Prediction and Recommendation in Online Media

by

Dawei Yin

A Dissertation
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy
in
Computer Science

Lehigh University
September 2013

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dawei Yin

Prediction and Recommendation in Online Media

____

Date

____

Professor Brian D. Davison, Dissertation Director, Chair

(Must Sign with Blue Ink)

____

Accepted Date

Committee Members

____

Professor Henry S. Baird

____

Professor Xiaolei Huang

____

Professor Wei-Min Huang

Acknowledgements

First and foremost, I heartily thank my advisor, Prof. Brian D. Davison, for his insightful guidance, encouragement, patience and suggestions throughout my Ph.D. process. Prof. Davison provides me the freedom through my Ph.D. to explore various research problems where he always gave stimulating and fruitful discussions. About research, he taught me a lot from writing and preparing submissions to seeing the big picture, finding the potential direction. Prof. Davison is not only an advisor for research, but also a mentor for many aspects in my life, especially for career planning. Without his advice and support, I would not be where I am today.

Parts of this dissertation were done when I was an intern in Xerox Research Center Europe and Microsoft Research Asia. I owe my gratitude to Dr. Cedric Archambeau, Dr. Boris Chidlovskii, Dr. Guillaume Bouchard and Dr. Shengbo Guo from Xerox Research Center Europe and Dr. Bin Cao and Dr. Jian-tao Sun from Microsoft Research Asia for offering me the internship opportunities at these companies. The internship experiences provided me valuable opportunities to work with, and learn from talented experts in my research area, to acquire first-hand knowledge in industry, and to gain access to large-scale, real-world data and high performance computing resources.

I am also grateful to my committee members, professors Henry S. Baird, Xiaolei Huang and Wei-Min Huang for their excellent guidance and suggestions. Prof. Baird was really insightful about research and gave me great help during the early stage of my Ph.D. studies.

Many thanks to my lab mates, Na Dai, Ovidiu Dan, Liangjie Hong, Xiaoguang Qi, Jian Wang, Zaihan Yang, Xiong Xiong and Zhenzhen Xue. I appreciate their help in providing discussions and doing experiments with me on large real data sets to test approaches proposed in this dissertation. As a senior member in the lab, Xiaoguang provided endless help for my life and research projects. I always enjoyed discussing with Liangjie on research. I also thank my friends, Lifeng Shang and Yong Zhang from The University of Hong Kong, who provided many helpful discussions on my research.

Finally, I am deeply indebted to my dear mother and father for their love and strong support during my graduate study. They have both been with me through the whole process with its ups and downs.

Much of the work presented in this dissertation was supported in part by a grant from the National Science Foundation under award IIS-0545875, as well as an equipment grant from Sun Microsystems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Dedication

To my parents, Haozhi Yin & Jianmei Chang.

Contents

List of Tables

List of Figures

Abstract

With billions of internet users, online media services have become commonplace. Prediction and recommendation for online media are fundamental problems in various applications, including recommender systems and information retrieval. As an example, accurately predicting user behaviors improves user experiences through more intelligent user interfaces. On the other hand, user behavior prediction in online media is also strongly related to behavior targeting and online advertisement which is the major business for most consumer internet services. Estimating and understanding users' click behaviors is a critical problem in online advertising.

In this dissertation, we investigate the prediction and recommendation problems in various online media. We find a number of challenges: high order relations, temporal dynamics, complexity of network structure, high data sparsity and coupled social media activities. We consider user behavior understanding and prediction in four areas: tag prediction in a social tagging system, link prediction in microblogging services, multi-context modeling in online social media and click prediction in sponsored search. In such topics, based on real world data, we analyze user behaviors and discover patterns, properties and challenges. Subsequently, we design specific models for online user behavior prediction in various online media: a probabilistic model for personalized tag prediction, a user-tag-specific temporal interests model for tracking users' interests over time in tagging systems, a personalized structure-based link prediction model for micro-blogging systems, a generalized latent factor model and Bayesian treatment for modeling across multiple contexts in online social media, a context-aware click model and framework for estimating ad group performance in sponsored search. Our extensive experiments on large-scale real-world datasets show our novel models advance the state-of-the-art.

Chapter 1

Overview

In this chapter, we provide an overview of the full dissertation. We start by briefly reviewing the evolution of the World Wide Web and online user behaviors in Web 2.0. Then we study the organization of some current online media and discuss problems and challenges in predicting online user behaviors. Finally, we summarize our contributions to various applications in four topics: tag prediction in social tagging, link prediction in microblogging, modeling across multiple contexts in online social media and click prediction in sponsored search.

1.1 Online User Behaviors

The traditional web (also called Web 1.0) was entirely made up of Web pages connected by hyperlinks. Users could only browse webpages but could not contribute to the content of the webpages. Few interactions could be made by users under this scheme. Although some forms such as e-mail and newsgroups provide basic user interactions and engagement in the early era of the web, the modern web (also called Web 2.0) reflects the development and evolution of web service and applications. With hundreds of millions of participants, online media services have become commonplace. Compared to the traditional web where users could only view webpages, the modern web provides many opportunities for user engagement, where users can generate their content and contribute to the World Wide Web with text (Wordpress[145, 134]), photos (Flickr[41]), videos (Youtube[164]), etc. Users are intensively involved in the online services: one could create a profile and become an entity in online services; users could share resources/opinions with others; users could declare friendship with other users, etc. Take Flickr as an example; in addition to being able to share content, users can often 1) rate content, 2) declare friendship with other users, 3) tag content with keywords, 4) comment on content, and 5) send personal or public messages to other users. Each of these activities provides valuable data to the service that can be used to model and predict future actions. For example, in a photo sharing site like Flickr, users of the service can add photos shared by others to their favorites. This is a form of rating, and so the service could examine a collection of user-photo pairings and build a model to predict whether the user would mark this photo as a favorite (e.g., what photos are preferred). Declaring friendship is similar to the more general activity of indicating the existence of a link between two entities (e.g., user-user rather than user-photo) which are a common context for recommender systems [22, 30, 93, 73].

Studying human behaviors is a traditional research topic in Sociology, Anthropology, Psychology, etc. However, most traditional research is based on surveys, field research or small scale datasets. For instance, in his famous social network experiments, Stanley Milgram challenged people to route postcards to a fixed recipient by passing them only through direct acquaintances [132]. Milgram found that that human society is a small-world-type network characterized by short path-lengths and the average number of intermediaries on the path of the postcards lay between

4.4 and 5.7, depending on the samples of people chosen. In that experiment, the connections of people are presented by mailing. However, such kinds of experiments are not easy to be conducted in a large scale. It is not feasible to involve millions of people to do paper-based survey or field research. When the experiments are conducted through the online social networks, e.g., Facebook and Twitter, it becomes much easier and feasible. For example, Backstrom et al. [8] report the results of the first world-scale social-network graph distance computations, using the entire Facebook network of active users (721 million users, 69 billion friendship links). The average distance we observe is 4.74, corresponding to 3.74 intermediaries or "degrees of separation".

Besides social network research, other studies in human behaviors, e.g., purchasing, conversations, also have been explored for decades. Nowadays, such behaviors have been more and more intensively associated to online media. For instance, in Facebook, not only users' social connections are presented, but also different behaviors (e.g., sharing photos, update status, personal messages) can be performed by users. Instead of going to a retail store, online shopping has become common and more convenient. By tracking online user behaviors, some similar experiments and research can be conducted more easily and accurately. However, although some classical theories and methodologies developed in traditional areas provide the foundations for modern analysis of online user behaviors, most of them cannot be directly applied to online settings due to their qualitative nature and also due to some of their case-by-case style of studies that cannot be scaled to the amount of data online. On the other hand, online user behaviors have their own properties and characteristics. Comparing traditional user behavior analysis, online media shows higher dynamics, such as higher speed information distribution and light-weight behaviors (e.g., clicks). Since classical behavior research was conducted prior to the time of popularity of the Internet, few characteristics of online behaviors were taken into consideration. The conclusions and results obtained through the classical methods are also needed to be re-verified in the new era as well.

It is clear that understanding online user behaviors is meaningful in traditional areas such as Sociology, Psychology etc. Moreover, the other applications of understanding online user behaviors, interests play important roles in three aspects: 1) accurately predicting user behaviors improves user experiences. For instance, in a social tagging system, when a user is trying to save a bookmark, we can predict the tags which the user will use on the bookmark and the user will select the suggested tags rather than typing tags. In an online video sharing system, when a user watches a video, we can predict which video that the user will watch subsequently and then we can preload that video for the user. 2) information filtering. The data and information in online media are growing explosively and it becomes difficult for users to capture the useful and interesting information effectively. Understanding user interests and filtering useful information for users in online media is more important than in traditional media. For instance, in a microblogging system, millions of messages may be generated in a very short time. A user can only received the messages produced by the friends that one follows, but there is likely to be a number of messages which the user does not receive but is interested in. Since it is impossible for user to browse all messages, understanding user interests and filtering out non-informational messages and mining the messages which users are interested in is quite important. 3) behavior targeting and online advertising. User behaviors or interests prediction is also strongly related to online advertising which is the major business for internet services. For instance, pay-per-click (PPC) is an advertising model that has been adopted by sponsored search markets. Under the PPC model, advertisers are charged when their advertisements are clicked by search engine users. More clicks bring more revenue to search engine companies [18], which has triggered research into related issues [13, 106]. Estimating and understanding users' click behaviors of a search ad is an important problem for modern commercial search engines.

1.2 Research Challenges

However, understanding user interests and behaviors in online media is not trivial, and analysis of data from social media services can be challenging, especially when considering 1) high order relations, 2) temporal dynamics, 3) complexity of network structure, 4) high sparsity of the data, and 5) co-related (coupled) activities. We now consider each of these challenges in more detail:

High order relations

In many social media contexts, user activities generate relations involving more than two types of entities. In contrast, most existing work is focused on traditional second order relations that involve just two types of entities. For instance, user-movie ratings involve pairs of users and movies.[1] However, in social media, typically data coupled only by second order relations are considered (presumably for simplicity). For example, by involving a temporal factor, user-movie-time ratings become third order relations, and thus naturally model users' preferences with temporal information [73, 148]. In social tagging systems, the posts (user-tag-item) are by nature third order data [114, 112, 159, 158]. Similarly, user-tag-item-time is an example of a fourth order relation. However, these types of higher order relations have rarely been studied due to the complexity and difficulty in modeling and inference. However, they are needed: if one is required to predict a user's comments on an item,

[1] The responses for pairs of users and movies are explicit ratings while others might be implicit e.g., user-user connections. Both implicit and explicit response data involves two types of entities, thus we define this as second order relational data.

traditional second order relations are not applicable. This comment prediction problem is related to but different from traditional opinion mining [34, 63], where the comment is classified or summarized.

Temporal dynamics

Past research on recommender system and user behavior prediction has shown that accuracy can be significantly improved by modeling users preferences [114, 112, 159, 158]. However, these preferences are usually treated as constant over time, neglecting the temporal factor within user interests. User interests may change and evolve dynamically [73]. Modeling temporal changes in user preferences brings unique challenges. The change of user preferences may be from two aspects: 1) Macro change, for example, the emergence of new products or services (e.g., seasonal changes, or specific holidays) will trigger the change of trends and most user preferences. 2) Micro change, the change of the individual's taste in movies and musics. All those changes cannot be captured and modeled easily, which also brings a challenge to current research.

Complexity of network structure

Random walk-based methods such as PageRank[19], HITS[69] have been used for traditional link analysis, especially for hyperlinks, but as online social networks and social media surging recently, the network structure became more complex. For instance, typical social network contains symmetric relationship and a number of properties (e.g., homophily) show the complexity of social network structures. More recently, in microblog services such as Twitter, Yammer and Weibo, relationships between two users may be asymmetric, leading to three types of link relationships. As a result, the network connecting users in a microblogging service will likely have properties of both a social network in which participants connect to establish or represent social relationships between them and an information network in which people connect so that information can be passed from a producer to a consumer (and perhaps retransmitted to many other users). It is even harder and more complex to analyze such an asymmetric micro-blogging network.

Sparse data

In most social media, the natural form of data representation results in significant data sparsity, and in extreme cases when a new entity is added there may be no information on which to build a prediction (a.k.a., the cold start problem [6, 163]). Indeed, data in typical recommendation tasks can be very sparse: Netflix users select from tens of thousands of movies, but relatively few of them are rated by any given user and there are even users who have not rated any movies [5, 163]. The sparsity of data is even more serious when the relations between entities involve many entities and are coupled.

Coupled social media activities

Many social media services (e.g., in Facebook, YouTube, and Flickr) involve user activities which create relations between entities. For example, users, items, comments, and messages are frequently used in multiple contexts. This will cause the problem that the relational data from different contexts are coupled together. Some activities are strongly related to each other: for instance, activities where users comment on items or where users rate items share two of the same types of entity—user and item. Another example is the temporal factor which is shared by all activities, and cannot be modeled separately in different contexts if one wishes to see the system evolve as a whole. Therefore, activities in social media contexts are often naturally coupled together, increasing the difficulty of the problem.

1.3 Network View of Online Media

In this dissertation, we will study the problem—understanding and prediction of online user behaviors—in different domains, such as social tagging system (e.g., Flickr, Bibsonomy), microblogging system (e.g., Twitter), movie rating system (e.g., Netflix, MovieLens) and online advertising system (e.g., Bing sponsored search). Before we dig into the details of the problems, let's review these online medias at first.

The resource sharing and structure of online services in Web 2.0 are often studied based on the types of networks. One type of network is a homogenous network, where the types of nodes are all the same. The most common one among these networks is the friendship network, where the users can make connections with other users. There is no direction information on the links. Once two users are connected, they will become mutual friends with each other. The typical web services of this network are Facebook [40], Linkedin [83] and Renren [115]. The problem will be to predict the potential friendship between users. The second type of the homogenous network is the directed network, where there is direction on the links. For example, user u may be interested in user v, but user v may not be interested in user v. The relationship between u and v will be $u \to v$. The typical web service of directed homogenous network is microblogging network, such as Twitter [135], Weibo [124] and Google Plus [48]. In microblogging services, participants form an explicit social network by following (subscribing to) another user and thus automatically receive the (short) messages generated by the target user. Unlike common online social networks such as Facebook, LinkedIn and Renren, a followed user has the option but not the requirement to similarly follow back. Thus, relationships in these social networks may be asymmetric, leading to three kinds of link relationships between users A and B. If A follows B, we say that A is a follower of B, and that B is a friend of A. If A and B both follow each other, we consider them mutual friends or reciprocal friends. Thus, user B in a microblog service can generate messages, which are generally public and searchable, and any followers of B, such as A, will automatically receive those messages along with messages generated by all other users that A follows. The combination of multiple message intentions and asymmetry of connections has led some to call microblogging services such as Twitter hybrid networks [76, 136]. They are hybrid not just because they can carry multiple types of messages, but also because participants create links for multiple reasons to be social (e.g., to connect online to existing offline social contacts) or to link to information sources.

Another type network is the heterogeneous network, in which the types of nodes can be different e.g., Netflix [102], Youtube [164]. Take Netflix as an example: users can connect with movies and that means, if some user likes a movie, then we consider this user has connected to the movie. From this view, the problem actually is a link prediction in bipartite graph problem which is also treated as a recommendation problem. In some web service, such as Netflix, not only to predict links, the users could rate the movie, in four levels—1,2,3,4. Higher ratings represent the user prefers the movie over other movies. The problem will be to predict the potential ratings which users rate the movies and then the system could recommend the movies to the users. Another similar application is online advertising and behavior targeting where we try to find the best ads for the specific users who are likely to click. Sponsored search (one type of online advertising) is the major business for todays commercial search engines, in which more clicks bring more revenue to search engine companies. Although the responses of traditional recommender system (e.g., movie, photo recommendation) and online advertising are different, the technical solutions are essentially the same under the view of heterogeneous network. In both problems, we study relation user-item, where item depends on the context (e.g., items are movies in Netflix and ads in online advertising). Generally speaking, such systems are based on one of two strategies [75]: 1) content filtering approach. It creates a profile for each user or item to characterize its nature. For example, movie features could be extracted from its genre, the participating actors and so forth. User profiles might include demographic information or answers provided on a suitable questionnaire. The profiles allow programs to associate users with matching products. The movie features and user profiles may not be easy to collect. 2) collaborative filtering[44]. It relies only on past user behaviors for example, previous transactions or product ratings without requiring the creation of explicit profiles.

In heterogeneous networks, another research topic is provided by social tagging systems, e.g., Flickr [41], Delicious [151]. In a social tagging system, the users could tag or bookmark the items which the users like. Unlike the movie recommendation system, the users could use a word or several words to describe the items rather than ratings. The item could be different types. For example, in Delicious, the items are webpages, in Flickr, the items are images, in Youtube the items are videos. Social bookmarking has already showed its value in many areas, such as query expansion [12], web search [10], personalized search [121, 149], web resource classification [162] and clustering [110]. A better understanding and prediction of tags on web pages is quite meaningful, especially in those areas. The problem in social bookmarking system, will not only be to recommend items, but also to recommend tags which the user will use on specific items. Tag recommenders can assist users with the tagging process by suggesting a set of tags that users are likely to use for a web resource. Personalized tag recommenders which take users' previous tagging behaviors into account when making suggestions usually have better performance compared with general tag recommenders. In short, the goal of a personalized tag recommender is to predict tags for each user specifically and effectively, given an item.

1.4 Contributions

In this dissertation, we discuss online user behavior prediction and its application in different contexts, such as social tagging prediction, friend recommendation and online advertising. We address many problems mentioned above and propose novel models to achieve state-of-the-art performance in a number of tasks in online media:

- In social tagging, we address the problem of tag prediction by proposing a probabilistic model for personalized tag prediction, which improves quality by over 30% compared to a leading algorithm on a public dataset.
- We investigate the dynamics of user interests in social tagging systems and propose a user-tag-specific temporal interests model for tracking user interests over time, which outperforms state-of-the-art tag prediction algorithms.
- In a micro-blogging system, by analyzing data collected over time, we find that 90% of new links are to people just two hops away and the dynamics of friend acquisition are also related to users' account age.
- We propose a novel structure-based personalized link prediction model for micro-blogging system, which outperforms fundamental and popular link prediction methods, including the state-of-the-art.
- In modeling multi-context in online media, we propose a generalized latent factor model and Bayesian treatment which handle the problems—coupled high order interaction, data sparsity and cold start on items, noticeably outperforming state-of-the-art approaches.

- In online advertising, to model user click behaviors, we propose a novel Context-Aware Click Model for sponsored search which outperforms the existing state-of-the-art.

- To estimate ad group performance in sponsored search, we design a novel framework that directly predicts group performance for lists of ads. Our methods noticeably outperform the existing state-of-the-art approaches.

We will discuss the contributions in detail:

At first, we study social tagging systems. We perform a time-sensitive sampling on an existing public dataset, resulting in a new scenario which is much closer to "real-world". We address the problem of tag prediction by proposing a probabilistic model for personalized tag prediction. The model is a Bayesian approach, and integrates three factors—an ego-centric effect, environmental effects and web page content. Two methods—both intuitive calculation and learning optimization—are provided for parameter estimation. Pure graph-based methods which may have significant constraints (such as every user, every item and every tag has to occur in at least p posts) cannot make a prediction in most real-world cases while our model improves the F-measure by over 30% compared to a leading algorithm on a publicly-available real-world dataset. Then we investigate the temporal dynamics of user interests in social tagging systems and propose a user-tag-specific temporal interests model for tracking user interests over time. Additionally, we analyze the phenomenon of topic switches in social bookmarking systems, showing that a temporal interests model can benefit from the integration of topic switch detection and that temporal characteristics of social tagging systems are different from traditional concept drift problems. We conduct experiments on three public datasets, demonstrating the importance of personalization and user-tag specialization in tagging systems. Experimental results show that our method can outperform state-of-the-art tag prediction algorithms. We also incorporate our model within existing content-based methods yielding significant improvements in performance.

The second contribution is about link analysis and prediction in microblogs. Unlike a traditional social network service, a microblogging network like Twitter is a hybrid network, combining aspects of both social networks and information networks. Understanding the structure of such hybrid networks and to predict new links are important for many tasks such as friend recommendation, community detection, and network growth models. By analyzing data collected over time, we find that 90% of new links are to people just two hops away and dynamics of friend acquisition are also related to users' account age. We compare two popular sampling methods which are widely used for network analysis and find that ForestFire, etc. does not preserve properties required for the link prediction task. We further compare recent and most popular methods and principles for link prediction and recommendation. We propose a novel structure-based personalized link prediction model and compare its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

In this dissertation, we also study higher-order multi-relational data in online social media. Most research in recommender systems focuses on modeling the interests as a function of two factors, such as predicting a rating given the user and the item. However, often there is one or more additional factors that have been ignored in the context of that recommendation. For instance, the activities of users commenting on items represent data with a third order coupling as it involves the user, the item, and the comment made by the user about this item. Moreover, the same factors (e.g., user, item, item-content, message) frequently appear in different contexts in social media. Our analyses demonstrate three principal challenges: coupled high order interaction, data sparsity and cold start on items. We tackle these problems by using a generalized latent factor model and Bayesian treatment and then test on three real-world data sets: Flickr, Bibsonomy and MovieLens. Our experiments on these data sets show that to achieve best predictive performance, one can employ a fully Bayesian treatment in modeling high order relations in social media. Our methods noticeably outperform state-of-the-art approaches.

Finally, we also discuss the application of user behavior prediction in online advertising. Sponsored search is the major business for today's commercial search engines. A critical problem in sponsored search is to understand and predict the browsing and click behaviors of users. In this dissertation, we analyze several factors influencing the CTR (click through rate) from the perspective of ad context, including the number of displayed ads, the content of the ads, the relationship between the query and ads, and the mutual influences between ads. Based on our data analysis, we propose a novel Context-Aware Click Model for sponsored search. We compare our Context-Aware Click Model with three strong baseline methods. The experiments show that our methods can outperform state-of-the-art methods. We then further challenge the current ad-delivery strategy (which is focused on predicting individual ad CTR) and study the problem—predicting the group performance (e.g., click yields) in sponsored search. To tackle all challenges in this problem—depth effects, interactive influence and cold start—we first investigate several effects and propose a novel framework that could directly predict group performance for lists of ads. To best leverage the text features and solve the sparseness issue in textual information, we embed a topic coding model into our framework to learn the topical information of short text for ads. Our extensive experiments on a large-scale real-world dataset from a commercial search engine show that we achieve significant improvement by solving the sponsored search problem from the new perspective. Our methods noticeably outperform the existing state-of-the-art approaches.

1.5 Organization

This dissertation is organized as follows.

In Chapter 2, we provide an overview of the social tagging system, and we address the problem of tag prediction by proposing a probabilistic model for personalized tag prediction. The model is a Bayesian approach, and integrates three factors—an ego-centric effect, environmental effects and web page content. We further investigate the temporal dynamics in social tagging system and analyze the phenomenon of topic switches in social bookmarking systems. Material in this chapter was published in three papers, presented at KDD 2010[159], WWW 2011[155] and AAAI 2011[158].

In Chapter 3, we study a hybrid network—microblogging system. We analyze the link formation in this microblogging network and compare two popular sampling methods. By analyzing data collected over time, we propose a novel structure-based personalized link prediction model and compare its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Material in this chapter is from two papers, presented at WWW 2011[157], CIKM 2011[156].

In Chapter 4, we investigate problems and challenges in predictive social media systems. Our analyses demonstrate three principal challenges: coupled high order interaction, data sparsity and cold start on items. We tackle these problems by using a generalized latent factor model and Bayesian treatment. Material in this chapter was a paper, presented at WSDM 2013[154].

In Chapter 5, we study problems in online advertising. We analyze several factors influencing the CTR from the perspective of ad context, and propose a novel Context-Aware Click Model for sponsored search. We further challenge the traditional strategy and study the problem predicting the ad group performance (e.g., click yields) in sponsored search.

Chapter 6 concludes the dissertation, summarizes what was learned, and discusses future directions.

Chapter 2

Personalized Tag Prediction in Social Tagging Services

In this chapter, we briefly introduce social tagging and study user behaviors in such systems. We first address the problem of tag prediction by proposing a probabilistic model for personalized tag prediction. The model is a Bayesian approach, and integrates three factors—an ego-centric effect, environmental effects and web page content. Two methods—both intuitive calculation and learning optimization—are provided for parameter estimation. We then investigate the temporal dynamics of user interests in tagging systems and propose a user-tag-specific temporal interests model for tracking user interests over time.

2.1 Introduction

Collaborative tagging systems have become increasingly popular for sharing and organizing web resources. In collaborative tagging systems, users add metadata in the form of descriptive terms, called tags, to describe web resources. Social tagging has already showed its value in many areas, such as query expansion [12], web search [10], personalized search [121, 149], web resource classification [162] and clustering [110]. A better understanding and prediction of tags on web pages is quite meaningful, especially in those areas.

Tag recommenders can assist users with the tagging process by suggesting a set of tags that users are likely to use for a web resource. Personalized tag recommenders which take a user's previous tagging behaviors into account when making suggestions usually have better performance compared with general tag recommenders. In short, the goal of a personalized tag recommender is to predict tags for each user specifically and effectively, given a web resource.

