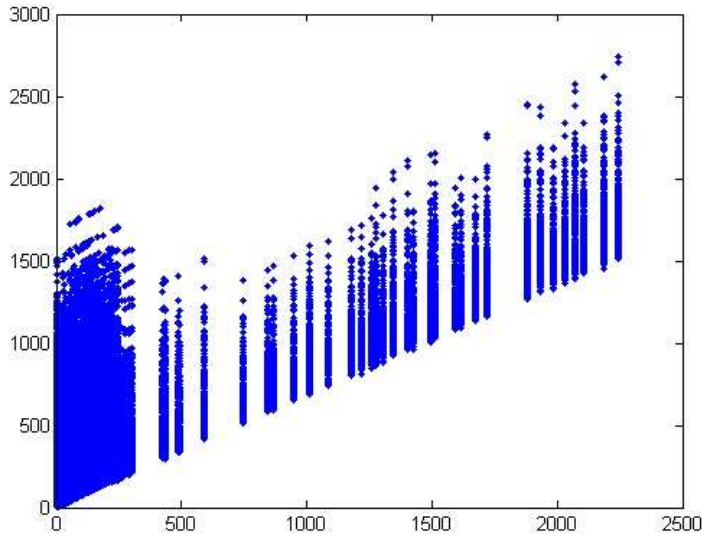


Figure 12: Q-Q plot of the preditive query sequence simulated with inferred Hawkes paramters versus the real query sequence.

5.3.3 Query Clustering

Along with search task identification, the proposed model simultaneously clusters queries into topics, and automatically labels identified search tasks, thus the performance of identifying and labeling search tasks mainly depends on how we cluster query words into different topics. In this series of experiments, we evaluate the quality of obtained query clusters/topics, which depends on their purity, or semantic coherence. Since no ground truth about the correct composition of a topic is available, we assess purity by the average similarity of each pair of queries within the same topic as:

$$\text{Purity} = \frac{1}{K}\sum_{k}\frac{\sum_{q_i,q_j\in t_k}\text{Sim}(q_i,q_j)}{N_k(N_k-1)/2} * 100\%,$$

where N_k is the number of queries in topic k.

We evaluate the query similarity based on their categorical labels from the Open Directory Project (ODP)[4]. The ODP , also known as DMOZ, is a human-edited directory of more than 4 million URLs. These URLs belong to over 590,000 categories organized in a tree-structured taxonomy where more general topics are located

[4]http://www.dmoz.org/

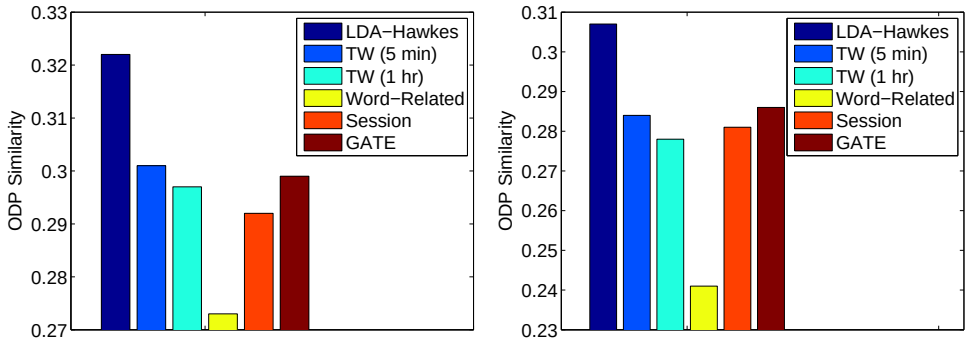
at higher levels. For instance, the URL {tech.groups.yahoo.com/group/amrc-l/} belongs to Top/Arts/Animation/Anime/Clubs_and_Organizations, {http://valleyofazure.tripod.com/} belongs to another directory Top/Arts/Animation/Anime/Characters. Hence, to measure how related two queries are, we can use a notion of similarity between the corresponding categories provided by ODP. In particular, we measure the similarity between category C_i of query q_i and category C_j of query q_j as the length of their longest common prefix $P(C_i, C_j)$ divided by the length of the longest path between C_i and C_j. More precisely, we define this similarity as:

ODP Similarity

$$\text{Sim}(q_i, q_j) = |P(C_i, C_j)| / \max(|C_i|, |C_j|),$$

where $|C|$ denotes the length of a path. For instance, the similarity between the two queries above is 3/5 since they share the path "Top/Arts/Animation" and the longest one is made of five directories. We evaluate the similarity between two queries by measuring the similarity between the most similar categories of the two queries, among the top 5 answers provided by ODP.

Figure 13 compares the purity of topics detected by LDA-Hawkes, alternative probabilistic models, and state-of-the-art query clustering approaches on AOL and Yahoo data sets. We can find that LDA-Hawkes outperforms all compared approaches. It improves over the second best method by up to 10%. `Gate` and TW(5 min) take the second place, both of them are slightly better than `Session-Similarity` and TW(1 hr), which again demonstrates that a small time window better benefits the LDA model in detecting semantically related queries. Word-Related performs significantly worse than other methods, which shows that considering only the co-occurrence of queries sharing words is very limited. Meanwhile, we find that compared with TW, LDA-Hawkes, `Session-Similarity`, and `Gate` perform relatively better on



(a) AOL (b) Yahoo

Figure 13: Query Clustering measured by Topic Purity. This metric relies on ODP Similarity to evaluate the pairwise similarity between queries.

Yahoo data set, which implies that LDA-Hawkes works for various real-world query logs. Notice that the absolute value of topic purity is not very high, since the ODP categories are fine-grained, the categories of queries from the same search task are very likely to be different, but share paths, i.e. have common prefix.

5.3.4 Search Task Identification

To justify the effectiveness of the proposed model in identifying search tasks in query logs, we employ a commonly used AOL data subset[5] with search tasks annotated, and also recruit eight editors to annotate search tasks in a chosen subset from the Yahoo data, which contains 100 users with around 50 queries per user. We measure the performance by a widely used evaluation metric,

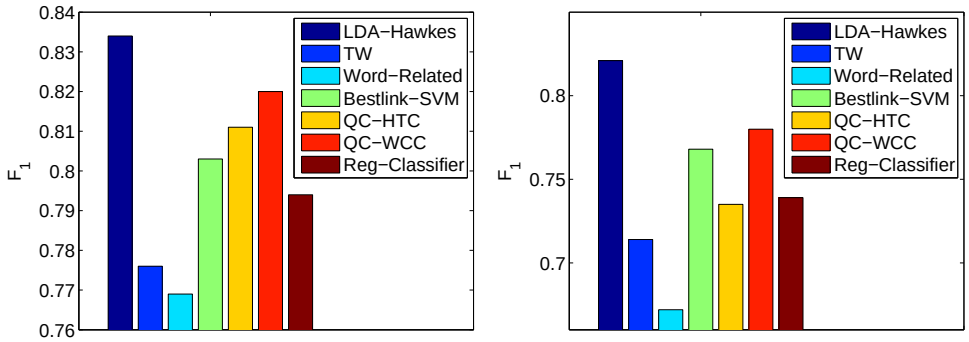
F_1 *score*

$$F_1 = \frac{2 * p_{pair} * r_{pair}}{p_{pair} + r_{pair}},$$

where p_{pair} denotes the percentage of query-pairs in our predicted search tasks that also appear in the same ground-truth task, while r_{pair} denotes the percentage of query-pairs in the ground-truth tasks that also appear in the same predicted task.

Figure 28 compares the proposed model with alternative probabilistic models and

[5]http://miles.isti.cnr.it/ tolomei/?page_id=35.



(a) AOL (b) Yahoo

Figure 14: Performance Comparison of Search Task Identification measured by F_1 Score.

state-of-the-art search task identification approaches by F_1 score. Here among TW models with various time-window size, we only include the "5 min" sized Time-Window in comparison, since it performs the best in both model fitness and query clustering. From Figure 28, we find that LDA-Hawkes performs the best among all compared approaches, and outperforms the second best approach by over 5%. Furthermore, LDA-Hawkes outperforms baselines in terms of both accuracy and recall. TW and Word-Related perform the worst since their assumptions on query-relationship within the same search task are too strong. Moreover, LDA-Hawkes's advantage over Bestlink-SVM and Reg-Classifier illustrates that employing self-exciting point processes like Hawkes to utilize the temporal information in query logs can be a better choice than incorporating temporal information in features. The advantage over QC-HTC and QC-WCC demonstrates that appropriate usage of temporal information in query logs can even better reflect the semantic relationship between queries, rather than exploiting it in some collaborative knowledge.

Case Study of Identified Search Tasks. In this part, we try to show a few examples that LDA-Hawkes identify and label search tasks in Yahoo query log. From Figure 15, we can find that both the word co-occurrence and temporal gap play a important role in predicting *influence* among sequential queries. Although chances

are very small that queries "aircract carrier" and "aircraft carrier" will co-occur, we predict an *influence* between them, since they are temporally close. On the other hand, query-pair "tibet" and "ryukyu islands", and query-pair "aircraft carrier" and "battleships us" are not consecutive, however, we predict that *influence* exist between those pairs of queries, as they co-occur in quite a few number of users' query sequences. Thus we may conclude that the existence of *influence* demands both temporal and sematic closeness. Queries linked by *influence* belong to the same search task since the user's *information need* is not satisfied by the former query, which makes the user additionally issue the later semantically related query, whose occurrence violates that user's regular query submission frequency. The figure also shows that LDA-Hawkes is able to assign the same label to different search tasks which are semantically related, despite that the temporal gap between them are very long.

5.4 Summary

In this chapter, we have presented a novel probabilistic model to integrate the L-DA model with Hawkes process for identifying and labeling search tasks. Basically, Hawkes processes utilize its self-exciting property to identify search tasks if influence exists among a sequence of queries for individual users, while the LDA model exploits query co-occurrence across different users to discover the latent information needed for labeling search tasks. By leveraging the temporally weighted query co-occurrence, our model not only guarantees sound performance by making full use of both textual and temporal information of the entire query sequences, but also enables the labeling of the identified search tasks since semantically related queries are clustered togeth-er through query links determined by co-occurrence. We have applied the proposed LDA-Hawkes model to analyze search tasks on both AOL and Yahoo query logs, and compare with several alternative approaches. Experimental results show that the improvements of our proposed model are consistent, and our LDA-Hawkes model

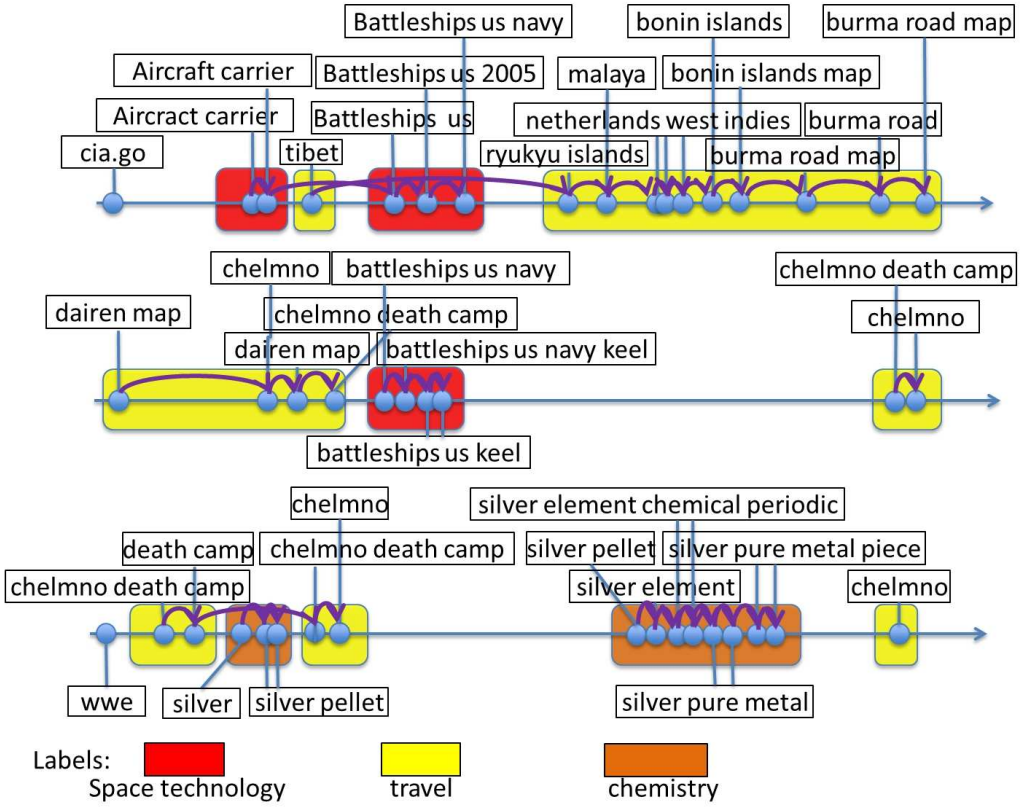


Figure 15: Case Study: Purple arrow line denotes the *influence* identified by the proposed model, rounded rectangle denotes the identified search tasks, rectangle denotes the labels our model assigns to search tasks.

achieves the best performance.

CHAPTER VI

ANALYZING USER'S SEQUENTIAL BEHAVIOR IN QUERY AUTO-COMPLETION VIA MARKOV PROCESSES

In the behavioral data studied in previous chapters, the underlying influence generally coincides with the self- & mutually exciting property that the occurrence of one event raises the probability of the occurrence of events in the near future. On the other hand, the influence in some behavioral data owns different properties, such as the Markov property. In this chapter, we consider one typical behavior — users' sequential interactions with search engines in the procedure of query auto-completion. Query auto-completion (QAC) plays an important role in assisting users typing less while submitting a query. The QAC engine generally offers a list of suggested queries that start with a user's input as a prefix, and the list of suggestions is changed to match the updated input after the user types each keystroke. Therefore rich user interactions can be observed along with each keystroke until a user clicks a suggestion or types the entire query manually. It becomes increasingly important to analyze and understand users' interactions with the QAC engine, to improve its performance. Existing works on QAC either ignored users' interaction data, or assumed that their interaction at each keystroke is independent from others.

In this chapter, we pay high attention to users' sequential interactions with a QAC engine in and across QAC sessions, rather than users' interactions at each keystroke of each QAC session separately. Analyzing the dependencies in users' sequential interactions improves our understanding of the following three questions: 1) how is a user's skipping/viewing move at the current keystroke influenced by that at the

previous keystroke? 2) how to improve search engines' query suggestions at short keystrokes based on those at latter long keystrokes? and 3) facing a targeted query shown in the suggestion list, why does a user decide to continue typing rather than click the intended suggestion? We propose a probabilistic model that addresses those three questions in a unified way, and illustrate how the model determines users' final click decisions. A variational inference algorithm is designed for parameter estimation of the proposed model. We evaluate our method based on real-world QAC logs. By comparing with state-of-the-art methods, our proposed model does suggest queries that better satisfy users' intents.

6.1 Modeling User's Sequential Behavior in Query Auto-Completion

In this section, we first introduce the concept of high-resolution QAC log, and then propose appropriate models to predict how likely a user will click a certain query at a certain location in a QAC session.

6.1.1 A High-Resolution QAC Log

Traditionally, the search query log only includes the submitted query and its associated search results, while it does not contain the sequential keystrokes (prefixes) user typed in the search box, as well as their corresponding QAC suggestions. In order to better analyze and understand real users' behavior, a high-resolution QAC log is introduced and analyzed in [156], which records users' interactions with a QAC engine at each keystroke and associated system respond in an entire QAC process. For each submitted query, there is only one record in a traditional search query log. However, in the high-resolution QAC log, each submitted query is associated with a **QAC session**, which is defined to begin with the first keystroke a user typed in the search box towards the final submitted query. The recorded information in each QAC session includes each keystroke a user has entered, the timestamp of a keystroke, the

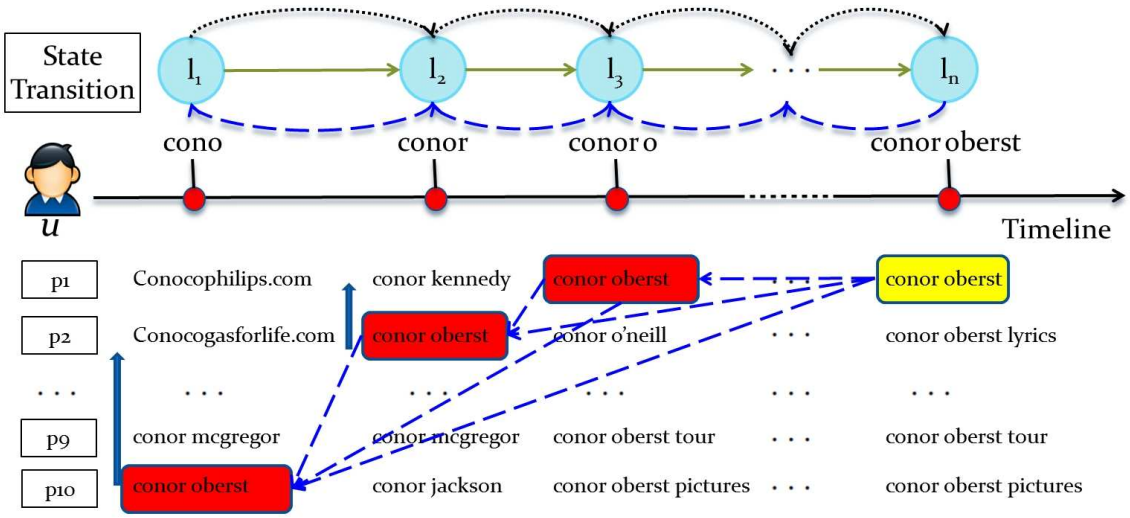


Figure 16: A Toy Example of a QAC Session in High-resolution QAC logs. Yellow tag highlights the query a user finally clicks, red tag highlights the user's intended query he/she doesn't click. Black dot line represents the dependency between users' skipping/viewing states captured by Markov process, and blue line denotes the influence of suggested query lists of latter keystrokes together with users' final click choices to the raise of the ranking of intended queries in the list of the current keystroke.

corresponding top 10 suggested queries to a prefix, the anonymous user ID, and the final clicked query.

Let us take a toy example to briefly introduce how a user interacts with a QAC engine and makes the final click in an entire QAC session. As shown in Figure 16, a **QAC session** contains S keystrokes and each keystroke has a suggested query list of length D.[1] A QAC session ends at the keystroke where the user clicks a query in the suggested query list, or when the prefix at that keystroke is exactly the query the user enters into the search engine. Among the $S \times D$ *slots* in each QAC session, where each slot q_{ij} is indexed by the i-th keystroke and the j-th position in the associated list, a user clicks at most one of them, although the user intended query may appear in many slots. Since users' clicked queries are usually their intended queries, appropriate modeling of users' click actions can be a good solution of the QAC problem. The ideal QAC engine should be able to rank the user intended query higher with less

[1] We experiment with real-world QAC logs where $D = 10$.

keystrokes or short prefixes. In this work, we leverage such a QAC log data to get better understanding of user sequential behavior in the QAC process.

6.1.2 Assumptions on QAC User Behavior

We view the QAC problem, which predicts a user's intended query, as the problem of predicting the query the user will click. Unlike existing works on query auto-completion, which paid no attention to those non-clicked suggested queries at the keystrokes before users' make clicks, or failed to straightforwardly analyze and reveal the difference between click cases and non-click cases, we propose a model which predicts the most likely slot a user will click in each QAC session by capturing the relationship between users' behaviors at different keystrokes.