Personalized tag prediction has become a popular research topic. The two main directions for these systems are content-based approaches and graph-based approaches. Content-based methods, which usually model users' preferences from textual information (e.g., web pages, academic papers and tags), can predict tags for new users and new web resources. Graph-based approaches, while often having stronger assumptions than content-based ones, typically provide better performance. For example, one such assumption is the CORE p [64] requirement, in which every user, every item and every tag has to occur at least p times in the training set. However, in most cases, such an assumption is not realistic. Actually, tag recommenders are often asked to recommend tags when the system knows nothing about the web resource or the user. Our analysis will show that in a real world scenario, the probability of a web resource being new to a tag recommender is more than 0.9. Comparing both kinds of approaches, the content-based approach has the advantage that it can predict tags for any user and any web resource, while the overall performance is not as good as graph-based approach. Graph-based approaches which require CORE p can have significantly better performance, but can only predict tags for certain groups of users and web resources, preventing them from being widely applicable. Thus, a better tag recommender should be able to recommend tags for new users or new web resources, and still have reasonably good performance. Our tag recommender has such functionality, in part by incorporating various factors. We believe that in the real world, when a user is tagging web pages, at least the following three factors will affect the choice of tags which he will finally use.

The ego-centric effect. A given user will have some specific interests and will tend to bookmark similar items with similar tags based on the user's vocabulary. Assume for example that a user is interested in "development" and he has already tagged many web pages about development by using "C++", "java", and "tutorials". When he bookmarks a new web page, intuitively, this item will be relevant to "development" with high probability. That is, the content of this new web page is very likely to be similar to web pages that the user tagged previously. In addition, on this web page, the user will also tend to use similar tags which he used before. We name this effect, which is from the user himself, the ego-centric effect.

Environmental effects. A user may be influenced by other users. When a user is tagging some web page, he may adopt tags which are used frequently by other similar users. For instance, a user may often use the tag "java" previously, but never use the tag "tutorial". Suppose that there is another user who is similar to this user—say, they both frequently use tag "java"—but in addition frequently uses the tag "tutorial". In this case, when the user is trying to bookmark an item which is similar to the items where the other user already has tagged as "tutorial", the probability of this user using both "java" and "tutorial" is higher (even if this user never used the tag "tutorial" before). In addition, some users may discover resources within the tagging system; that is, they find interesting items which other users have already tagged. In this case, the probability of this user using the same tags will be very high (most graph-based recommenders adopt this strategy). Another aspect is that some current tagging systems allow users to set up relationships with other users, e.g., delicious [151, 137]. This also strengthens the influence of neighboring users. We call all of these kinds of effects environmental effects.

Item content. For an item which already exists in past bookmarks, we can get some prediction hints from the tags which have already been used on this item. However, we have found that tag prediction systems may need to face new items more than 90 percent of the time. Thus, strictly graph-based recommenders will not work on these new items. When facing a new item, content analysis is necessary. Even if the item is not a new item, content analysis is still quite useful, because other items with similar topics will provide hints for tag prediction.

Temporal factor. some, such as [56, 165], attempt to study the temporal characteristics of tagging systems in a global sense and suggest that the more frequently and recently used tags should be favored for tag suggestion, due to the fact that users may re-use tags in a short time frame and the scope of users' interests might change over time. This implies that it is not appropriate to make suggestions simply based on all past data, as most current methods do, and the possibility to advance the state-of-the-art approaches by carefully modeling temporal dynamics of individual user's interests. However, none of these is rigorously studied in the literature. In other forms of collaborative filtering and recommendation systems, the performance of recommenders has been greatly improved by incorporating temporal factors into the models. For instance, in the problem of movie recommendation, Koren [73] showed that users' inclinations are evolving over time and proposed a latent factor model with temporal dynamics which can better recommend items for users. Xiang et al. [146] also found that temporal information can contribute to improved recommendation in collaborative filtering by fusing the temporal factor into a random walk graph-based model.

In this chapter, we first propose a probabilistic model for tag prediction which integrates three factors—an ego-centric effect, environmental effects, web page content. Then we systematically investigate the temporal dynamics of user interests in social tagging systems and propose a novel approach for the tag prediction problem by modeling users' long-term and short-term preferences in a principled manner. More specifically, we associate each user and tag pair with a kernel function to characterize their temporal changes and show an effective estimation process to embed this idea into the probabilistic model.

In this chapter, our contributions are as follows:

- We perform time-sensitive sampling on an existing public dataset, and propose a new use case of tag prediction which is closer to real world cases.
- We present a novel probabilistic model for personalized tag prediction.
- We verify the existence of short-term interests through the exploration of simple models.
- Our experiments show that our methods which are only based on personal historical tagging sequences can outperform the state-of-the-art in the presence of concept drift, personalized tag prediction and content-only tag prediction.
- By combining methods, state-of-the-art algorithms can get significant improvements in predictive quality in the online tag prediction task.

The chapter is organized as follows: Section 2.2 precisely defines the problem and notation. Section 2.6 introduces a general model for capturing users' current interests. Section 2.4 reports our experiments. Section 2.8 concludes and outlines future work. In Section 2.9, we briefly review some recent work on tag recommendation, temporal analysis and related topics.

2.2 Definitions

In a social tagging system, users can bookmark web pages by assigning tags to them. The system can also retrieve the content of a web page which the user is bookmarking and based on content, the system can recommend to the user some personalized tags. The task of recommending tags to users is called tag recommendation.

A similar task is tag prediction which needs to predict the tags which the user will use on some bookmarks. This can also be personalized; that is, given a user and a set of bookmarks without tags, the algorithm should predict which tags the user will use on each bookmark. In order to predict or recommend tags for a specific user precisely, the recommender should first understand the user well. Because different users have different preferences and interests, for some users, the bookmarks the user saves may tend to be similar or in the same topic. In addition, on similar bookmarks, the tags which different users use may be similar. But for other users, even if they save the same or similar web pages, they may use different tags because of different perspectives and different preferences.

Here, we formalize the definitions. Let U be the set of all users, I be the set of all items (they sometimes are also called objects, resources, or web pages in other literature) and T be the set of all tags. For past tagging information, we have existing ternary relations S, and $S \subseteq U \times I \times T$. Thus, each single record $(u, i, t) \in S$ means that user u has tagged an item i with the tag t. Here, we also define P_s as all the past distinct user-item combination:

$$P_s = \{(u, i) | \exists t \in T : (u, i, t) \in S\}$$

Thus, when the current user u_c is trying to add an item i_c, the task is to recommend a list of tags to the potential post (u_c, i_c), based the past posts S, which we also call training data.

2.3 Probabilistic Model

The tag prediction problem can be treated as the reverse of web search. In web search, users submit a list of terms as a query, and then the relevant web pages i will be retrieved and the web pages can be ranked by $P(i|t)$, the probability of the page i being relevant to the query t. Here, the list of terms can be considered as a list of tags. Without considering personal information (non-personalized tag prediction), the general tag prediction could be that given a web page i, retrieve a list of potential tags. The tags can be ranked by $P(t|i)$. According to Bayesian theory, we have

$$P(t|i) = \frac{P(i|t) \cdot P(t)}{P(i)} \tag{2.1}$$

In Equation 2.1, $P(t|i)$ means the probability of using tag t given an item i. $P(i|t)$ means the frequency of item i in a set of items which are tagged by t. $P(t)$ is the prior probability of tag t. If the tag t appears more frequently, it will hold a higher prior probability. If the item i exists in past posts which can be considered as the training data, then $P(i|t)$ can be easily estimated by simply using the number of occurrence of (i, t)—$N_{i,t}$. However, if the item i does not exist in the past posts, that is, i is a new item, it is difficult to estimate the probability $P(i|t)$. One possible solution is to use the content of the item.

The content of item i can be represented by a language model. The most straightforward model is a unigram language model. The item i is treated as a bag of words $W = \{w|w \text{ appears in item } i\}$. Here, if the word-independence assumption is made, the probability of item i given the tag t will be:

$$P(i|t) = \prod_{w \in W_i} P(w|t) \tag{2.2}$$

According to Equation 2.2, we know that the probability $P(i|t)$ can be broken down into the production of word-level probabilities $\prod_{w \in W_i} P(w|t)$. $P(w|t)$ means the likelihood of the word w appearing in the item's content, given a tag t. Given a item i, the number of occurrence of w is denoted as $N_{w,i}$. Given a tag t, the number of occurrence of w is denoted as $N_{w,t}$ which can be calculated as follows:

$$N_{w,t} = \textstyle\sum_{i \in I} N_{w,i} \cdot N_{i,t}$$

To estimate $P(w|t)$, we can assume that words obey the following distribution:

$$P(w|t) = \tfrac{N_{w,t}}{N_t}$$

Then, maximum likelihood estimation (MLE) can be used to estimate the parameter N. To maximize the probability of the word w, we have:

$$N_t = \textstyle\sum_w N_{w,t}$$

By combining Equations 2.1 and 2.2, general tag prediction can be expressed as:

$$P(t|i) = \frac{\prod_{w \in W_i} P(w|t) \cdot P(t)}{P(i)} \tag{2.3}$$

2.3.1 Personalized Tag Prediction

While we have shown how to perform general tag prediction, personalized tag prediction is more preferable. In social tagging systems, individual users may have specific interests and tend to bookmark similar web pages by using similar tags. For different users, the prior probability of tags is often different, and the language model of tags is also different. Even if two users bookmark the same item, the tags they use can also be different because of their various interests, perspectives and preferences. Rendel et al. [112] show that personalized tag prediction systems empirically outperform the theoretical upper bound for any non-personalized tag recommender. In our probabilistic model, the general tag prediction can be simply extended to personalized prediction by involving the ego-centric effect. Given a user u, the personalized tag prediction can be:

$$P(t|i,u) = \frac{P(i|t,u) \cdot P(t|u)}{P(i|u)} \tag{2.4}$$

Here, $P(t|i,u)$ means that given a user u, the probability that tag t is applied to the item i. $P(i|t,u)$ means the likelihood of item i given a tag t and user u. $P(t|u)$ is also the conditional prior probability of tag t, given the user u. It can be easily understood that Equation 2.4 is based on a set of past posts S—that is, for the specific user u_c, $S_{u_c} = \{(u_c, i, t) | (u_c, i, t) \in S\}$. Similar to non-personalized tag prediction, to incorporate the content of items, replacing $P(i|t,u)$ in Equation 2.4, the personalized tag prediction will become:

$$P(t|i,u) = \frac{\prod_{w \in W_i} P(w|t,u) \cdot P(t|u)}{P(i|u)} \tag{2.5}$$

However, in this model, if a user has not yet used a tag, we cannot rank it. Because if tag t has not been used by user u, the prior probability $P(t|u) = 0$, and we cannot get the $P(i|t,u)$ either. According to Equation 2.5, for this new tag t, the $P(t|d,u)$ will be always 0. Thus, the candidate tags will be constrained to the set of tags which the user has used before. Obviously, such candidate tags are often quite limited.

In addition, when users are trying to bookmark some web pages, the three factors mentioned previously will affect the tags which the user will finally use: the ego-centric effect, environmental effects and item content. In Equation 2.5, the ego-centric effect is modeled by the whole equation and item content is modeled by $\prod_{w \in W_i} P(w|t,u)$. To model environmental effects, we involve the probability of neighbor effects $P(u_k|u)$, that is, given the current user u, the probability of user u_k affecting user u. When $u_k = u$, then $P(u|u)$ represents the exact weight of the ego-centric effect. Thus,

$$\sum_{u_k} P(u_k|u) = 1 \tag{2.6}$$

When we integrate the environmental effects into Equation 2.5, we get

$$\begin{aligned} &P(t|i,u) = \\ &\frac{\prod_{w \in W_i} \sum_{u_k} P(w|t,u_k)P(u_k|u) \times \sum_{u_k} P(t|u_k)P(u_k|u)}{\sum_{u_k} P(i|u_k)P(u_k|u)} \end{aligned} \tag{2.7}$$

This equation enlarges the tag candidates for tag prediction and also integrates the environmental effects. Given a user u and an item i, the probability of tag t being used will be $P(t|i,u)$. Our algorithm will rank the tags by the value $P(t|i,u)$. Because the evidence $P(i|u)$ is the same for all tag candidates, then

$$\begin{aligned} &P(t|i,u) \\ &\propto \prod_{w \in W_i} \sum_{u_k} P(w|t,u_k)P(u_k|u) \times \sum_{u_k} P(t|u_k)P(u_k|u) \\ &\propto \sum_{w \in W_i} \log \sum_{u_k} P(w|t,u_k)P(u_k|u) + \log \sum_{u_k} P(t|u_k)P(u_k|u) \end{aligned} \tag{2.8}$$

We refer to Equation 2.8 as,

$$\begin{aligned} y_{u,i,t} = &\sum_{w \in W_i} \log \sum_{u_k} P(w|t,u_k)P(u_k|u) \\ &+ \log \sum_{u_k} P(t|u_k)P(u_k|u) \end{aligned} \tag{2.9}$$

Then, given a user u and an item i, our algorithm will rank the tags by the value $y_{u,i,t}$. If we define the probability $P(u|u)$ as α or $p_{u,u}$ and environmental effects $P(u_k|u)$ as $p_{u_k,u}$, then split the ego-centric effect part and environmental effects part and Equation 2.9 can be rewritten as

$$\begin{aligned} y_{u,i,t} = &\sum_{w \in W_i} \log(\sum_{u_k \neq u} p_{u_k,u} \cdot P(w|t,u_k) + \alpha \cdot P(w|t,u)) \\ &+ \log(\sum_{u_k \neq u} p_{u_k,u} \cdot P(t|u_k) + \alpha \cdot P(t|u)) \end{aligned} \tag{2.10}$$

where $\sum_{u_k \neq u} p_{u_k,u} + \alpha = 1$ and $\sum_{u_k \neq u} p_{u_k,u}$ can be also called the weight of environmental effects and α can be called the weight of ego-centric effect or ego weight. To avoid zero probability, for $P(w|t, u_k)$, we use simple Laplace smoothing in our experiments.

2.3.2 Parameter Estimation and Optimization

In our model, Equation 2.9, we have already introduced the unigram language model for $P(w|t, u_k)$. Another $P(t|u_k)$ can be calculated through the number of occurrence of tag t within the posts of user u_k. The hard problem is to estimate the ego-centric effect and environmental effects $P(u_k|u)$.

Intuitively calculating $P(u_k|u)$

Given a user u, to calculate the probability of another user's influence—$P(u_k|u)$, we consider that users can be represented in tag space. In the set of past posts S, each user has a set of tags, which describes the interests of the user. In other words, each user has a distribution of tags. The vector of tag occurrences can be used to represent to the user. For the user u_k,

$$V_{u_k} = [n_{u_k,t_0}, n_{u_k,t_1}, ...n_{u_k,t_i}, ...n_{u_k,t_m}]$$

Here, n_{u_k,t_i} means the number of times of user u_k uses tag t_i. For the user u_k, the prior probability of the tag t_j can be calculated by

$$P(t_j|u_k) = \frac{n_{u_k,t_j}}{\sum_{i=0}^{m} n_{u_k,t_i}}$$

If two users have similar interests, then they may have similar distributions of tags and there will be a higher probability of affecting each other. Here, for user u, if we assume that the similarity of interests between user u and user u_k is directly proportional to the probability of u_k affecting u—$P(u_k|u)$, then

$$P(u_k|u) \propto sim(u, u_k) = \frac{V_u \cdot V_{u_k}}{|V_u| \times |V_{u_k}|}$$

Where $sim(u, u) = 1$, the ego weight will be always larger than the weight of other individual users. After normalizing similarity, we can simply set the

$$P(u_k|u) = \frac{sim(u, u_k)}{\sum_{u_i} sim(u, u_i)} \tag{2.11}$$

We refer to this calculation as user-tag-user similarity. We can also manually cut off users by using a threshold. For experiments, the number of neighbor users can be set as runtime parameter k. Only the most similar k neighbor users will be counted.

Learning algorithm

Our intuitive estimation is only a rough method of estimating $P(u_k|u)$. In some cases, it may not be precise. For example, user u_k may use the same tags that user u used on item i, to tag another item i_k, while the content i and i_k are completely different. Thus, different users may use the same tags with different intents or perspectives for tagging web pages. Our previous method will over-estimate the probability $P(u_k|u)$ in this situation. To solve this problem, we design a learning algorithm to calculate $P(u_k|u)$ iteratively. For a post (u, i), the algorithm ranks tags by $y_{u,i,t}$. We use the similar objective function as in [112], which uses the "post-based ranking interpretation" and maximizes the ranking statistic AUC (area under the ROC-curve).

$$\begin{aligned} &AUC(\hat{\theta}, u, i) = \\ &\frac{1}{|T^+_{u,i}||T^-_{u,i}|} \sum_{t^+ \in T^+_{u,i}} \sum_{t^- \in T^-_{u,i}} H_{0.5}(y_{u,i,t^+} - y_{u,i,t^-}) \end{aligned} \tag{2.12}$$

where

$$H_\beta = \begin{cases} 0, & x < 0 \\ \beta, & x = 0 \\ 1, & x > 1 \end{cases} \tag{2.13}$$

$T^+_{u,i}$ is the set of tags which the user u adds on the item i while $T^-_{u,i}$ is the set of tags which the user u does not add on the item i. The overall optimization task with respect to the ranking statistic AUC and the observed data is then:

$$\arg\max_{\hat{\theta}} \sum_{(u,i)\in P_s} AUC(\hat{\theta}, u, i) \tag{2.14}$$

Then, we use the continuous sigmoid function to replace H:

$$s(x) = \frac{1}{1+e^{-x}} \tag{2.15}$$

Then using gradient descent, AUC has to be differentiated with respect to all model parameters and for each post $(u,i) \in P_s$ the model parameters $P(u_k|u)$ are updated and renormalized.

$$\begin{aligned} &\frac{\partial}{\partial p_{u_k,u}} AUC(\hat{\theta}, u, i) \\ &= \frac{\partial}{\partial p_{u_k,u}} \frac{1}{|T^+_{u,i}||T^-_{u,i}|} \sum_{t^+\in T^+_{u,i}} \sum_{t^-\in T^-_{u,i}} s(y_{u,i,t^+} - y_{u,i,t^-}) \\ &= z \sum_{t^+\in T^+_{u,i}} \sum_{t^-\in T^-_{u,i}} w_{t^+,t^-} \frac{\partial}{\partial p_{u_k,u}} (y_{u,i,t^+} - y_{u,i,t^-}) \end{aligned}$$

with:

$$\begin{aligned}
w_{t^+,t^-} &= s(y_{u,i,t^+} - y_{u,i,t^-})(1 - s(y_{u,i,t^+} - y_{u,i,t^-})) \\
z &= |T^+_{u,i}||T^-_{u,i}| \\
y_{u,i,t^+} - y_{u,i,t^-} &= (\sum_{w \in W_i} \log \sum_{u_k} P(w|t^+, u_k)P(u_k|u) \\
&\quad + \log \sum_{u_k} P(t^+|u_k)P(u_k|u)) \\
&\quad -(\sum_{w \in W_i} \log \sum_{u_k} P(w|t^-, u_k)P(u_k|u) \\
&\quad + \log \sum_{u_k} P(t^-|u_k)P(u_k|u))
\end{aligned}$$

And

$$\frac{\partial}{\partial p_{u_k,u}}(y_{u,i,t^+} - y_{u,i,t^-}) =$$

$$\sum_{w \in W_i} \frac{P(w|t^+, u_k)}{\sum_{u_k} P(t^+|u_k)P(u_k|u))} + \frac{P(t^+|u_k)}{\sum_{u_k} P(t^+|u_k)P(u_k|u)}$$

$$- \sum_{w \in W_i} \frac{P(w|t^-, u_k)}{\sum_{u_k} P(t^-|u_k)P(u_k|u))} - \frac{P(t^-|u_k)}{\sum_{u_k} P(t^-|u_k)P(u_k|u)}$$

Then, the derivation of $p_{u_k,u}$ is

$$\frac{\partial AUC}{\partial p_{u_k,u}} = z \sum_{t^+ \in T^+_{u,i}} \sum_{t^- \in T^-_{u,i}} w_{t^+,t^-} Y_{t^+,t^-}$$

where

$$\begin{aligned}
w_{t^+,t^-} &= s(y_{u,i,t^+} - y_{u,i,t^-})(1 - s(y_{u,i,t^+} - y_{u,i,t^-})) \\
z &= |T^+_{u,i}||T^-_{u,i}| \\
Y_{t^+,t^-} &= \frac{\partial}{\partial p_{u_k,u}}(y_{u,i,t^+} - y_{u,i,t^-})
\end{aligned}$$

Thus, for each post $(u, i) \in P_s$ the model parameters $P(u_k|u)$ are updated as follow.

$$\hat{p}_{u_k,u} \leftarrow \frac{\hat{p}_{u_k,u} + \gamma \cdot \frac{\partial AUC}{\partial p_{u_k,u}}}{\eta}$$

where η is a normalization factor $\eta = \sum_{u_j}(\hat{p}_{u_j,u} + \gamma \cdot \frac{\partial AUC}{\partial p_{u_j,u}})$ and γ is a learn rate.

2.3.3 Processing New Users

Our model is designed for personalized tag prediction, especially for existing users. However, in the real world, we still may face users who have not been seen by the tagging system previously. A simple method to predict tags for new users is to just use the general model Equation 2.3.

A better option is that instead of using the general model, we can build a language model for the new user u_{new} from the item i. Given a new user and an item (u_{new}, i), even if we do not know the past information of the user, we can still get some implication from the content of item i. For existing users, a similar language model is extracted from the items which the users tagged previously. Then the language models are used to represent users' interests. For user u_k,

$$W_{u_k} = [n_{u_k,w_0}, n_{u_k,w_1}, ...n_{u_k,w_i}, ...n_{u_k,w_m}]$$

Here, n_{u_k,w_i} means the number of times of user u_k has used the word w_i. Similarly,

$$P(u_k|u_{new}) \propto sim(u_{new}, u_k) = \frac{W_{u_{new}} \cdot W_{u_k}}{|W_{u_{new}}| \times |W_{u_k}|}$$

Where $sim(u_{new}, u_{new}) = 0$, for new users, there will be no ego effect. All the information should be from environmental effects and item content. After normalizing similarity, we can simply set the

$$P(u_k|u_{new}) = \frac{sim(u_{new}, u_k)}{\sum_{u_i} sim(u_{new}, u_i)} \tag{2.16}$$

We refer to this calculation as user-lan-user similarity. For new users, we cannot use learning algorithm to refine $P(u_k|u_{new})$.

Table 2.1: Offline Statistics

Training Data		Test Data	
Total Posts	262,336	Total Posts	668
Total Records	914,162	Total Records	2,307
Total Users	2,677	New/Total Users	2/169
Total Items	234,764	New/Total Items	564/668
Total Tags	56,370	New/Total Tags	54/1,224

2.4 Experiments

In this section, we describe the details of datasets and experiments. We also compare our approach with two other algorithms.

2.4.1 Dataset

In our experiments, we use the bookmark dataset of the ECML PKDD 09 Challenge Workshop[1]. The dataset S includes 2,679 users, 263,004 items, 56,424 tags, 262,336 posts and 1,401,104 records. All of the posts contain timestamps. We uniformly sample 668 posts along the time line as our test dataset S_{test} and the remaining posts constitute the training dataset S_{train}.

In Figure 2.1 we show the tag and item frequencies over the full dataset. In the plot on the right, the large vertical gap between the two leftmost points means that 93.6% items appear only once and only 6.4% of items appear more than once. Thus, most graph-based methods which require more than CORE-2 (users, tags and items appear at least twice) cannot work on it. For tags, 49.4% of tags appear only once; 50.6% of tags appear more than once.