To predict how likely a user will click a certain slot, there are mainly three issues we need to solve:

- Whether the user has viewed the slot;
- Whether the query shown on the slot satisfies the user's intent; and
- Whether the user is willing to click the slot.

We use Figure 17 to illustrate how the above three issues together determine users' click choices among all potential slots. Figure 17 shows a QAC session which contains $S(=4)$ columns/keystrokes, and each keystroke contains $D(=4)$ positions, thus makes a total number of 16 potential slots to click. Among those 16 potential slots, the queries in 12 of them do not satisfy the user's intent, thus the user will not click it anyway. On the other hand, the user's intended query appears in the other four slots, where each keystroke contains one appearance. The user viewed the suggested query lists at both the first and second keystrokes, however, the intended query is ranked at a relatively low position, which the user failed to pay attention to, or thought this position costs him/her too much effort to click, thus he/she didn't click it. At the

third keystroke, although the intended query is ranked at the top position, the user didn't view this suggestion, thus he/she missed the information and failed to click. Finally, at the last keystroke, the user viewed it, and found his/her intended queries lie at the top position, then he/she clicked that query.

6.1.3 Modeling Clicks in Query Auto-Completion

In the following, we show how to better address the above issues by taking the relationship between users' behaviors at different keystrokes into account. We also introduce some analysis conducted on the real-world high-resolution QAC log, to show why our proposed ideas are reasonable.

Skipping or viewing a keystroke? Existing works [32, 27] mostly failed to address the first (and the third) issue since they lack the information of users' interactions with a QAC engine before making a click, i.e., the high-resolution QAC log. Generally, it is assumed that a user makes a decision to either viewing or skipping at each keystroke, which depends only upon the user's interaction with the QAC engine at that keystroke. Based on its definition that "skipping behavior happens when the final clicked query is ranked within top 3 in the suggestion list of any of the prefixes except the final prefix", we count the sequential appearance of skipping and viewing states statistically on a real-world high-resolution QAC log collected from a commercial search engine, and find that 78.4% of the states are followed the same type of states. Thus the transitions between skipping and viewing states are not random, and the error of inferring state type based only upon user behavior at that keystroke can be significant. Thus, our work, on the other hand, assumes that the user's decision will also be influenced by his/her decision at the previous keystroke besides his/her interaction. As shown in Figure 16, we use a hidden Markov model to capture such influence, and take viewing and skipping as $K = 2$ states. We assume that T state

transition rules exists in the observed QAC logs, where each rule is actually a probability matrix corresponding to a certain type of viewing/skipping behavior. Users' interactions with the QAC engine, such as typing speed and reaching word boundary, at the current keystroke influence their choice of transition rules at this keystroke. For instance, for a fast typing user, the probability of transition from skipping to skipping can be very small. When a user reaches a word boundary, he is very likely to follow the transition rule where the probability of transition from skipping to viewing is significant. Assuming T user interaction patterns lie in the entire QAC log, and each user interaction pattern corresponds to one state transition rule. Then at each keystroke, we analyze a user's interaction with the QAC engine and find which interaction pattern it belongs to, then choose accordingly the corresponding move transition patterns. Since at each keystroke, typing speed, word

How to capture users' intent? We adopt logistic regression (LR) to model how likely a query satisfies a user's intent under the current prefix. Existing works [156] already demonstrated a set of relevance features, which characterize the relevance between a certain prefix-query pair, to be effective in predicting the user's intent given certain prefixes. However, we find those features not enough due to two drawbacks: 1) those features mainly reflect the general interest of majority of users, and care little about users' personal history and preference; and 2) those features rarely imply the relationship between different prefixes, which makes them difficult to utilize users' preference of queries in the suggested lists of other keystrokes to improve the ranking at the current keystroke. As shown in Figure 16, a user's intended query, should get a higher rank at previous keystrokes where this query appears, despite that it is not clicked at that time. Actually, the real-world QAC log shows that 29.4% of users' submitted queries (this number counts redundant appearances) have been submitted more than 3 times by a user, while among those queries, only 18.4% of them has been submitted multiple times by more than 25% of users, i.e. different users favor

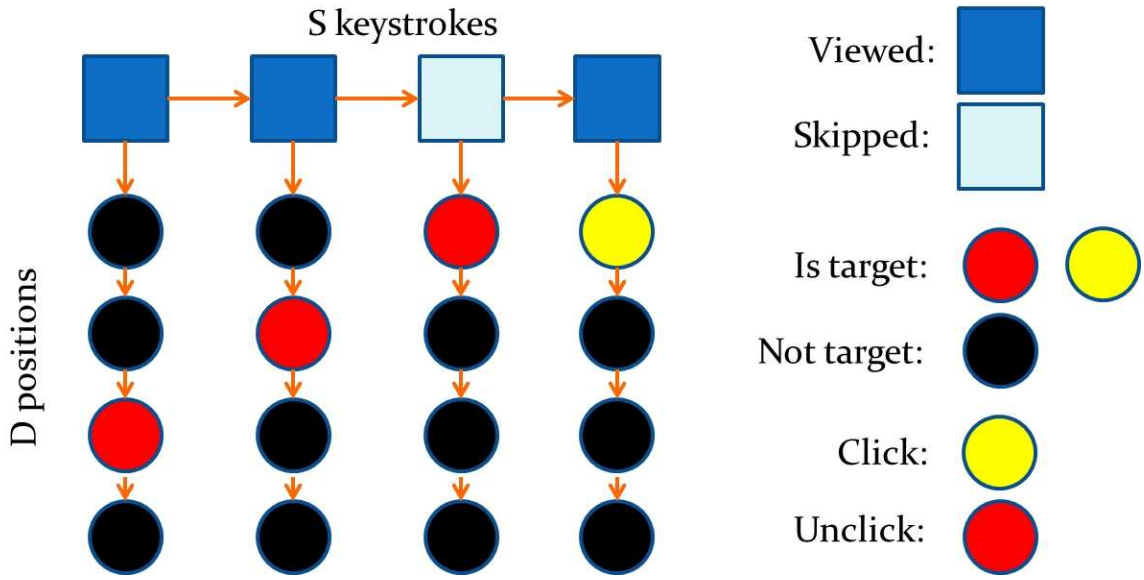


Figure 17: How Users Choose to Click Suggested Queries.

different query sets. Thus, besides existing features, we also employ a set of user-specific relevance features, which are designed to capture users' personal preference of queries and their corresponding relationship with keystrokes.

We summarize these search behavior features in Table 7. Our features generally originate from statistical counting of users' interactions with the QAC engine in their own historical sessions. For each user, given a certain query, we measure the number of times the same query has been clicked by that user in the past (denoted as Query Clicks). For users who have some queries daily issued, such as "facebook" or "youtube", this feature is capable of predicting his intent at the first few keystrokes. We also measure the average length of queries the user has clicked in the past (denoted as Query Length), the average number of words in queries that a user clicked (denoted as Query Word Number). In addition, we define the ratio between the length of a prefix and that of a query as *Prefix/Query Length Ratio (PQLR)*, and calculate the distribution of the associated PQLR of queries the user clicked in the past. For each new coming query, we estimate the percentage of the appearance of the associated PQLR in the user's history (denoted as Prefix/Query Ratio).

Table 7: User-Specific Relevance Features

Feature x	Description
Query Length	The average length of queries a user clicked.
Query Word Number	The average number of words in queries that a user clicked.
Query Clicks	The number of clicks a user makes on the current query.
Prefix/Query Ratio	The percentage of the appearance of $PQLR$ of the current prefix-query pair in the history.

Clicking a position or continue typing? In addressing the third issue, we find it very necessary to take a user's tendency of viewing and clicking a certain position or continuing typing into consideration. On the real-world QAC log, we find that when a user's intended query (the click this user finally clicks in a QAC session) is ranked within the top 2 positions, 37.6% of them will be clicked by users. On the other hand, if this intended query is ranked out of the top 2 positions, only 13.4% of them will be clicked by users. Furthermore, such tendencies can be very different for different groups of users. For PC users, 42.9% of intended queries will be clicked if they are ranked within the top 2 positions, otherwise, 23.0% of them will be clicked. While for mobile users, 35.1% of intended queries will be clicked if they are ranked within the top 2 positions, otherwise, 11.7% of them will be clicked. Here we can clearly find that on mobile, users' are more likely make clicks when their intended queries appear on the suggestion lists, even if they are ranked at low positions, while PC users prefer typing, since typing on PC is much more convenient than on mobile.

We use a D-dimensional Dirichlet prior to learn the relative cost ratio of viewing and clicking a certain position against typing. Such a prior allows each user to have distinguished position bias in the preference of viewing and clicking than typing. Users who prefer clicking than typing will have a higher average *clicking/typing cost ratio*

(CTCR) than users who prefer typing than clicking. The gaps of CTCR between high position and low position of users' who rarely click suggested queries in low positions are generally larger than those of users who tend to click their intended queries the first time they are suggested no matter how lower their positions are.

Let us consider a typical scenario where M users issue M corresponding query sequences, where for each user m, the QAC log records N_m QAC sessions. The n-th QAC session of user m contains $S_{m,n}$ keystrokes, where each keystroke contains D suggested queries. In total, each QAC session has $S \times D$ potential slots where a user can click, and we use $z_{m,n,s,d} = 1$ to denote that a user clicks the slot ranked d at the s-th keystroke, and $z_{m,n,s,d} = 0$ otherwise. We also use $x_{m,n,s,d}$ to denote the relevance feature of a prefix-query pair, where the prefix is at the s-th keystroke and the query is ranked d at that keystroke. Then given the weights ω of those relevance features, we can derive the relevance score of each suggestion as $\omega x_{m,n,s,d}$. In addition, we denote $x'_{m,n,s}$ as a user's interaction with the QAC engine at the s-th keystroke.

Based on our proposed solution of the listed three issues and above definition, finally, we present our generative model as follows:

- For each user m, draw a D dimensional membership vector $\beta_m \sim \text{Dirichlet}(\alpha)$.
- For each move transition pattern t, draw a move transition matrix $\delta_t \sim \text{Dirichlet}(\alpha')$, where each pattern is associated with a user interaction pattern ω'_t.
- For the n-th query issued by user m, and for the prefix at step s,
  - Draw the user's move transition pattern membership $\pi_{m,n,s} \sim \text{Multinomial}(\theta)$, where θ is the prior distribution of move transition pattern membership.
  - Draw the user's interaction with QAC engine through $x'_{m,n,s} \sim \text{Gaussian}(\omega'_{\pi_{m,n,s}}, \sigma)$.

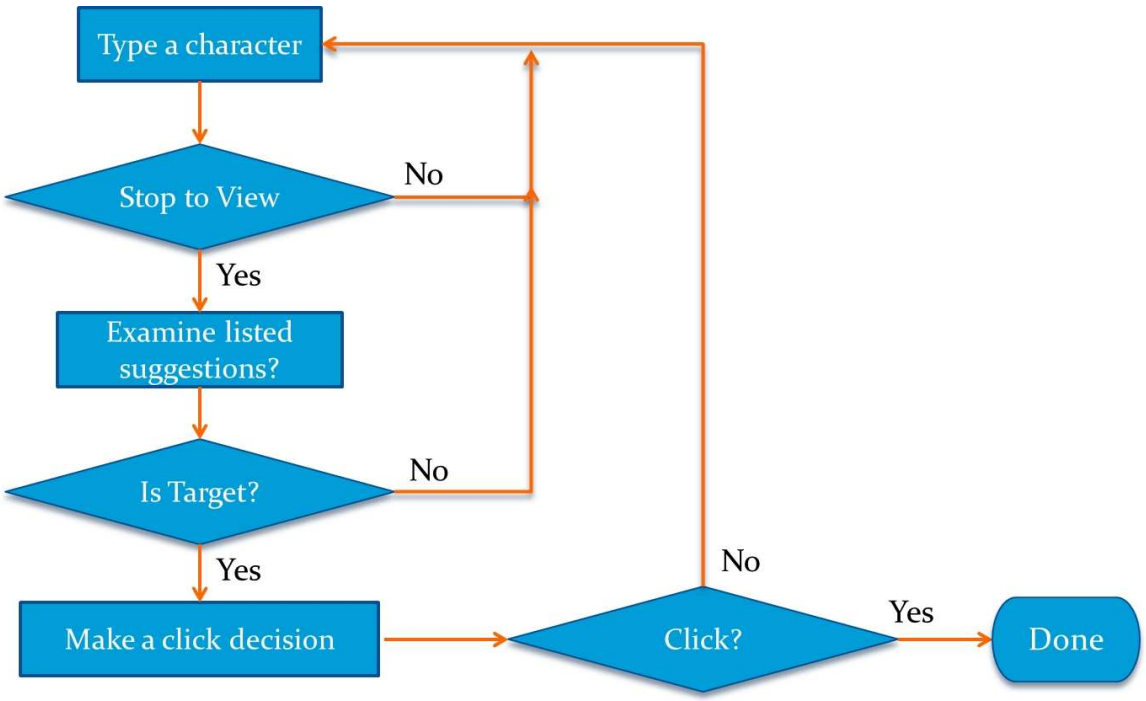


Figure 18: RBCM Flowchart.

- Draw the user's next move, which is either type or view, $Y_{m,n,s} \sim \text{Multinomial}(\delta_{\pi_{m,n,s-1},Y_{m,n,s-1}})$. If we have $Y_{m,n,s} = 1$, continue to type; otherwise, stop and view the results.
- For each position d in the suggestion list of the current prefix, draw the user's clicked suggestion through $z_{m,n,s,d} \sim \text{Multinomial}(\text{LR}(\beta_{m,d}\omega x_{m,n,s,d}))$. If we have $z_{m,n,s,d} = 1$, select the suggestion in position d to click, then go to the $n+1$-th query; otherwise, continue to type, i.e., go to the next prefix.

Notice that the proposed model is a combination of three parts that address the listed issues respectively. We name this probabilistic model as relationship-based click model (RBCM). To better illustrate the generative process of the proposed RBCM model, we show the flowchart of user behaviors in Figure 18.

Under our RBCM model, the joint probability of data $Z = \{\{z_{m,n}\}_{n=1}^{N_m}\}$, $X = \{\{x_{m,n}\}_{n=1}^{N_m}\}$, $X' = \{\{x'_{m,n}\}_{n=1}^{N_m}\}$ and latent variables $\{\beta_{1:M}, \pi, Y\}$ can be written as

follows:

$$
\begin{aligned}
& p(Z, X, X', \beta_{1:M}, Y, \pi | \alpha, \alpha', \omega, \omega', \theta) \\
= & \prod_{m,n,s,d} P(z_{m,n,s,d} | x_{m,n,s,d}, \beta_{m,d}, \omega, Y_{m,n,s}) \\
& \times \prod_{m,n,s} P(Y_{m,n,s} | Y_{m,n,s-1}, \delta, \pi_{m,n,s-1}) P(x'_{m,n,s} | \pi_{m,n,s-1}, \omega') \\
& \times \prod_{m,n,s} P(\pi_{m,n,s} | \theta) \prod_{t} P(\delta_t | \alpha'_t) \prod_{m} P(\beta_m | \alpha).
\end{aligned}
$$

6.2 Inference

Despite that a tremendous amount of work on inference of topic models have been published, none of them are designed to address topic model combined with point processes. In this section, we derive a mean-field variational Bayesian inference algorithm for our proposed RBCM model.

6.2.1 Variational Inference

Under the RBCM model, given observations of both click information $Z = \{Z_m\} = \left\{\{z_{m,n}\}_{n=1}^{N_m}\right\}$, the relevance features X, and user interaction features X', the log-likelihood for the complete data is given by $\log P(Z, X, X' | \alpha, \alpha', \omega, \omega')$. Since this true posterior is hard to infer directly, we turn to variational methods [38], whose main idea is to posit a distribution over the latent variables with variational parameters, and find the settings of the parameters so as to make the distribution close to the true posterior in Kullback-Leibler (KL) divergence. In Figure 19, the lower part shows the variational distribution that approximates the data likelihood. We choose to introduce a distribution of latent variables q specified as the mean-field fully factorized family as follows:

$$
\begin{aligned}
& q(Y, \pi, \beta_{1:M} | \rho, \phi, \gamma_{1:M}) \\
= & \prod_{m} \prod_{n} \prod_{s} q_1(Y_{m,n,s} | \rho_{m,n,s}) q_1(\pi_{m,n,s} | \phi_{m,n,s}) \prod_{m} q_2(\beta_m | \gamma_m)
\end{aligned}
$$

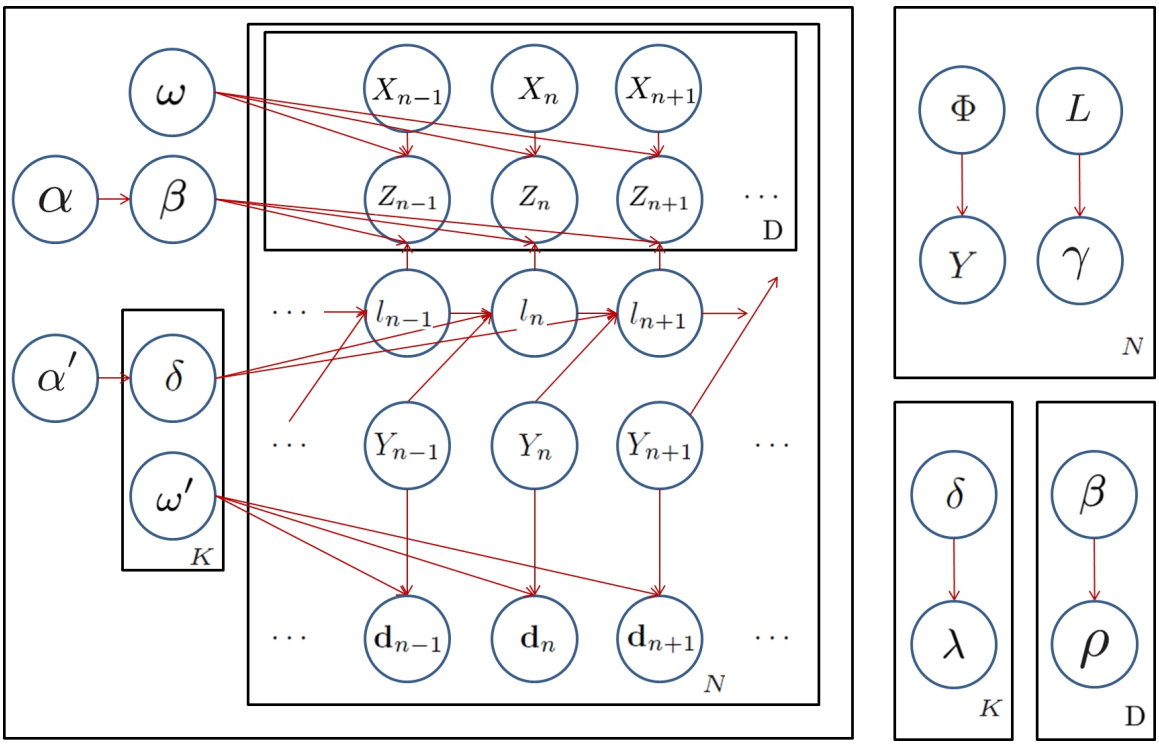


Figure 19: Graphical model representation of RBCM and the variational distribution that approximates the likelihood. The left box shows the graphical model representation of RBCM, while the right boxes show the variational distribution that approximates the likelihood.

where q_1 is a multinomial, q_2 is a Dirichlet, and $\{\rho, \phi, \gamma_{1:M}\}$ are the set of variational parameters. We optimize those free parameters to tight the following lower bound $\mathcal{L}'$ for our likelihood:

$$\begin{aligned}\log p(Z, X, X'|\alpha, \alpha', \omega, \omega', \theta) \geq & E_q \left[\log p(Z, X, X', \beta, Y, \pi|\alpha, \alpha', \omega, \omega', \theta)\right] \\ & - E_q \left[\log q(Y, \pi, \beta|\rho, \phi, \gamma_{1:M})\right]. \end{aligned} \tag{14}$$

Under a coordinate descent framework, we optimize the lower bound as in Eqn (15) against each variational latent variable and the model hyper-parameter. For variational latent variables, we have the following process

- update rules for γ's as:

$$\gamma_{m,d} = \alpha_d + \sum_n \sum_s \log\left(1 + \exp\left(z_{m,n,s,d}\omega_m x_{m,n,s,d}\right)\right);$$

- update rules for ρ's as:

$$\rho_{m,n,s,k=1} \propto \exp\left(-\sum_{d} z_{m,n,s,d} - b_{m,n,s,1}\right);$$

$$\rho_{m,n,s,k=2} \propto \exp\left(b_{m,n,s,2} - \sum_{d} \log\left(1 + \exp\left(z_{m,n,s,d} - \omega_m x_{m,n,s,d}\left[\Phi(\gamma_{m,d}) - \Phi\left(\sum_{d} \gamma_{m,d}\right)\right]\right)\right)\right).$$

where

$$b_{m,n,s,k} = \sum_{k'}\sum_{t} \phi_{m,n,s,t}\left[\Phi(\eta_{t,k',k}) - \Phi\left(\sum_{k''} \eta_{t,k',k''}\right)\right] + \sum_{k'}\sum_{t} \phi_{m,n,s+1,t}\left[\Phi(\eta_{t,k,k'}) - \Phi\left(\sum_{k''} \eta_{t,k.k''}\right)\right]$$

- update rules for ϕ's as:

$$\phi_{m,n,s,t} \propto \exp\left(\sum_{k,k'} \rho_{m,n,s,k}\rho_{m,n,s+1,k'}\left[\Phi(\eta_{t,k,k'}) - \Phi\left(\sum_{k''} \eta_{t,k,k''}\right)\right] - \frac{1}{2\sigma^2}\left\|x'_{m,n,s} - \omega'_t\right\|_2^2\right)$$

- update rules for η's as:

$$\eta_{t,k,k'} = \alpha'_t + \sum_{m}\sum_{n}\sum_{s} \phi_{m,n,s,t} \sum_{k,k'} \rho_{m,n,s,k}\rho_{m,n,s-1,k'}$$

6.2.2 Learning

We use a variational expectation-maximization (EM) algorithm [69] to compute the empirical Bayes estimates of the topic model hyper-parameters α and α' in our RBCM model. This variational EM algorithm optimizes the lower bound as in Eqn (15) instead of the real likelihood, and iteratively approximates the posterior by fitting the variational distribution q and optimizes the corresponding bound against the parameters.

In updating α, we use a Newton-Raphson method, since the approximate maximum likelihood estimate of α doesn't have a closed form solution. The Newton-Raphson method is conducted with a gradient and Hessian as follows:

$$\frac{\partial \mathcal{L}'}{\partial \alpha_d} = N\left(\Psi\left(\sum_d \alpha_d\right) - \Psi(\alpha_d)\right) + \sum_m \left(\Psi(\gamma_{m,d}) - \Psi\left(\sum_d \gamma_{m,d}\right)\right),$$

$$\frac{\partial \mathcal{L}'}{\partial \alpha_{d_1}\alpha_{d_2}} = N\left(\mathbb{I}_{(d_1=d_2)}\Psi'(\alpha_{d_1}) - \Psi'\left(\sum_d \alpha_d\right)\right).$$

Similar update rules can be derived for α'.

On the other hand, to obtain the approximate maximum likelihood estimation of ω, we employ the stochastic gradient descent to update ω in each interaction based on the observed click data z, relevance features x, and the inferred latent variables ρ and γ. On the other hand, the approximate maximum likelihood estimation of ω' will lead to the following update rule,

$$\omega'_t = \frac{\sum_m \sum_n \sum_s \phi_{m,n,s,t} x'_{m,n,s}}{\sum_m \sum_n \sum_s \phi_{m,n,s,t}}$$

Our variation inference algorithm, named RBCM, can be interpreted intuitively in the following way. The CTCR distribution γ of each user is determined by both the topic prior and the accuracy that the learned weights of relevance features predict users' intended queries. Users' viewing/skipping states ρ at each keystroke is determined by the influence from users' states at the previous keystroke and that from users' states at the next keystroke. The state transition at each keystroke is determined by the transition prior, users' interaction patterns at that keystroke, and users' states at that keystroke and the keystroke before that. The probability of the interaction of a user m at the s-th keystroke in the n-th QAC session belonging to interaction pattern k is jointly determined by the state transition between the current and the next keystroke and the users' interaction at the current keystroke.

In our mean-field variation inference algorithm, the computational cost of the inference of variational variables is $O(N * \bar{S} * T + M * D)$, where $\bar{S}$ is the average

number of keystrokes in a QAC session, and $N = \sum_m N_m$ is the total number of QAC sessions in the entire high-resolution QAC log. The computational cost of the estimation of topic model hyper-parameters is $O(M * D)$. The computational cost of the estimation of weights of relevance features is $O(N * \bar{S} * D)$, while the cost of the estimation of user interaction patterns is $O(N * \bar{S} * T)$. Thus the total computational cost of our algorithm is $O(N * \bar{S} * (T + D))$. Since T and D are both small constants, we can view the computational cost as linear in the total number of keystrokes in all sessions in the QAC log.

6.3 Experiments

We evaluated our RBCM model on real-world data sets, and compared the performance with the following baselines: three alternative probabilistic models that only use two parts of the proposed model and four state-of-the-art QAC algorithms:

Alternative-A: This model does not use the hidden Markov model to capture the state transition of skipping/viewing moves. The state of skipping/viewing is determined by users' interactions with the QAC engine only.

Alternative-B: This model avoids using user-dependent features in the logistic regression part to capture users' real preference of suggested queries. It only utilizes user-independent features in the logistic regression part.

Alternative-C: This model avoids using a Dirichlet prior to model users' CTCR. Instead, it assumes users have no preference in clicking different positions as well as typing, i.e., the probabilities of clicking any positions and typing are all equal, and user's clicked suggestion is drawn via $z_{m,n,s,d} \sim \text{Multinomial}(\text{LR}(\omega x_{m,n,s,d}))$.

MPC [27, 217]: This method, named MostPopularCompletion, is a widely used baseline in Query Auto-Completion, and employed as one main feature in many QAC engines.

UBM [78]: This User Browsing Model proposes a number of assumptions on user

Table 8: Log Predictive Likelihood on Real-world Data

Model/Data set	Platform	RBCM	Alter-A	Alter-B	Alter-C	TDCM
LargeQAC	PC	-185.37	-192.83	-189.43	-191.84	-208.65
LargeQAC	Mobile	-177.95	-187.64	-184.91	-185.50	-195.04
SmallQAC	PC	-206.32	-218.14	-214.98	-216.69	-234.83

browsing behavior that allows the estimation of the probability of observing a document. It depends on statistical counting of prefix-query pairs, thus unable to predict unseen prefix-query pairs.

BSS [96]: This Bayesian Sequential State model uses a probabilistic graphical model to characterize the document content and dependencies among the sequential click events within a query with a set of descriptive features. This is a content-aware model which is able to predict unobserved prefix-query pairs.

TDCM [156]: This is a two-dimensional click model which emphasize two kinds of user behaviors. It consists of a horizontal model which explains the skipping behavior, and a vertical model that depicts the vertical examination behavior.

6.3.1 Real-world Data

We conducted extensive experiments on two real-world QAC logs collected from a commercial search engine. The first data set, which we name *LargeQAC*, contains QAC logs from May 2014 to July 2014. The collection consists of a sample of 7.4 million QAC sessions from about 40,000 users over a 3-month period. We randomly selected a subset of users who submitted over 500 QAC sessions during this period, and collected their corresponding search activities, including the anonymized user ID, query string, timestamp, and the clicked URL. As a result, we collected 3,954 users with 2.6 million queries, and their activities span from 22 days to 3 months. According to the platform each QAC session belongs to, we separate the entire data set into two subsets. One is PC, which contains 1.6 million QAC sessions, while the other is mobile phones, which contains 1.0 million QAC sessions.

The second data set is also collected from a commercial search engine. We name this data *SmallQAC* to distinguish it from the previous one. This data set is constituted of random sampled high-resolution QAC logs dating from Nov 2013 to Jan 2014. The log contains 125 thousand QAC sessions from PCs. Since existing QAC algorithms utilizing high-resolution QAC logs have already shown rich results on the QAC log, we utilize both data sets to evaluate our proposed model and compare with the state-of-the-art methods in the following section.

6.3.2 Model Fitness.

This section evaluates the fitness of our proposed model on real-world data, and compares our model with probabilistic model based methods. We split the data based on the time information: the QAC sessions occurred in the first 90% of the time period are used as the training data, while the remaining 10% used as the test data. Table 11 shows the log predictive likelihood on sessions falling in the final 10% of the total time of QAC log data. According to Table 11, RBCM fits the real-world data better than the three alternative probabilistic models and TDCM. This illustrates that in the proposed RBCM model, all three parts play an important role in capturing the relationship between users' behaviors at different keystrokes. Alter-A performs the worst among all three alternative probabilistic models, which shows the importance of using the Markov process to model the state transition between skipping and viewing. Alter-B performs the best among three alternative probabilistic models, which shows that user-specific features do not fully utilize the relationship between users' behaviors. The reason may be that, even when the relevance features and their associated weights fail to reflect users' real preference of suggested queries, the other two parts of the proposed model can reduce the harm of those mispredictions by inferring reasonable skipping/viewing states and user-specific CTCR. TDCM performs the worst, since it fails to utilize the relationship between users' behaviors from any aspects.

6.3.3 Query Auto-Completion.

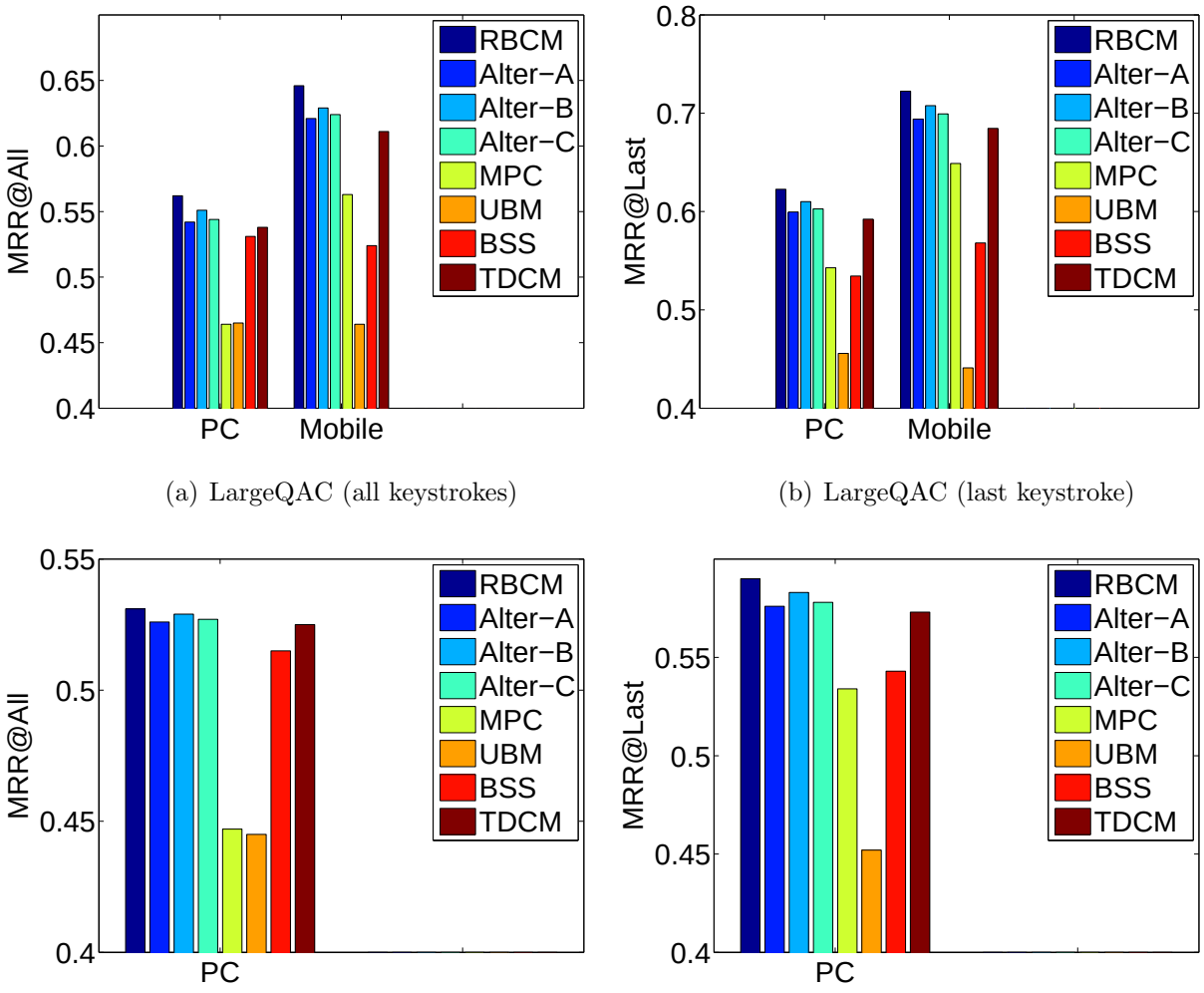
To evaluate the effectiveness of the proposed model in suggesting users intended queries in each QAC session, we compare the proposed model with both alternative probabilistic models and state-of-the-art QAC algorithms. We employ the Mean Reciprocal Rank (MRR) as the relevance measurement, which is a widely used evaluation metric in measuring QAC performance [27, 217, 156],

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q},$$

where Q is the set of queries a user finally submitted, and rank_q denotes the rank of the query q in the suggested query list.

Notice that among the suggested query lists of all keystrokes, those lists that do not contain users' finally submitted queries are removed from our experimental analysis. Since our experiments are conducted on high-resolution QAC data, we report both the average MRR score of all keystrokes, and the average MRR of the last keystroke only, since this is the keystroke where the users' click occurs. Notice that existing works which didn't make use of high-resolution QAC logs usually used the MRR of the last keystroke to measure their performance. In the following experiments, the whole dataset is divided evenly into a training set and a test set for different settings.

Figure 20 compares the proposed model with alternative probabilistic models and state-of-the-art QAC algorithms by MRR. We can observe that RBCM performs the best among all compared approaches, and outperforms existing QAC algorithms by over 6%, for all the data sets and different settings. In addition, the three alternative probabilistic models generally performs better than existing QAC algorithms. Such phenomenon demonstrates the effectiveness of making use of the relationship between users' interactions in different keystrokes in solving the QAC task. Essentially, appropriate modeling of such relationships together makes the proposed model much better



(a) LargeQAC (all keystrokes)

(b) LargeQAC (last keystroke)

(c) SmallQAC (all keystrokes)

(d) SmallQAC (last keystroke)

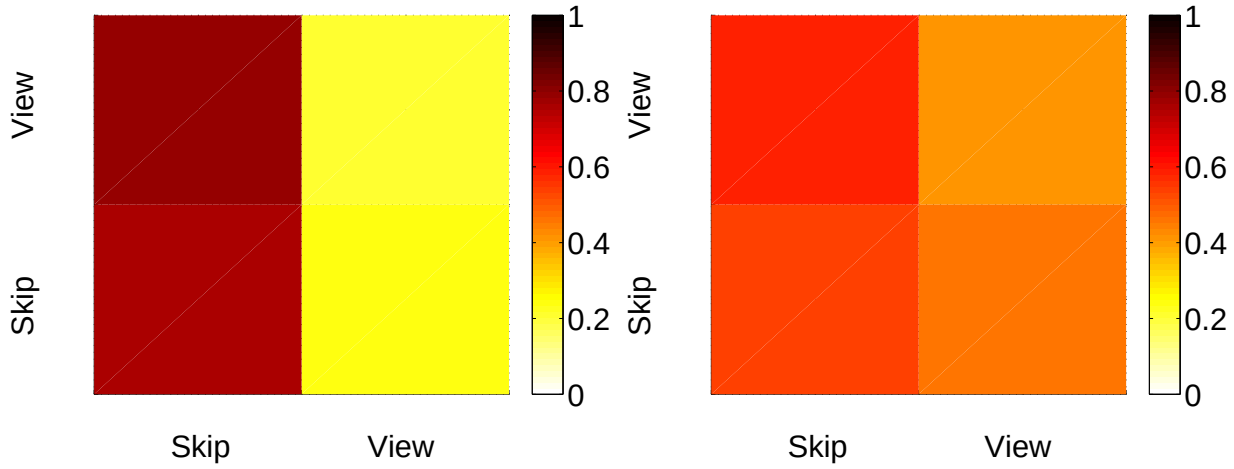
Figure 20: Performance Comparison of QAC Methods.

than those alternative baselines. Besides our proposed model and its alternatives, T-DCM performs better than the rest of existing QAC algorithms, which we attribute to the usage of high resolution QAC logs. BBS outperforms UBM since it adopts the content-aware relevance model. MPC performs the worse, since it pays little attention to users' behaviors in QAC logs. By comparing with the performance using all the keystrokes and last keystroke only, we find that the advantages of the proposed model are ever more significant when measured by MRR@All. It indicates that the proposed model can recommend user intended queries higher with less keystrokes.

6.3.4 State Transition of Skipping/Viewing.

Based on the state transition matrices and the corresponding user interaction patterns that learned by the proposed RBCM model from real world QAC logs, we provide a detailed analysis on the difference of transition rules between skipping/viewing and the associated user interaction patterns. Among all learned state transition rules, we pick two of them which differ the most in probability. Figure 21 shows the state transition rules and their corresponding user interaction patterns, where each block is the transition probability from previous state (vertical) to next state (horizontal). Intuitively, we find Figure 21(a) represents *skipping user*s, i.e., users who prefer skipping than viewing, as these users have a much higher probability to skip at the current keystroke no matter the previous state is skipping or viewing, while Figure 21(b) represents *viewing user*s, i.e., users who prefer viewing than skipping, because these users are more likely to switch to the viewing state from previous skipping or viewing status. Notice that *skipping* and *viewing user*s only refer to the tendency of users' at each keystroke. A user who always skips suggested query lists will behave like *skipping user* consistently, while a user who has no habit in querying may alternatively switch between *skipping* and *viewing user*s from time to time.

From Figure 21, we find that no matter what the state of the current keystroke is,



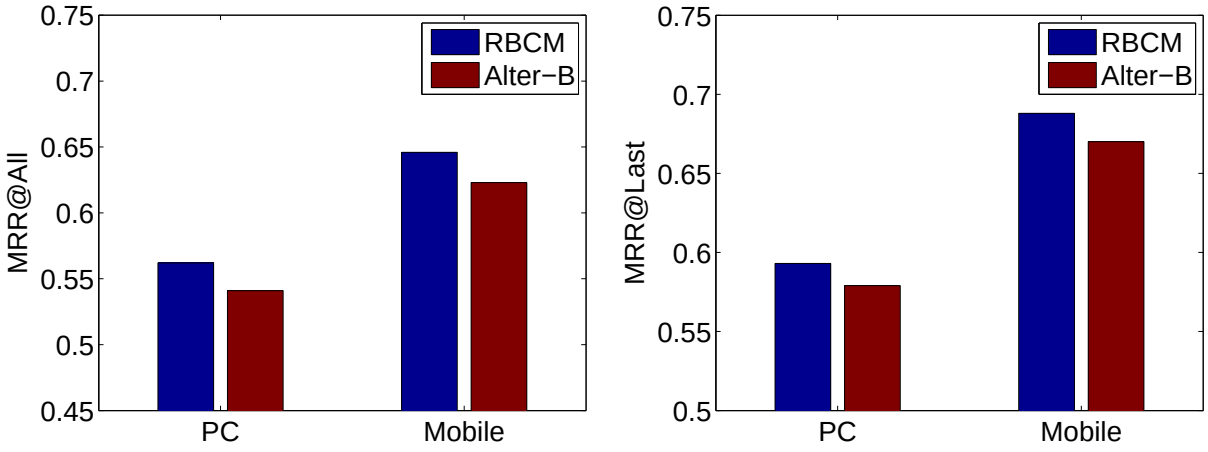
(a) Skipping User (b) Viewing User

Figure 21: State Transition of Skipping/Viewing.