In comparison, if we ignore time information and assume a traditional fixed training and test split (e.g., use the dataset as an "offline" dataset), a test post may

[1] http://www.kde.cs.uni-kassel.de/ws/dc09/

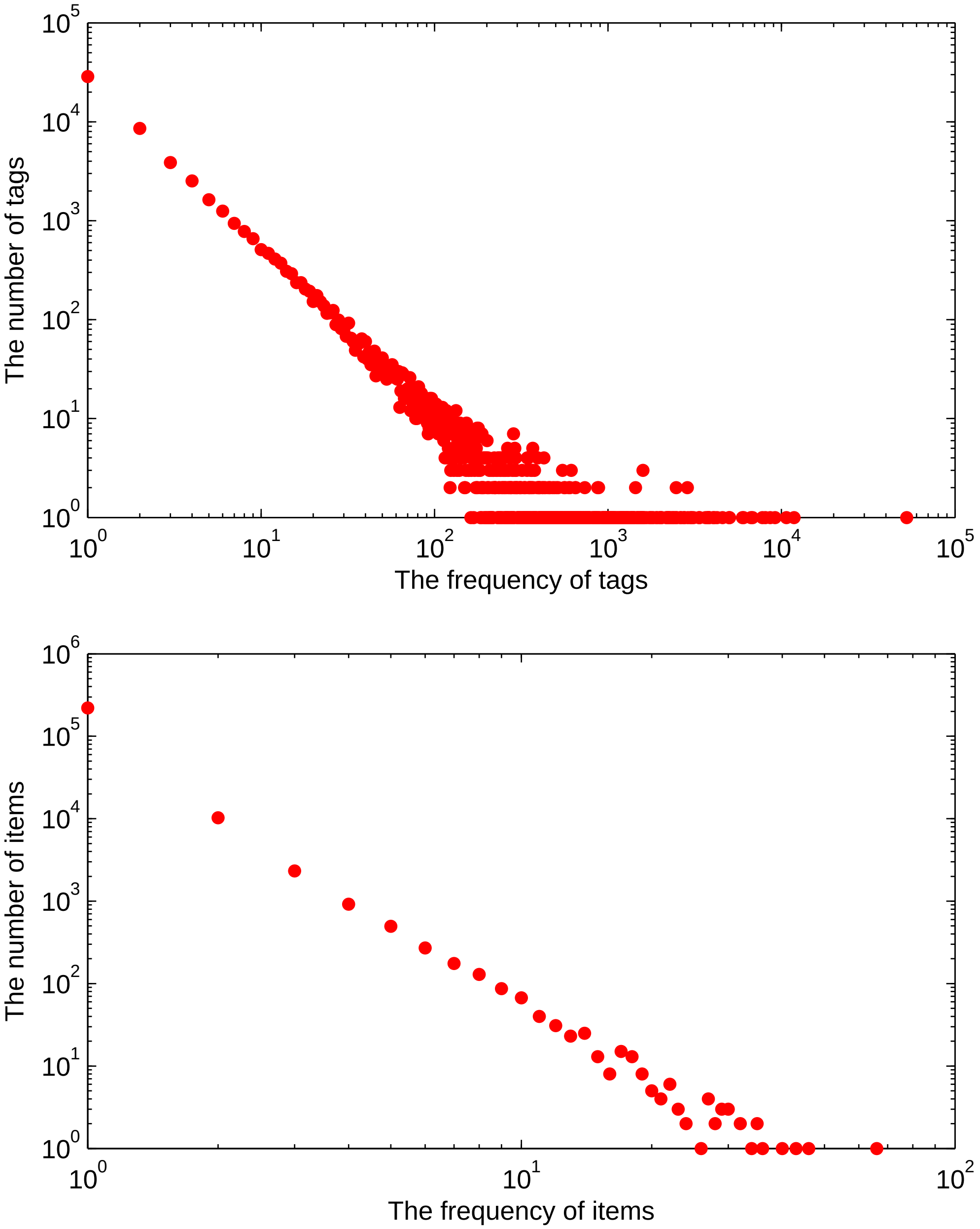


Figure 2.1: Dataset Statistics.

Table 2.2: Online Statistics

Total Posts	668
Total Records	2307
Old/Total Users	627/668
New/Total Users	41/668
Old/Total Items	66/668
New/Total Items	602/668
Old/Total Tags	1,986/2,307
New/Total Tags	321/2,307

have occurred prior to some training posts, effectively using the future to build a model to predict the past. Table 2.1 provides statistics regarding the training data and the number of "new" items seen in the test data. We find that there are only 2 new users out of 169 users and 54 new tags out of 1,224 tags in the test dataset. However, there are 564 new items out of 668 items even in the offline statistics. Here, "new" means that it does not exist in training data. While the offline analysis can give us some impression of the dataset, it is different from the real world, because in the real world, we cannot use future data as training data to recommend tags for users.

2.4.2 Online Evaluation

Besides the offline test, another testing method which is often used in tag prediction evaluation is that of fixing a time point—posts whose timestamp is earlier than that time will be used as training data while posts whose timestamps are later than that time will be used as test data. The ECML PKDD Challenge Workshop employed this approach. There are still some problems for this method. For example, if a user never tagged items before that time point and then tagged M posts after that time point, in this test mode, the all M posts of this user will be treated as the posts of a new user to training data. Thus, there will be too many "new user cases" which in

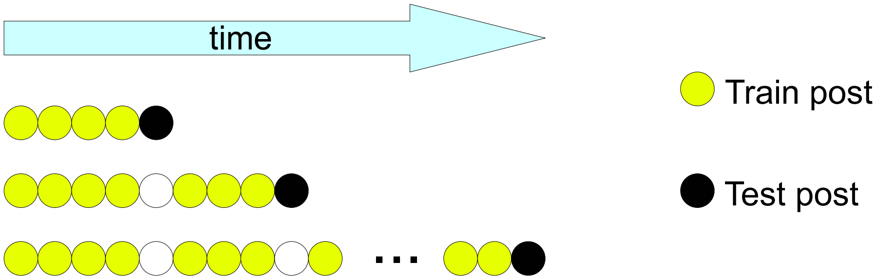


Figure 2.2: Online framework.

the real world is actually existing users. In the real world, after the user tagged his first item, the system should know this user and be able to retrieve the list of tags which this user has previously used. In addition, for users who tagged items both before and after the time point, their interests may not always stay the same and may even change frequently; in the real world, the system can again retrieve the latest tags which can represent the latest interests of this user. Such information should also be considered to make better prediction of tags.

We introduce a better evaluation method which is much closer to the real world and call it the "online" framework in this chapter. Figure 2.2 illustrates the online framework. Like online machine learning [16] which has been used widely, in our online mode[2], the tagging system operates in an incremental mode and the test posts are randomly sampled from the whole dataset along the timeline. In other words, for users and items in our test dataset, we only use the training posts which have earlier timestamps than the test posts, and the available training data is different

[2] In this chapter, online mode means a incremental mode of real tagging system rather than real-time tag prediction.

for each test post. Under this setting, for items tagged early in time, fewer training data is available. The online statistics (shown in Table 2.2) demonstrate that we still face a new user (a user which is not in the training set) in 6.1% of the cases, and in 90.1% of the time users are trying to bookmark a new item. In addition, there is .139 probability that users would use new tags (which do not appear in the system before). Thus, in the real world, the principal difficulty is to process cases in which existing users which try to tag new items. Overall, this online mode is more like a real tag prediction system, permitting the system to learn user behaviors incrementally rather than existing evaluation procedures with a fixed dataset split.

To evaluate performance of predictive model, we use the common evaluation scheme of F-measure in Top N lists, where $N = 5$ is mainly used as our measurement. The precision, recall and F-measure are calculated as follows.

$$\begin{aligned}
\text{Prec}(S_{test}, N) &= \underset{(u,i)\in P_{S_{test}}}{\text{avg}} \frac{|\text{Top}(u,i,N) \cap \{t|(u,i,t) \in S_{test}\}|}{N} \\
\text{Rec}(S_{test}, N) &= \underset{(u,i)\in P_{S_{test}}}{\text{avg}} \frac{|\text{Top}(u,i,N) \cap \{t|(u,i,t) \in S_{test}\}|}{|\{t|(u,i,t) \in S_{test}\}|} \\
\text{F}_1(S_{test}, N) &= \frac{2 \times \text{Prec}(S_{test}, N) \times \text{Rec}(S_{test}, N)}{\text{Prec}(S_{test}, N) + \text{Rec}(S_{test}, N)}
\end{aligned}$$

2.4.3 Comparison

From our analysis, in the real world, the graph-based method cannot work on most posts. Most graph-based algorithms require that users, tags and items appear at least twice in training set. We compare our approach with Liczak's method [84], which took the first place in the "content-based" recommendation task, and took third place in "graph-based" recommendation task in ECML PKDD Discovery Challenge[39]. They have two versions respectively for the "content-based" task and the "graph-based" task. In this chapter, we call their "content-based" version *LHKM-C* and their "graph-based" version *LHKM-G*, corresponding to the authors'

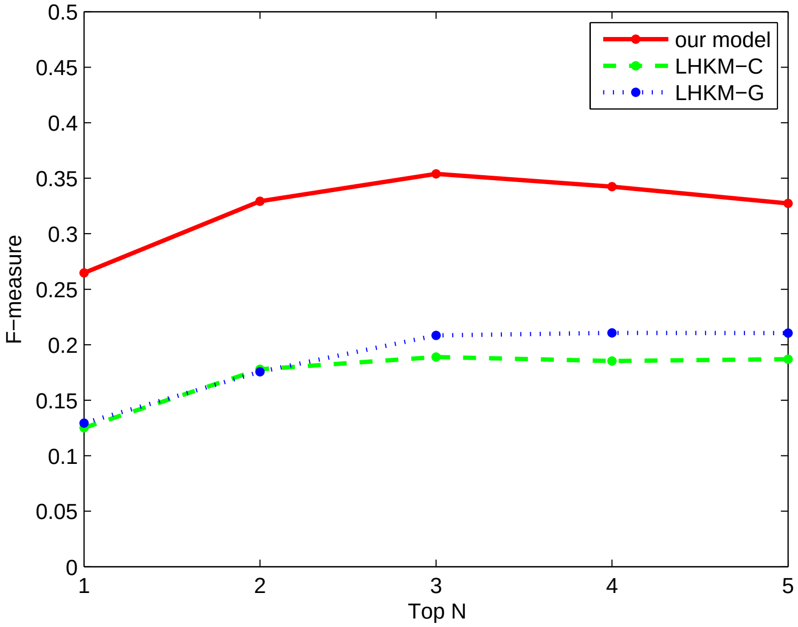


Figure 2.3: Comparison with offline.

initials of [84]. For LHKM-C and LHKM-G, we use the same parameters as they used in the Challenge Workshop. For our model, we only use the most similar 30 neighbors for each test user. In the $P(u_k|u)$ part, we use user-tag-user similarity mode to estimate ego-centric effect and environmental effects for existing users and user-lan-user similarity mode for new users.

In Figure 2.3 and 2.4, we show the comparison between online and offline tests. For each we also show the difference between performances whens recommending various number of tags (known as Top N). We see that as expected, the results of the offline test are always better than the results of the online test, because in the offline test, more training data (even future data) can be used. The results of LHKM-G are slightly better than the results of the LHKM-C, because in LHKM-G, "graph-information" is used. Our method outperforms both LHKM-C and LHKM-G. In offline test, the F-measure of our model is around 11% higher than LHKM-G

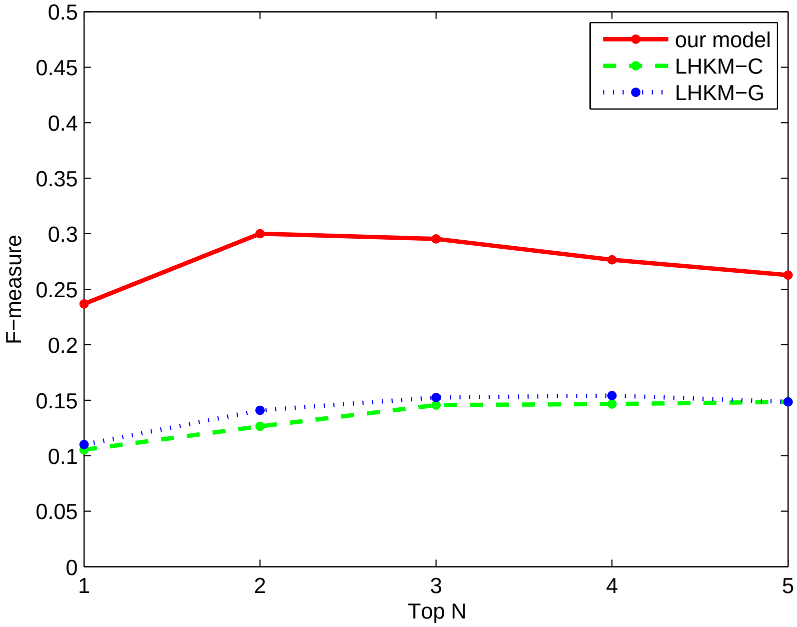


Figure 2.4: Comparison with online.

and 14% higher than LHKM-C. In online test, the F-measure of our model is 12% higher than LHKM-C and LHKM-G. In the following experiments, all the evaluation of F-measure in Top N lists will be based on $N = 5$.

2.4.4 Optimization Analysis

In this section, we use gradient descent to optimize parameters which can more accurately represent the environmental effects and ego-centric effects. We run the learning algorithm on the offline test. In our optimization, although it shows some improvement on the results, it is very time-consuming. For each user, we only use 10 training items to optimize the environmental effects of 30 neighbors and the ego-centric effect—α. The learning rate is set to 1.

There are two versions: the first is opt-Alpha which only tries to optimize α, the

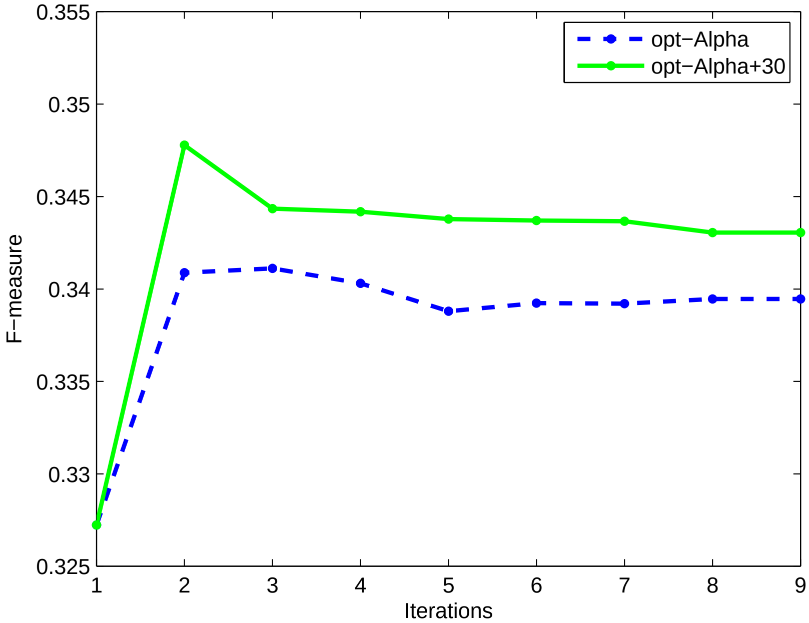


Figure 2.5: Results on iterative optimization.

second is opt-Alpha+30N which tries to optimize α and all 30 neighbors; that means, in total 31 parameters will be optimized. The initial values of $P(u_k|u)$ are the same as the section 2.4.3, using user-tag-user similarity for old users and user-lan-user similarity for new users.

Figure 2.5 shows the results of the iterative learning algorithm. The x-axis is the number of iterations and y-axis is F-measure. As expected, both optimization methods can improve the results of initial value a little (2-3%) and opt-Alpha+30N always outperforms opt-Alpha. This is because in opt-Alpha+30N, 31 parameters can be optimized while in opt-Alpha, only Alpha is optimized. From Figure 2.5, we also notice that after 1 or 2 times iteration, both opt-Alpha+30N and opt-Alpha get the best results and then the F-measure decreases slightly and converges. We hypothesize that this situation may be caused by overfitting. Another possible reason is that the learning procedure is time consuming, and we only use 10 items to optimize the parameters. Some users tagged thousands of items, so 10 training items may not be sufficient. In addition, better objective function and optimization methods are necessary for further improvement on both F-measure and running time.

2.4.5 Parameter Analysis

Compared to individual neighbors, the user's ego weight α should be the most important part. It decides the ego weight and relative impact of environmental effects. We consider that usually user's ego weight should be very high. Also the number of neighbors may affect the results of our model.

We find that the optimization process always generates higher α. In this experiment, we fix the number of neighbors to 100 and tune the ego weight alpha, from 0 to 1. The weights of neighbors will be normalized as follow.

$$p_{u,u_j} \leftarrow (1-\alpha) \cdot \frac{p_{u,u_j}}{\sum_{u_j \neq u} p_{u,u_j}} \tag{2.17}$$

Ego-centric effect analysis

For $P(u_k|u)$, we use user-tag-user similarity for existing users and user-lan-user similarity for new users. We use the most similar 100 users as environmental effects. Figure 2.6 shows the results. In this figure, the straight lines are from LHKM-C and LHKM-G for comparison. (In the online test, the F-measure of LHKM-C and LHKM-G is quite similar, so they only show a single line in the figure.) Our results on the offline tests and on the online tests are highly consistent. When $\alpha = 0$, that means, all information is from the most 100 similar neighbors, the F-measure is still slightly better the LHKM-C and LHKM-G on online test, but slightly worse on offline test. When α is set to 0.05, F-measure dramatically increases, and become

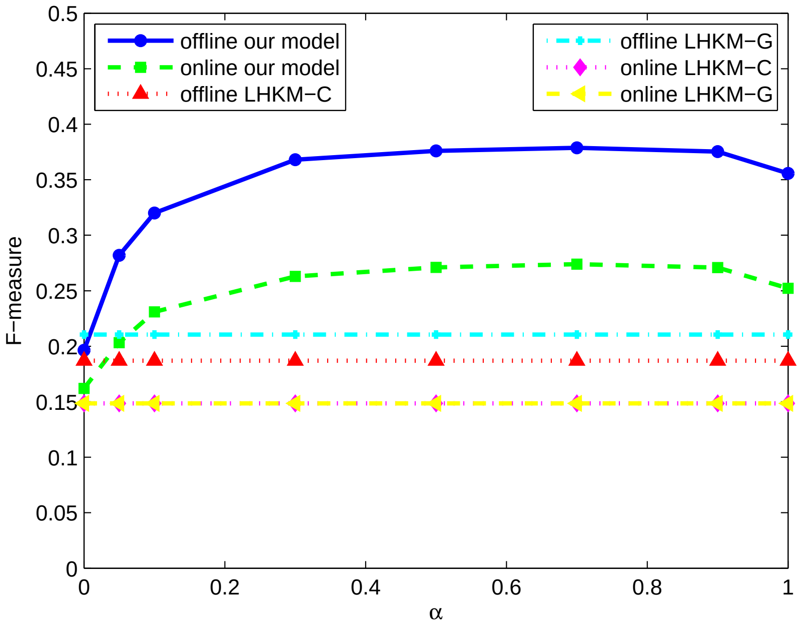


Figure 2.6: Ego-centric effect analysis.

higher than that of LHKM-C and LHKM-G in offline test. As α increases, the F-measure increases and achieves the best result when the α is set to 0.7. In offline test, it is around 37.8% (16% higher than LHKM-G) and in online test, it is 27.3% (12% higher than LHKM-G). Another interesting point is that even if α is set to 1, the performance of our model is still much better than LHKM-C and LHKM-G. In online test, regardless of how α is set, our model always outperforms Liczak's methods. These results verify our conjecture that the users' ego weight should be very important in tag prediction.

Environmental effects analysis

Then, we fix $\alpha = 0.5$, and tune the number of effective neighbors from 0 to unlimited—that is, we use all possible users and in our model, for existing users, as long as user-tag-user similarity is non-zero, then this user will be treated as an

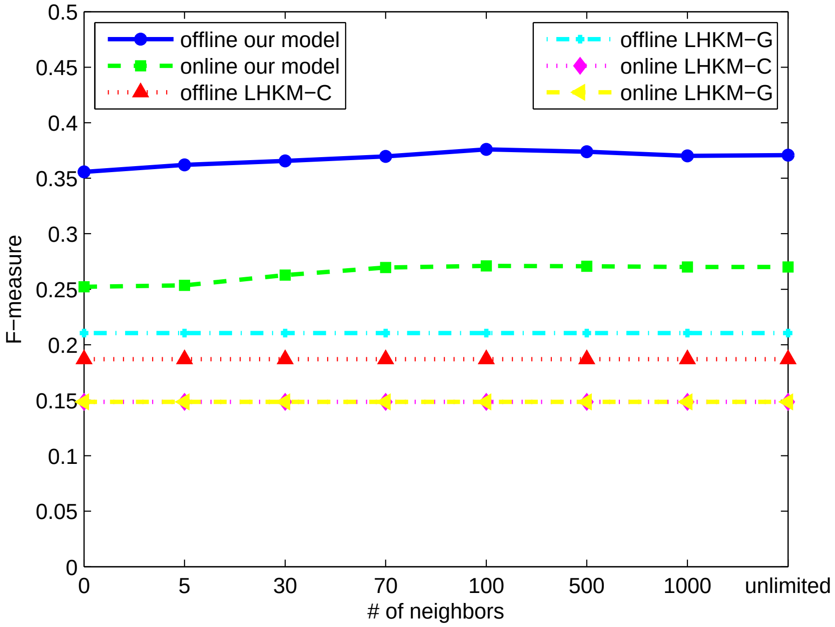


Figure 2.7: Environmental effects analysis.

effective neighbor. The results are showed in Figure 2.7. The straight lines are also from LHKM-C and LHKM-G. From Figure 2.7, in the beginning, as the number of neighbors increase, the F-measure increase. When the number of neighbors is set to 100, our model achieves the best F-measure on both offline test and online test, which are 37.5% and 27.1% respectively and also much better than LHKM-C and LHKM-G. We also notice that compared to α, the number of neighbors affects the results less. Thus, the number of neighbors is less important than the ego weight α and it can be simply set to 100 to get the best performance.

Online experiment

Based on the manually tuned α, we also try to optimize the ego weight to get the highest F-measure on the online test for real world performance. In this case, the manually tuned α and $P(u_k|u)$ will be used as initial values for the learning

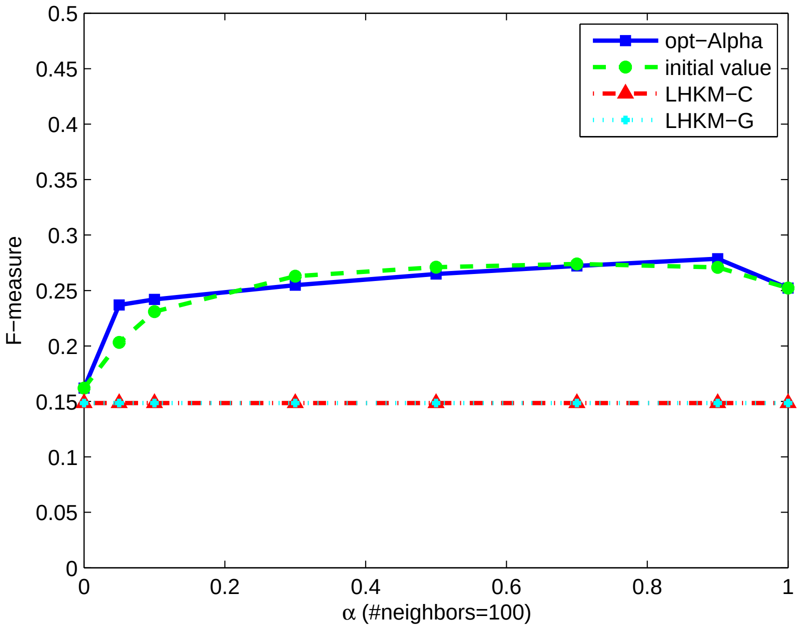


Figure 2.8: Online Experiments.

algorithm. For each test user, we still use 10 training items to optimize the ego weight. The learning rate is 1. The results are showed in Figure 2.8 opt-Alpha is the optimization version while the "initial value" is the same as the one in 2.4.5. The straight lines are also from LHKM-C and LHKM-G. From Figure 2.8 we can see that at some points, e.g., $\alpha = 0.05$, 0.1 and 0.9, opt-Alpha improves the F-measure and there are also some points where the performance of opt-Alpha and initial value are similar. Here, we also get the highest F-measure 27.9% on the online test which is 13% higher than LHKM-C (an improvement of more than 85%). Comparing to the results of learning algorithm, the results of manually tuned α are good enough and it runs much faster. At this moment, we suggest to manually tune α.

Table 2.3: 5-Fold Cross Validation

	LHKM-C	LHKM-G	our model
Test 1	0.193	0.202	0.348
Test 2	0.194	0.213	0.348
Test 3	0.193	0.210	0.347
Test 4	0.194	0.211	0.347
Test 5	0.195	0.211	0.348
mean	0.1938	0.2094	0.3476

2.4.6 Five-Fold Cross Validation

Because our test set is relatively small, in order to show the robustness of our model, k-fold cross validation was used to compare the performance of our model vs. LHKM-C and LHKM-G. In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained for testing the model and the remaining $k-1$ subsamples are used as training data. The cross-validation process is applied a total of k times (the folds), with each of the k subsamples used exactly once as the test data. In our experiment, $k = 5$ and we do offline testing. The number of neighbors is set to unlimited and alpha is set to 0.5. The parameters of LHKM-G and LHKM-C are the same as previous experiments. For each part of test and training data, the training data contains around 210,000 posts, 2,400 users, 190,000 items and 50,000 tags. and test data contains around 52,000 posts, 1,600 users, 50,000 items and 24,000 tags. Among them there are around 300 new users, 45,000 new items and 63,00 new tags. This is also consistent with our small test set.

In Table 2.5, we can see that our model outperforms the LHKM-C and LHKM-G by more than 10% on F-measure. The 5 results are quite similar and this also demonstrates that our model can generate better results than LHKM-C and LHKM-G stably.

2.5 Temporal Factors Analysis

In this section, we systematically investigate the temporal dynamics of user interests in social tagging systems and propose a novel approach for the tag prediction problem by modeling temporal preferences in a principled manner. Our method stands on techniques introduced to address "concept drift" [77], which imposes a continuous smoothing scheme over the timeline. However, we show that this smoothing scheme may lead to sub-optimal predictions due to the phenomenon that users may suddenly change interests and topics while using social bookmarking systems, as we suggest in Yin et al. [155]. We tackle the problem by explicitly modeling session-like behaviors and incorporate such models into our prediction process.