*skipping user*s are more likely to skip the suggested query list of next keystroke than *viewing user*s. On the contrary, *viewing user*s are more likely to view the suggested query list of next keystroke than *skipping user*s under all circumstances. We also find that no matter which type a user belongs to, if he/she already viewed the suggested query list of the current keystroke, he/she will be more likely to skip the next keystroke than that under the situation where he/she skipped the current keystroke. Moreover, the corresponding user interaction patterns of different state transition rules appear quite different. *Skipping user*s generally have faster typing speed, and come across less word boundaries, and enter more navigational queries.

6.3.5 Users' Real Preference of Suggested Queries

. We use this series of experiments to discuss how our designed user-specific features enable the proposed model to understand users' real preference of suggested queries. We compare the proposed model with Alter-B on the QAC task measured by MRR, so as to illustrate the importance of using user-specific features. From Figure 22, we find that the proposed model performs better than Alter-B in recommending users' queries that satisfy their intent. In addition, we list a subset of learned weights of



(a) LargeQAC (all keystrokes)

(b) LargeQAC (last keystroke)

Figure 22: Comparison of RBCM with Alter-B.

those designed user-specific features in Figure 23. From here, we can find that the history of queries that a user clicked plays a very important role in predicting the future queries he/she will click, especially when certain prefixes are given. Meanwhile, the length of the queries a user used to click is a significant signal for learning users' real preference of queries in the suggested query lists. The reason can be that such a signal implies users' clicking habit from some aspects. A skipping user who always clicks long queries will probably ignore his/her intended queries shown to him/her, before he/she enters enough number of keystrokes. Under such situation, the signal of query length will be capable of capturing users' real preference at keystrokes with short prefixes, which enable the proposed model to rank the intended queries at higher positions than those shorter queries. On the other hand, when a user has no preference in clicking long queries, this signal will not take effect, and the proposed model will recommend popular queries according to the frequency of the occurrence of prefix-query pairs. Such suggested queries are usually not that much longer than the given prefixes.

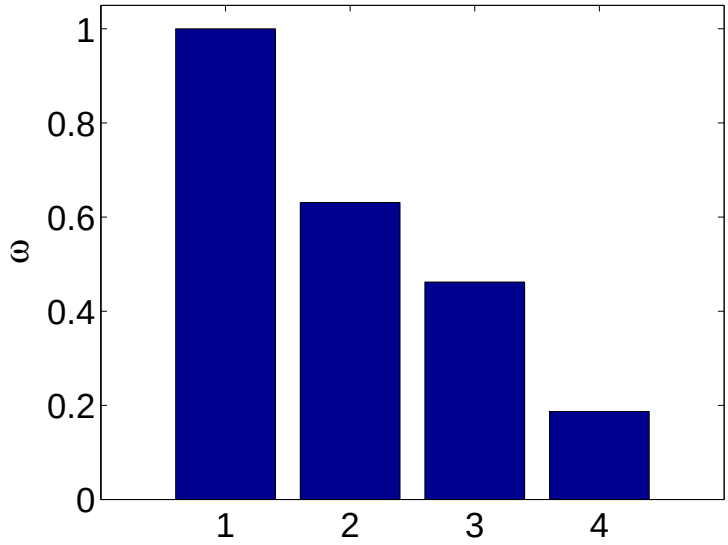
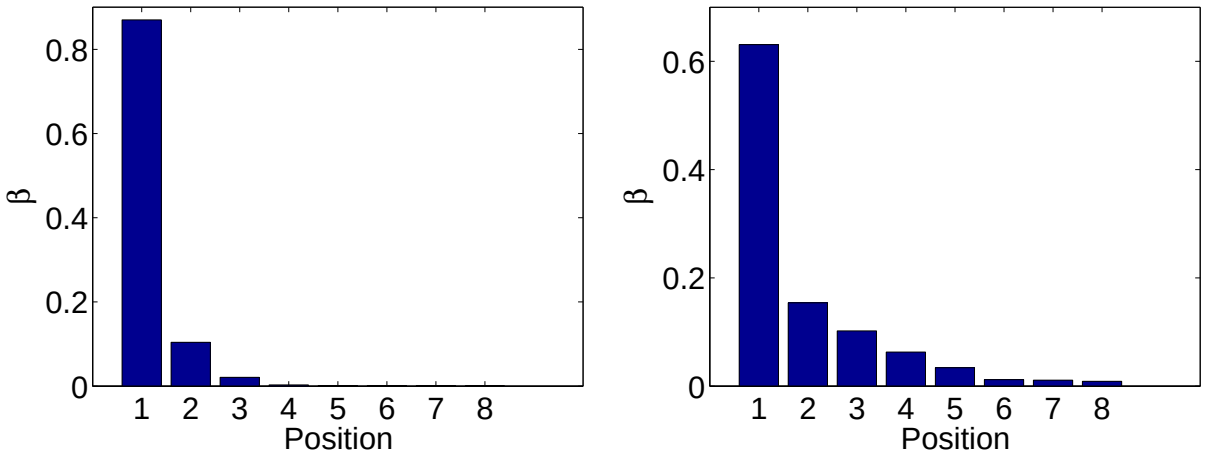


Figure 23: Weights of Relevance Features Learned by RBCM. Indices of selected user-specific relevance features: 1-Query Clicks, 2-Query Length, 3-Prefix/Query Ratio, 4-Query Word Number. The values of weights are scaled to the range of $[0, 1]$ to clarify the comparison of relative importance of different features.

6.3.6 User-specific Cost Between Position Clicking and Typing

. Based on the latent variable γ learned by the proposed model, we analyze the specific cost balance between position clicking and typing of different users. We select two subsets of users: one named *typing user*, which is formed by the top 20% users with the highest average click positions, and another, *clicking user*, is formed by the bottom 20% users with the lowest average click positions. We calculate the averaged *CTCR* of both two subsets of users, separately, and plot the results in Figure 24. From here, we find that the learned CTCR γ of *typing user*s and *clicking user*s have very different distributions. Although users from both subsets are most likely to click the top query of a suggested query list, *clicking user*s also occasionally clicks queries located at the middle positions of a suggested query list, while *typing user*s rarely click those positions.

Furthermore, we try to distinguish the difference between *typing user*&*clicking*



(a) Typing User (b) Clicking User

Figure 24: User-Specific Clicking/Typing Cost Ratio.

Table 9: Overlap between *typing user-skipping user* pair and *clicking user-viewing user* pair

Overlap	Percentage
typing user∩*skipping user*	34.2%
clicking user∩*viewing user*	22.7%

user and *viewing user*&*skipping user*. We select the top 20% users with the largest percentage of viewing states , and the bottom 20% users with the smallest percentage of skipping states. We compared the selected subsets of *typing user*s with *skipping user*s, and the selected subset of *clicking user*s with *viewing user*s. Table 9 shows the overlaps of the two pairs of compared subsets. According to Table 9, we find that there exists overlap between *typing user*s and *skipping user*s, *clicking user*s and *viewing user*s. For users sharing the same tendency of skipping/viewing state choices, the difference of users in choosing clicking and typing becomes smaller. Skipping and typing are two reasons that a user does not click his/her intended query when it appears. Skipping users are more likely to click upper position queries, but this is not always true. A user who owns a high typing speed may not only prefer skipping than viewing but also prefers typing than clicking. However, he/she may also click the intended query the first time he/she views it.

6.3.7 Case Study of Query Auto-Completion.

Now we show a few examples that illustrate how RBCM recommends users better intended queries by capturing the relationship between users' behaviors at different keystrokes. Figure 25 shows a QAC session where user finally submit "star wars", which is the user's intended query in this session. From Figure 25, we can find that the proposed model generally ranks users' intended queries higher than TDCM, especially at keystrokes with shorter prefixes. For example, with the prefix 'st', RBCM ranks the intended query at the position 3 while TDCM ranks it at the position 10. The reason is that the proposed model utilize the user's preference of the clicked queries at the last keystroke to improve its ranking at the previous keystrokes by modeling the relationship between users' behaviors at different keystrokes. Our designed relevance features show that the query "star wars" has been issued many times by this user. Thus, although for the entire user collection, the generally frequencies of the appearances of "star wars" are relatively low given short prefix, such as "st" or "sta", the proposed model ranks this query at the higher position for the specific user than other users which rarely search "star wars". We also notice that queries of similar intent, such as "star wars the old republic" and "star wars episode 7" are also ranked higher by our proposed model than by TDCM, which emphasizes that our model can better capture users' personal interests. Actually, through the analysis of the historical log of this particular user, we can find that he/she is a science fiction fan, which explains why our model also ranks "star trek" higher than TDCM. Thus we can conclude that appropriate modeling of such relationships is critical for predicting users' intended queries by typing less keystrokes.

6.4 Summary

In this chapter, we have presented a probabilistic model to solve the query auto-completion (QAC) task by capturing the relationship between users' behaviors at

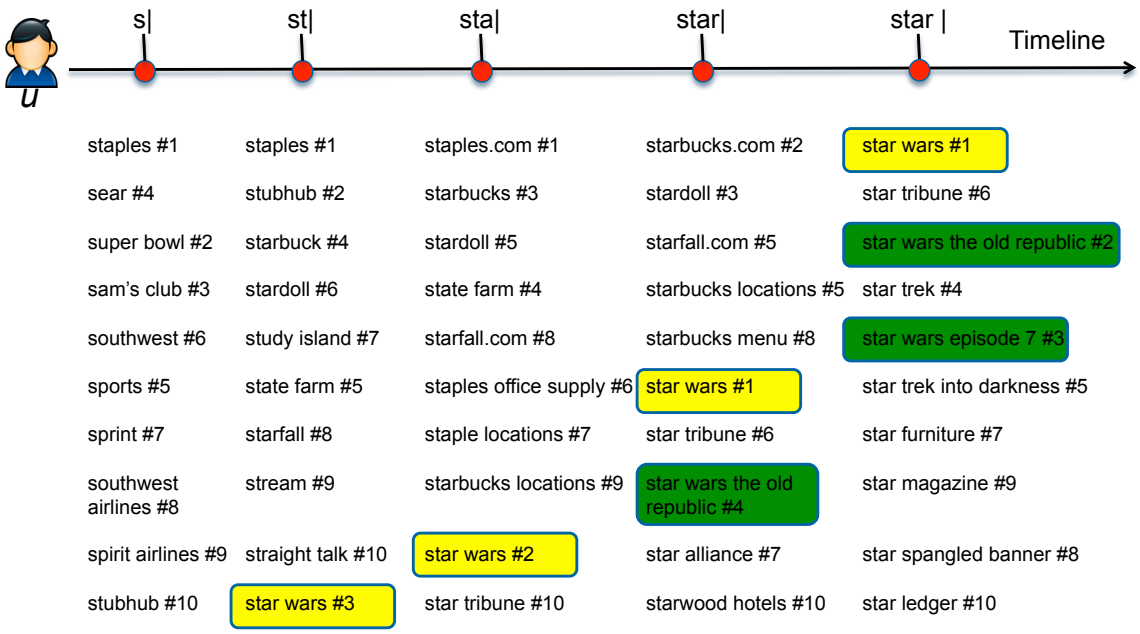


Figure 25: Case Study: The position of list queries from top to down shows the ranking of suggested queries predicted by TDCM, while the number tagged with # behind each query show its ranking given by the proposed model. The yellow box highlights the user's intended query, and the green box highlights queries satisfy similar user intent. Notice that "—" is the cursor.

different keystrokes in high resolution QAC logs. The proposed model integrates three parts, each addressing a single aspect of the above relationship, and illustrates how the three parts together determines users' final click decisions. We have applied the proposed model to predict users' intended queries on real world high-resolution QAC logs collected from a commercial search engine, and compare it with several alternative approaches. Experimental results show that the improvements of our proposed model are consistent, and our model achieves the best performance.

CHAPTER VII

EXPLORING THE MUTUAL INFLUENCE BETWEEN QAC AND CLICK BEHAVIORS FOR CONTEXTUAL-AWARE WEB SEARCH AND QUERY SUGGESTION

Previous chapters addressed several issues in the modeling of influence in one single type of behavioral data. In real world scenarios, influence can also exist between different types of behaviors that are related in some way. Modeling such mutual influence can be very necessary due to two reasons: 1) it enables the usage of two types of data to solve the applications on each single type; 2) such mutual influence can be more straightforward than the influence between the behaviors of a single type. For instance, search engine users' behaviors in submitting a query is usually followed by their behaviors on clicking the returned documents of the issued query, instead of their behaviors in submitting the next query. Thus the influence between users' submission of those two queries is not straightforward, as users' clicking behaviors on the first query may alter their motivations in submitting the next one. In this chapter, we consider the modeling of mutual influence between the behaviors of search engine users in query auto-completion and those in clicking the returned web documents of users' issued queries. We view each type of behavioral data as the context of the other type, and explore how such context influences the modeling of both QAC and click behaviors.

Contextual data plays an important role in modeling search engine users' behaviors on both query auto-completion (QAC) logs and normal query (click) logs. User's recent history on each log has been widely used individually as the context to benefit

the modeling of users' behaviors on that log. However, no existing work has explored and incorporated both logs together for extracting contextual data. As QAC logs and click logs actually record users' sequential behaviors in query submission, the available context of a user's current behavior on one type of log can also benefit from the users' recent history on the other type of log.

In this chapter, we propose to model user behavior on both QAC logs and click logs simultaneously by using QAC logs and click logs as the contextual data of each other. The key idea is to capture the correlation between users' behavior patterns on QAC logs and users' behaviors patterns on click logs. We model such correlation through a novel probabilistic model based on LDA. The learned users' behavior patterns on both QAC logs and click logs are utilized to address the application of query auto-completion on QAC logs, and the prediction of clicks of web documents on click logs. Experiments on real-world logs collected from Yahoo demonstrate the performance improvement on both applications over state-of-the-art approaches using a single type of log alone.

7.1 Contextual-Aware Web Search and Query Suggestion

In this section, we first analyze the relationship between QAC log and click log in describing users' sequential interactions with a search engine. Then, we come up with methods for modeling users' behaviors on both logs simultaneously by using QAC log and click log as the contextual data for each other.

7.1.1 Relationship between QAC Log and Click Log

As mentioned in the previous chapter, unlike previous QAC logs, high definition QAC logs record users' detailed interactions with QAC engines in the procedure of query submission, which provides rich resource for studying users' instantaneous status, inclination, and interests in searching. Thus those logs can provide detailed forecast for users' following behaviors on clicking returned web documents, and vice versa.

For each query issued by a user, two types of behaviors are recorded by search engine logs. One is the keystrokes typing and suggested query clicking recorded by the high definition QAC log, before the user issues the query; the other is the web document clicking recorded by the click log, after the user issued the query. Figure 16 shows a toy example of QAC logs and click logs aligns in the timeline. From the figure, we can find that the QAC session of a query is followed by the click session of that query, and that click session is followed by the QAC session of the next query. Such sequential orders highlight us the opportunity of exploring appropriate relationship between QAC logs and Click logs. Although the user's behaviors on QAC logs and click logs are of different types, the underlying relationship between the user and the query they imply are the same, such as whether the issued query satisfies the user's intent, and how familiar is the user with the issued query or the domain that query belongs to. For instance, if a user is familiar with the issued query, in QAC log, he/she can type the query very fast, and in click log, he/she will spend little time on looking for the satisfying web pages, and spend much time on reading those pages. Thus in the QAC log, if a user behaves like he/she is familiar with the issued query, we can foretell his/she later behaviors of clicking the returned web document of the query on the click log.

On the other hand, the user's behavior on one type of log can be used as the contextual data of the behavior on the other type of log across different query sessions, since users generally behave consistently in nearby time slots. For instance, if in the click log, a user behaves as he is very familiar with the query, then in the QAC session of the next query, if the issued query is under the same topic, the user will probably typing the new query fluently.

In the following, we summarize a set of features describing users' behaviors on QAC and click logs, separately in Table 10. Among features of QAC behaviors, we expect "Type Speed Deviation" to reflect the stability of a user's typing speed.

A user who looks for his/her intended queries from QAC engine's suggestions from time to time will not maintain a stable typing speed, even if the user is skilled at typing. On the other hand, a user who plan to type the entire query at the first will show a stable typing speed. "Typing completion" is designed to show whether a user preferring typing than clicking suggestions. On the other hand, among features of click behaviors, we use "Search Time" to show how fast a user can find his/her intended web documents after submitting a query. An experienced user is very likely to make the first click after only a short while.

7.1.2 Contextual Topic Distribution

To detect user behavior patterns from logs, we choose graphical models like LDA [41], which has been proven to be effective in topic discovery by clustering words that co-occur in the same document into topics. First we consider how to use LDA to cluster user behaviors based on one single type of log only (either QAC log or click log). One straightforward idea is to treat each user's query sequence as a document, and cluster user behaviors that co-occur frequently in the same query sequence into topics, since each user maintains certain behavior patterns in query submission, and different groups of users prefer different behavior patterns. Our LDA model assumes K behavior patterns lie in the given query sequences, and each user m is associated with a randomly drawn vector π_m, where $\pi_{m,k}$ denotes the probability that the user behavior in a query session of user m belongs to behavior pattern k. For the n-th query in the query sequence of user m, a K-dimensional binary vector $Y_{m,n} = [y_{m,n,1}, \ldots, y_{m,n,K}]^T$ is used to denote the pattern membership of the user behavior in that query session. One challenge we encounter in the inference of pattern membership Y is that, user's choice of behavior patterns in each query session is not only decided by users' own preferences of behavior patterns, but also influenced by the context of the current query session.

Table 10: User Behaviors on QAC and Click Logs

Log Type	Behavior	Description
QAC Log	Typing Speed	Average typing speed at keystrokes in a QAC session
	Type Speed Deviation	The deviation of typing speed at keystrokes in a QAC session
	Intent Position	The average position of the appearance of queries satisfying users' search intent in a QAC session.
	Typing Completion Ratio	The percentage of entered keystrokes of the submitted query
	Typing Completion	Whether a user finish typing the entire query or clicks some suggestions
	Time Duration	The time duration of the current QAC session.
	Highest Non-Click Position	The highest position of the appearance of queries satisfying users' search intent but the user does not click.
Click Log	Click Number	The total number of clicks on the returned web documents of query q.
	Dwell Time	The average time between the current click and the next click in the current query session.
	Click Position	The average position of clicks in the current query session.
	Time Duration	The time duration of the current query session.
	Click Speed	The number of clicks divided by the time duration of a query session.
	Scanned Pages	The number of result pages user scanned for a issued query.
	Time Interval	The time interval between the current query session and the next query session.
	Search Time	The time interval between a user's query submission and his/her first click of web documents in a query session.