2.5.1 Data Sets

We use three public datasets. The first is the Bibsonomy dataset of the ECML PKDD 09 Challenge Workshop[3] which includes item content. The remaining two datasets are Delicious and Flickr datasets crawled by Gorlitz et al. [49][4]. There is no item content in the Delicious and Flickr datasets while all three contain timestamps. In order to observe the versatility of user interests on three datasets, for each user, we calculate and plot the total number of tags, and the total number of posts. In Figure 2.9, we can see that the three datasets have different properties and users form three clusters. In Bibsonomy, users typically apply a larger variety of tags across fewer posts, suggesting that their interests are more varied. In contrast, the users in Flickr use fewer tags and their interests are more focused, by reusing their tags many times. This implies that it may be easier to track the user interests in

[3] http://www.kde.cs.uni-kassel.de/ws/dc09/

[4] https://www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/
Research/DataSets/PINTSExperimentsDataSets/

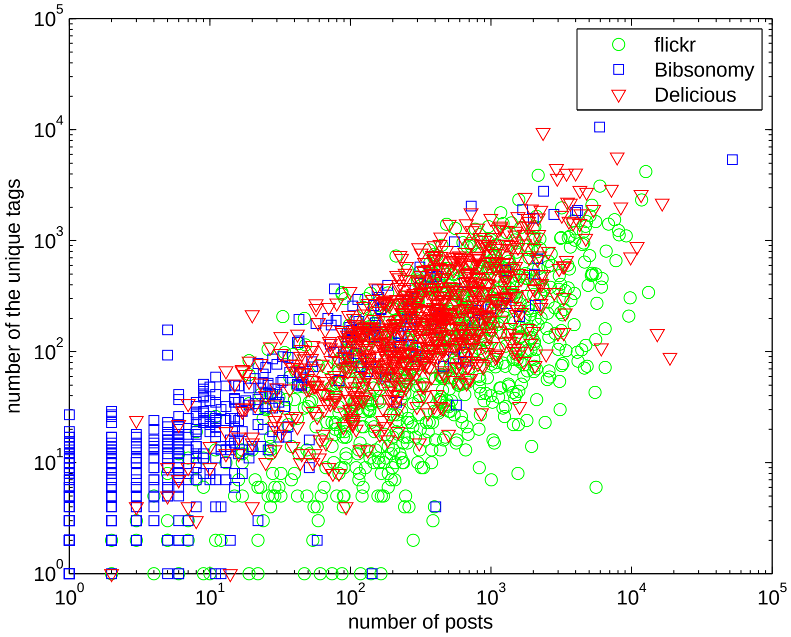


Figure 2.9: The number of tags against the number of posts.

Flickr.

2.5.2 Time-Sensitive Sampling

As in Section 2.4, we employ online evaluation, in which only training posts which have earlier timestamps than those of the test posts are used. Note that this implies that the available training data is different for each test post and for items tagged earlier in the timeline, fewer training data are available. While the online evaluation approach naturally fits the real-world case in which every post is used for testing a model trained on all prior posts, its feasibility depends highly on the efficiency of the training method as a new model may be necessary for each post. Instead, we can estimate the performance of the complete system by performing evaluation on only a sample of test posts, and largely avoid model-building efficiency concerns.

Table 2.4: Fractions of new users, items, or tags

	Bibsonomy	Delicious	Flickr
New/Total Users	41/668	16/1000	23/1000
New/Total Items	602/668	712/1000	1000/1000
New/Total Tags	321/2207	181/2920	175/4123

Similarly as in Section 2.4, we also use the common F_1-measure as our principal metric.

Here, we utilize the online evaluation model and conduct time-sensitive sampling experiments (which are similar to the experiments in Section 2.4) on three data sets. For the Bibsonomy dataset, we use the same sampling dataset as in Yin et al. [159] which includes 668 test posts. For Delicious and Flickr, we randomly choose 1000 posts. In all cases we effectively simulate a system running—the tagging system operates in an incremental mode. The Bibsonomy data set statistics (shown in Table 2.4) stay the same as Section 2.4. The other two datasets also show similar distributions. This shows that most of the time (i.e., 86.1% of posts) it is feasible to predict tags based only on past tags. Thus, in the real world, the principal difficulty is to handle cases in which existing users try to tag new items and therefore strictly graph-based recommenders (e.g., [112, 114]) will not be able to make recommendations most of the time.

2.5.3 The Baselines

Let U be the set of users u, I be the set of items i being tagged, T be the set of tags t and M be the set of timestamps τ. Additionally, S is the set of all records s, representing the relations among the four types of objects, $S \subseteq U \times I \times T \times M$. Each record $(u, i, t, \tau) \in S$ means that user u has tagged an item i with the tag t at time τ. Here, we also define P_s as all the distinct user-item-time combinations:

$P_s = \{(u, i, \tau) | \exists t \in T : (u, i, t, \tau) \in S\}$.

Long-Term Interests Model. If we assume that users' interests are not drifting over time, then users' interests can be modeled as long-term interests. We assume that the users' interests—$P(t|u)$ the probability of tags occurring—follows a multinomial distribution, from which the MLE gives us a simple representation of $P_{\tau_p}(t|u) = \frac{\sum_{p' \in P'_u} c(t, p'|u)}{\sum_{t'} \sum_{p' \in P'_u} c(t', p'|u)}$, where $c(t', p'|u)$ is the number of times that tag t' occurs on post p', and typically users use a tag only once per post. P'_u is the set of u's posts whose timestamps are earlier than the current time. Long-term interest models simply recommend the most frequent tags used in the past.

Short-Term Interests Model. Users' interests may change over time; thus users' recent behaviors can better represent users' current preferences. We model short-term interests using a sliding window which is common in temporal methods. $P_{\tau_p}(t|u)$ will be calculated only based on recent data (e.g., within three days). The size, σ, of the time window corresponds to the lifetime of short-term interests. Based on this Short-Term Interests model, we tune the parameter—the size of the time window.[5] The results are shown in Figure 2.10. We find that in Bibsonomy, the best performance is achieved when $\sigma = 30$ days. Overall, the more recent the data, the more accurate the estimate of users' interests.

2.6 Temporal Interests Model

The experiments using the Short-Term Interests Model show that the users' interests are continuous and similar within a time slot. However, the above time window methods may not fit the real case in which their interests are drifting over time, that is, $P_\tau(t|u)$ varies with changing τ. If we assume that the tagging behaviors of different users are independent, then for a specific user, we can only focus on

[5] Dataset and evaluation are the same as in Yin et al. [159]

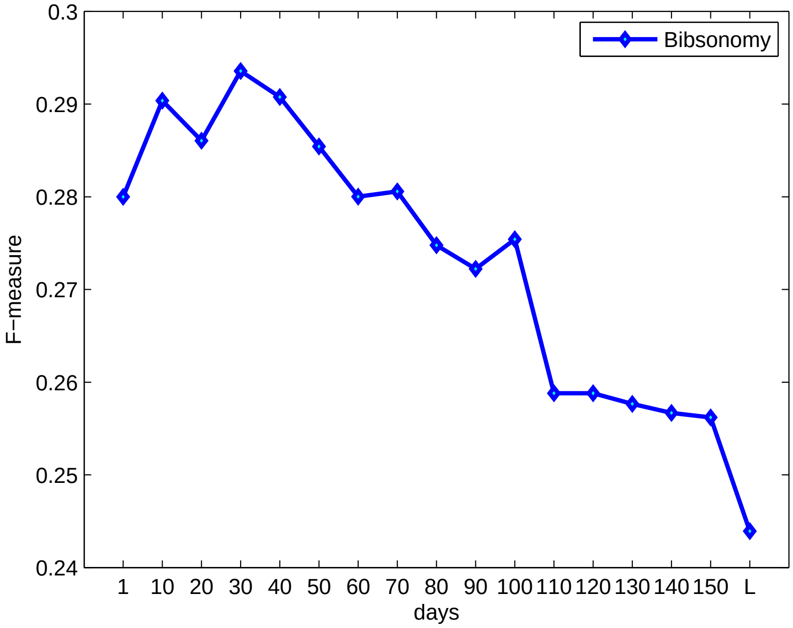


Figure 2.10: F-measure as a function of the time window size.

the user's past behaviors. The occurrences of tags $P(t|u)$ can be generated by a multinomial distribution or n-gram extension. We further make the assumption that the lifetimes for different tags are independent. Then, in post p, the tags are generated by a multinomial distribution and from a definite set T. Let $\theta_{t,u}$ refer to $P(t|u)$.

$$P_\tau(p|u) \propto \prod_{t \in T_p} \theta_{t,u}^{c(t,p|u)}$$

To model the dynamics of users' interests, we use the standard kernel smoothing technique and the likelihood at time τ is smoothed or weighted on users' data D_u by a non-negative smoothing kernel $K : \mathbb{R} \to \mathbb{R}$. By further assuming that the number of tags on posts is independent of t, the local likelihood can be written as

$$\begin{aligned} l_\tau(\eta|D_u) &\overset{\text{def}}{=} \sum_{\tau'\in M} K(\tau-\tau') \sum_{p'\in P_{\tau,u}} \log P(p';\eta) \\ &= \sum_{\tau'\in M} K(\tau-\tau') \sum_{p'\in P_{\tau,u}} \sum_{t\in T_{p'}} c(t,p'|u)\log \eta_t \end{aligned}$$

At each time τ, for user u, the estimation of each θ is derived by maximizing the local likelihood.

$$\hat{\theta}_{\tau,u} = \arg\max_{\eta\in\Theta_u} l_\tau(\eta|D_u)$$

There is a closed form expression for the local likelihood maximizer $\hat{\theta}_{\tau,u}$ which can be obtained by setting the gradient of the Lagrangian to 0.

$$0 = \frac{1}{[\hat{\theta}_{\tau,u}]_t} \sum_{\tau'\in M} K(\tau-\tau') \sum_{p'\in P_{\tau,u}} c(t,p'|u) + \lambda_t$$

By solving the above equation, we obtain

$$[\hat{\theta}_{\tau,u}]_t = \frac{\sum_{\tau'\in M} K(\tau-\tau') \sum_{p'\in P_{\tau,u}} c(t,p'|u)}{\sum_{\tau'\in M} K(\tau-\tau') \sum_{p'\in P_{\tau,u}} \sum_{t\in T_p'} c(t,p'|u)} \tag{2.18}$$

We can see that the present distribution $[\hat{\theta}_{\tau,u}]_t$ is actually the fraction of occurrences weighted by the kernel function. There are several choices for the kernel function [89, 77]. Usually, the kernel function is symmetric, like the uniform kernel ($K(\tau) = \mathbf{1}_{\{|\tau|<\sigma\}}$) and the Gaussian kernel. Because our task is to estimate the user's present distribution $[\hat{\theta}_{\tau,u}]_t$ based only on the past data, the kernel is only the right half of the symmetric kernel function and it can be also considered as decaying of interests. The speed of decay measures the probability of the user staying on the same topic over time. Unlike traditional approaches to concept drift which try to track global trends across the whole dataset and use a fixed kernel function, a very essential problem in social tagging systems is personalization. In particular, different users may have different decay speeds for short-term interests. Even for the same user, the behaviors on different tags are different. Thus, we propose a personalized method and moreover a personalized tag-specific model. It is more reasonable to model the problem as tag lifetime rather than as a simple kernel smoothing problem. Intuitively, once an interest appears, it will stay for a while and then become weaker and weaker. Assuming that the lifetime of the short-term interests follow the exponential distribution, then at time τ_i, the topics emerged and the probability of interests still staying at time τ_j is $P_\tau(t \texttt{ stay}|u) = \int_\tau^\infty \frac{1}{\sigma_{u,t}} e^{-\tau'/\sigma_{u,t}} = e^{-\tau/\sigma_{u,t}}$. Using this equation as the kernel function results in:

$$K_t(\tau|u) = e^{-\tau/\sigma_{u,t}} \tag{2.19}$$

where $\sigma_{u,t}$ is the user-tag specific parameter. For each user-tag pair, there will be a specific $\sigma_{u,t}$ to control the decay of this tag for the user. Equation 2.18 can be interpreted as the fraction of the weighted remaining interests. If we make the assumption that the same user has the same decay and lifetime distribution on all different tags, that is, $\sigma_u = \sigma_{u,t_1} = \sigma_{u,t_2} ... = \sigma_{u,t_i}$, the model becomes a personalized model.

For simplicity, we rewrite Equation 2.18 as follows.

$$P_{\tau_p}(t|u) = \frac{c'(t, p|u)}{\sum_t' c'(t', p|u)}$$

where $c'(t, p|u) = \sum_{p' \in P_u, p' \neq p, \tau_p \geq \tau_{p'}} c(t, p'|u) K_t(\tau_p - \tau_{p'}|u)$ and τ_p is the timestamp on post p. The problems is how to estimate the parameters $\sigma_{u,t}$.

2.6.1 Estimation of Parameters

From the assumption that the lifetime of the short-term interests follow the exponential distribution, we know that $\sigma_{u,t}$ is the mean lifetime of tag t for user u. We will consider a continuous sequence of posts where user u keeps using t as an event of topic t occurring.

Formally, for user u, let $p_1, p_2, ..., p_i, ..., p_n$ represent the sequence of u's posts in chronological order so far. Their timestamps $\tau_1 \leq \tau_2 \leq ...\tau_i \leq ... \leq \tau_n$. Let $s = (p_i, p_{i+1}...p_{j-1}, p_j)$ be a subsequence with maximum length where all posts contain tag t. At time τ_i, user u starts to use tag t, and at τ_j, and the user stops using tag t. In this event, the lifetime of tag t is $\tau_s = \tau_j - \tau_i$. Let $S_{u,t}$ represent the set of all such subsequences of tag t for user u. The parameter σ can be estimated as $\hat{\sigma}_{u,t} = \frac{1}{|S_{u,t}|} \sum_{s \in S_{u,t}} \tau_s$. It is consistent with the intuition that in the past, once user u starts to be interested in tag t and the interest always stays for a long while, then recent use of tag t will hold a strong signal that t will be used again. However, the above estimation may cause too much emphasis on personalization and so smoothing and controlling the weight of personalization are required:

$$\hat{\sigma}_{u,t} \quad = \quad \lambda \frac{1}{|S_{u,t}|} \sum_{s \in S_{u,t}} \tau_s + (1 - \lambda)\tau_a + \epsilon$$

In the above equation, λ is a factor which controls the tradeoff between personalization and non-personalization and τ_a is the average tag lifetime over all users and all tags. ϵ is a smoothing factor and is usually set to a small value. In fact, it is not only a smoothing factor, but also controls the tradeoff between short-term and long-term interests. If it is infinity, the model will be equivalent to the long-term interests model. The larger the ϵ, the smaller the differences of decays among different tags. In Figure 2.11, 2.12 and 2.13, we track 20 tags for a random user in Flickr. The x-axis is the time (day), and the y-axis is the $P_\tau(t|u)$. Three tags "2005", "rockandroll" and "livebands" are highlighted. From Figure 2.11, we can see the change of $P_\tau(t|u)$ of these tags. Because the data is from 2004 to 2005, we can see from the middle, the tag "2005" emerged and because of continuous usage of "2005", $P_\tau(2005|u)$ grows higher and higher. The tag "livebands" in first half

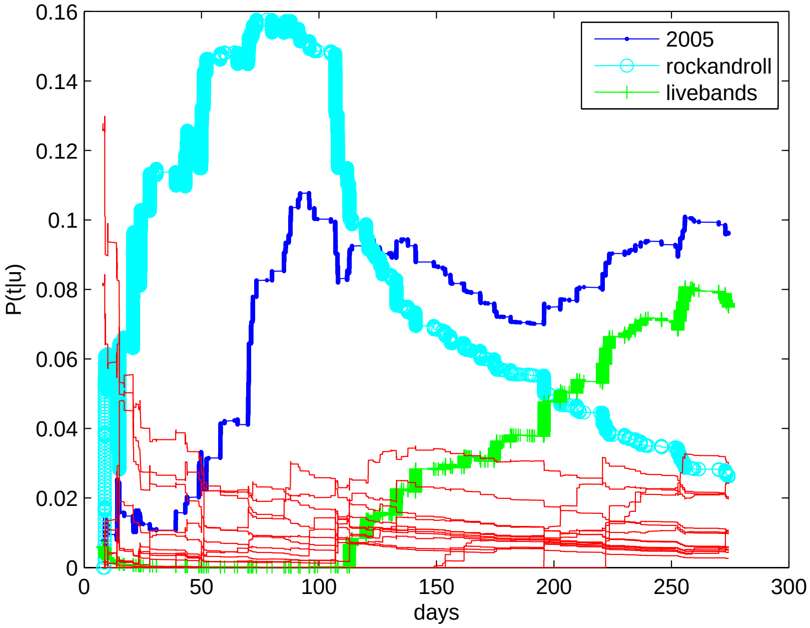


Figure 2.11: Tracking users interests $\epsilon = 100$.

is zero, because the user never uses that tag before 2005, and later user u became very interested in "livebands". Comparing the three figures, we notice that from $\epsilon = 100$, $\epsilon = 10$ to $\epsilon = 1$, the tracking become more and more detailed. Because as ϵ becomes lower, the local interests start to outperform the global interests and $P_\tau(t|u)$ becomes more sensitive to the short-term behaviors. For larger ϵ, it can capture the long-term trends of tags, and for smaller ϵ, it may better predict tags for current posts. It is difficult to determine which one is better and it depends on the task: when you try to capture trends of user interests, larger ϵ is suitable, and when you want to find the accurate tags on the test posts, smaller ϵ may be more suitable.

Similarly, for the non-tag-specific model, σ is the overall mean lifetime on all

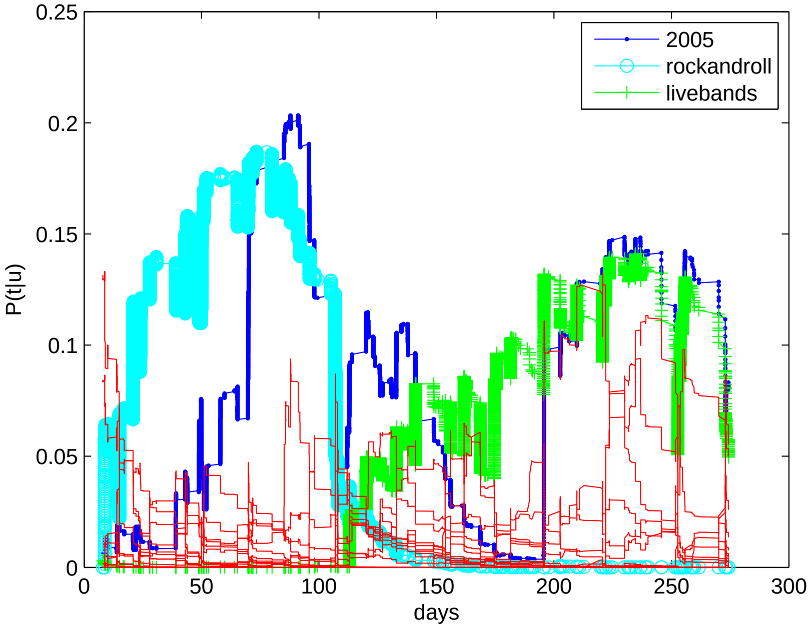


Figure 2.12: Tracking users interests $\epsilon = 10$.

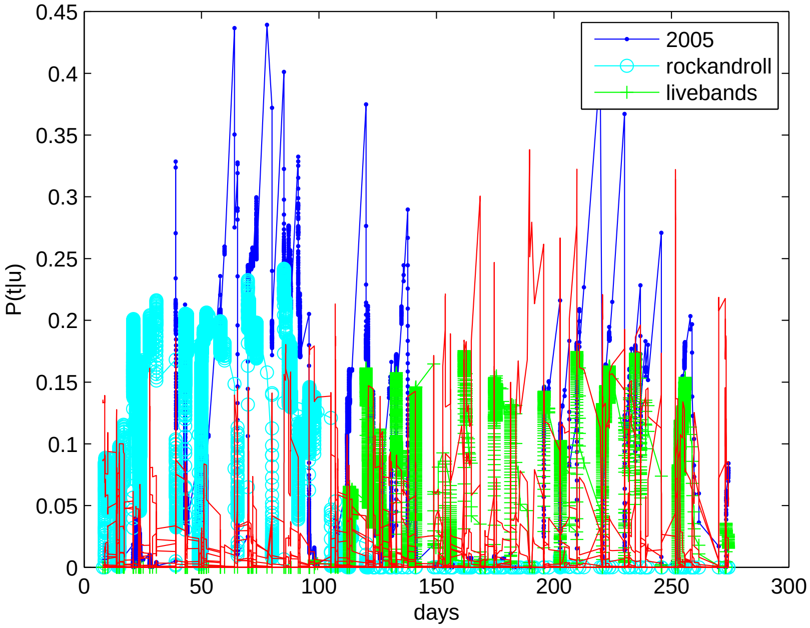


Figure 2.13: Tracking users interests $\epsilon = 1$.

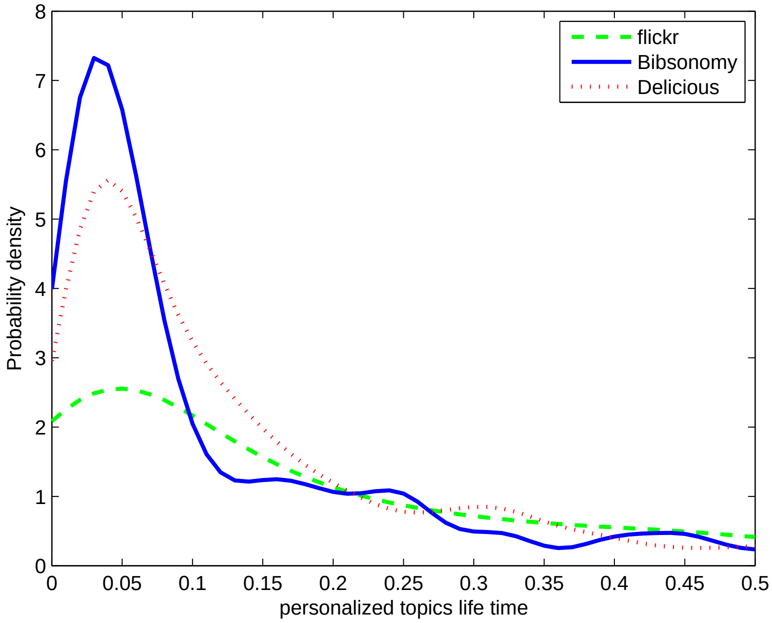


Figure 2.14: Estimated probability density of personalized topic lifetime.

tags, resulting in:

$$\hat{\sigma}_u = \lambda \frac{1}{\sum_t |S_{u,t}|} \sum_t \sum_{s \in S_{u,t}} \tau_s + (1-\lambda)\tau_a + \epsilon$$

When considering the whole data set, the variance of tag lifetime is large, making it difficult to determine a single lifetime for all users. Thus, we calculated a personalized tag lifetime for each user. Figure 2.14 shows the probability density of personalized tag lifetime. We can see that more users in Flickr hold longer tag lifetimes.

2.6.2 Capturing Topic Switches

From our observations, in personal tagging data, there often exist some topic switches—session-like behaviors as users switch between several subtopics. For the task of capturing the trends of users interests, the effects of topic switches are not so important as in task of tag prediction which require more accurate models of short-term interests.

Users may become interested in some new topics suddenly or switch back to some older topics because of some unknown external effects. We first assume that the current post (from the test set) is not a topic switch post, meaning the user continues the most recent session of tags on a particular topic. As in Yin et al. [155], we use a threshold on the tag similarity as measured by Jaccard's coefficient to define topic switches. For a given user, let p_{i-1}, p_i be two consecutive posts, whose timestamps are $\tau_{i-1} \leq \tau_i$ and tag sets are T_{i-1} and T_i. Use J_{p_{i-1},p_i} as the measurement of the possibility of a topic switch at post p_i: $J_{p_{i-1},p_i} = \frac{|T_{i-1} \cap T_i|}{|T_{i-1} \cup T_i|}$. The personalized session lengths for each user are controlled by a global threshold κ. If $J_{p_{i-1},p_i} < \kappa$, the post p_i is considered to be a topic switch. For each test post p, our method will find the post p_i from which the latest session begins, and then the kernel smoothing will be only effective from p_i. Although κ is a shared parameter among all users, it generates personalized session lengths for users.

In the above session model, we made an assumption that the current test post is not a topic switch post; however, in fact, the current post may be the start of a new session. We believe that the time interval from the current test post to the most recent post can help predict such a case. Intuitively, the longer the interval is, the higher the probability of a new session starting. To measure whether the current post p_c is the start of a new session, we propose a function $J_{p_c} = f(\tau_c), \mathbb{R} \to \mathbb{R}$ where J_{p_c} is the predicted tag similarity between the current test post p_c and the most recent post based on the elapsed time. For the current test post p_c of user u, we have all past posts of user u—P_u. For every two consecutive posts p_{i-1}, p_i, we have a time interval $\tau_i = \tau_{p_{i,u}} - \tau_{p_{i-1}}$ and their similarity value $J_i = J_{p_{i-1},p_i}$. Then we have a set of samples $(\tau_1, J_1), (\tau_2, J_2), ..., (\tau_n, J_n)$, from which we need to learn

Table 2.5: Validation Results

Method	Bibsonomy	Delicious	Flickr
Long-term model	.245	.161	.369
TIM	.325	.258	.726
User-tag TIM	**.334**	**.283**	**.733**
User-tag TIM (w/o κ)	.302	.276	.726
LZ (uniform)	.291	.191	.448
LZ (triagular)	.301	.237	.616

the function $J_{p_c} = f(\tau_c)$. While there are many regression methods, we use a non-parametric technique—the nearest neighbor method. Compared to kernel methods, the nearest neighbor method defines points local to τ_c not through the fixed kernel bandwidth, but instead on a set of points closest to τ_c, measured by the distance $d_{i,c} = |\tau_i - \tau_c|$. Then the regression at τ_c is calculated as $J_{p_c} = \frac{\sum_i w_i \cdot J_i}{\sum_i w_i}$ where w_i is a tri-cube weight function

$$w_i = \begin{cases} (1 - (\frac{d_{i,c}}{d_{k,c}})^3)^3 & d_{i,c} \leq d_{k,c} \\ 0 & d_{i,c} > d_{k,c} \end{cases}$$

where only k of n points closest to τ_c are considered as the neighborhood and $d_{k,c}$ is the distance of the furthest τ_c. Following the previous definition: if $J_{p_c} \geq \kappa$, the current test post will still stay in the current session and the session-based prediction method will be employed while if $J_{p_c} < \kappa$, we will treat this test post as the start of a new session and so at this moment, other methods which do not depend on temporal information can be employed, such as content-only methods [84]. In the following experiments, we will also discuss combinations of methods.

2.7 Experiments on Temporal Factor

On all three data sets, we split the whole data into two parts: earlier data and test data (the last 30 days data). Validation data in which 1000 posts are sampled from earlier data at random is used to tune and analyze the parameters. Then based on the last 30 days data, we perform completely online evaluation to simulate the tagging system running (evaluate each post over time and after that the post will be treated as an additional training post). In our interests lifetime model, there are two models: the personalized temporal interests model which assumes the users' behaviors on different tags are the same, and the personalized user-tag-specific temporal interests model in which users have different behaviors on different tags. We call them TIM and User-tag TIM.

We compare our method with three kinds of leading algorithms, which are from Lebanon and Zhao's [77] method of temporal document modeling (LZ), Yin et al.'s [159] method of personalized tag prediction (YXHD, also described in Section 2.3), and Lipczak et al.'s [84] method of content-only tag prediction (LHKM). Lipczak's method took the first place in the "content-based" recommendation task in ECML PKDD Discovery Challenge [39]. We use the common F-measure function of precision and recall to evaluate prediction performance as we used previously [159]. F-Measure is measured in break even point.

2.7.1 Parameter Analysis

Here we describe the parameter tuning process using the validation data (prior to the final month). In the predictive model, there are three parameters: ϵ is a smoothing factor, λ controls the personalization weight and κ is the factor of session detection. If $\epsilon = \infty$ and $\kappa = \infty$, the model is exactly the long-term interests model. On all three data sets, the effects of the three parameters are similar: for the tag prediction task, smaller ϵ is more suitable and can capture local interests better. λ tends to be better near to one. In Bibsonomy, the maxima appears when $\lambda = 1.0, \epsilon = 0.001, \kappa = 0.1$. In Delicious and Flickr datasets, the maximas appear at $\lambda = 0.8, \epsilon = 0.0001, \kappa = 0.3$

Table 2.6: Results on 30 day test data

Method	Bibsonomy	Delicious	Flickr
Long-term model	.118	.163	.312
User-tag TIM	**.501**	**.267**	**.835**
LZ (uniform)	.431	.203	.419
LZ (triangular)	.497	.232	.701

and $\lambda = 0.9, \epsilon = 0.0001, \kappa = 0.6$ respectively.

We also compare several variations of our methods to analyze the effects of each part. At first, we compare user-tag TIM with TIM where all tags of the same user share the same σ. In Table 2.5, the results show that user-tag TIM can outperform the default personalized model. Because the computational cost for the two algorithms is the same, we will use user-tag TIM in the following experiments. We also find that session-like behaviors are an important factor. In the tag prediction task, performance can be improved significantly over the version without topic switch detection (w/o κ).

The comparison method LZ is also carefully tuned, resulting in $h = 5$ in Bibsonomy and $h = 1$ in Delicious and Flickr. The triangular kernel and uniform kernel are used in local likelihood: the uniform kernel—$K_h(\tau) = 2^{-1} \cdot \mathbf{1}_{\{r<h\}}$ and the triangular kernel $K_h(\tau) = \frac{(1-\frac{r}{h})}{h} \cdot \mathbf{1}_{\{r<h\}}$.

2.7.2 Simulating the Real System

We simulate the real tagging system running on the last 30 days of data—performing completely online evaluation on the test data. There are 4,742 posts and 17,785 records in Bibsonomy, 21,916 posts and 76,213 records in Delicious and 110,551 posts and 517,949 records in Flickr. The results are shown in Table 2.6.

The results on test data are better than the results on validation data because the system has more historical information. It show that our user-tag TIM is better than

Table 2.7: Results on Bibsonomy

Method	F_1	Method	F_1	p-value
LZ	.306	User-tag TIM	**.341**	.0498
LHKM	.136	LHKM w. TIM	**.369**	7.56e-004
YXHD	.309	YXHD w. TIM	**.357**	.0033

the baselines and LZ on all three data sets. In Flickr, the performance achieves over 80% which is consistent with the fact that Flickr users' interests are more focused and easier to be tracked. Interestingly, it suggests that in real tagging systems, we can make effective recommendation through users' temporal interests analysis only.

2.7.3 Incorporating Content

In this section, we compare our temporal interests model with two successful content-related methods—YXHD (described in Section 2.3) and LHKM [84]. We use the Bibsonomy data set—the same data set as in Section 2.3, the same evaluation methods[6] and the same parameter tuning. Table 2.7 presents the results. Our temporal interests model can outperform the two content-related methods. The p-value is also calculated by two-sample t-test, compared to the state-of-art YXHD. We can see that TIM gets significant improvement.

YXHD treats the tag prediction problem as the reverse problem of web searching and start from the basic Bayes rule, integrating three factors—an ego-centric effect, environmental effects and web page content. Because users' preferences on each tags are drifting over time, intuition suggests that temporally adjusting the prior can get better results. To incorporate the content, we combine the two methods by replacing the $P(t|u)$ with the temporal prior $P_\tau(t|u)$ which has already been shown to better capture users' current preferences. The combined methods achieve an F-measure

[6] Under online evaluation mode, we also calculated Top-5 F-measure, and the results are similar.

of 0.357, which is significantly better than either YXHD or our temporal interests model.

LHKM only uses content to recommend tags. The advantage of the LHKM algorithm is when processing new items and during topic switches. Because it is a content-only method, it does not distinguish whether the item has appeared in training data or not. Even if the current user suddenly changes interests, the algorithm can also obtain stable performance. In the detecting topics switches section, we describe a non-parametric method for simply combining TIM with other method. The results show that the combined LHKM can achieve the best performance. We also notice that because YXHD has already involved a high weight on the ego-centric effect, the improvement is not as high as LHKM.

2.8 Summary

In this chapter, we suggest that social tagging by nature is an incremental process, and perform a time-sensitive sampling on an existing public dataset. Our analysis shows that in the real world, the problem of tag prediction is dominated by the need to predict tags for existing users when they tag new items. Most graph-based methods require CORE p, and thus may simply not function in such situations.

We proposed a novel probabilistic model for personalized tag prediction. Our online experiments and 5-fold cross validation experiments indicate that our model achieves over 30% improvement on F-measure compared to a leading method, in the "real-world" test scenario.

We then investigated the temporal dynamics of user interests in tagging systems, and proposed a user-tag-specific temporal interests model for tracking users' interests. Using three public datasets we showed the impact of personalization and user-tag specification.

Based on our experiments, we are able to conclude that our temporal user interests model, generated only from the temporal tag sequence, can achieve an F-measure of 0.341 and outperform the state-of-the-art which is 0.309 for Bibsonomy data. Combining with existing methods YXHD and LHKM, performance further improved to 0.357 and 0.369, respectively. All three methods incorporating TIM can outperform the state-of-the-art as well as a leading algorithm addressing concept drift.

2.9 Bibliographic Notes

Personalized tag recommendation, as a special case of collaborative filtering, is a recent topic in recommender systems. The two main directions for these systems are content-based approaches and graph-based approaches.

Content-based methods, which usually encode users' preferences from textual information (e.g., web pages, academic papers, tags), can predict tags for new users and new items. One state-of-the-art content-based tag recommendation system [84] utilized several tag sources including item content and user history to build both profiles for users and tags.

Graph-based approaches, which usually have stronger assumptions than content-based ones (e.g., requiring every user, every item and every tag to occur in at least p posts), can provide better performance. Earlier work like FolkRank, introduced by Hotho et al. [62], is an adaptation of PageRank that can generate high quality recommendations which are shown empirically to be better than other previously proposed collaborative filtering models [64]. Guan et al. [51] proposed a framework based on a graph Laplacian to model interrelated multi-type objects involved in the tagging system. Recently, factorization models (also considered as graph-based approaches) show very successful evaluation results on personalized tag recommendation problems [112, 114, 130].

Non-personalized tag recommenders—i.e., for a given item they recommend to all users the same tags—have also attracted a lot of attention (e.g., [58, 127]). Garg et al. [42] propose a personalized interactive tag suggestion system which suggests tags based on the ones that a user entered most recently. They employ a naive Bayes classifier which is only based on tag co-occurrences.

An important factor not considered by any of the above methods is the temporal dynamics of users' short-term interests. Recent research also shows that users are much more likely to use their recently used tags. Zhang et al. [165] investigate the recurrence dynamics of social tagging. Time information is also important to recommend high-quality tags to users. In recommender systems and collaborative filtering, temporal information has already shown its success. Ding et al. [35] simply add a time weight on collaborative filtering through a decay function. Koren [73] demonstrates that users' interests evolve and presents a model which can track the temporal behavior to better recommend items for users. Xiang et al. [146] model long-term and short-term preferences by creating a session node on user-item graph and then use temporal personalized random walk to recommend items for users. Modeling long-term and short-term interests is also related to the problem of concept drift which needs to find the balance between temporal effects and long-term trends [120, 144]. Lebanon et al. [77] introduce a local likelihood model for concept drift which weights the local likelihood by using a kernel function. Another similar method is positional language model which is proposed by Lv and Zhai [89]. Both models are proximity-based methods. In this kind of method, the smoothing actually models the lifetime of users' short-term interests.

Chapter 3

Structural Link Analysis and Prediction in Microblogs

In the above chapter, we study the personalized tagging prediction and temporal dynamics in social tagging systems. In this chapter, we will investigate another popular system—microblogging system. We first analyze the link structure in Twitter, and then by analyzing data collected over time, we compare most popular and recent methods and principles for link prediction. Finally, we propose novel structural methods to calculate the probability of a link being created by examining the current user's local network structure. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

3.1 Introduction

The use of online social networks and social media in general has surged in recent years. In this chapter, we focus on the understanding of the use of one particular type of social service—that of the microblogging network. In microblog services such as Twitter, Yammer and Google Buzz, participants form an explicit social network by "following" (subscribing to) another user and thus automatically receive the (short) messages generated by the target user. Unlike some online social networks such as Facebook, LinkedIn or Myspace, a followed user has the option but not the requirement to similarly follow back. Thus, relationships in these social networks may be asymmetric, leading to three kinds of link relationships between users A and B. If A follows B, we say that A is a follower of B, and that B is a friend of A. If A and B both follow each other, we consider them mutual friends or reciprocal friends.

Thus, user B in a microblog service can generate messages, which are generally public and searchable, and any followers of B, such as A, will automatically receive those messages along with messages generated by all other users that A follows. The combination of multiple message intentions and asymmetry of connections has led some to call microblogging services such as Twitter "hybrid networks" [76, 117]. They are hybrid not just because they can carry multiple types of messages, but also because participants create links for multiple reasons—to be social (e.g., to connect online to existing offline social contacts) or to link to information sources. With multiple types of users, it may be difficult to understand how microblogging networks grow and evolve. In a hybrid social-information network, there are two viewpoints to consider. In an information network, the link prediction problem is like the recommendation problem, which is to recommend an information source to an information consumer. In a social network, the problem is to recommend friends to the users, as introduced by Liben-Nowell and Kleinberg [80]. If we can predict the next link that a user will likely create, we will 1) have a model of the user's interests that may be of value in recommending new links (e.g., as in Twitter's recently introduced "Who to follow" friend suggestions, and many third-party suggestion services) and in detecting communities; 2) be closer to modeling the network's overall growth processes; and, 3) be able to simplify the task of adding that link when the user wishes to do so.

In this chapter, we analyze link structures in Twitter to predict future links. Our contributions are as follows.

- We analyze real Twitter data collected over time to answer the question of from where the new links come. We additionally compare three sampling methods for the link prediction task.
- We are the first to experimentally compare many popular link prediction methods in a microblogging network. Furthermore, we also compare with matrix factorization—the most popular method of recommender systems.
- We propose a novel structure-based link prediction method. Empirical results on ego-centric networks of Twitter users show that our methods can outperform state-of-the-art methods on this task.

3.2 Link Analysis

In this section, we define concepts and examine characteristics of the Twitter relationship graph.

3.2.1 Definitions and Data Sets

As with any network, it is convenient to model the hybrid network as a directed graph $G(V, E)$. Users in the network are represented as nodes V and the links are represented as directed edge E. If user v_u follows v_c and v_c does not follow back in a microblogging system, there will be an edge $(v_u \Rightarrow v_c)$, we call v_c is a friend-only of v_u. If user v_u is followed-only by v_c in the microblogging system, there will be an edge ($v_u \Leftarrow v_c$)—we call it v_c is a follower-only of v_u and if user v_u and v_c follow each other, then ($v_u \Leftrightarrow v_c$), and we call it v_u is a reciprocal friends of v_c.

In our experiments, we randomly sampled 1000 English-tweeting users out of 9,026,165 active users between early February and the end of March 2010. Though users may appear multiple times in the public timeline, we sampled by name, not by tweets, so highly active users had no additional selection advantage. In the end, we had 979 users as our target users.[1] We monitored daily the changes in the selected users' ego-centric networks on Twitter. That means that each day we recorded changes to their friends and followers. The number of immediate friends and followers of the 979 target users was nearly 200,000. Since it was not possible to monitor daily such a large number of users, we decided to crawl the ego-networks of this group of users monthly. The data we used in this dissertation is from April 5th to May 12th, 2010.[2] This data set helped us better and more accurately understand from where new links come.

Our background data is the Twitter data set collected by KAIST researchers [76]. To build their data set, they crawled the entire Twitter site in July 2009 and obtained 41.7 million user profiles and 1.47 billion social relations. For some analysis, we use this data set and assume it represents the whole Twitter network.

3.2.2 Links Formation Analysis

By examining the changing networks on a daily basis, we can determine from where new links come. We monitored the changes of ego-networks for each of the 979 users. We collected a total of 18,777 new friends for the 979 users. Most new users

[1] During monitoring, 21 users changed their privacy setting to "protect", preventing us from continuing to collect their information.

[2] The data is prior to the introduction of Twitter's friend recommendation system which may introduce a link formation bias.

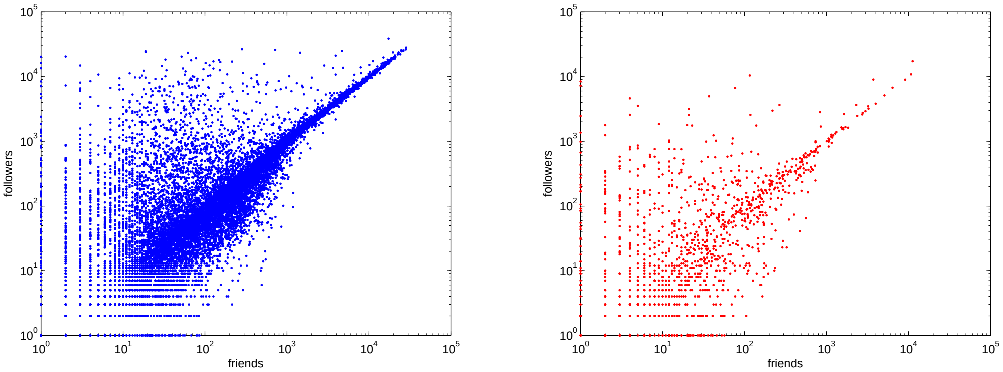
type	Dynamic	Static	ForestFire	Uniform
Unknown	0.08126	0.05939	0.90350	0.64741
⇔⇔	**0.48712**	**0.49710**	0.00028	**0.16158**
⇔⇒	0.03974	0.02341	0.00102	0.03321
⇔⇐	0.04082	0.06068	0.00264	0.03823
⇒⇔	0.01636	0.05391	0.00155	0.03582
⇒⇒	0.03706	0.06889	**0.04457**	0.01684
⇒⇐	0.01112	0.03830	0.01955	0.01213
⇐⇔	0.17471	0.12875	0.00087	0.03312
⇐⇒	0.02700	0.00869	0.00456	0.00769
⇐⇐	0.08477	0.06084	0.02141	0.01392

Table 3.1: The distribution of relationship types for new links.

are friends of friends. In particular, 17,251 (91.78%) new friends were second level neighbors within the target user's ego-network and the remaining 1,526 ($P_{unknown}$ = 8.12%) new friends were of unknown relationship (i.e., more than two hops away).

We also find that only 12% follower-only users appear in the new friends, thus it is reasonable to use the follower-only users as negative samples in some learning algorithms. For each of the new friends, we further check their relationship type with the 979 target users. Table 3.1 shows the distribution of relationship patterns for each new friend, prior to link creation where ⇔⇔ means target users ⇔⇔ new friend. For example, ⇔⇔ means the new friend is the reciprocal friend of some reciprocal friend of the target user. Similarly, ⇒⇒ means the new friend is the following-only friend of some following-only friend of the target user. It is similar for other types. Based on the results shown in Table 3.1, if we use two-hop neighbors in the ego-network as candidates to recommend as friends, we will only miss 8.12% of new friends. In previous research, researchers typically [9] only use second hop users as candidates for link prediction, and here we verify that that choice in hybrid-network also works.

Our observation confirms that it is reasonable to only use second hop neighbors



(a) Degree analysis of new friends that were two hops from the ego.

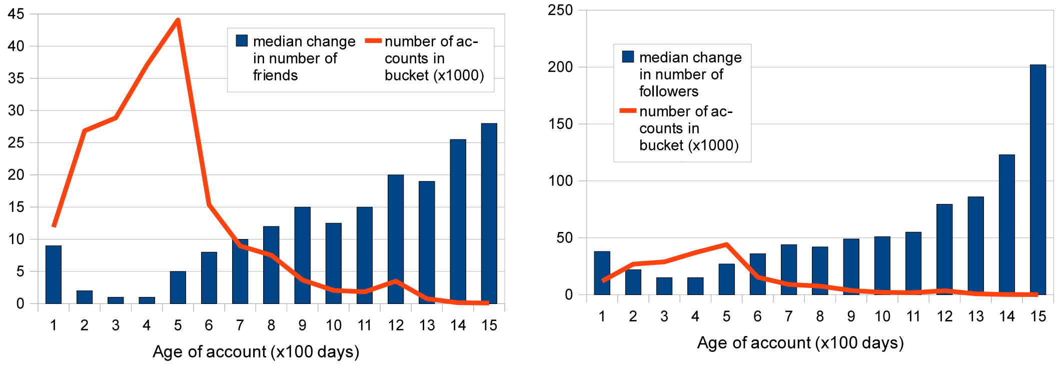
(b) Degree analysis of new friends that were more than two hops from the ego.

Figure 3.1: Scatterplots of friend-degree versus follower-degree.

as candidates for link prediction. Another interesting result is that even if two users share an audience, it does not suggest that they are interested in each other. Among two-hop users, most of the new friends have relationships ⇔⇔, ⇐⇔, ⇐⇐. In Romero and Kleinberg [117], they only find ⇒⇒ as an factor in link formation. We extend their observations and find ⇔⇔ is by far the most important indicator for future link formation. We also compare two-hop new friends with those more than more than two hops away, assuming that those nearby neighbors would be more likely to be social connections and the far-away connections would be to information producers. However, Figure 3.1 (Scatterplots of friend-degree versus follower-degree for friends that were two-hops away and for new friends that were more than two hops away) shows that from degree analysis, two-hop new friends and more remote new friends are very similar.

3.2.3 Age Analysis

We analyze the relationship between the changes in the size of a user's network and user's account age. We compare two snapshots (April 5th 2010 and August 20th



(a) Change in number of friends.

(b) Change in number of followers.

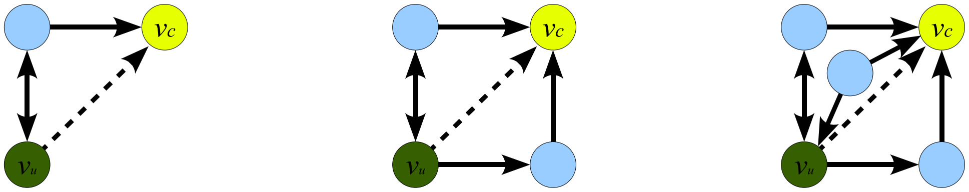
Figure 3.2: Account age analysis in Microblogging system.

2010) of profiles of 200,000 users. The results are shown in Figure 3.2 (the changes in the median number of neighbors as a function of account age). The x-axis is the user's account age and y-axis is the median change in the number of neighbors. We find that in early participation (accounts no older than 100 days), users add many friends and for more experienced users (100-400 day account ages), their friends become more stable. For long-time users (more than 500 days), we surprisingly find that their number of new friends is larger and larger. For followers, Figure 3.2(b) shows a *rich get richer* pattern; the older the user, the larger the increase in followers. A more detailed analysis (not shown for space) reveals that young accounts (e.g., less than two years) have a larger (but decreasing over time) change in followers and friends, while more established accounts (from about two years on) have a more consistent relative growth rate.

3.2.4 Comparing Sampling Methods

Link prediction experiments are usually based on a sampled graph rather than the whole graph before deploying it on a real system. However, real dynamic data usually is not available, in which case, artificial data is necessary, and is used in many studies that attempt to "predict" links that have been removed from a static graph [28, 160, 161]. We consider three sampling approaches. ForestFire [79] is a popular sampling method, preserving many graph properties of the sampled graph, such as some static properties (e.g., degree distributions, clustering coefficients), temporal properties (e.g., shrinking diameters) and cascading properties (shown in [33]). We crawled a graph by ForestFire, which contains 1,607,178 users, and 2,900,516 links and remove 10% links at random as test data. Another data set is sampled from the whole twitter network of the KAIST data[76]: we randomly select 1,600,000 users, and then put the links back—total 2,008,519 links and also remove 10% links at random as test data, which we call Uniform data. The last data set is based on the April 5th snapshot of ego-centric network. For 1000 egos, we remove 10% of the links at random as test data, which we call Static data. Treating those removed data as the new links, we perform the same analysis experiment as Section 3.2.2 to analyze whether the artificial data can retain properties consistent with real data. The results are also shown in Table 3.1. The static data keeps the consistent distribution among all kinds of the two-hop relationships. For ForestFire and Uniform, the properties are quite different from Dynamic: fewer than 10% of the candidates are found on the second hop in ForestFire and fewer than 40% of the candidates are found on the second hop in Uniform, causing ego-network based structural methods, such as common neighbors and Jaccard coefficient to fail on such cases.

Furthermore, we find that the distribution of two-hop relationships in ForestFire are also changed. The fraction of $\Leftrightarrow\Leftrightarrow$ becomes very low and $\Rightarrow\Rightarrow$ becomes higher. We also draw the distribution of the number of 2-hop paths from the target users to the 'new' friends after removing links, and Figure 3.3 shows how to count the 2-hop paths. The distributions of four data sets are shown in the Figure 3.4 (Distribution of links with exactly x 2-hop paths in the graph generated by each sampling method).



(a) $l = 1$ (b) $l = 2$ (c) $l = 3$

Figure 3.3: The counting of 2-hop paths l

We can see that the distribution of static data is very similar to the real data—Dynamic, but the distribution of ForestFire is different from the real (dynamic) data. That means that even when we only use the candidates which are still available via two hops from the target users as our test data, the algorithms may still generate different performance, compared with the real dynamic data. These experimental results suggests that for the link prediction task, the common evaluation method which is based on ForestFire sampled data may not produce the same results as real data. Finally, we run a simple but popular method—Common Neighbors on the fore data sets.[3] The predictive performance results are 0.116 in Dynamic, 0.071 in Static, 0.0013 in ForestFire data and 0.011 in Uniform data respectively. As we expect, prediction fails to be competitive with ForestFire data set and Uniform data while the performances on Dynamic and Static data are significantly higher. Although ForestFire and Uniform sample does keep some good properties for link analysis such as PageRank, temporal properties, they may not be suitable for the link prediction task.

[3] We use the standard F1-measure in the break even point.