To model the influence of the context to user's choice of behavior pattern in the current query session, we assume users' preferences of behavior patterns depend on the context, rather than the user. That is to say, a document in LDA model does

not contain the user behaviors in all query sessions of a user, but only the behaviors in those query sessions that the user conducts under the same status, for instance, in the same mood, or sharing the same topic. In this chapter, we focus on studying how using one type of log as context can benefit the user behavior modeling on the other type of log. Thus, in the detection of user behavior patterns in one type of log, we define each status mentioned above to be the user behavior pattern in the other type of log. That is to say, instead of building a behavior pattern distribution π_m for each user m, and accordingly draw the user's behavior in each query session of that user, we construct a QAC (or click) behavior pattern distribution π_k for each click (or QAC) pattern. Then after we inferred the pattern membership of a user's behavior on click (or QAC) log, we obtain the corresponding QAC (or click) behavior pattern distribution, and in the next QAC (or Click) session, draw the QAC (or click) pattern accordingly.

7.1.3 Context-LDA Model

Let us consider a typical scenario where M users issue M corresponding query sequences. For each query n, we have the QAC log records a user's behaviors $\omega_{m,n}$ in the QAC engine before submitting the query and the click log records a user's behaviors $d_{m,n}$ on returned web documents after the query is issued. We assume that K QAC behavior patterns exist in the QAC log, and K' click behavior patterns exist in the click log.

Finally, we present our generative model as follows:

- For each query behavior pattern k', draw a K dimensional membership vector $\pi_{k'} \sim \text{Dirichlet}(\alpha)$.
- For each QAC behavior pattern k, draw a K' dimensional membership vector $\pi'_k \sim \text{Dirichlet}(\alpha')$.

- For each query behavior pattern k', draw a T' dimensional distribution vector ${\theta'}_{k'}$.
- For each QAC behavior pattern k, draw a T dimensional distribution vector θ_k.
- For the n-th query session issued by user m,
  - Draw the user's query session behavior $d_{m,n} \sim \text{Gaussian}(\theta'_{Y_{m,n}})$;
  - Draw the user's next QAC behavior pattern membership $Z_{m,n+1} \sim \text{Multinomial}(\pi_{Y_{m,n}})$;
- For the $n+1$-th QAC session issued by user m,
  - Draw the user's QAC session behavior $\omega_{m,n+1} \sim \text{Gaussian}(\theta_{Z_{m,n+1}})$;
  - Draw the user's next query behavior pattern membership $Y_{m,n+1} \sim \text{Multinomial}(\pi'_{Z_{m,n+1}})$;

Here T is the number of features of QAC behaviors, and T' is the number of features of click behaviors. We name the proposed model Context-LDA.

Under our Context-LDA model, the joint probability of data $D = \{D_m\} = \{\{d_{m,n}\}_{n=1}^{N_m}\}$, $\omega = \{\{\omega_{m,n}\}_{n=1}^{N_m}\}$, and latent variables $\{Y, Z\}$ can be written as follows:

$$
\begin{aligned}
& p(D, \omega, \pi, \pi', Y, Z | \alpha, \alpha', \theta, \theta') \\
= & \prod_m \prod_n P(d_{m,n}|Y_{m,n}, \theta') P(\omega_{m,n}|Z_{m,n}, \theta) \\
& \times \prod_m \prod_n P(Z_{m,n}|Y_{m,n-1}, \pi) P(Y_{m,n}|Z_{m,n}, \pi') \\
& \times \prod_k P(\pi_k|\alpha) \prod_{k'} P(\pi'_{k'}|\alpha').
\end{aligned}
$$

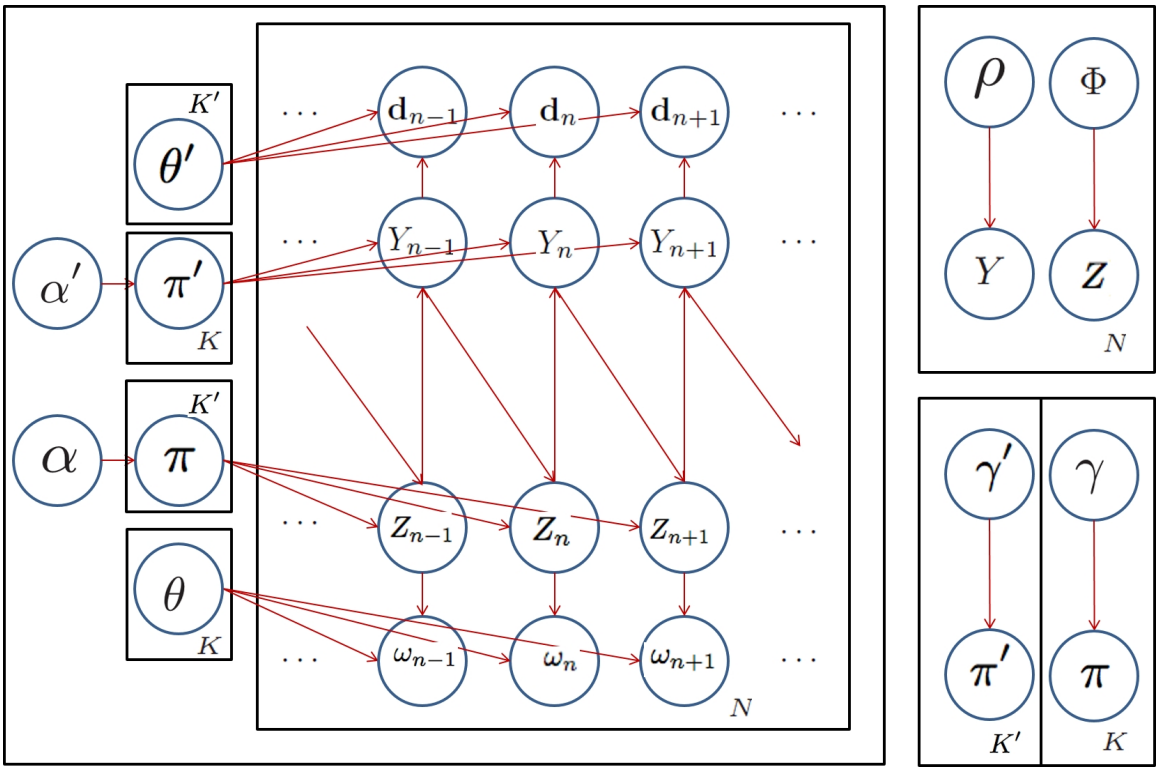


Figure 26: Graphical model representation of Context-LDA and the variational distribution that approximates the likelihood. The left figure shows the graphical model representation of Context-LDA, while the right figure shows the variational distribution that approximates the likelihood.

7.2 Algorithm

In the proposed Context-LDA model, provided observations of both the high definition QAC log $D = \{D_m\} = \{\{d_{m,n}\}_{n=1}^{N_m}\}$ and the click log $\omega = \{\{\omega_{m,n}\}_{n=1}^{N_m}\}$, the log-likelihood for the complete data is given by $\log P(D, \omega|\alpha, \alpha', \theta, \theta')$. We turn to variational methods [38], since this true posterior is hard to infer directly. In Figure 26, the right part shows the variational distribution that approximates the data likelihood. A distribution of latent variables q specified as the mean-field fully factorized family is introduced as follows:

$$q(Y, Z, \pi, \pi'|\rho, \phi, \gamma, \gamma') = \prod_m \prod_n q_1(Y_{m,n}|\rho_{m,n}) q_1(Z_{m,n}|\phi_{m,n})$$
$$\prod_{k'} q_2(\pi_{k'}|\gamma_{k'}) \prod_k q_2(\pi'_k|\gamma'_k)$$

where q_1 is a multinomial, q_2 is a Dirichlet, and $\{\Phi, \rho, \gamma, \gamma'\}$ are the set of variational parameters. We optimize those free parameters to tight the following lower bound $\mathcal{L}'$ for our likelihood:

$$\begin{aligned}\log p(D, \omega|\alpha, \alpha', \theta, \theta') \geq & E_q[\log p(D, \omega, \pi, \pi', Y, Z|\alpha, \alpha', \theta, \theta')] \\ & - E_q[\log q(Y, Z, \pi, \pi'|\rho, \phi, \gamma, \gamma')]. \end{aligned} \tag{15}$$

We optimize the lower bound as in Eqn (15) against each variational latent variable and the model hyper-parameter under a coordinate descent framework. For the updating of latent variables γ, γ', ρ, and ϕ, the following rules are use:

- update rules for γ's as:

$$\gamma_{k',k} = \alpha_k + \sum_m \sum_n \phi_{m,n+1,k} \rho_{m,n,k'};$$

- update rules for γ''s as:

$$\gamma'_{k,k'} = \alpha'_{k'} + \sum_m \sum_n \phi_{m,n,k} \rho_{m,n,k'};$$

- update rules for ρ's as:

$$\begin{aligned}\rho_{m,n,k'} \propto \exp \Bigg(& -\frac{1}{2\sigma^2} \sum_{m,n} (d_{m,n} - \theta'_{m,n,k'})^2 \\ & + \sum_k \phi_{m,n+1,k} \left[\Phi(\gamma_{k,k'}) - \Phi\left(\sum_{k'} \gamma_{k,k'} \right) \right] \\ & + \sum_k \phi_{m,n,k} \left[\Phi(\gamma'_{k',k}) - \Phi\left(\sum_k \gamma'_{k',k} \right) \right] \Bigg),\end{aligned}$$

- update rules for ϕ's as:

$$\begin{aligned}\phi_{m,n,k} \propto \exp \Bigg(& -\frac{1}{2\sigma^2} \sum_{m,n} (\omega_{m,n} - \theta_{m,n,k})^2 \\ & + \sum_{k'} \rho_{m,n-1,k'} \left[\Phi(\gamma_{k,k'}) - \Phi\left(\sum_{k'} \gamma_{k,k'} \right) \right] \\ & + \sum_{k'} \rho_{m,n,k'} \left[\Phi(\gamma'_{k',k}) - \Phi\left(\sum_k \gamma'_{k',k} \right) \right] \Bigg),\end{aligned}$$

We compute the empirical Bayes estimates of the LDA hyper-parameters α and α' in our Context-LDA model using expectation-maximization (EM) algorithms [69]. In updating α, we use a Newton-Raphson method, since the approximate maximum likelihood estimate of α doesn't have a closed form solution. The Newton-Raphson method is conducted with a gradient and Hessian as follows:

$$\frac{\partial \mathcal{L}'}{\partial \alpha_k} = K\left(\Psi\left(\sum_k \alpha_k\right) - \Psi(\alpha_k)\right) + \sum_{k'}\left(\Psi(\gamma_{k'k}) - \Psi\left(\sum_k \gamma_{k'k}\right)\right),$$
$$\frac{\partial \mathcal{L}'}{\partial \alpha_{k_1}\alpha_{k_2}} = N\left(\mathbb{I}_{(k_1=k_2)}\Psi'(\alpha_{k_1}) - \Psi'\left(\sum_k \alpha_k\right)\right).$$

Similar update rules can be derived for α'.

On the other hand, to obtain the approximate maximum likelihood estimation of parameters describing QAC and click behavior patterns θ and θ', we optimize the lower bound as in Eqn (15) against each parameter, and update θ and θ independently with closed-form solutions as follows:

$$\theta'_{k'} = \frac{\sum_{m,n}\rho_{m,n,k'}d_{m,n}}{\sum_{m.n}\rho_{m,n,k'}}, \qquad \theta_k = \frac{\sum_{m,n}\phi_{m,n,k}\omega_{m,n}}{\sum_{m.n}\phi_{m,n,k}};$$

In our mean-field variation inference algorithm, the computational cost of inferring variational variables is $O((\sum_m N_m)KK')$. The computational cost of the estimation of LDA hyper-parameters is $O(KK')$. The computational cost of the estimation of behavior patterns is $O(\sum_m N_m(K+K'))$, Thus the total computational cost of our algorithm is $O((\sum_m N_m)KK')$. Since we can control the value of KK' by limiting the number QAC and click behavior patterns, this total computational cost can be viewed as linear to the number of queries in the entire log.

7.3 Experiments

We evaluated our Context-LDA model on both synthetic and real-world data sets, and compared the performance with two alternative LDA-based probabilistic models that are also capable of learning QAC and click behavior patterns:

LDA: This model uses a normal LDA to learning users' behavior patterns on QAC and click logs, separately. No contextual data is utilized in the process of pattern learning.

HMM: This is a hidden Markov model that builds a hidden state for each QAC and click session, and the users' behaviors in each QAC and click session as observation. The transition matrix between hidden states learned by the HMM model is expected to capture the effect of using one type of log as the contextual data of modeling the other type of log.

Synthetic data. Given parameters $(M, N, K, K', \alpha, \alpha')$, we sample the synthetic data according to the proposed generative model. Our synthetic data are simulated with two different settings: 1) `Small`: $M = 100$, $N = 12{,}000$, $K = 10$, $\hat{\alpha} = 0.1$, $\hat{\alpha'} = 0.1$. Simulations were run 1,000 times; 2) `Large`: $M = 10{,}000$, $N = 1{,}000{,}000$, $K = 50$, $\hat{\alpha} = 0.1$, $\hat{\alpha'} = 0.1$. Simulations were run 5 times. To test the robustness of our method, we add noise to the original synthetic data:

`Behavior Noise`: Instead of using $\omega_{m,n}$ and $\mathbf{d}_{m,n}$ to simulate user m's QAC and click behaviors at the n-th query session, respectively, we use noisy values $\omega'_{m,n}$ and $\mathbf{d}'_{m,n}$, which is obtained by adding Gaussian noises on $\omega_{m,n}$ and $\mathbf{d}_{m,n}$, separately, as shown in Eqn (16).

$$\omega'_{m,n} = \omega_{m,n} + \mathbf{e},\ \mathbf{e} \sim \mathcal{N}(0, \sigma). \tag{16}$$

The default value of σ is set to be 1. A similar equation is used for generating $\mathbf{d}'_{m,n}$.

Real-world QAC and Click Logs. We conducted extensive experiments on a real-world QAC log and the corresponding click log collected from a commercial search engine. This data set contains QAC and click logs from May 2014 to July 2014. The collection consists of a sample of 7.4 million query sessions from about 40,000 users over a 3-month period. We randomly selected a subset of users who submitted over 500 query sessions during this period, and collected their corresponding search activities, including the anonymized user ID, query string, timestamp, and the clicked

Table 11: Log Predictive Likelihood on Both Synthetic and Real-world Data

Data set	Context-LDA	HMM	LDA
Small Synthetic	-128.53	-141.32	-148.83
Small Behavior Noise	-143.85	-159.02	-164.37
Large Synthetic	-188.53	-211.35	-217.74
Large Behavior Noise	-202.74	-230.86	-236.57
Yahoo	-241.74	-272.85	-280.29

URL. As a result, we collected 3,954 users with 2.6 million queries, and their activities span from 22 days to 3 months. According to the platform each query session belongs to, we separate the entire data set into two subsets. One is PC, which contains 1.6 million query sessions, while the other is mobile phones, which contains 1.0 million query sessions. On query auto-completion experiments, we evaluate the performance on those separate subsets, since users' behavior on QAC engines are significantly influenced by the platform they use.

Model Fitness. Table 11 shows the log predictive likelihood on events falling in the final 10% of the total time of query data. According to Table 11, Context-LDA and HMM fits both synthetic and real-world data better than LDA. This illustrates that the effectiveness of using one type of log as the contextual data for modeling the behaviors in the other type of log. Context-LDA performs better than HMM, which shows that the proposed model can better utilize the contextual data to help the behavior modeling through appropriate modeling of relationship between QAC and click logs. On both noisy data sets, the performances of all models become worse. However, the decrease of the performance of Context-LDA is smaller than that of HMM and LDA, which demonstrates the robustness of our proposed model.

7.3.1 Contextual-Aware Query Auto-Completion.

In this series of experiments, we show how to utilize the pattern membership inferred by the proposed model to enhance the performance of query auto-completion. We

design a new QAC method based on a two-dimensional click model (TDCM) [156], which is known to be the first model proposed for solving the QAC task using high definition QAC logs. Instead of learning a TDCM model on the entire QAC log, our new method separates the log according to the behavior pattern membership in each QAC session, and learns separate TDCM models on each subset. To justify how effectively appropriate search patterns help solving the QAC task, we compare the performance of the above method with those of methods using a similar strategy using the search patterns learned by LDA and HMM. We compare the performance with several state-of-the-art QAC algorithms, where two of them are context-aware QAC algorithms:

MPC [27, 217]: This method, named MostPopularCompletion, is a widely used baseline in Query Auto-Completion, and employed as one main feature in many QAC engines.

BSS [96]: This Bayesian Sequential State model uses a probabilistic graphical model to characterize the document content and dependencies among the sequential click events within a query with a set of descriptive features. This is a content-aware model which is able to predict unobserved prefix-query pairs.

TDCM [156]: This is a two-dimensional click model which emphasize two kinds of user behaviors. It consists of a horizontal model which explains the skipping behavior, and a vertical model that depicts the vertical examination behavior. It is the first work that utilizes high definition QAC logs.

Hybrid [78]: This is a context-sensitive query auto completion algorithm, which outputs the completions of the user's input that are most similar to the context queries. The similarity is measured by representing queries and contexts as high-dimensional term-weighted vectors and resort to cosine similarity.

We employ the Mean Reciprocal Rank (MRR) as the relevance measurement, which is a widely used evaluation metric in measuring QAC performance [27, 217, 156],

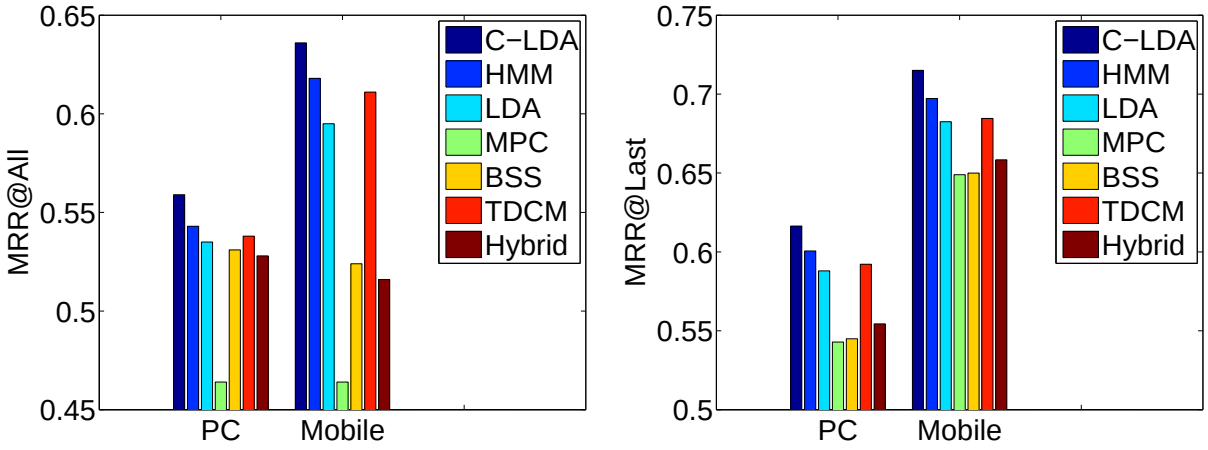
$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q},$$

where Q is the set of queries a user finally submitted, and rank_q denotes the rank of the query q in the suggested query list.