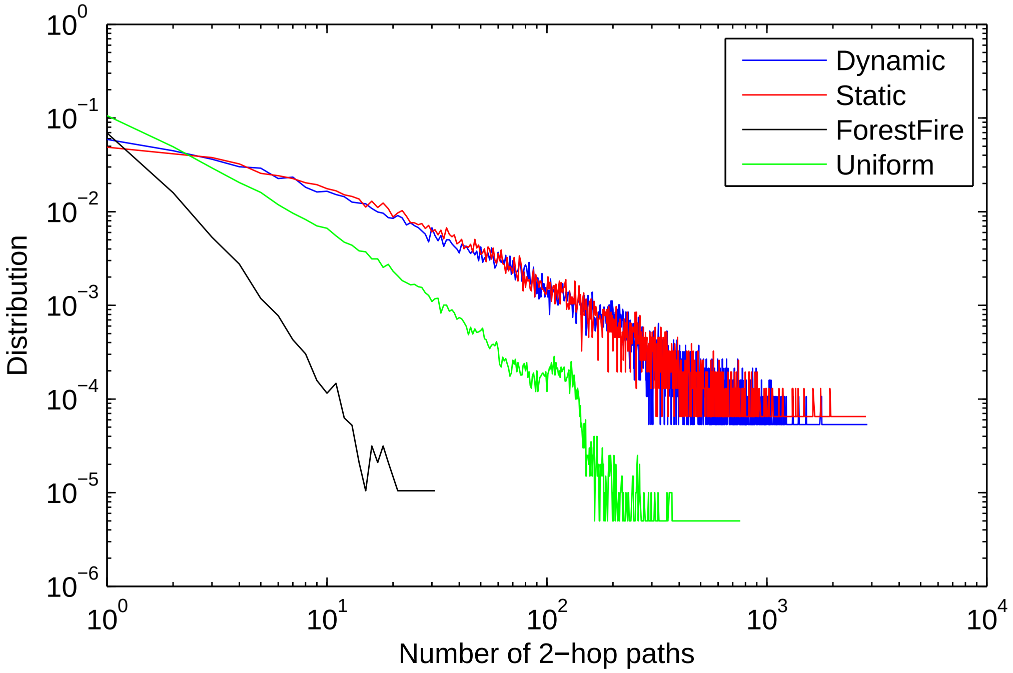
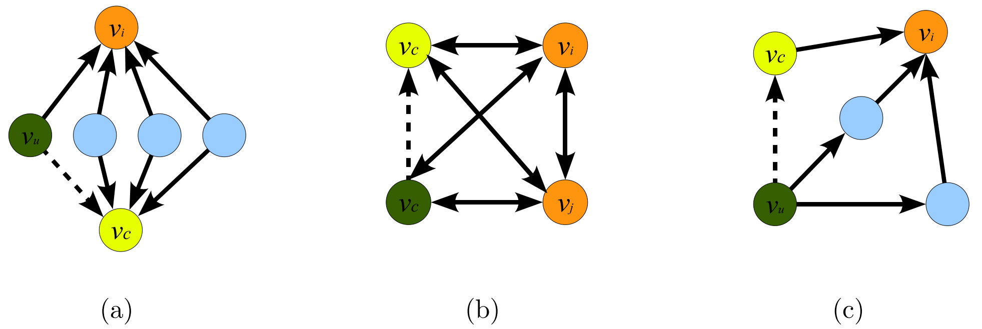


Figure 3.4: Distribution of links

3.3 Link Prediction

In this section, we introduce our prediction framework based on link structures. In a hybrid social-information network, structures can reflect many scenarios that may be useful for capturing users' interests and predicting potential links. In Figure 3.5, we can see some examples of various structural meanings: a) User v_u may be interested in v_c, because other similar users with v_u are following v_c. b) User v_u may want to follow v_c, because they may be friends with each other in real life and they are willing to use microblog as social networks. c) User v_u may want to follow v_c, because v_u is following other users which are following v_i while v_c is also following v_i and they may share the same interests. With these three examples, we have already seen some meanings of structures. We wish to design a model to exhaust such structural information for predicting new links.



(a) (b) (c)

Figure 3.5: Examples of relationships between user v_u and candidate v_c.

Suppose that we want to recommend other users which user u may know or be interested in following. The problem we consider is that given a user u and the whole network G, what is the probability that user v_u follows user v_c: $P(v_u \rightarrow v_c|G)$. We will rank candidate users according to this equation, and the top N users will be recommended to user v_u. To calculate $P(v_u \rightarrow v_c|G)$, theoretically, each intermediate user/vertex v_i can contribute some structural information which represents two parts: the link structure between v_i and v_c and the link structure between v_u and v_i. Now let us define the set of target users to which we will recommend some friends V_u, the set of intermediate users which we will exhaust the structural information V_i, and the set of candidate users for recommendation V_c. Assuming that $P(v_u \rightarrow v_c|G)$ is the linear combination of all possible intermediate users/vertices' contribution, we have

$$P(v_u \rightarrow v_c|G) = \textstyle\sum_{v_i \in V_i} b_{v_i,v_c} \cdot a_{v_u,v_i}$$

Let b_{v_i,v_c} represent the contribution of the structural information between v_i and v_c, which can be considered as the strength of v_i's recommendation for v_c and a_{v_u,v_i} represent the contribution of the structural information between v_u and v_i, which can be considered as the score of v_u liking a recommendation of v_i. We will denote with A the matrix with elements $A_{v_u,v_i} = a_{v_u,v_i}$ and with A_{v_u} the column of A corresponding to v_u. Similarly, $B \in \mathcal{R}^{|V_i| \times |V_c|}$ with column vector B_{v_c}. Let $R_{v_u} = [r_{v_u,v_1}, r_{v_u,v_2}, ...r_{v_u,v_n}]$ represent the current friends snapshot of v_u in which $r_{v_u,v_i} = 1$ means v_i is a current friend of v_u and $r_{v_u,v_i} = 0$ means v_i is the current follower-only of v_u.

In section 3.2.2, we report that only 12% of follower-only users of all follower-only users become new friends; thus it is perhaps reasonable to use follower-only users as negative samples. Then,

$$\hat{R} = A^T B$$

In probabilistic view, we define the conditional distribution over the current friends.

$$p(R|A, B, \sigma^2) = \prod_{v_u \in V_u} \prod_{v_c \in V_c} \left[\mathcal{N}(R|A_{v_u}^T B_{v_c}, \sigma^2)\right]^{I_{v_u,v_c}}$$

where $\mathcal{N}(R|A_{v_u}^T B_{v_c}, \sigma^2)$ is the probability function of the gaussian distribution with mean $A_{v_u}^T B_{v_c}$ and variance σ^2. I_{v_u,v_c} is the indicator function for selecting observed training data. For pair (v_u, v_c), if we use it as our training data, then $I_{v_u,v_c} = 1$, otherwise, $I_{v_u,v_c} = 0$. We also place zero-mean spherical Gaussian priors on the two structural parts A and B

$$p(A|\sigma_A^2) = \prod_{v_u \in V_u} \mathcal{N}(A_{v_u}|0, \sigma_A^2 \mathbf{I})$$
$$p(B|\sigma_B^2) = \prod_{v_c \in V_c} \mathcal{N}(B_{v_c}|0, \sigma_B^2 \mathbf{I})$$

The log of the posterior distribution over R,A and B is given by

$$\begin{aligned}
& \ln p(A, B|R, \sigma^2, \sigma_A^2, \sigma_b^2) \\
= & -\frac{1}{2\sigma^2} \sum_{v_u \in V_u} \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 \\
& -\frac{1}{2\sigma_A^2} \sum_{v_u \in V_u} A_{v_u}^T A_{v_u} - \frac{1}{2\sigma_B^2} \sum_{v_c \in V_c} B_{v_c}^T B_{v_c} \\
& -\frac{1}{2} \left(\sum_{v_u \in V_u} \sum_{v_c \in V_c} I_{v_u,v_c} \right) \ln \sigma^2 - \frac{1}{2}|V_u||V_i| \ln \sigma_A^2 \\
& -\frac{1}{2}|V_c||V_i| \ln \sigma_B^2 + C
\end{aligned}$$

where σ_A and σ_B control the smoothing factor of A and B. Let $\sigma_B = \sigma_A$, and then maximizing the log of the posterior distribution is equivalent to

$$\begin{aligned}
\min_{A,B} \quad & \sum_{v_u} \sum_{v_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 \\
& +\lambda_1(\|A\|_{Fro}^2 + \|B\|_{Fro}^2)
\end{aligned}$$

where $\lambda_1 = \sigma^2/\sigma_A^2$, is actually the smoothing factor and $\|\cdot\|_{Fro}^2$ denotes the Frobenius norm. Next, we need to involve structural regularization into the objective function.

3.3.1 Structural Regularization

In section 3.2, we show that more than 90% of new links go to people two hops away from user (the ego). Intuitively, if two users v_i and v_j are far away on the graph, that is, the shortest path between v_i and v_j is too long, their structural information can be ignored. We can define the set of the effective structures S^e. For example, if we define that the structures with only one hop are effective, the set of effective structures will be $S^e = \{\Leftarrow, \Rightarrow, \Leftrightarrow\}$ and if we define that all structures with up to two hops are effective, then the set of effective structures will be $S^e = \{\Leftarrow, \Rightarrow, \Leftrightarrow, \Rightarrow\Rightarrow, \Rightarrow\Leftarrow, \Rightarrow\Leftrightarrow, \Leftarrow\Leftarrow, \Leftarrow\Rightarrow, \Leftarrow\Leftrightarrow, \Leftrightarrow\Leftarrow, \Leftrightarrow\Rightarrow, \Leftrightarrow\Leftrightarrow\}$. Let S_{v_i,v_j} represent the set of all possible structures from v_i to v_j and $S^e_{v_i,v_j}$ represent the set of all effective structures from v_i to v_j—$S^e_{v_i,v_j} = S_{v_i,v_j} \cap S^e$. Thus, if $S^e_{v_u,v_i} = \emptyset$ where $v_u \in V_u$ and $v_i \in V_i$, then let $a_{v_u,v_i} = 0$ and similarly, if $S^e_{v_i,v_c} = \emptyset$ where $v_i \in V_i$ and $v_c \in V_c$, then let $b_{v_i,v_c} = 0$.

Beginning at some user $v_u \in V_u$, intuitively, if the structures of $(v_u \leftrightsquigarrow v_i)$ and $(v_u \leftrightsquigarrow v_j)$ are similar or same, the contribution scores of a_{v_u,v_i} and a_{v_u,v_j} should be similar. Following this intuition, we make constraints on structural scores matrix A, and define a structural regularization function $\mathcal{S}(A)$ to constrain similar scores on similar structures.

$$\mathcal{S}(A) = \frac{\sum_{v_u \in V_u} \sum_{v_i \in V_i} \sum_{v_j \in V_i} \mathcal{W}_{v_u}(v_i,v_j)(a_{v_u,v_i} - a_{v_u,v_j})^2}{\sum_{v_u \in V_u} \sum_{v_i \in V_i} \sum_{v_j \in V_i} \mathcal{W}_{v_u}(v_i,v_j)}$$

where $\mathcal{W}_{v_u}(v_i, v_j)$ is the measurement of similarity on structures attached on v_u: the more similar the structures of $(v_u \leftrightsquigarrow v_i)$ and $(v_u \leftrightsquigarrow v_j)$ are, the higher value the $\mathcal{W}_{v_u}(v_i, v_j)$ is. There are many kinds of methods to measure the structural similarity. Here, We list two:

Binary weighting if $S^e_{v_u,v_i} = S^e_{v_u,v_j}$, then $\mathcal{W}_{v_u}(v_i, v_j) = 1$, otherwise $\mathcal{W}_{v_u}(v_i, v_j) = 0$.

Cosine weighting let $N_{S^e_{v_u,v_i}}$ represent the vector of quantified effective structures of $(v_u \leftrightsquigarrow v_i)$, that is, $N_{S^e_{v_u,v_i}} = [n_{v_u \Rightarrow v_i}, n_{v_u \Leftarrow v_i}, n_{v_u \Leftrightarrow v_i}, n_{v_u \Rightarrow\Rightarrow v_i} ...]$, where $n_{v_u \Rightarrow\Rightarrow v_i}$ is the number of $\Rightarrow\Rightarrow$ path from v_u to v_i. Then, the cosine similarity is calculated as $\mathcal{W}_{v_u}(v_i, v_j) = \frac{N_{S^e_{v_u,v_i}} \cdot N_{S^e_{v_u,v_j}}}{\|N_{S^e_{v_u,v_i}}\| \cdot \|N_{S^e_{v_u,v_j}}\|}$

We also notice that if we take $S^e = \{\Leftarrow, \Rightarrow, \Leftrightarrow\}$, the two kinds of weighting are equivalent, because $n_{v_u \Rightarrow v_i}, n_{v_u \Leftarrow v_i}$ and $n_{v_u \Leftrightarrow v_i}$ only can be 0 or 1. Similarly, we add the structural constraints to B, and we have

$$\mathcal{S}(B) = \frac{\sum_{v_i \in V_i} \sum_{v_c \in V_c} \sum_{v_k \in V_c} \mathcal{W}_{v_i}(v_c,v_k)(b_{v_i,v_c} - b_{v_i,v_k})^2}{\sum_{v_i \in V_i} \sum_{v_c \in V_c} \sum_{v_k \in V_c} \mathcal{W}_{v_i}(v_c,v_k)}$$

The objective function $\mathcal{O}$ becomes

$$\begin{aligned} \min_{A,B} \mathcal{O} &= \sum_{v_u \in V_u} \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 \\ & +\lambda_1 \|A_{v_u}\|_{Fro}^2 + \lambda_1 \|B\|_{Fro}^2 \\ & +\lambda_2 \mathcal{S}(A) + \lambda_2 \mathcal{S}(B) \end{aligned} \tag{3.1}$$

where λ_2 is the structural factor tuning the weight of structural regularization. In the above model, we see the two parameters λ_1 controls the weight of smoothing and λ_2 controls the weight of regularization. The selected training links are represented by I_{v_u,v_c}.

3.3.2 Prediction in Ego-centric Networks

We call the above model the global model because the prediction is from the global network and performs collaborative filtering among all V_u. The global model will run on the whole graph to make predictions for a specific user and it will take a relatively long time to finish the computation; however, sometimes users perform interactive behaviors—such as requesting lvingthemodan instant recommendation. In this case, the global model may not work because of such long term computation. Secondly, the friendship network of some users may be already stable [157] and they may not want to add new friends.

It is necessary to make instant prediction for the users who are eager to get new friends. Unfortunately, directly reducing the model to fit the local structures of user v_u will likely cause overfitting. Thus, here we introduce a local model.

Considering the extreme case that only one user v_u requests new friends, the matrix R and A will reduce to only vectors R_{v_u} and A_{v_u} and a personalized method is necessary. We recall that the meaning of A and B, a_{v_u,v_i} can be considered as the probability of v_u trusting the recommendation of v_i and b_{v_i,v_c} can be considered as the probability of v_i recommending v_c. For b_{v_i,v_c}, because we know the current friendship network of v_u and also the structural information of v_i, we can make B personalized for v_u—B_{v_u}, that is, b_{v_u,v_i,v_c} means the probability of v_i recommending v_c for v_u, given the structure information of v_i. For some specific user v_u, we assume that v_u is interested in all his friends. Given the structure of some path between v_i and v_c ($v_i \leftrightsquigarrow v_c$), we can use the following equation to get the approximation value of b_{v_u,v_i,v_c}:

$$\beta_{v_u,v_c,v_i} = \frac{\sum_{v_k \in V_{v_u \rightarrow}} \mathcal{W}_{v_i}(v_c,v_k)}{\sum_{v_k \in V} \mathcal{W}_{v_i}(v_c,v_k)}$$

where $v_u \in V_u$, $v_c \in V_c$ and $v_i \in V_i$. The above actually calculates the fraction of the number of v_u's friends who share similar structures with v_c over the number of all users who share similar structures with v_c. If the value β_{v_u,v_c,v_i} is larger, then there will be a larger probability that v_u will follow v_c. Then similarly as in section 3.3, for some specific targeting user v_u we let

$$\begin{aligned} p(A_{v_u}|\sigma_A^2) &= \mathcal{N}(A_{v_u}|0, \sigma_A^2 \mathbf{I}) \\ p(B|\beta_{v_u}, \sigma_K^2) &= \prod_{v_c \in V_c} \mathcal{N}(B_{v_c}|\beta_{v_u,v_c}, \sigma_B^2 \mathbf{I}) \end{aligned}$$

Then we have the objective function $\mathcal{O}_{v_u}$ for v_u:

$$\begin{aligned} \min_{A,B} \mathcal{O}_{v_u} &= \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 + \lambda_1 \|A_{v_u}\|^2 \\ &\quad + \lambda_1 \|B - \beta_u\|_{Fro}^2 + \lambda_2 \mathcal{S}(A_{v_u}) + \lambda_2 \mathcal{S}(B) \end{aligned} \tag{3.2}$$

3.3.3 Solving the Model

Solutions for Equations 3.1 and 3.2 are quite similar. One simple method is gradient descent. Intuitively, the structure rarely contributes negative effects and usually a user v_u likes some kinds of users or does not care about some other kinds of users. In the quantified observed matrix R, we also use 1 for current v_u's friends and 0 to represent links that v_u does not care about. From the Section 3.3.2, we also involve a guidance value—β. All these reasons lead us to constrain A and B to be nonnegative. Nonnegative matrix factorization has been researched for many years [78, 21].

The objective function $\mathcal{O}$ and $\mathcal{O}_{v_u}$ in Eq. 3.1 and Eq. 3.2 are not convex in both A and B together and it is realistic to expect an algorithm to find the global minima. The process we use for solving $\mathcal{O}$ and $\mathcal{O}_{v_u}$ in Equations 3.1 and 3.2 is to use an iterative algorithm following the methods in in [78, 21] to derive multiplicative update rules. The proof by Lee and Seung [78] suggests that the objective function will be nonincreasing under such update rules.

3.4 Experiments

In this section, we describe our prediction experiments. Our method is only based on structural information of the social graph; thus for comparison methods, we also mainly focus on structure-based methods which do not involve user properties or content.

3.4.1 Data Set and Evaluation

In link prediction experiments, we use the same 979 Twitter users as in Section 3.2 and their immediate neighbors (979 ego users and their neighbors) that were collected to build a network for the link prediction task. In total, there are 211,559 unique users. For our experiments, we employed two kinds of evaluation methods.

Static Evaluation. Based on the 979 users' ego network snapshot on April 5th

2010, for each target user whose number of friends is larger than ten, we remove five links to friends. The prediction task is then to use the pruned networks to find the missing links. This evaluation method is widely used in the link prediction literature [28, 160, 161]. We use this process both for parameter tuning and for model analysis.

Dynamic Evaluation. We also monitored the changes in the 979 users' friendships and recorded the new links established between April 5th and May 12th. Here, the prediction task is based on the April 5th network snapshot to predict new friends in the following months.

For validation purposes, we also run our experiments on a second static Twitter data set (described below in Section 3.4.6). Precision, recall and F-measure are calculated in the standard manner, and our main measurement is the F-measure based in the break even point.

3.4.2 Baselines

In this section, we analyze and discuss simple predictors and principles to show the difficulty of this problem. Golder et al. [46, 45] discuss link prediction in Twitter, analyzing several principles for link prediction, such as shared interests, shared followers, and mutuality. Romero and Kleinberg [117] also introduce the directed closure process in Twitter tie formation. Here, we re-implement and compare the simple predictors which are from the principles described in [117, 46, 45].

To represent the principle *Shared Interests*, we use the predictor: the number of shared friends. A shared interest is best represented by the relationship chain $v_u \rightarrow X \leftarrow v_c$. Similarly, *Shared Audience* ($v_u \leftarrow X \rightarrow v_c$) is measured by the number of shared followers. For *Transitivity* [46, 45] or the *Directed Closure Process* [117], we use Katz's methods with degree length $l = 2$, which is equivalent to the

Method	**Static**	**Dynamic**
Shared followers	0.078	0.119
Shared friends	0.061	0.083
Shared mutual	0.074	0.086
Common neighbors	0.071	0.116
Katz (l=2)	0.094	0.086

Table 3.2: Simple predictor analysis (F-measure).

number of paths $v_u \rightarrow X \rightarrow v_c$. We also test *Shared Mutual Friends*. *Shared Neighbors* is just the count of the total number of neighbors (both friends and followers) without considering direction. The results are shown in Table 3.2.

From Table 3.2 we can see that all simple predictors provide similar performance—around .10 F-measure. We notice that the shared friends predictor performs worse than others, and that implies that two users sharing the same interests may not be particularly interested in following each other. Overall, simply using any single predictor cannot generate good results. Better methods are necessary.

3.4.3 Parameter Analysis

In this section, we analyze our two models, and tune parameters on static data. In the experiments, we use the snapshots of the target user's friendship network to construct the observation matrix R: if user v_c is a friend of v_u, we will set the entry $r_{v_u,v_c} = 1$ and if user v_c is a follower-only of v_u, we will set the entry $r_{v_u,v_c} = 0$. Because we already know that more that 90% of new links are from second level neighbors, our effective structures are defined in one-hop; that is, each edge will have two parameters a and b respectively in A and B, and in the global model, it will generate prediction in two hops. In the local model, full structural information is captured in v_u's two-hop ego network. Initial values of A and B are all set to the same value. We finally find that when smoothing parameter $\lambda_1 = 100$ and

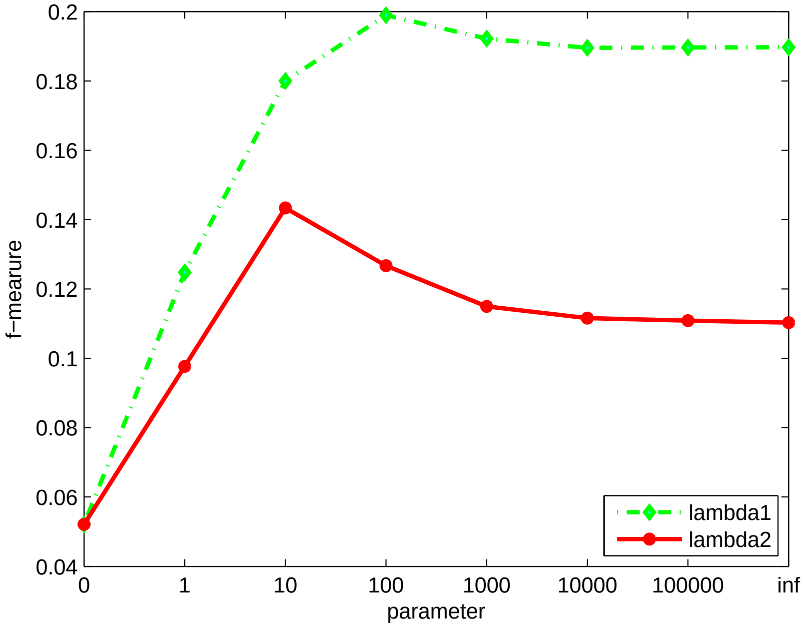


Figure 3.6: F-measure as a function of $\lambda_1(\lambda_2 = 0)$ and $\lambda_2(\lambda_1 = 0)$

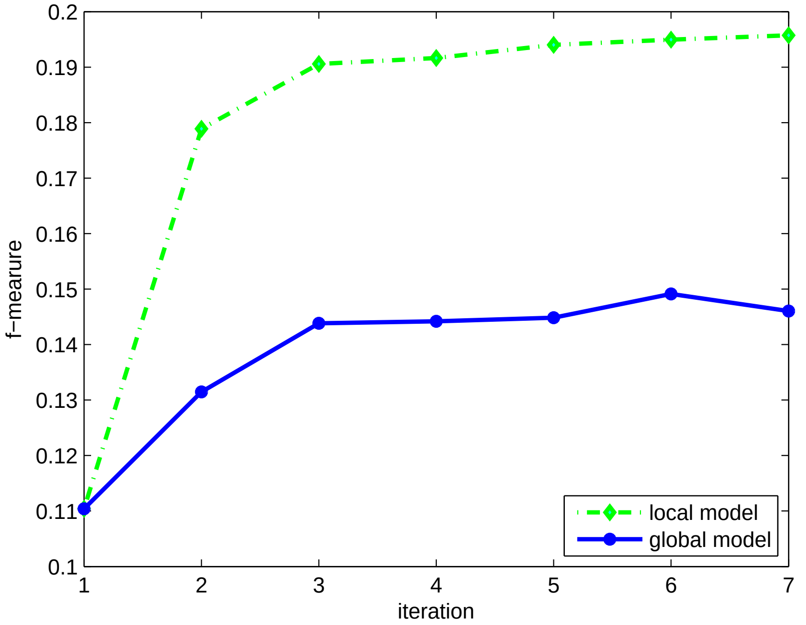


Figure 3.7: F-measure as a function of No. of iterations

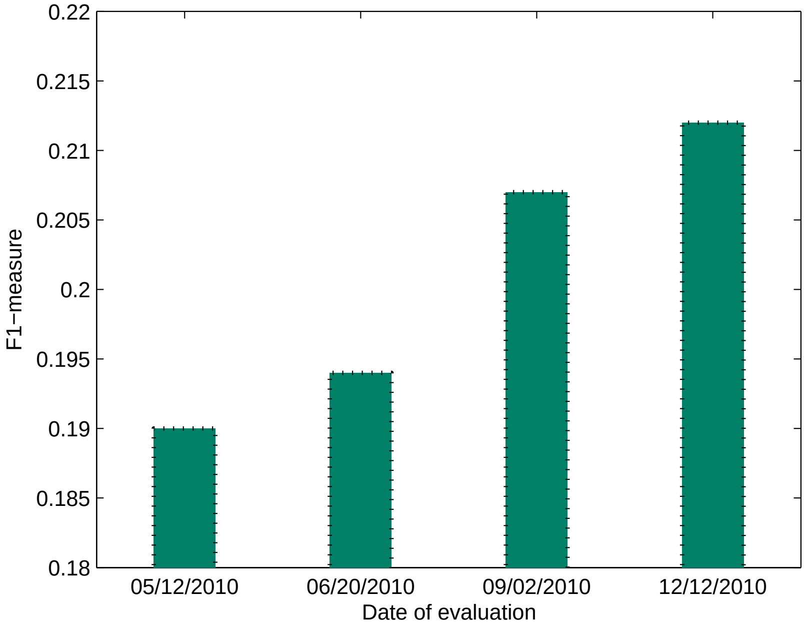


Figure 3.8: Evaluation on different time periods

regularization parameter $\lambda_2 = 100$ in the local model and $\lambda_1 = 1000$, $\lambda_2 = 100$ in the global model, the best performance is achieved. The model usually can converge within 10 iterations, and Figure 3.7 shows the performance changes as a function of iteration number. Because the current network is an ego-centric network which can provide full structural information, but the set of target users—V_u—is relatively small and may not provide good collaborative filtering, the performance of the local model is .197, which is better than the global model—.15. In the following, we use the local model for comparison.

Based on the local model and static data, we also analyze the effects of λ_1 and λ_2. Figure 3.6 shows the results. For the curve for λ_1, we set $\lambda_2 = 0$, and then tune λ_1 from 0 to infinity. We can see that it achieves the best performance when it is set to 100. We also note that when λ_2 is infinite, the model is reduced to the simple methods where links of the same type will share the same value. The performance of this model is still competitive, although its F-measure is lower than the best performance. Similarly, the performance of tuning λ_2 is shown in the same figure.

3.4.4 Comparing to Link Prediction Methods

In this section, we compare recent and popular methods which have been already used widely in link prediction problem. Six methods are implemented for comparison. The *Common neighbors* method simply counts the number of common neighbors. The *Jaccard coefficient* is calculated through dividing the number of common neighbors by the total number of neighbors. *Adamic/Adar* [2] refines the simple counting features by weighting rarer features more heavily. *Preferential attachment* scores are the product of v_c in-degree and v_u out-degree. We also compare the latest method—*PropFlow* [82]. In both Katz's method [68] and PropFlow, we tune the parameters l from 1 to 5 in static data.[4] Finally, we compare them on real dynamic data. The results are shown in the Static and Dynamic columns of Table 3.3.

In static evaluation, the results of *PropFlow*, *Common Neighbors*, *Jaccard Coefficient* and *Adamic/Adar* are similar and *PropFlow* which is a relatively newer method, gets better results than the other five competitors. *Jaccard Coefficient* shows competitive results which is similar with *PropFlow.* Preferential Attachment fails to predict missing links. For Preferential Attachment, because it only depends on the in-degree of the candidates, in the case of the information producers (with higher in-degrees), it may work. However, in real world, we know that individual

[4] In their paper, they also proposed a supervised method. Here, we select PropFlow for two reasons: First, for Lichtenwalter et al.'s supervised methods, there are many parameters to tune and selecting features is also a problem. Second, in their paper, PropFlow is used as a feature, and for most supervised methods, our method can also be used as a feature.

users are more prevalent than information producers. Thus, we can imagine the failure of Preferential Attachment. In dynamic evaluation, a point which we have to note is that unlike in static evaluation, the *Jaccard coefficient* works very well and even better than *PropFlow*. Within ego-networks, the *Jaccard coefficient* is a competitive method and also simple to calculate. We also find the failure of *Preferential Attachment*. Our method outperforms all other methods in both static evaluation and dynamic evaluation.

3.4.5 Comparing to Matrix Factorization

As mentioned earlier, another direction to solve the link prediction problem in a hybrid network is to use the techniques of the traditional recommender systems. Matrix Factorization is a popular method which is widely used in recommender systems [73, 75]. Here we employ the standard matrix factorization methods with smoothing. The observation matrix R is the same as the one in our model and the objective function is as follows.

$$\min_{A,B} \textstyle\sum_{v_u \in V_u} \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 + \lambda(\|A\|_{Fro}^2 + \|B\|_{Fro}^2)$$

To solve this optimization, we used stochastic gradient descent. Based on the static data, we tune the number of hidden features from 20 to 300, find the optimal parameter for comparison and set $\lambda = 0.05$. The results are shown in the last row of Table 3.3. On the static data, matrix factorization only achieves around .09 F-measure but the performance of matrix factorization on real data is also competitive at .163. Our model can outperform the standard matrix factorization in both static data and dynamic data because our method essentially incorporates matrix factorization techniques with structural information.

3.4.6 Validating Results

To test for sensitivity to our data set and sampling methods, we also ran our experiments on a subset of the large Twitter follow graph collected by Kwak et al. at KAIST [76]. We randomly sample 2,000 test users and extract their ego networks as in Section 3.4.3. There are, in total, 81,580 users and almost 10 million edges within this second test network. We again compared our methods with the other seven methods using the static dataset methodology. The results are shown in the rightmost column of Table 3.3 and are consistent with our earlier experiments. PropFlow is also better than other comparison methods. Our approach consistently outperforms all other tested methods.

3.4.7 Discussion

We have demonstrated many of the challenges of link prediction in a hybrid network and also noticed that the overall performance is relatively low, compared to results presented in some link prediction papers on other datasets. However, even when considering "social networks", most existing work does not directly examine online social networks, but rather networks of co-authorship or similarly constructed

Method	**Static**	**Dynamic**	**KAIST**
Our model	**0.197**	**0.190**	**0.127**
PropFlow	0.124	0.099	0.081
Katz	0.094	0.086	0.077
Jaccard coefficient	0.098	0.169	0.079
Adamic/Adar	0.090	0.128	0.069
Common neighbors	0.071	0.116	0.051
Pref. Attachment	0.012	0.012	0.023
Matrix factorization	0.082	0.163	0.074

Table 3.3: Comparing link prediction methods (F-measure).

networks reflecting some social relationship or record of activity.

On the other hand, the links in an online social network may reflect relationships (friends, family) that are not visible in a record of activity, and in a microblogging network with hybrid characteristics is even more complex.

As a result, previous methods which may work well on traditional social networks or co-authorship networks may not work as well on hybrid networks. Our results shows that the F-measure of many popular methods on our real-world data is only around 0.10.

Another cause for low performance of link prediction is that the microblogging network continues to grow. Each day, there may be many new links created [157]. In our experiments, we only evaluate new links within the following one month, so performance may be underestimated. It is possible that users are actually interested in those predicted links but they may not create those links within the following one month due to the fact that users may not discover those potential friends in a short period of time. In other words, users may create those links later, after our initial evaluation period. We conduct a simple experiment to test this: we make predictions based on the same training data—the 04/05/2010 snapshot, but we evaluate on different snapshots from different times. Figure 3.8 shows the results, and we find that after 05/12/2010, target users continue to create links which we had predicted, so measured performance grows higher and higher.

Another thing we can notice is that performance on the three test sets are different. For example, matrix factorization works well on the dynamic data but not well on the static data. We can imagine that static evaluation and dynamic evaluation have different properties such that some methods are better suited for one or the other. For the prediction task, dynamic evaluation is a more accurate estimate of future performance than static evaluation. However, if recommendation is the true end goal, it is difficult to tell which (if any) is better without involving a user study.

3.5 Summary

In this chapter, we examined the link structure and link prediction task within the Twitter microblogging network. In daily monitoring experiments, we analyzed properties of new links and saw from where in the network those links come and compared three sampling methods for the link prediction task. We proposed a novel personalized structure-based link prediction model and compared its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

3.6 Bibliographic Notes

There are several fundamental kinds of link prediction methods, such as structural methods, random walk methods and supervised methods. Liben-Nowell et al. surveyed an array of methods for link prediction in online social networks [80, 81].

One branch of structural methods is based on the local structure, such as *common neighbors*, *Jaccard coefficient* and *Adamic/Adar* [2] which refines the simple counting features by weighting rarer features more heavily. The *preferential attachment* method supposes that the likelihood that a new edge involves node v is proportional to $\Gamma(v)$, the number of neighbors of v. Based on global structure information, Clauset et al. [28] present a general technique for inferring hierarchical structure from network data and show that the existence of the hierarchy can simultaneously explain and quantitatively reproduce many commonly observed topological properties of networks.

Another approach utilizes random walk methods such as *Rooted PageRank* [80, 81] which is a variation of PageRank [105] that measures the stationary probability of each node in a random walk that returns to the root with some probability in each step. Weng et al. [141] try to identify influential users of micro-blogging services by using LDA to analyze user interests. Yin et al. propose a method which augments the original graph with attribute nodes, and then uses random walk to calculate link relevance [160, 161]. *SimRank* [65] recursively defines the similarity of two nodes and can also be interpreted in terms of a random walk. The most recent random walk-like method is *PropFlow* [82] which calculates the probability that a restricted random walk starting at node v_i ends at v_j in l steps. Katz [68] proposes a path-based method, which defines a measure that sums over the collection of all paths from v_i to v_j, and assigns more weight to shorter paths. Recently, Backstrom et al. [9] proposed a supervised random walk method which combines information from the network structure with node and edge level attributes. Supervised learning then adjusts the weights on different attributes to guide a random walk on the graph.

In supervised methods, the link prediction problem is usually considered as a classification problem. Such methods extract features from training data and can include both topological features (as in [67]) and node features. Hasan et al. [57] use different kinds of features, such as proximity features, aggregated features and topological features, and also compare different kinds of classifiers. More recently, Lichtenwalter et al. [82] examine important factors in the link prediction problem and present a classification framework which employs their *PropFlow* as a feature.

If you consider link prediction as a recommendation problem, a popular method is matrix factorization [75, 73, 72, 92] where the algorithms find hidden features for users and items by factorizing the observation matrix. However, those methods are designed for a user-item pair, and never before used for link prediction in social network.

There is other related research about link prediction [100, 136] and hybrid networks. Kwak et al. [76] find that the relationship of following and being followed on Twitter is not reciprocal, unlike most other social networking sites such as Myspace and Facebook. Romero and Kleinberg [117] also introduce the hybrid network concept and explore the directed closure process in Twitter. Recently, Golder et al. [46] discuss prediction specifically in Twitter. They analyze several principles for link prediction, such as shared interests, shared followers, and mutuality. They also discuss their user study results in [45].

Chapter 4

Multi-Relational Data Analysis in Online Social Media

In previous chapters, we have studied tag prediction and link prediction in social tagging systems and microblogging systems. However, they are focused on a single task or view. In this chapter, we will systematically investigate the coupled activities of users and their mutual effects in online social media. Our explanatory analyses demonstrate three principal challenges: coupled high order interaction, data sparsity and cold start on items. We tackle these problems by using a generalized latent factor model and Bayesian treatment. To evaluate performance, we test our methods on three real-world data sets—Flickr, Bibsonomy and MovieLens. Our experiments on these data sets show that to achieve best predictive performance, one can employ a fully Bayesian treatment in modeling high order relations in social media. Our methods noticeably outperform state-of-the-art approaches.

4.1 Introduction

Online social media services, such as Facebook, Flickr, YouTube and Twitter are designed to provide many opportunities for user engagement. Thus, in addition to being able to share content, users can often 1) rate content, 2) declare friendship with other users, 3) tag content with keywords, 4) comment on content, and 5) send personal or public messages to other users. Each of these activities provides valuable data to the service that can be used to model and predict future actions. For example, in a photo sharing site like Flickr, users of the service can add photos shared by others to their favorites. This is a form of rating, and so the service could examine a collection of user-photo pairings and build a model to predict whether the user would mark this photo as a favorite (e.g., what photos are preferred).

The second activity (declaring friendship) is similar to the more general activity of indicating the existence of a link between two entities (e.g., user-user rather than user-photo). Two-entity relations (or equivalently, relations of order two) are a common context for recommender systems [22, 30, 93, 73]. In fact, each of these activities could be approached as a recommendation or prediction problem by predicting zero (absence) or one (presence) of the link. The third activity (tagging) produces a non-numeric value, but it can also be viewed as a generalization of two-entity relations to three-entity relations. For example, when considering only the user-item relation, it might be more simply represented as a triple: user-item-tag, and the relation then signifies the use of that tag by that user for that item. The fourth type of behavior (commenting) is similar, but suggests an even harder problem: predicting a user's comments. The last activity type reveals relationships among users, as well as a potential message content prediction task.

Thus, all five activities provide contexts for prediction or recommendation, including rating prediction [3, 6, 75, 90, 119], tagging prediction [114, 112, 113] or link prediction [9, 80]. The existence of multiple activities across a single site (e.g., as can be found in YouTube) suggests that a model of the full system might be valuable to provide prediction or recommendation across multiple contexts [5]. The desire to provide recommendations even when the user or item is new (i.e., the cold start problem) provides additional rationale to co-relate relations of variable order. Thus, modeling multiple social media activities in a unified framework will not only solve multiple recommendation tasks simultaneously, but address data sparsity and cold start problems as well.

We have briefly discussed the challenges of analysis of data from social media services in Chapter 1 such as high order relations, high sparsity of the data, and co-related (coupled) activities. While past research has considered each of these issues separately, the work presented in this chapter addresses them together under a new, high-performing model.

To address such coupled higher order relations simultaneously as is often required for social media services, we propose a probabilistic generative model with the aim of inferring missing relation instances. Two problems are studied and analyzed by our methods: 1) connecting comments and tags in social tagging systems. We systematically investigate the coupled activities of users and their mutual effects in a social tagging system. Our explanatory analyses demonstrate three main challenges in modeling tagging systems: coupled high order interaction, data sparsity, and cold start on items. We tackle these problems by proposing a generalized Bayesian probabilistic latent factor model which can be tailored to fit the tagging system. We conduct empirical evaluations on two public data sets—Flickr and Bibsonomy. The experiments show that in social tagging systems, a user's commenting behaviors on item and user's tagging behaviors are highly correlated and can be mutually inferred, which has not been explored previously. 2) Incorporate temporal factor and external information in collaborative filtering, such as MovieLens and Netflix. Extensions to traditional recommender systems based on second order coupled entities fit naturally into our higher order relational model, such as temporal information [73, 148], user profile and item features [4, 152, 125, 163]. Our model can do so simultaneously and demonstrate superiority over state-of-the-art methods [119, 125, 148].

Before presenting the details of our proposed model and algorithm, we summarize the contributions of this chapter:

1. We propose a generalized probabilistic framework for high-order and multi-context relational data which is common in social media services.
2. By using our proposed framework, we can jointly model multiple activities, such as tagging logs, user comments, temporal rating history and social networks. Two application problems (social tagging system and temporal collaborative filtering) are analyzed and solved effectively by adopting our methods.
3. Experiments show that our Bayesian inference can achieve much better performance than the point estimation (MAP) of the parameters of our model, due to the sparsity of the high-order relational data in social media. Our model significantly outperforms state-of-the-art methods.

This chapter is organized as follows: we first present an application problem in Section 4.2. Sections 4.3 and 4.5 present the proposed model, followed by describing an efficient and scalable approach developed for estimating the model parameters in Section 4.4. Section 4.6 describes the fully Bayesian treatment for the model. Section 4.7 presents the empirical evaluations of the proposed approach on three data sets. Section 4.8 summarizes this chapter. Section 4.9 discusses related work.

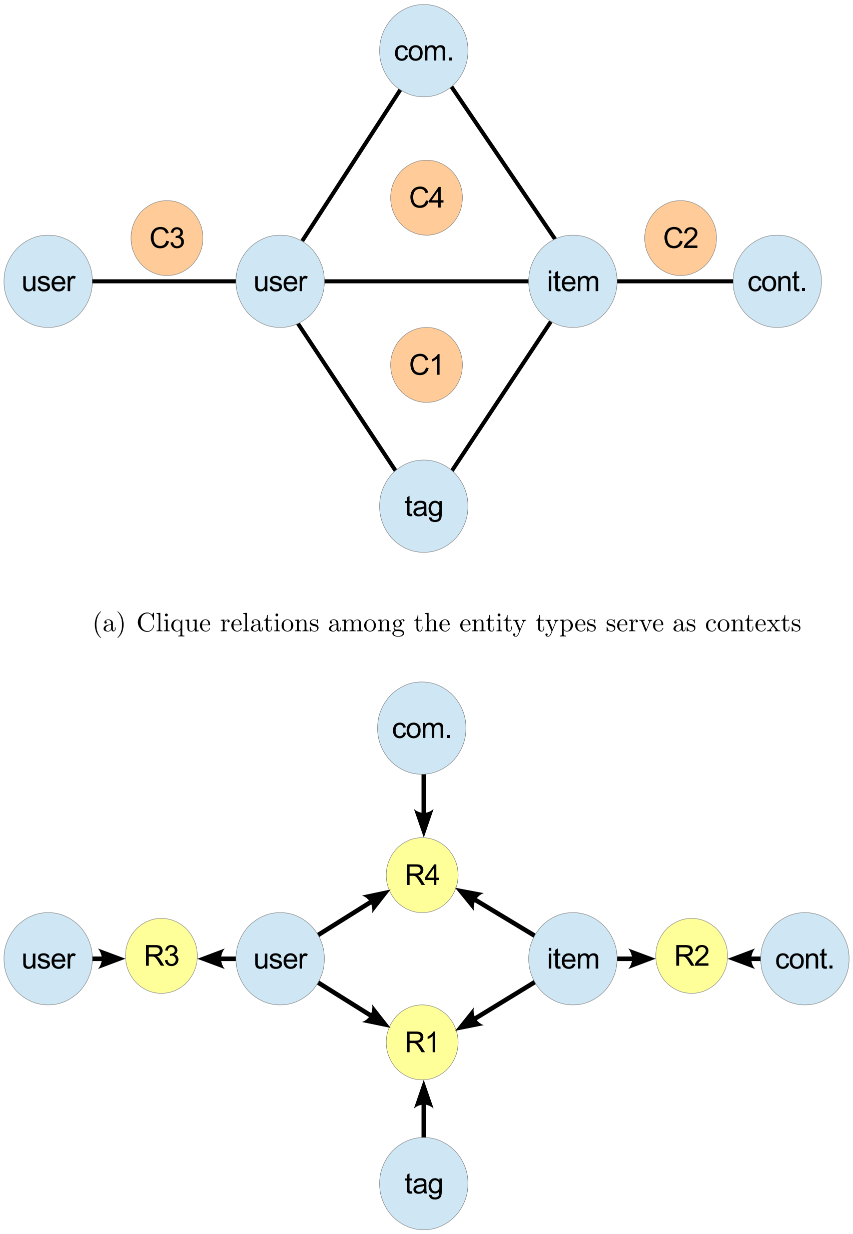
4.2 An Application: Connecting Comments and Tags

In this section, we take social tagging as an example. At first, we review the characteristics of social tagging systems and describe the challenges and problems in modeling user behaviors across multiple contexts such as tagging and commenting. Personalized tag prediction has been studied for several years, but personalized comment prediction, which is quite different from traditional opinion mining, is rarely investigated. For example, while Agarwal et al. [6] developed personalized comment recommendation via factor models, they do not predict the content (e.g., term frequency) of personalized comments which could potentially help in the interpretation of comments and improve sentiment analysis of comments. On the other hand, little is known about the connection and correlations among these behaviors and contexts in social tagging systems.

In addition to user-generated tags and comments, users are also able to denote friendship (via links) with other users. All of these activities provide potential hints for tag prediction, comment prediction and prediction of other user behaviors. By analyzing all of these activities, we can better capture users' preferences and make more accurate recommendations, but many of these activities are coupled and that coupling is not easily modeled.

4.2.1 Comparing with Traditional Collaborative Filtering

Unlike traditional collaborative filtering and recommendation tasks, in social tagging systems, a user's tagging and commenting activities generate relations involving more than two types of entities. In contrast, most traditional work focuses on second order relations that involve just two types of entities (e.g., user-item). In



(a) Clique relations among the entity types serve as contexts

(b) Bipartite graph between relations and entity types

Figure 4.1: An example of four relations on five entity types.

social tagging systems, the posts (that is, each tag produced by a user for an item) are by nature third order data [114, 112, 159, 158] that we consider as a triple (user-tag-item). Figure 4.1(a) show that in a tagging system, users, tags and items pairwise interact and compose a clique. C_1 is the tag post context (user-tag-item), C_2 is the item-content context (item-content feature), C_3 is the social network context (user-user) and C_4 is the comment context (user-comment-item). Similarly, users, tags and comments also interact pairwise. For tag/comments prediction task, we cannot drop any one of user, tag/comments, or item. By involving the temporal factor, it even becomes fourth order data [158]. However, these types of higher order relations have rarely been studied due to the complexity and difficulty in modeling and inference.

On the other hand, the relational data from different contexts are coupled together. In Figure 4.1(a), we can see the social tagging entity relations: there exist four cliques in this social tagging system (user-tag-item, user-comment-item, user-user, item-content). Within these cliques, all involved entities interact with all others. Among these cliques, they are strongly correlated with each other: for instance, activities where users comment on items or where users rate items share two of the same types of entity—user and item. With Figure 4.1(a), after recognizing the cliques, we can define them as contexts. Each context can be considered a type of observation individually and generated by the associated entities. In Figure 4.1(a), we see the directed bipartite graph, which describes which entities contribute to the process of generating each context. These contexts are frequently coupled together by sharing the same entities, increasing the difficulty of tackling the problem.

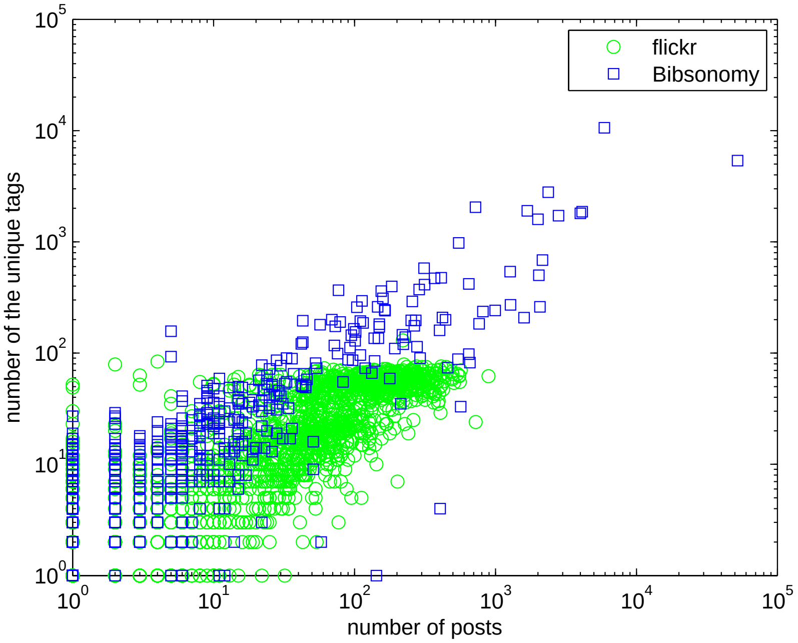


Figure 4.2: The number of tags as a function of the number of posts.

4.2.2 Data Sets

In this section, we conduct some simple analyses on two data sets: Flickr and Bibsonomy. The main data set is from Flickr. We crawl the data from **Flickr** by using the Flickr API[1]. This data set includes 2,866 users, 60,339 tags, 32,752 comment terms and 46,733 items (e.g., images), leading to the four relations shown in Figure 4.1(b). The other dataset is the **Bibsonomy** dataset is from the ECML PKDD 09 Challenge Workshop[2] which includes two relations: user-tag-item and item-content. The Bibsonomy data sets are the same as the data sets in Chapter 2. In Figure 4.2, we can see that the two datasets have different properties and users form two clusters. Similarly as in Chapter 2, Bibsonomy, users typically apply a larger variety of tags across fewer posts, suggesting that their interests are more

[1] http://www.flickr.com/services/api/

[2] http://www.kde.cs.uni-kassel.de/ws/dc09/

varied. In contrast, users in Flickr use fewer tags and their interests are more focused, by reusing their tags many times. This suggests that it may be easier to track user interests in Flickr.

4.2.3 Coupled Higher-order Systems

We conduct some analysis experiments for different relations. First, we examine the distribution of tags and terms in the comments. Figure 4.3 shows a linear relationship between the of number of tags/terms and the frequency of tags/terms in log scale. We can see that the distributions over both tags and comment terms are very similar and show two straight lines with essentially the same slope in the log-log plot. In the (user-comment-item) relation, among the 21,881 records in which a user comments on an item, 8,307 records show that the user also tags the same item, meaning that if a user comments on an item there will be around a 1/3 chance that the user will also contribute a tag for that item. This evidence shows the strong connection between the relation (user-comment-item) and the relation (user-tag-item).

Figure 4.4 (For each user, number of friends as a function of the number of commented items in Flickr) shows the coupling between user-user interaction and commenting activity. From the figure, we can see that most users are located in the upper left half of the Figure. Some users with many friends may NOT comment at all (or very little) but users who frequently comment on items usually have many friends. We also note that the inverse does not apply.