Figure 27 compares Context-LDA with alternative probabilistic models, and state-of-the-art QAC algorithms on Yahoo data sets. We can find that Context-LDA outperforms all compared approaches. It improves over the second best method by up to 5%. `TDCM` take the second place, which demonstrates that high definition QAC logs provide rich additional information for the modeling of users interactions with QAC engines than normal QAC logs. Context-LDA performs better than HMM and LDA, which shows the importance of appropriate modeling of user behaviors, and appropriate behavior patterns play a very positive effect in solving QAC tasks. HMM performs better than LDA, since HMM utilizes the contextual relationship between QAC and click logs, while LDA models user behaviors on each log, separately. BSS and Hybrid generally perform better than MPC, which demonstrates the effectiveness of using contextual data for user behavior modeling and prediction of suggestions in query auto-completion.

7.3.2 Contextual-Aware Click Prediction on Web Documents.

In this series of experiments, we show how to utilize the pattern membership inferred by the proposed model to enhance the prediction of clicks on web documents returned search engines. We design a new QAC method based on BSS, which is known to be an efficient context-aware click model. Instead of learning a BSS model on the entire QAC log, our new method separate the log according to the behavior pattern membership in each click session, and learns separate BSS models on each subset. To justify how effectively appropriate search patterns help solving the QAC task, we compare the performance of the above method with those of methods using a



(a) MRR@All (b) MRR@Last

Figure 27: Performance of Query Auto-Completion. In the figure we use C-LDA to denote Context-LDA

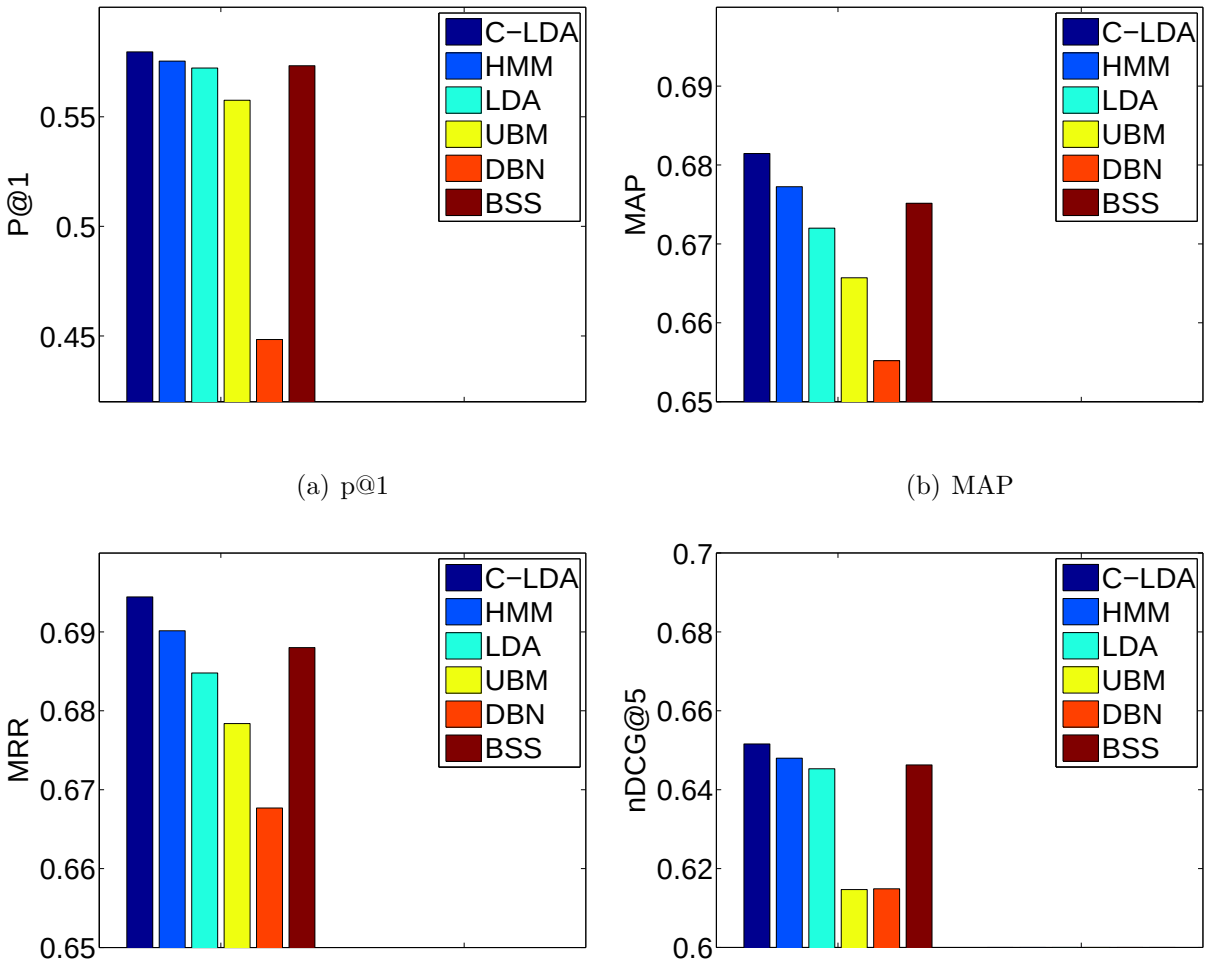
similar strategy using the search patterns learned by LDA and HMM. We compare the performance with several state-of-the-art click models, where one of them is a context-aware click model:

UBM [78]: This User Browsing Model proposes a number of assumptions on user browsing behavior that allows the estimation of the probability of observing a document. It depends on statistical counting of query-document pairs, thus unable to predict unseen query-document pairs.

DBN [53]: This Dynamic Bayesian Network model provides unbiased estimation of the relevance from the click logs. This model also relies on the counting of query-document pairs.

BSS [96]: This Bayesian Sequential State model uses a probabilistic graphical model to characterize the document content and dependencies among the sequential click events within a query with a set of descriptive features. This is a content-aware model which is able to predict unobserved query-document pairs.

Figure 28 compares the proposed model with alternative probabilistic models and



(a) p@1

(b) MAP

(c) MRR

(d) NDCG@5

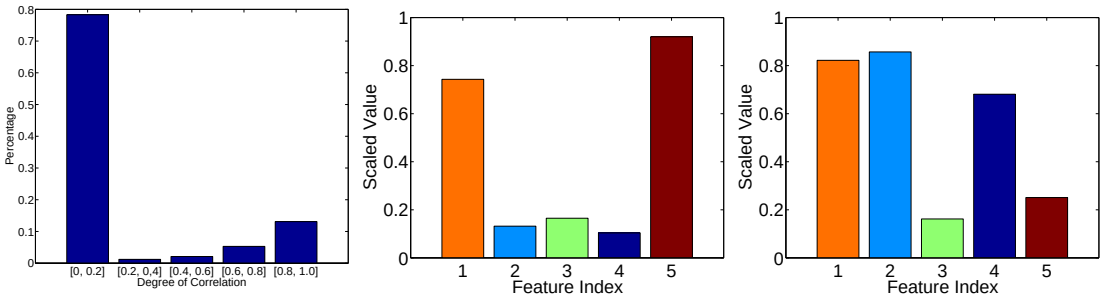
Figure 28: Performance Comparison of Prediction of Clicks in Web Documents. In the figure we use C-LDA to denote Context-LDA

state-of-the-art click models. We find that the proposed model performs the best among all compared approaches. Context-LDA's advantage over HMM and LDA illustrates the importance of appropriate click behavior patterns in predicting the clicks of web documents. Meanwhile, BSS performs better than UBM and LDA, which shows the effectiveness of using contextual data for click modeling.

7.3.3 Correlation between Behavior Patterns in QAC and Click Logs.

In this part, we analyze the correlation between behavior patterns in QAC logs and click logs based on the inferred conditional pattern membership distribution, and then try to show a few examples of highly correlated QAC behavior patterns and click behavior patterns in Yahoo logs. First, we estimate the degree of correlation between QAC and click behavior patterns based on the inferred conditional pattern membership distributions $\{\pi_{k'}\}$ and $\{\pi'_k\}$. Here we take $\pi_{k',k}$ and $\pi'_{k,k'}$ as the directed partial correlations of the k-th QAC behavior pattern and the k'-th click behavior pattern. Then, we statistically count the number of pattern pairs whose degree of correlation falls in the range of [0, 0.2], [0.2, 0,4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0], separately. Finally, we show the percentage of pattern pairs in each bin in Figure 29(a). From the figure, we can find that between most pairs of behavior patterns, the degree of correlation is very small. And among the rest pattern pairs with significant correlations, between most of them the corresponding degrees of correlation are larger than 0.8, i.e., there is a one to one mapping between those pattern pairs. Such phenomenon shows that a lot users retain the same behavior mode for quite a while.

Figure 29(b) and (c) show an example pair of QAC and click behavior patterns between which the degree of correlation is larger than 0.8. From the QAC behavior pattern shown in (b), we find that 1) the user's typing speed is very fast; 2) the time cost of completing a QAC session is very small; 3) the user does not like to click suggested queries even if they satisfy his/her search intent and ranked at top positions; 4) the user types keystrokes in a consistent speed; and 5) most of time, the user typing his/her intended query completely instead of stopping to click the suggestions returned by QAC engines. Based on the above behaviors, we can conclude that this is probably a user who is proficient in searching or his/her intended topic. From the click behavior pattern shown in (c), we find that 1) the user clicks a lot of web documents returned by the search engine; 2) the user spends a lot of time in



(a) Correlation (b) QAC Pattern (c) Click Pattern

Figure 29: Correlation between QAC Behavior Patterns and Click Behavior Patterns. The left figure shows the percentage of the number of pattern pairs with the degrees of correlation in different ranges. The middle and right figures together show a pair of highly correlated QAC and click behavior patterns. The middle figure shows the scaled value of features in the QAC behavior pattern. Indices of selected features of QAC behaviors are: 1-'Typing Speed', 2-'Time Duration', 3-'Highest Non-Click Position', 4-'Type Speed Deviation', 5-'Typing Completion', The right figure shows the scaled value of features in the Click behavior pattern. Indices of selected features of click behavior patterns are: 1-'Click Number', 2-'Dwell Time', 3-'Click Speed', 4-'Scanned Pages', 5-'Search Time'.

viewing the clicked web documents, 3) the user scans several pages of results, and 4) it does not take a lot of time for the user to find his intended web documents after submitting the query. Those behaviors also illustrate that this is a proficient user. Thus the correlation the proposed model captured from real-world QAC and click logs is appropriate and meaningful.

7.4 Summary

In this chapter, we presented a LDA-based probabilistic model to study users' behaviors on both QAC logs and click logs simultaneously by using QAC and click logs as the contextual data of each other. The model is designed to capture the correlation between users' behavior patterns on QAC logs and that on click logs. The learned users' behavior patterns on both QAC and click logs are utilized to benefit the query auto-completion task and the prediction of users' click on web documents. We have applied the proposed Context-LDA model to model user behavior on both

real-world QAC logs and click logs collected from Yahoo, and compare with several alternative approaches. Experimental results show that our proposed model offers a better context-aware solution to both the applications of query auto-completion and click prediction.

CHAPTER VIII

CONCLUSIONS AND DISCUSSION

8.1 Summary

In this thesis, we investigate the applications and extensions of methods based on probabilistic models to solve the problem from real-world applications. In particular, we analyze the major factors that implicate the particular influence we are to model under each scenario. Specifically, we propose to using time-varying features describe instant individual property or pairwise relationship to model the current influence among individuals in diffusion networks. Moreover, we introduced probabilistic models based on point processes, Latent Dirichlet Allocation (LDA), and Markov processes to explore the influence among electronic appliance usage, users' sequential query submissions, users' sequential interactions with query auto-completion (QAC) engines, and users' behaviors across QAC and click logs, separately.

We briefly summary the contributions of this thesis as follows:

- **Learning Parametric Models for Social Infectivity in Multi-dimensional Hawkes Processes.** We propose a novel multi-dimensional Hawkes model that parameterizes pairwise infectivity using linear combinations of time-varying features. Alternating direction method of multipliers (ADMM) is employed to estimate the proposed features' coefficients, which are regularized by a ℓ_1 norm to select effective features.

- **Explore Energy Usage Behavior Modeling in Energy Disaggregation via Marked Hawkes Process.** We formulated the task of energy disaggregation into the modeling of marked event sequences. We presented a probabilistic

model that integrates topic models with Hawkes processes to capture the influence from the occurrence and the mark of an event to the occurrences and the marks of future events.

- **Explore Energy Usage Behavior Modeling in Energy Disaggregation via Marked Hawkes Process.** We integrated the LDA model with Hawkes process for identifying and labeling search tasks. Basically, Hawkes processes utilize its self-exciting property to identify search tasks if influence exists among a sequence of queries for individual users, while the LDA model exploits query co-occurrence across different users to discover the latent information needed for labeling search tasks. By leveraging the temporally weighted query co-occurrence, our model not only guarantees sound performance by making full use of both textual and temporal information of the entire query sequences, but also enables the labeling of the identified search tasks since semantically related queries are clustered together through query links determined by co-occurrence.

- **Analyzing User's Sequential Behavior in Query Auto-Completion via Markov Processes.** We have presented a probabilistic model to solve the query auto-completion (QAC) task by capturing the relationship between users' behaviors at different keystrokes in high resolution QAC logs. The proposed model integrates three parts, each addressing a single aspect of the above relationship, and illustrates how the three parts together determines users' final click decisions. We have applied the proposed model to predict users' intended queries on real world high-resolution QAC logs collected from a commercial search engine, and compare it with several alternative approaches. Experimental results show that the improvements of our proposed model are consistent, and our model achieves the best performance.

- **Exploring QAC and Click log for Contextual-Aware Web Search and**

Query Suggestion. We presented a LDA-based probabilistic model to study users' behaviors on both QAC logs and click logs simultaneously by using QAC and click logs as the contextual data of each other. The model is designed to capture the correlation between users' behavior patterns on QAC logs and that on click logs. The learned users' behavior patterns on both QAC and click logs are utilized to benefit the query auto-completion task and the prediction of users' click on web documents. We have applied the proposed Context-LDA model to model user behavior on both real-world QAC logs and click logs collected from Yahoo, and compare with several alternative approaches. Experimental results show that our proposed model offers a better context-aware solution to both the applications of query auto-completion and click prediction.

8.2 Discussions and Future Directions

The proposed Para-Hawkes model generally utilize time-varying features based on statistical counting of the appearance of a certain pattern involving one individual or one individual pair, it would be interesting to consider additional time-varying features, and investigate the performance of the proposed model in other kinds of social networks.

To improve our marked Hawkes process, we plan to consider other marks, e.g., the attributes of appliances, into this framework, and investigate the performance of M-Hawkes in other domains. In addition, we'll attempt to directly model the behavior of users instead appliances, and the influence in-between.

For modeling users' sequential behaviors in QAC, we are to explore more complex relationships between users' behaviors at different keystrokes. For instance, we could increase the number of states from simply skipping or viewing to viewing to a certain position d, which enables a more precise modeling of users' behaviors in QAC. Meanwhile, we also find it interesting to consider the usage of additional user

behavior features in the proposed model. Finally, we plan to investigate alternative models that can effectively capture the relationship between user behaviors on QAC logs and those on click logs.

The proposed models in this thesis only explored the usage of Hawkes processes and Markov processes in modeling the influence between users' historical behaviors and their current behaviors. We would also like to justify the effectiveness and efficiency of other appropriate models, such as cox proportional hazards model, in social influence modeling, and compare its strengths and weaknesses with our explored ones.

REFERENCES

[1] "http://archive.ics.uci.edu/ml/databases/nsfabs."

[2] "http://www.grouplens.org/."

[3] "http://www.informatik.uni-trier.de/ ley/db."

[4] "www.shuijunwang.com."

[5] "www.zhubajie.com."

[6] A. Cichocki, A.-H. P., "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics*, vol. 92, pp. 708–721, 2009.

[7] A. Cichocki, R. Zdunek, A.-H. P. S. A., "Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation," (New York, USA), Wiley, 2009.

[8] A. Cichocki, R. Z. and Amari, S., "Nonnegative matrix and tensor factorization," *IEEE Signal Processing Magazine*, vol. 25, pp. 142–145, 2008.

[9] A. Perotte, N. Bartlett, N. E. and Wood, F., "Hierarchically supervised latent dirichlet allocation," in *NIPS*, pp. 2609–2617, 2011.

[10] ACLED, "http://www.acleddata.com," 2010.

[11] Agichtein, E., White, R. W., Dumais, S. T., and Bennett, P. N., "Search, interrupted: understanding and predicting search task continuation," in *SIGIR*, pp. 315–324, 2012.

[12] Aiello, L. M., Donato, D., Ozertem, U., and Menczer, F., "Behavior-driven clustering of queries into topics," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, (New York, NY, USA), pp. 1373–1382, ACM, 2011.

[13] Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P., "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, jun 2008.

[14] Ait-Sahalia, Y., Cacho-Diaz, J., and Laeven, R., "Modeling financial contagion using mutually exciting jump processes," *Tech. rep.*, 2010.

[15] Ali, S. M. and Silvey, S. D., "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

[16] Anick, P., "Using terminological feedback for web search refinement: a log-based study," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, (New York, NY, USA), pp. 88–95, ACM, 2003.

[17] AOL, "http://gregsadetsky.com/aol-data/."

[18] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W., "On smoothing and inference for topic models," in *UAI*, pp. 27–34, AUAI Press, 2009.

[19] Bacrya, E., Delattreb, S., Hoffmannc, M., and Muzyd, J. F., "Modeling microstructure noise with mutually exciting point processes," *Quantitative Finance*, 2012.

[20] Baeza-yates, R., Hurtado, C., and Mendoza, M., "Query recommendation using query logs in search engines," in *In International Workshop on Clustering Information over the Web, Creete*, pp. 588–596, Springer, 2004.

[21] Baeza-yates, R., Hurtado, C., and Mendoza, M., "Query recommendation using query logs in search engines," in *In International Workshop on Clustering Information over the Web, Creete*, pp. 588–596, Springer, 2004.

[22] Baeza-Yates, R. and Tiberi, A., "Extracting semantic relations from query logs," KDD, (New York, NY, USA), pp. 76–85, ACM, 2007.

[23] Bagga, A., Hu, J., Zhong, J., and Ramesh, G., "Multi-source combined-media video tracking for summarization," *Pattern Recognition, International Conference on*, vol. 2, p. 20818, 2002.

[24] Banerjee, A., "Optimal bregman prediction and jensens equality," in *In Proc. International Symposium on Information Theory (ISIT)*, p. 2004, 2004.

[25] Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J., "Clustering with bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, December 2005.

[26] Baptista, M., Fang, A., Prendinger, H., Prada, R., and Yamaguchi, Y., "Accurate household occupant behavior modeling based on data mining techniques," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 1164–1170, 2014.

[27] Bar-Yossef, Z. and Kraus, N., "Context-sensitive query auto-completion," in *Proceedings of the 20th international conference on World wide web*, pp. 107–116, ACM, 2011.

[28] Barbu, V. and Limnios, N., *Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications: Their Use in Reliability and DNA Analysis.* Springer Publishing Company, Incorporated, 1 ed., 2008.

[29] Barker, S., Mishra, A., Irwin, D., Cecchet, E., Shenoy, P., and Albrecht, J., "Smart*: An open data set and tools for enabling research in sustainable homes.," in *Proceedings of the 2012 Workshop on Data Mining Applications in Sustainability (SustKDD 2012), Beijing, China*, 2012.

[30] Barth, W., Martin, R. S., and Wilkinson, J. H., "Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection," *Numer. Math.*, vol. 9, pp. 386–393, 1967.

[31] Bast, H., Majumdar, D., and Weber, I., "Efficient interactive query expansion with complete search," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 857–860, ACM, 2007.

[32] Bast, H. and Weber, I., "Type less, find more: fast autocompletion search with a succinct index," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 364–371, ACM, 2006.

[33] Beitzel, S. M., Jensen, E. C., Lewis, D. D., Chowdhury, A., and Frieder, O., "Automatic classification of web queries using very large unlabeled query logs," *ACM Trans. Inf. Syst.*, vol. 25, 4 2007.

[34] Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., and Frieder, O., "Hourly analysis of a very large topically categorized web query log," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 321–328, ACM, 2004.

[35] Benthem, M. H. V. and Keenan, M. R., "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of Chemometrics*, vol. 18, pp. 441–450, 2004.

[36] Berkman, L., Singer, B., and Manton, K., "Black/white differences in health status and mortality among the elderly.," in *Demography*, vol. 26, pp. 661–678, Nov 1989.

[37] Bhatia, S., Majumdar, D., and Mitra, P., "Query suggestions in the absence of query logs," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, (New York, NY, USA), pp. 795–804, ACM, 2011.

[38] Blei, D. and Jordan, M., "Variational inference for dirichlet process mixtures," in *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[39] Blei, D. and Jordan, M., "Variational inference for dirichlet process mixtures," in *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[40] Blei, D. M., Griffiths, T. L., and Jordan, M. I., "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, pp. 1–30, Feb 2010.

[41] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, March 2003.

[42] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[43] Blundell, C., Heller, K. A., and Beck, J. M., "Modelling reciprocating relationships with hawkes processes," *NIPS*, 2012.

[44] Bowdler, H., Martin, R. S., Reinsch, C., and Wilkinson, J. H., "The qr and ql algorithms for symmetric matrices," *Numer. Math.*, vol. 11, pp. 293–306, 1971.

[45] Boyd, S., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[46] Bro, R. and Jong, S. D., "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, pp. 393–401, 1997.

[47] Broyden, C. G., Dennis, J. E., and More, J. J., "On the local and superlinear convergence of quasi-newton methods," *J. Inst. Math. Appl*, vol. 12, pp. 223–246, 1973.

[48] Cai, Y. and Li, Q., "Personalized search by tag-based user profile and resource profile in collaborative tagging systems," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, (New York, NY, USA), pp. 969–978, ACM, 2010.

[49] Candes, E. J., Li, X., Ma, Y., and Wright, J., "Robust principal component analysis?," 2009.

[50] Cao, H., Jiang, D., Pei, J., Chen, E., and Li, H., "Towards context-aware search by learning a very large variable length hidden markov model from search logs," in *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pp. 191–200, 2009.

[51] Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., and Li, H., "Context-aware query suggestion by mining click-through and session data," in *KDD*, pp. 875–883, 2008.

[52] CaRdenas-PenA, D., Orozco-Alzate, M., and Castellanos-Dominguez, G., "Selection of time-variant features for earthquake classification at the nevado-del-ruiz volcano," *Comput. Geosci.*, vol. 51, pp. 293–304, 2 2013.

[53] Chapelle, O. and Zhang, Y., "A dynamic bayesian network click model for web search ranking," in *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, (New York, NY, USA), pp. 1–10, ACM, 2009.

[54] Chapelle, O. and Zhang, Y., "A dynamic bayesian network click model for web search ranking," in *Proceedings of the 18th international conference on World wide web*, pp. 1–10, ACM, 2009.

[55] Chen, B.-W., Wang, J.-C., and Wang, J.-F., "A novel video summarization based on mining the story-structure and semantic relations among concept entities," *Multimedia*, vol. 11, pp. 295–312, Feb. 2009.

[56] Cichocki, A. and Zdunek, R., "Nmflab for signal and image processing," in *tech. rep, Laboratory for Advanced Brain Signal Processing*, (Saitama, Japan), BSI, RIKEN, 2006.

[57] Cichocki, A., Zdunek, R., and Amari, S., "Csiszar's divergences for non-negative matrix factorization: Family of new algorithms," in *LNCS*, pp. 32–39, Springer, 2006.

[58] Cohen, G., Amir, A., Ponceleon, D., Blanchard, B., Petkovic, D., and Srinivasan, S., "Using audio time scale modification for video browsing," in *HICSS*, (Washington, DC, USA), p. 3046, IEEE Computer Society, 2000.

[59] Crane, R. and Sornette, D., "Robust dynamic classes revealed by measuring the response function of a social system," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 41, pp. 15649–15653, 2008.

[60] Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B., "An experimental comparison of click position-bias models," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 87–94, ACM, 2008.

[61] Csiszar, I., "Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten," *Magyar. Tud. Akad. Mat. Kutato Int. Kozl*, vol. 8, pp. 85–108, 1963.

[62] Cucerzan, S. and White, R. W., "Query suggestion based on user landing pages," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, (New York, NY, USA), pp. 875–876, ACM, 2007.

[63] Cuppen, J. J. M., "A divide and conquer method for the symmetric eigenproblem," *Numer. Math.*, vol. 36, pp. 177–195, 1981.

[64] D. Kim, S. S. and Dhillon, I., "Fast newton-type methods for the least squares nonnegative matrix approximation problem," in *in Data Mining, Proceedings of SIAM Conference on*, (Minneapolis, Minnesota, USA), pp. 343–354, April 2007.

[65] Dai, W., Chen, Y., Xue, G.-R., Yang, Q., and Yu, Y., "Translated learning: Transfer learning across different feature spaces," in *NIPS*, pp. 353–360, 2008.

[66] Dai, W., Xue, G.-R., Yang, Q., and Yu, Y., "Transferring naive bayes classifiers for text classification," in *AAAI*, pp. 540–545, AAAI Press, 2007.

[67] Darby, S., "The effectiveness of feedback on energy consumption.," *Technical report, Environmental Change In- stitute, University of Oxford.*, 2006.

[68] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R., "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, June 1990.

[69] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[70] Deng, H., King, I., and Lyu, M. R., "Entropy-biased models for query representation on the click graph," SIGIR, (New York, NY, USA), pp. 339–346, ACM, 2009.

[71] Detyniecki, M. and Marsala, C., "Video rushes summarization by adaptive acceleration and stacking of shots," in *TVS*, (New York, NY, USA), pp. 65–69, ACM, 2007.

[72] Dhillon, I. S. and Sra, S., "Generalized nonnegative matrix approximations with bregman divergences," in *Neural Information Proc. Systems*, pp. 283–290, 2005.

[73] Ding, C., He, X., and Simon, H. D., "On the equivalence of nonnegative matrix factorization and spectral clustering," *In Proceedings of SIAM Internationall Conference on Data Mining*, no. 12, 2005.

[74] Ding, C., Li, T., and Peng, W., "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," *Comput. Stat. Data Anal.*, vol. 52, no. 8, pp. 3913–3927, 2008.

[75] Donato, D., Bonchi, F., Chi, T., and Maarek, Y. S., "Do you want to take notes?: identifying research missions in yahoo! search pad," in *WWW*, pp. 321–330, 2010.

[76] Donoho, D. L. and Johnstone, I. M., "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[77] Duan, H. and Hsu, B.-J. P., "Online spelling correction for query completion," in *Proceedings of the 20th international conference on World wide web*, pp. 117–126, ACM, 2011.

[78] Dupret, G. E. and Piwowarski, B., "A user browsing model to predict search engine click data from past observations.," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 331–338, ACM, 2008.

[79] Easley, D. and Kleinberg, J., *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge, UK: Cambridge University Press, 2010.

[80] Eckstein, J. and Bertsekas, D. P., "On the douglas rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[81] Eguchi, S., "U-boosting method of classification and information geometry," in *SRCCS International Statistical Workshop*, Seoul National University, June 2002.

[82] Errais, E., Giesecke, K., and Goldberg, L. R., "Affine point processes and portfolio credit risk," *SIAM J. Fin. Math.*, vol. 1, pp. 642–665, Sep 2010.

[83] Feng, S., Banerjee, R., and Choi, Y., "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, (Stroudsburg, PA, USA), pp. 171–175, Association for Computational Linguistics, 2012.

[84] Fevotte, C., Bertin, N., and Durrieu, J.-L., "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural Comput.*, vol. 21, pp. 793–830, March 2009.

[85] Frigyik, B. A., Srivastava, S., and Gupta, M. R., "Functional bregman divergence and bayesian estimation of distributions," *Information Theory, IEEE Transactions*, vol. 54, pp. 5130–5139, Nov 2008.

[86] Frigyik, B. A., Srivastava, S., and Gupta, M. R., "An introduction to functional derivatives," *UWEE Tech Report*, vol. 0001, 2008.

[87] Fu, W. J., "The bridge versus the lasso," in *Journal of Computational and Graphical Statistics*, vol. 7, pp. 397–416, 1998.

[88] G.-Rodriguez, M., Balduzzi, D., and Schölkopf, B., "Uncovering the temporal dynamics of diffusion networks," in *ICML*, pp. 561–568, 2011.

[89] Gayo-Avello, D., "A survey on session detection methods in query logs and a proposal for future evaluation," *Inf. Sci.*, vol. 179, no. 12, pp. 1822–1843, 2009.

[90] Gillis, N. and Glineur, F., "Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization," *Neural Computation*, vol. 24, pp. 1085–1105, 2012.

[91] Golub, G. H. and Van Loan, C. F., *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[92] Google-Directory, "http://directory.google.com/."

[93] Granka, L. A., Joachims, T., and Gay, G., "Eye-tracking analysis of user behavior in www search," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 478–479, ACM, 2004.

[94] Griffiths, T. and Steyvers, M., "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004.

[95] Grippo, L. and Sciandrone, M., "On the convergence of the block nonlinear gauss-seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, 2000.

[96] H. Wang, C. Zhai, A. D. and Chang, Y., "Content-aware click modeling.," in *WWW*, 2013.

[97] Hajda, J. M., *The effect of time-variant acoustical properties on orchestral instrument timbres*. PhD thesis, UNIVERSITY OF CALIFORNIA, LOS ANGELES, 6 1999.

[98] Hawkes, A. G., "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, pp. 83–90, 1971.

[99] He, D., Göker, A., and Harper, D. J., "Combining evidence for automatic web session identification," *Inf. Process. Manage.*, vol. 38, no. 5, pp. 727–742, 2002.

[100] He, L., Sanocki, E., Gupta, A., and Grudin, J., "Auto-summarization of audio-video presentations," in *MULTIMEDIA*, (New York, NY, USA), pp. 489–498, ACM, 1999.

[101] Hegemann, R. A., Lewis, E. A., and Bertozzi, A. L., "An "estimate & score algorithm" for simultaneous parameter estimation and reconstruction of missing data on social networks," *accepted in Security Informatics.*, 2012.

[102] Hestenes, M. R. and Stiefel, E., "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, 1952.

[103] Hoffman, M. D., Blei, D. M., and Bach, F., "Online learning for latent dirichlet allocation," in *NIPS*, 2010.

[104] Hofmann, T., "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1/2, pp. 177–196, 2001.

[105] Hofmann, T., "Probabilistic latent semantic indexing," SIGIR, (New York, NY, USA), pp. 50–57, ACM, 1999.

[106] Hsieh, C.-J. and Dhillon, I. S., "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, (New York, NY, USA), pp. 1064–1072, ACM, 2011.

[107] Hsu, B.-J. P. and Ottaviano, G., "Space-efficient data structures for top-k completion," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 583–594, International World Wide Web Conferences Steering Committee, 2013.

[108] Hu, H., Zhang, M., He, Z., and Wang, P., "Diversifying query suggestions by using topics from wikipedia," *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 1, pp. 139–146, 2013.

[109] Hua, W., Song, Y., Wang, H., and Zhou, X., "Identifying users' topical tasks in web search," in *WSDM*, pp. 93–102, 2013.

[110] Huang, Q., Liu, Z., Rosenberg, A., Gibbo, D., and Shahraray, B., "Automated generation of news content hierarchy by integrating audio, video, and text information," in *International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[111] Hunter, D. R. and Lange, K., "A tutorial on mm algorithms," *Amer. Statist*, pp. 30–37, 2004.

[112] Jacobi, C. G. J., "Uber ein leichtes verfahren die in der theorie der sacularstroungen vorkommendern gleichungen numerisch aufzulosen," *Crelle's J.*, vol. 30, pp. 51–94, 1846.

[113] Jacobsen, M., "Point process theory and applications: Marked point and piecewise deterministic processes," 2006.

[114] Jiang, J. and Zhai, C., "Instance weighting for domain adaptation in nlp," in *ACL*, The Association for Computer Linguistics, 2007.

[115] Jiang, J.-Y., Ke, Y.-Y., Chien, P.-Y., and Cheng, P.-J., "Learning user reformulation behavior for query auto-completion," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 445–454, ACM, 2014.

[116] Johansson, F., Färdig, T., Jethava, V., and Marinov, S., "Intent-aware temporal query modeling for keyword suggestion," in *Proceedings of the 5th Ph.D. Workshop on Information and Knowledge*, PIKM '12, (New York, NY, USA), pp. 83–86, ACM, 2012.

[117] Jones, R. and Klinkner, K. L., "Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs," in *CIKM*, pp. 699–708, 2008.

[118] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K., "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 11 1999.

[119] Ju, S. X., Black, M. J., Minneman, S., and Kimber, D., "Summarization of video-taped presentations: Automatic analysis of motion and gesture," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, pp. 686–696, 1998.

[120] Kersting, K., Wahabzada, M., Thurau, C., and Bauckhage, C., "Hierarchical convex nmf for clustering massive data," in *Proceedings of the 2nd Asian Conference on Machine Learning*, ACML-10, (Tokyo, Japan), Nov 8–10 2010.

[121] Keshavan, R. H., Montanari, A., and Oh, S., "Matrix completion from a few entries," 2009.

[122] Kharitonov, E., Macdonald, C., Serdyukov, P., and Ounis, I., "User model-based metrics for offline query suggestion evaluation," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 633–642, ACM, 2013.

[123] Khuller, S., Moss, A., and Naor, J., "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[124] Kim, H., Marwah, M., Arlitt, M. F., Lyon, G., and Han, J., "Unsupervised disaggregation of low frequency power measurements.," in *SDM*, pp. 747–758, SIAM / Omnipress, 2011.

[125] Kim, H. and Park, H., "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 713–730, July 2008.

[126] Kim, H., Park, H., and Elden, L., "Non-negative tensor factorization based on alternating large-scale non-negativity-constrained least squares," in *Proceedings of IEEE 7th International Conference on Bioinformatics and Bioengineering (BIBE07)*, pp. 1147–1151, 2007.

[127] Kim, H. and Park, H., "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, pp. 1495–1502, June 2007.

[128] Kim, J., He, Y., and Park, H., "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," in *Under review*.

[129] Kim, J. and Park, H., "Sparse nonnegative matrix factorization for clustering," *CSE Technical Reports*, 2008.

[130] Kim, J. and Park, H., "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," *Data Mining, IEEE International Conference on*, vol. 0, pp. 353–362, 2008.

[131] Kim, J. and Park, H., "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, (Washington, DC, USA), pp. 353–362, IEEE Computer Society, 2008.

[132] Kim, J. and Park, H., "Fast active-set-type algorithms for l1-regularized linear regression,," in *AISTATS10*, 5 2010.

[133] Kim, J. and Park, H., "Fast nonnegative matrix factorization: An active-set-like method and comparisons," in *SIAM Journal on Scientific Computing*, vol. 33, 2011.

[134] Kim, J. and Park, H., "Least squares solution on probability simplex," Unpublished note, 2011.

[135] Kolter, J. Z., Batra, S., and Ng, A. Y., "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pp. 1153–1161, 2010.

[136] Kolter, J. Z. and Jaakkola, T., "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, pp. 1472–1482, 2012.

[137] Kolter, J. Z. and Johnson, M. J., "Redd: A public data set for energy disaggregation research.," in *Proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, 2011.

[138] Koren, Y., "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, (New York, NY, USA), pp. 426–434, ACM, 2008.

[139] Kotov, A., Bennett, P. N., White, R. W., Dumais, S. T., and Teevan, J., "Modeling and analysis of cross-session search tasks," in *SIGIR*, pp. 5–14, 2011.

[140] Kuang, D., Ding, C., and Park, H., "Symmetric nonnegative matrix factorization for graph clustering," SDM, 2012.

[141] Kulkarni, A., Teevan, J., Svore, K. M., and Dumais, S. T., "Understanding temporal query dynamics," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 167–176, ACM, 2011.

[142] Lawrence, N. D. and Platt, J. C., "Learning to learn with the informative vector machine," in *ICML*, (New York, NY, USA), p. 65, ACM, 2004.

[143] Lawson, C. L. and Hanson, R. J., "Solving least squares problems," *Society for Industrial Mathematics*, 1995.

[144] Le, D.-T. and Bernardi, R., "Query classification using topic models and support vector machine," in *Proceedings of ACL 2012 Student Research Workshop*, ACL '12, (Stroudsburg, PA, USA), pp. 19–24, Association for Computational Linguistics, 2012.

[145] Lebanon, G., "Axiomatic geometry of conditional models," *Information Theory, IEEE Transactions*, vol. 51, pp. 1283–1294, April 2005.

[146] Lee, D. D. and Seung, H. S., "Algorithms for non-negative matrix factorization," in *NIPS*, pp. 556–562, MIT Press, 2000.

[147] Lewisa, E. and Mohlerb, G., "A nonparametric em algorithm for multiscale hawkes processes," *Journal of Nonpara-metric Statistics*, vol. 1, 2011.

[148] Li, H., Adali, T., and Wang, W., "Non-negative matrix factorization with orthogonality constraints for chemical agent detection in raman spectra," in *Machine Learning for Signal Processing, 2005 IEEE Workshop on*, pp. 253–258, 9 2005.

[149] Li, L., Lebanon, G., and Park, H., "Fast bregman divergence nmf using taylor expansion and coordinate descent," KDD, (New York, NY, USA), pp. 307–315, ACM, 2012.

[150] Li, L. and Zha, H., "Dyadic event attribution in social networks with mixtures of hawkes processes," CIKM, (New York, NY, USA), pp. 1667–1672, ACM, 2013.

[151] Li, L. and Zha, H., "Learning parametric models for social infectivity in multi-dimensional hawkes processes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 101–107, 2014.

[152] Li, L., Zhou, K., Xue, G.-R., Zha, H., and Yu, Y., "Enhancing diversity, coverage and balance for summarization through structure learning," in *WWW*, (New York, NY, USA), pp. 71–80, ACM, 2009.

[153] Li, L., Deng, H., Dong, A., Chang, Y., and Zha, H., "Identifying and labeling search tasks via query-based hawkes processes," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 731–740, 2014.

[154] Li, W. and McCallum, A., "Pachinko allocation: Dag-structured mixture models of topic correlations," 2006.

[155] Li, Y., Lee, S.-H., Yeh, C.-H., and Kuo, C. C. J., "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," *Signal Processing Magazine*, vol. 23, no. 2, pp. 79–89, 2006.

[156] Li, Y., Dong, A., Wang, H., Deng, H., Chang, Y., and Zhai, C., "A two-dimensional click model for query auto-completion," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, (New York, NY, USA), pp. 455–464, ACM, 2014.

[157] Liao, Z., Song, Y., He, L.-w., and Huang, Y., "Evaluating the effectiveness of search task trails," in *Proceedings of the 21st international conference on World Wide Web*, pp. 489–498, ACM, 2012.

[158] Lin, C. Y. and Hovy, E., "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *NAACL*, (Morristown, NJ, USA), pp. 71–78, Association for Computational Linguistics, 2003.

[159] Lin, C.-J., "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, October 2007.

[160] Liu, C., Guo, F., and Faloutsos, C., "Bayesian browsing model: Exact inference of document relevance from petabyte-scale data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 4, p. 19, 2010.

[161] Liu, Y. and Stone, P., "Value-function-based transfer for reinforcement learning using structure mapping," in *AAAI*, pp. 415–420, AAAI Press, 2006.

[162] Lloyd, S. P., "Least square quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[163] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[164] Lucchese, C., Orlando, S., Perego, R., Silvestri, F., and Tolomei, G., "Identifying task-based sessions in search engine query logs," in *WSDM*, pp. 277–286, 2011.

[165] Lucchese, C., Orlando, S., Perego, R., Silvestri, F., and Tolomei, G., "Discovering tasks from search engine query logs," *ACM Trans. Inf. Syst.*, vol. 31, no. 3, p. 14, 2013.

[166] M. Sonka, V. Hlavac, R. B., *Image Processing, Analysis, and machine vision*. 2007.

[167] Ma, H., Lyu, M. R., and King, I., "Diversifying query suggestion results," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI, 2010.

[168] Mazumder, R., Hastie, T., and Tibshirani, R., "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 99, pp. 2287–2322, August 2010.

[169] Mei, Q., Zhou, D., and Church, K., "Query suggestion using hitting time," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, (New York, NY, USA), pp. 469–478, ACM, 2008.

[170] Mei, Q., Klinkner, K. L., Kumar, R., and Tomkins, A., "An analysis framework for search sequences," in *CIKM*, pp. 1991–1994, 2009.

[171] Melucci, M., "Contextual search: A computational framework," *Foundations and Trends in Information Retrieval*, vol. 6, no. 4-5, pp. 257–405, 2012.

[172] Mendez, M. O., Matteucci, M., Castronovo, V., Strambi, L. F., Cerutti, S., and Bianchi, A. M., "Sleep staging from heart rate variability: time-varying spectral features and hidden markov models," *International Journal of Biomedical Engineering and Technology*, vol. 3, 1 2010.

[173] Merritt, M. and Zhang, Y., "An interior-point gradientmethod for large-scale totally nonnegative least squares problems," *J. Optimization Theory and Applications*, vol. 126, pp. 191–202, 2005.

[174] Mihalkova, L., Huynh, T., and Mooney, R. J., "Mapping and revising markov logic networks for transfer learning," in *AAAI*, pp. 608–614, AAAI Press, 2007.

[175] Mills, M., Cohen, J., and Wong, Y. Y., "A magnifier tool for video data," in *CHI*, (New York, NY, USA), pp. 93–98, ACM, 1992.

[176] Minka, T. and Lafferty, J., "Expectation-propagation for the generative aspect model," in *UAI*, (San Francisco, CA, USA), pp. 352–359, 2002.

[177] Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E., "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, pp. 100–108, Jan 2011.

[178] Morimoto, T., "Markov processes and the h-theorem," *Journal of the Physical Society of Japan*, vol. 18, no. 3, pp. 328–331, 1963.

[179] Motoyama, M., McCoy, D., Levchenko, K., Savage, S., and Voelker, G. M., "Dirty jobs: the role of freelance labor in web service abuse," in *Proceedings of the 20th USENIX conference on Security*, SEC'11, (Berkeley, CA, USA), pp. 14–14, USENIX Association, 2011.

[180] Mukherjee, A., Liu, B., and Glance, N., "Spotting fake reviewer groups in consumer reviews," in *WWW*, (New York, NY, USA), pp. 191–200, ACM, 2012.

[181] Myers, S. A. and Leskovec, J., "On the convexity of latent social network inference.," in *NIPS*, pp. 1741–1749, Curran Associates, Inc., 2010.

[182] Nallapati, R., Lafferty, J., and Cohen, W., "Multi-scale topic tomography," 2007.

[183] Neenan, B. and Robinson, J., "Residential electricity use feedback: A research synthesis and economic frame- work.," *Technical report, Electric Power Research Institute.*, 2009.

[184] Ngo, C.-W., Ma, Y.-F., and Zhang, H.-J., "Video summarization and scene detection by graph modeling," *IEEE transactions on circuits and systems for video technology*, vol. 15, no. 2, pp. 296–305, 2005.

[185] Nocedal, J. and Wright, S. J., "Numerical optimization," Springer, 2006.

[186] Ogata, Y., "The asymptotic behaviour of maximum likelihood estimators for stationary point processes," *Annals of the Institute of Statistical Mathematics.*, vol. 30, no. 1, pp. 243–261, 1978.

[187] Ogata, Y., "On lewis' simulation method for point processes," *IEEE Trans. Inf. Theor.*, vol. 27, pp. 23–31, 1 1981.

[188] Ogata, Y., "Statistical models for earthquake occurrences and residual analysis for point processes," *Journal of the American Statistical Association.*, vol. 83, no. 401, pp. 9–27, 1988.

[189] Ogata, Y., "Space-time point-process models for earthquake occurrences," *Annals of the Institute of Statistical Mathematics.*, vol. 50, pp. 379–402, Jun 1998.

[190] Oliva, A. and Torralba, A., "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[191] Open-Directory-Project, "http://www.dmoz.org/."

[192] Ott, M., Choi, Y., Cardie, C., and Hancock, J. T., "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th*

Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, (Stroudsburg, PA, USA), pp. 309–319, Association for Computational Linguistics, 2011.

[193] Ozmutlu, H. C. and Çavdur, F., "Application of automatic topic identification on excite web search engine data logs," *Inf. Process. Manage.*, vol. 41, no. 5, pp. 1243–1262, 2005.

[194] Parson, O., Ghosh, S., Weal, M., and Rogers, A., "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 22 -26, 2012, Toronto, Canada*, pp. 356–362, 2012.

[195] Peharz, R., Stark, M., and Pernkopf, F., "Sparse nonnegative matrix factorization using l0-constraints," in *MLSP*, pp. 83–88, Aug 2010.

[196] Perry, P. O. and Wolfe, P. J., "Point process modeling for directed interaction networks," in *Journal of the Royal Statistical Society*, 2013.

[197] Pietra, S. D., Pietra, V. D., and Lafferty, J., "Duality and auxiliary functions for bregman distances," tech. rep., School of Computer Science, Carnegie Mellon University, 2002.

[198] Piwowarski, B., Dupret, G., and Jones, R., "Mining user web search activity with layered bayesian networks or how to capture a click in its context," in *WSDM*, pp. 162–171, 2009.

[199] Porter, M. D. and White, G., "Self-exciting hurdle models for terrorist activity," *The Annals of Applied Statistics*, vol. 6, no. 1, pp. 106–124, 2011.

[200] Radlinski, F. and Joachims, T., "Query chains: learning to rank from implicit feedback," in *KDD*, pp. 239–248, 2005.

[201] Raleigh, C., Linke, A., Hegre, H., and Karlsen, J., "Introducing acled: An armed conflict location and event dataset special data feature," *Journal of Peace Research*, vol. 47, pp. 651–660, July 2010.

[202] Rasmussen, J. G., "Bayesian inference for hawkes processes," *Methodology and Computing in Applied Probability*, vol. 15, p. 623642, 9 2013.

[203] Rennie, J. D. M. and Srebro, N., "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*, ICML '05, (New York, NY, USA), pp. 713–719, ACM, 2005.

[204] Richardson, M., Dominowska, E., and Ragno, R., "Predicting clicks: estimating the click-through rate for new ads," in *Proceedings of the 16th international conference on World Wide Web*, pp. 521–530, ACM, 2007.

[205] Rodriguez, M. G., Balduzzi, D., and Scholkopf, B., "Uncovering the temporal dynamics of diffusion networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (Getoor, L. and Scheffer, T., eds.), (New York, NY, USA), pp. 561–568, ACM, 2011.

[206] Russell, D. M., "A design pattern-based video summarization technique: Moving from low-level signals to high-level structure," *Hawaii International Conference on System Sciences*, vol. 3, p. 3048, 2000.

[207] S. Feng, L. X., Gogar, A., and Choi, Y., "Distributional footprints of deceptive product reviews," in *International AAAI Conference on WebBlogs and Social Media*, ICWSM '12, 2012.

[208] Saad, Y. and Schultz, M., "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput*, vol. 7, pp. 856–869, 1986.

[209] Salakhutdinov, R. and Mnih, A., "Probabilistic matrix factorization," 2007.

[210] Salakhutdinov, R. and Mnih, A., "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*, ICML '08, (New York, NY, USA), pp. 880–887, ACM, 2008.

[211] Schoenberg, F., "Introduction to point processes," *Wiley Encyclopedia of Operations Research and Management Science*, pp. 616–617, 2010.

[212] Settles, B. and Craven, M., "An analysis of active learning strategies for sequence labeling tasks," in *EMNLP*, (Morristown, NJ, USA), pp. 1070–1079, Association for Computational Linguistics, 2008.

[213] Shahnaz, F., Berry, M. W., Pauca, V. P., and Plemmons, R. J., "Document clustering using nonnegative matrix factorization," *Inf. Process. Manage.*, vol. 42, pp. 373–386, March 2006.

[214] Shen, X., Dumais, S., and Horvitz, E., "Analysis of topic dynamics in web search," in *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, (New York, NY, USA), pp. 1102–1103, ACM, 2005.

[215] Shen, X., Tan, B., and Zhai, C., "Context-sensitive information retrieval using implicit feedback," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 43–50, ACM, 2005.

[216] Shokouhi, M., "Detecting seasonal queries by time-series analysis," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in information retrieval*, pp. 1171–1172, ACM, 2011.

[217] Shokouhi, M., "Learning to personalize query auto-completion," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 103–112, ACM, 2013.

[218] Shokouhi, M. and Radinsky, K., "Time-sensitive query auto-completion," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 601–610, ACM, 2012.

[219] Sieg, A., Mobasher, B., and Burke, R., "Web search personalization with ontological user profiles," in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, (New York, NY, USA), pp. 525–534, ACM, 2007.

[220] Simma, A. and Jordan, M., "Modeling events with cascades of poisson processes," *UAI*, 2010.

[221] Singh, P. A. and Gordon, G. J., "A unified view of matrix factorization models," ECML PKDD, (Berlin, Heidelberg), pp. 358–373, Springer-Verlag, 2008.

[222] Sloan, M. and Wang, J., "Iterative expectation for multi period information retrieval," *arXiv preprint arXiv:1303.5250*, 2013.

[223] Smith, M. A. and Kanade, T., "Video skimming and characterization through the combination of image and language understanding techniques," pp. 370–382, 2001.

[224] Song, Y., Zhou, D., and He, L., "Query suggestion by constructing term-transition graphs," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, (New York, NY, USA), pp. 353–362, ACM, 2012.

[225] Spink, A., Koshman, S., Park, M., Field, C., and Jansen, B. J., "Multitasking web search on vivisimo.com," in *ITCC (2)*, pp. 486–490, 2005.

[226] Stewart, G. W., "Accelerating the orthogonal iteration for the eigenvalues of a hermitian matrix," *Numer. Math.*, vol. 13, pp. 362–376, 1969.

[227] Stomakhin, A., Short, M. B., and Bertozzi, A. L., "Reconstruction of missing data in social networks based on temporal patterns of interactions," *Inverse Problems.*, vol. 27, Nov 2011.

[228] Strohmaier, M., Kröll, M., and Körner, C., "Intentional query suggestion: Making user goals more explicit during search," in *Proceedings of the 2009 Workshop on Web Search Click Data*, WSCD '09, (New York, NY, USA), pp. 68–74, ACM, 2009.

[229] Sun, A. and Lou, C., "Towards context-aware search with right click," in *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pp. 847–850, 2014.

[230] Swan, R. and Allan, J., "Extracting significant time varying features from text," in *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, (New York, NY, USA), pp. 38–45, ACM, 1999.

[231] Taniguchi, Y., Akutsu, A., Tonomura, Y., and Hamada, H., "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *MULTIMEDIA*, (New York, NY, USA), pp. 25–33, ACM, 1995.

[232] Taskiran, C. M., Pizlo, Z., Amir, A., Ponceleon, D., and Delp, E. J., "Automated video program summarization using speech transcripts," *Multimedia*, vol. 8, no. 4, pp. 775–791, 2006.

[233] Taylor, M. E. and Stone, P., "Cross-domain transfer for reinforcement learning," in *ICML*, (New York, NY, USA), pp. 879–886, ACM, 2007.

[234] Teevan, J., Adar, E., Jones, R., and Potts, M. A. S., "Information re-retrieval: repeat queries in yahoo's logs," in *SIGIR*, pp. 151–158, 2007.

[235] Teh, Y., Jordan, M., Beal, M., and Blei, D., "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 576, pp. 1566–1581, 2006.

[236] Tibshirani, R. and Hinton, G., "Coaching variables for regression and classification," *Statistics and Computing*, vol. 8, no. 1, pp. 25–33, 1998.

[237] Truong, B. T. and Venkatesh, S., "Generating comprehensible summaries of rushes sequences based on robust feature matching," in *TVS*, (New York, NY, USA), pp. 30–34, ACM, 2007.

[238] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y., "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, pp. 1453–1484, 2005.

[239] Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J., "Video manga: generating semantically meaningful video summaries," in *MULTIMEDIA*, (New York, NY, USA), pp. 383–392, ACM, 1999.

[240] Vallet, D. and Castells, P., "Personalized diversification of search results," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, (New York, NY, USA), pp. 841–850, ACM, 2012.

[241] Vu, D. Q., Asuncion, A. U., Hunter, D. R., and Smyth, P., "Dynamic egocentric models for citation networks," in *ICML*, pp. 857–864, 2011.

[242] Wang, H., Song, Y., Chang, M.-W., He, X., White, R. W., and Chu, W., "Learning to extract cross-session search tasks," in *WWW*, pp. 1353–1364, 2013.

[243] Wang, S. and Schuurmans, D., "Learning continuous latent variable models with bregman divergences," in *In Proc. IEEE International Conference on Algorithmic Learning Theory*, p. 2004, 2003.

[244] Wang, X. and Grimson, E., "Spatial latent dirichlet allocation," NIPS '07, 2007.

[245] Wang, Y., Agichtein, E., and Benzi, M., "Tm-lda: efficient online modeling of latent topic transitions in social media," in *KDD*, (New York, NY, USA), pp. 123–131, ACM, 2012.

[246] Watts, D. J. and Strogatz, S. H., "Collective dynamics of 'small-world' networks," *Nature*, pp. 440–442, 6 1998.

[247] White, R. W., Chu, W., Hassan, A., He, X., Song, Y., and Wang, H., "Enhancing personalized search by mining and modeling task behavior," in *WWW*, pp. 1411–1420, 2013.

[248] White, R. W. and Marchionini, G., "Examining the effectiveness of real-time query expansion," *Information Processing & Management*, vol. 43, no. 3, pp. 685–704, 2007.

[249] Whiting, S. and Jose, J. M., "Recent and robust query auto-completion," in *Proceedings of the 23rd international conference on World wide web*, pp. 971–982, International World Wide Web Conferences, 2014.

[250] Wytock, M. and Kolter, J. Z., "Contextually supervised source separation with application to energy disaggregation," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 486–492, 2014.

[251] Xing, E. P., Fu, W., and Song, L., "A state-space mixed membership blockmodel for dynamic network tomography," in *the Annals of Applied Statistics*, pp. 535–566, the Institute of Mathematical Statistics, 2010.

[252] Xiong, Z., Radhakrishan, R., Divakaran, A., and Ishikawa, Y., "Generation of sports highlights using motion activity in combination with a common audio feature extraction framework," in *ICIP*, 2003.

[253] 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.

[254] Yan, J., Wang, Y., Zhou, K., Huang, J., Tian, C., Zha, H., and Dong, W., "Towards effective prioritizing water pipe replacement and rehabilitation," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 2931–2937, AAAI Press, 2013.

[255] Yan, J., Zhang, C., Zha, H., Gong, M., Sun, C., Huang, J., Chu, S., and Yang, X., "On machine learning towards predictive sales pipeline analytics," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.*, 2015.

[256] Yang, S. and Zha, H., "Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes," in *AISTATS*, pp. 641–649, 2013.

[257] Yang, S. and Zha, H., "Mixture of mutually exciting processes for viral diffusion," in *ICML*, 2013.

[258] Yeung, M. M. and Yeo, B. L., "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 771–785, 1997.

[259] Yu, C.-N. J. and Joachims, T., "Learning structural svms with latent variables," in *ICML*, (New York, NY, USA), pp. 1169–1176, ACM, 2009.

[260] Yue, Y. and Joachims, T., "Predicting diverse subsets using structural svms," in *ICML*, (New York, NY, USA), pp. 1224–1231, ACM, 2008.

[261] Yun, S. and Woo, H., "Linearized proximal alternating minimization algorithm for motion deblurring by nonlocal regularization," *Pattern Recogn.*, vol. 44, pp. 1312–1326, Jun 2011.

[262] Z.-Mangion, A., Dewarc, M., Kadirkamanathand, V., and Sanguinetti, G., "Point process modelling of the afghan war diary," *PNAS*, vol. 109, pp. 12414–12419, July 2012.

[263] Zdunek, R. and Cichocki, A., "Non-negative matrix factorization with quasi-newton optimization," *Springer LNAI*, vol. 4029, pp. 870–879, 2006.

[264] Zdunek, R. and Cichocki, A., "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, pp. 1904–1916, 2007.

[265] Zhang, J., "Divergence function, duality, and convex analysis," *Neural Comput.*, vol. 16, pp. 159–195, January 2004.

[266] Zhang, X. and Mitra, P., "Learning topical transition probabilities in click through data with regression models," in *Procceedings of the 13th International Workshop on the Web and Databases*, WebDB '10, (New York, NY, USA), pp. 11:1–11:6, ACM, 2010.

[267] Zhang, Y., Chen, W., Wang, D., and Yang, Q., "User-click modeling for understanding and predicting search-behavior," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1388–1396, ACM, 2011.

[268] Zhang, Z. and Nasraoui, O., "Mining search engine query logs for query recommendation," in *Proceedings of the 15th international conference on World Wide Web*, WWW '06, (New York, NY, USA), pp. 1039–1040, ACM, 2006.

[269] Zheng, X., Lai, Y. M., Chow, K. P., Hui, L. C. K., and Yiu, S. M., "Sockpuppet detection in online discussion forums," in *Proceedings of the 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP '11, (Washington, DC, USA), pp. 374–377, IEEE Computer Society, 2011.

[270] Zhou, K., Zha, H., and Song, L., "Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes.," in *AISTATS*, vol. 31 of *JMLR Proceedings*, pp. 641–649, JMLR.org, 2013.

[271] Zhu, X., Fan, J., Elmagarmid, A. K., and Wu, X., "Hierarchical video content description and summarization using unified semantic and visual similarity," *Multimedia Syst.*, vol. 9, no. 1, pp. 31–53, 2003.

[272] Zhu, Z. A., Chen, W., Minka, T., Zhu, C., and Chen, Z., "A novel click model and its applications to online advertising," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 321–330, ACM, 2010.

[273] Zhuang, J., Ogata, Y., and Jones, D. V., "Stochastic declustering of space-time earthquake occurrences," *Journal of the American Statistical Association.*, vol. 97, no. 458, pp. 369–380, 2002.

VITA

Liangda Li is a fifth year Ph.D. student in the School of Computer Science at Georgia Institute of Technology, under the supervision of Professor Hongyuan Zha. He received his B.S. degree in Computer Science from Shanghai Jiao-Tong University in 2010.

His research interest includes machine learning and its applications in web search and social network. In particular, his focus in on modeling influence in various real-world behavioral data.

He has hold intern positions in Google and Yahoo Labs in 2011-2014, working on online advertising, data validation, video recommendation, and query auto-completion, respectively.