4.2.4 Cold Start

As in Chapter 2, we employ online evaluation in which only training posts which have earlier timestamps than those of the test posts are used. Note that this implies

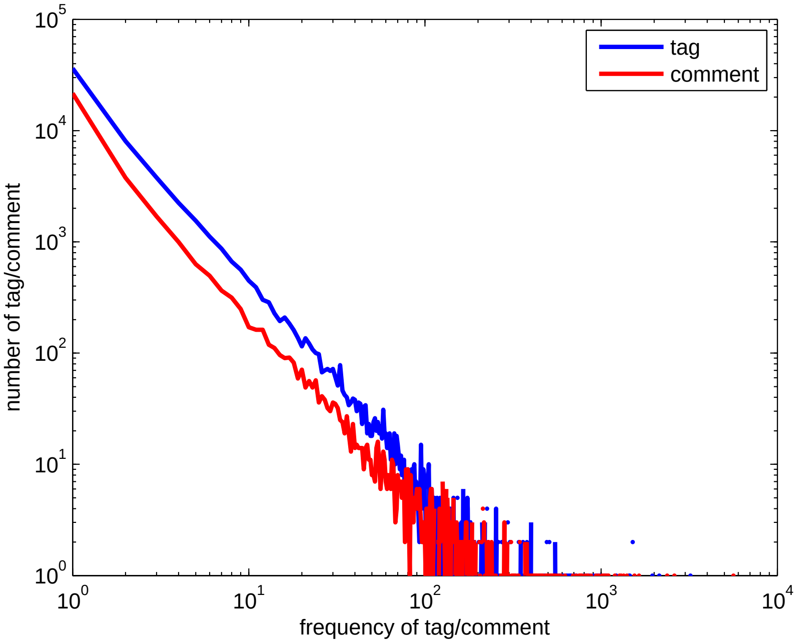


Figure 4.3: Distribution of tag/comments frequency in Flickr.

Table 4.1: Fractions of new users, items, or tags in samples from each data set.

	Bibsonomy	Flickr
New/Total Users	41/668	23/1000
New/Total Items	602/668	1000/1000
New/Total Tags	321/2207	175/4123

that the available training data is different for each test post and, for items tagged earlier in the timeline, fewer training data are available. While the online evaluation approach naturally fits the real-world case in which every post is used for testing a model trained on all prior posts, its feasibility depends highly on the efficiency of the training method as a new model may be necessary for each post. Instead, we can estimate the performance of the complete system by performing evaluation on only a sample of test posts, and largely avoid model-building efficiency concerns for the purpose of evaluation of effectiveness.

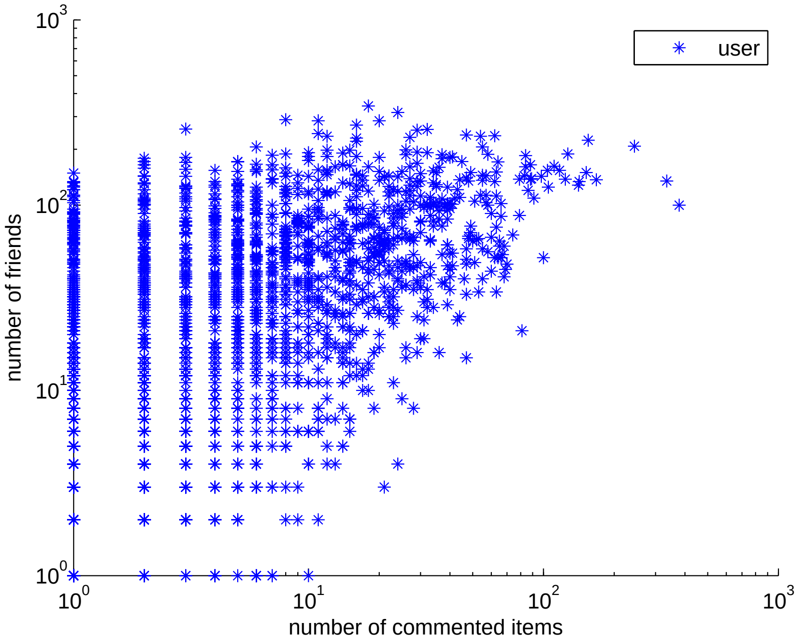


Figure 4.4: The number of friends against the number of commented items.

We utilize the online evaluation model and conduct time-sensitive sampling experiments on two data sets. For the Bibsonomy dataset, we use the same sampling dataset as in Yin et al. [159] which includes 668 test posts. For Flickr, we randomly choose 1000 posts. In all cases we effectively simulate a system running—the tagging system operates in an incremental mode. The data set statistics (shown in Table 4.1) demonstrate that in Bibsonomy data, we face a new user (a user which is not in any prior data) in 6.1% of the cases, and in 90.1% of the time users are trying to bookmark a "new item" not previously seen by the system. In addition, there is 13.9% chance that users would use new tags (which do not appear in the system before).

This shows that most of the time (i.e., 86.1% of posts) it is feasible to predict tags based only on previously seen tags. The other dataset also shows similar distributions. Thus, in the real world, the principal difficulty is to handle cases in which existing users try to tag new items and therefore strictly graph-based recommenders (e.g., [112, 114]) will not be able to make recommendations most of the time. This also suggests that incorporating external information, such as item content or comments into the model might help process these cold start cases.

4.2.5 Data Sparsity

Another notorious problem in most social media systems is data sparsity. Here, we define the number of observations over the total number of entries in the relations. For comparison, in one MovieLens data set[3], there are 1,000,000 ratings for 6,000 users and 4,000 movies, so the data density is 4.17%. The sparsity of data is even more serious when the relations is higher-order and coupled in social tagging system: in our Flickr data, there are 373,125 records in user-tag-item relation, so the density is 4.6170×10^{-8} $(373125/(2866 \times 60339 \times 46733))$; in context user-tag-item and for comments, there are 218161 records in user-tag-comments, so the density is 3.8518×10^{-8} $(218161/(2866 \times 60339 \times 32752))$. Similarly in our Bibsonomy data, the data density is 3.52×10^{-8}. Thus, data sparsity is considerably more severe in social tag data than the traditional 2-dimensional recommendation problem. The serious problem of sparsity in higher order relations strongly suggests Bayesian treatment. Previous work has already shown the significant advantage of the Bayesian approach in processing sparse data, such as in the comparison of LDA [15] to PLSA [60] and BPMF [118] to PMF [119].

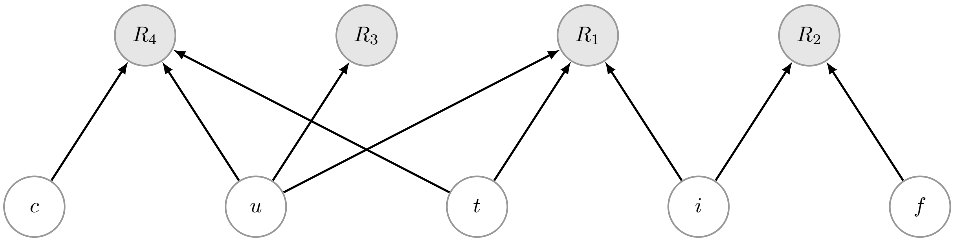


Figure 4.5: A bipartite graph representation of Fig 4.1(b)

4.3 Multi-Relational Data Model

With respect to all problems described above in Section 4.2, in this section we propose a latent factor model to model coupled higher-order data in the social tagging system. To handle the data sparsity, Bayesian treatment is employed to learn the parameters in the model.

An activity performed by a user in a specific social tag context induces a relation; for instance, the activity consisting of the triple (user-comment-item) is a 3-order relation with three types of entities. Let us consider a coupled higher order relational dataset with K types of entities. There are V possible relations among the entities and, for each entity type $k \in \{1, \ldots, K\}$, there are N_k possible entities. Each relation $v \in \{1, \ldots, V\}$ is associated with the list S_v of the entity types involved in relation v, that is $S_v = (S_{v1}, \ldots, S_{v|S_v|})$ with $S_{vj} \in \{1, \ldots, K\}$. Relations are then encoded by multi-dimensional arrays, where dimension j is indexed by entity type S_{vj}. The data associated with relation v are the observed triplets $\mathcal{D} = (v_m, \mathbf{i}_m, r_m)_{m=1}^{M}$ where for the m^{th} observation, $v_m \in \{1, \ldots, V\}$ is the index of the relation and $\mathbf{i}_m = (i_{m1}, \ldots, i_{m|S_{v_m}|})$ is a list of entity indices identifying the observation with value $r_m \in \mathbb{R}$.

Our probabilistic multi-relational data model assumes that each entity can be represented by a latent (i.e., unobserved) continuous feature vector in $\mathbb{R}^D$, where

[3] http://www.grouplens.org/node/73

K	Number of entity types.
N_k	Number of entities of type k.
D	Latent feature dimension.
V	Number of relations.
Θ_k	Latent features for entities of type k.
R_v	Set of M_v values corresponding to relation v observations.
M_v	Total number of observations of relation v.
S_v	List of indices identifying the types of relation v.
α_v^{-1}	Variance of the observations of relation v.

Table 4.2: Summary of the notation used in Figure 4.7.

D is typically small (e.g., of the order of 10 or 100). The low-dimensional latent features are denoted by $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_K)$, where $\Theta_k = (\boldsymbol{\theta}_{k1}, \ldots, \boldsymbol{\theta}_{kN_k})^T \in \mathbb{R}^{N_k \times D}$ contains the feature vectors associated to entity type k.

A summary of notation is shown in Table 4.2. To facilitate understanding of the notation, we consider the example described in Figure 4.5 where there are four relations and five entity types: u for users, i for items, f for item features, t for tags and c for comment terms. The four relations are coupled together by linking the same types of entities. Two of these four relations linking different entity types forms a 3-dimensional array, while the other two relations are encoded as two 2-dimensional arrays. To this end, we can define S as $\{S_1, S_2, S_3, S_4\}$, where $S_1 = \{u, i, t\}$, $S_2 = \{i, f\}$, $S_3 = \{u, u\}$ and $S_4 = \{u, c, i\}$.

Figure 4.6 shows the graphical model for multi-relational data factorization. The model assumes multi-linear links in order to predict the mean of the observations given the latent features of the corresponding entities. Formally, this means that for an observation of relation v with indices $\mathbf{i} = (i_1, \ldots, i_{|S_v|})$, the mean of the observation r is a multi-dimensional dot-product $\langle ., \cdots, . \rangle$ defined as

$$\langle \mathbf{\Theta_i} \rangle = \langle \theta_{i_1}, \cdots, \theta_{i|S_v|} \rangle = \sum_{d=1}^{D} \prod_{k \in S_v} \theta_{k i_k d} \ .$$

Note that for binary relations, this is equivalent to a standard vectorial dot-product. In this chapter, the distribution of the observations is assumed to be Gaussian with relation-dependent variances α_v^{-1}. This assumption can be relaxed easily to model other types of generalized linear models such as Poisson, Bernoulli distributions or other exponential family distributions[29]. Assuming independent observations, the likelihood is given by

$$\begin{aligned} p(\mathcal{D}|\mathbf{\Theta}) &= \prod_{(v,\mathbf{i},r) \in \mathcal{D}} p(r|\theta_{S_{v1} i_1}, \ldots, \theta_{S_{v|S_v|} i_{|S_v|}}, \alpha_v) \\ &= \prod_{(v,\mathbf{i},r) \in \mathcal{D}} \mathcal{N}(r| \sum_{d=1}^{D} \prod_{k \in S_v} \theta_{k i_k d}, \alpha_v^{-1}) \\ &= \prod_{(v,\mathbf{i},r) \in \mathcal{D}} e^{-\ell(\sum_{d=1}^{D} \prod_{k \in S_v} \theta_{k i_k d}, r; \alpha_v)}, \end{aligned}$$

where $\ell(\bar{r}, r; \alpha) = \frac{\alpha}{2}(r - \bar{r})^2 - \frac{1}{2} \log \frac{\alpha}{2\pi}$ is the quadratic loss.

We also assume that the prior distributions over $\Theta_1, \ldots, \Theta_K$ are independent isotropic Gaussian distributions with type-dependent variances $\sigma_1^2, \ldots, \sigma_K^2$:

$$p(\Theta_k|\sigma_k^2) = \prod_{j=1}^{N_k} \mathcal{N}(\theta_{kj}|0, \sigma_k^2 \mathbf{I}).$$

Distribution of the Response (Loss function)

In reality, there are several types of the response such as term frequency, rating, binary case. Here, we propose the corresponding distribution assumption for these types of the response.

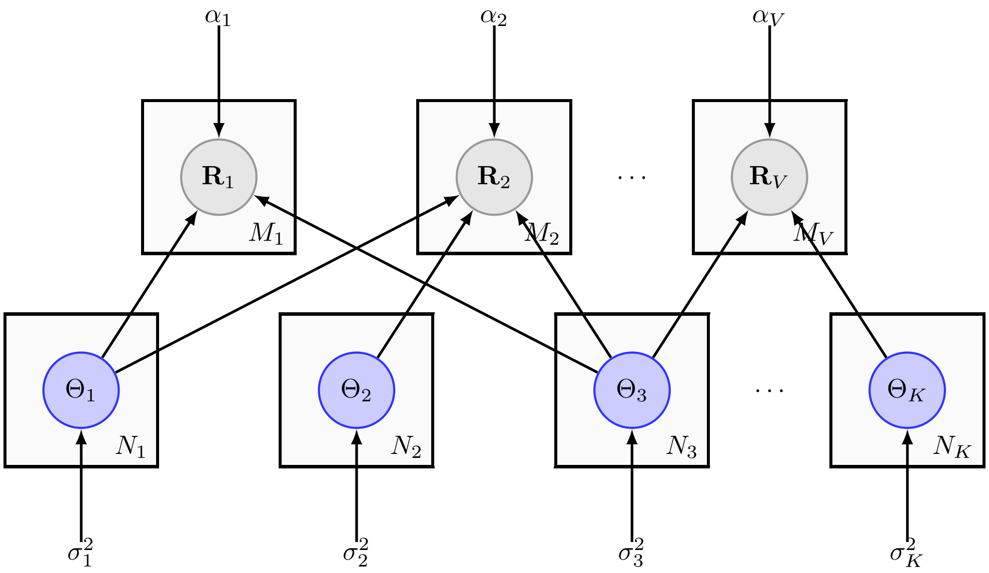


Figure 4.6: Probabilistic multi-relational data graphical model.

If the response is the real number such as item feature, we can make an assumption of Gaussian distribution

$$r \sim \mathcal{N}(\langle \mathbf{\Theta_i} \rangle, \alpha_v^{-1})$$

If the response is the positive integer, such as term frequency, we can make an assumption of Poisson distribution

$$r \sim \mathcal{P}(\langle \mathbf{\Theta_i} \rangle)$$

If the response is binary 0/1, such as link prediction, we can make an assumption of Bernoulli distribution

$$r \sim \mathcal{B}(s(\langle \mathbf{\Theta_i} \rangle))$$

where s is logistic function $s(x) = \frac{1}{1+e^{-x}}$. All these distribution assumptions are equivalent to specific loss functions.

Temporal Factor

Social media by nature are incremental processes. For a number of data sets, there is temporal structure in the data. In this section, we process the temporal factor, which is slightly different from the regular factor. Let t be the type of temporal factor among all K types, $t \in \{1, ..., K\}$. Because the time factor represent the evolution of global trends, a reasonable assumption is that they change smoothly over time rather than independently. Therefore we further assume that each time factor depends only on its immediately predecessor and use the following prior for time factor:

$$\begin{aligned} \theta_{tj} &\sim \mathcal{N}(\theta_{tj-1}, \sigma_t^2 \mathbf{I}), \quad j = 2...N_t \\ \theta_{t1} &\sim \mathcal{N}(\mu_t, \sigma_t^2 \mathbf{I}) \end{aligned}$$

$$p(\Theta_t) = \mathcal{N}(\theta_{t1}|\mu_t, \sigma_t^2 \mathbf{I}) \prod_{j=2}^{N_t} \mathcal{N}(\theta_{tj}|\theta_{tj-1}, \sigma_t^2 \mathbf{I})$$

4.4 Parameter Estimation

Now that we have presented the model, the remaining problem is to infer the latent variables $\boldsymbol{\Theta}$ given the observations. We first derive the *Maximum a Posteriori* (MAP) estimator of $\boldsymbol{\Theta}$. As the MAP estimator is easy to scale to very large datasets. The problem is therefore a simple minimization problem of a smooth and differentiable objective function equal to the negative log-likelihood:

$$\min_{\boldsymbol{\Theta}} \mathcal{O} \quad \text{, where} \quad \mathcal{O} := -\log p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\alpha}) - \log p(\boldsymbol{\Theta}|\boldsymbol{\sigma}), \tag{4.1}$$

and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_V)$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)$.

Two approaches to solve the optimization problem are *stochastic gradient descent* (SGD) and *alternating least squares* (ALS). ALS is a block-coordinate descent algorithm which minimizes Equation (4.1) with respect to one of the types, say Θ_k by fixing all others and repeats the same procedure for each Θ_k sequentially, ensuring that each step decreases the objective function. The procedure is repeated until convergence. The inner optimization problems are ordinary least squares which can be solved optimally. However, there is evidence from the tensor factorization literature that this procedure is not always effective because there are often strong dependencies between the feature values of the different types [109]. In addition, our method targets very large data sets for which even one pass through the data can be slow. This setting favors SGD-type algorithms since every gradient estimation is much cheaper than their batch counterpart (i.e., using standard unconstrained optimization tools such as L-BFGS [104]). This type of first-order optimization technique can be formally justified by a bias-variance argument, remarking that the ultimate goal of the estimation procedure is not the minimization of the objective (4.1), but the minimization of its expectation $\mathbb{E}[\mathcal{O}]$ under the sample distribution [17].

4.4.1 Parameter Learning for Θ

The SGD algorithms minimize large sum functions of the form $\mathcal{O} = \sum_{m=1}^{M} \mathcal{O}_m$ where M is typically large. The idea is that at each iteration of the algorithm, we only need the gradient of a single element of the sum, say $\mathcal{O}_m$. To apply this algorithm to our case, we need to decompose the objective in terms of a sum. The negative log-likelihood term $-\log p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\alpha}) = -\sum_{m=1}^{M} \log p(r_m|v_m, i_m, \boldsymbol{\Theta}, \boldsymbol{\alpha})$ has a suitable form, but more care is needed when dealing with the penalization term $-\log p(\boldsymbol{\Theta}|\boldsymbol{\sigma})$. Previous methods[152, 75] which are based on SGD algorithms set the same penalization term. That will cause the problem that the parameters will receive the different weights of the penalization. Here, we derive the exact the penalization term for each parameter. Denoting ν_{kn} the number of observations for the n^{th} entity of type k, that is to say

$$\sum_{(v,\mathbf{i},r)\in\mathcal{D}} \sum_{j}^{|S_v|} \frac{I(S_{vj}=k, i_j=n)}{\nu_{kn}} = 1, \forall(k,n).$$

Hence, we can combine the penalization term and the individual likelihood terms to obtain the following expression:

$$\begin{aligned}
\mathcal{O} &= \sum_{(v,\mathbf{i},r)\in\mathcal{D}} \ell\left(\sum_{d=1}^{D}\prod_{k\in S_v}\theta_{ki_kd}, r; \alpha_v\right) + \sum_{k=1}^{K}\sum_{n=1}^{N_k}\log p\left(\Theta_{kn}\right)\times 1 \\
&= \sum_{(v,\mathbf{i},r)\in\mathcal{D}} \ell\left(\sum_{d=1}^{D}\prod_{k\in S_v}\theta_{ki_kd}, r; \alpha_v\right) + \\
&\qquad \sum_{k,n}\sum_{(v,\mathbf{i},r)\in\mathcal{D}}\sum_{j}^{|S_v|}\frac{I(S_{vj}=k, i_j=n)}{\nu_{kn}}\log p\left(\Theta_k\right) \\
&= \sum_{(v,\mathbf{i},r)\in\mathcal{D}} \left(\ell\left(\sum_{d=1}^{D}\prod_{k\in S_v}\theta_{ki_kd}, r; \alpha_v\right) + \right. \\
&\qquad \left. \sum_{j}^{|S_v|}\sum_{k,n}\frac{I(S_{vj}=k, i_j=n)\log p\left(\Theta_k\right)}{\nu_{kn}}\right) \\
&= \sum_{(v,\mathbf{i},r)\in\mathcal{D}} \left(\ell\left(\sum_{d=1}^{D}\prod_{k\in S_v}\theta_{ki_kd}, r; \alpha_v\right) + \sum_{j}^{|S_v|}\frac{\log p\left(\Theta_{S_{vj}i_j}\right)}{\nu_{S_{vj}i_j}}\right) \qquad (4.2)
\end{aligned}$$

where $I(\cdot)$ is an indicator function. Our objective function (4.2) has the form of $\mathcal{O} = \sum_{m=1}^{M}\mathcal{O}_m = \sum_{m=1}^{M}\mathcal{O}_{(v_m\mathbf{i}_mr_m)}$ required by SGD if we set

$$\mathcal{O}_{(v,\mathbf{i},r)} := \ell(\sum_{d=1}^{D}\prod_{k\in S_v}\theta_{ki_kd}, r; \alpha_v) + \sum_{j=1}^{|S_v|}\frac{\|\boldsymbol{\theta}_{S_{vj}i_j}\|^2}{2\sigma^2_{S_{vj}}\nu_{S_{vj}i_j}}. \qquad (4.3)$$

It is now straightforward to compute the gradient with respect $\boldsymbol{\theta}_{kn}$ for every observation $(v, \mathbf{i}, r)$:

$$\nabla \boldsymbol{\theta}_{kn} \mathcal{O}_{(v,\mathbf{i},r)} = \ell'(\sum_{d=1}^{D} \prod_{k \in S_v} \theta_{ki_kd}, r; \alpha_v) \frac{\prod_{j=1}^{|S_v|} \boldsymbol{\theta}_{S_{vj}i_j}}{\boldsymbol{\theta}_{kn}} + \frac{\boldsymbol{\theta}_{kn}}{\sigma^2_{S_{vj}} \nu_{kn}}$$

if $k \in S_v$ and $n \in \mathbf{i}$, and 0 otherwise. The function ℓ' denotes the first derivative of the loss with respect to the first parameter, i.e. $\ell'(\bar{r}, r; \alpha) = \nabla_{\bar{r}} \ell(\bar{r}, r; \alpha) = \alpha(\bar{r} - r)$. Note that this gradient can be efficiently computed since it only requires the latent feature values of the entities involved in the current observation (i.e., with indices given by $\mathbf{i}$). If observations are chosen at random irrespective of the relation, on average we recover the exact gradient (up to a M^{-1} factor) of the full objective function defined in Equation (4.1):

$$\mathbb{E}\left[\nabla_{\theta_{kn}} O_{(v,\mathbf{i},r)}\right] = \frac{1}{M} \nabla \theta_{kn} \mathcal{O} \ .$$

4.4.2 Parameter Learning for α and σ^2

Under the Gaussian distribution assumption, the hyperparameters $\alpha_1, \cdots, \alpha_V$ correspond to a weighting of the different relations and the hyperparameters $\sigma_1^2, \cdots, \sigma_K^2$ correspond to a weighting of the regularization. We can manually set α_v and σ_k^2, or they can be learned as follows:

Like in alternating least squares, for α and σ^2, if we fix Θ, we take the derivative respect α and σ^2 and set to zero respectively. Solving the equations, we can get

$$\alpha_v^{-1} = \frac{1}{M_v} \sum_{(v,\mathbf{i},r) \in \mathcal{D}} (\sum_{d=1}^{D} \prod_{k \in S_v} \theta_{ki_kd} - r)^2. \tag{4.4}$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{j=1}^{N_k} \theta_{kj}^T \theta_{kj}. \tag{4.5}$$

Eq. 4.4 and Eq. 4.4 suggest an algorithm to learn α and σ^2 as well when Θ is estimated by swiping the observations.

4.4.3 Learning Algorithm

The pseudo-code of our learning algorithm is summarized in Algorithm 2. Given a properly chosen step size sequence $\boldsymbol{\eta} = (\eta_l)_{l \geq 1}$, SGD updates the latent features Θ for which the gradient is non-zero at every step l. Each update can be interpreted as follows. For the m-th training observation, SGD predicts the rating $\bar{r}_m$ and makes a small gradient step $\eta_\ell(r_m - \bar{r}_m)\boldsymbol{\theta}_{ki_j}$ for each entity i_j involved in the observation in order to minimize the future prediction errors for this training observation.

The proposed method can be viewed an extension to PARAFAC tensor factorization [71]. PARAFAC tensor factorization algorithms are also based on gradient descent schemes and are only guaranteed to converge to local minima, but there is no global guarantee of the solution. Note that recently, several authors proposed convex formulations of tensor completion, but their decomposition into low-rank factors (and hence equivalence with the PARAFAC model) is still an open problem[131]. To avoid being trapped in a local minimum, we considered multiple random restarts (we can refer to some experiments with and without multiple random restarts).

The time complexity of updating the latent parameter $\boldsymbol{\theta}_{kn}$ per observation is $O(KD)$. A single pass on the data is of $O(KDM)$. Assuming the maximum number of sweeps is L, then the total time complexity is $O(KDML)$. Hence, since K and D are constants, the time complexity is linear in the number of observations, $O(ML)$. In the experiments, we use early stopping to decide the number of iterations for SGD where an empirically experiment on validation data split (20%) from training data is conducted. For example, on the Flickr data set this occurs after approximately 100 sweeps of the training data.

Algorithm 1 SGD for Multi-Relational Database Factorization

```
INPUT: prior parameters $\sigma_1^2, \sigma_2^2 \dots \sigma_K^2$ and $\alpha_1, \alpha_2 \dots \alpha_V$
OUTPUT: model parameters $\Theta_1, \Theta_2, \dots \Theta_K$
INITIALIZE model parameters $\Theta_1, \Theta_2, \dots \Theta_K$
for $\ell = 1, ..., L$, do
    SHUFFLE the sequence of observations $(v, \mathbf{i}, r)$ at random.
    for all $(v, \mathbf{i}, r) \in \mathcal{D}$ do
        UPDATE all associated $\Theta_{k,t}$ at parallel.
            $$\boldsymbol{\theta}_{kn}^{(\ell+1)} \leftarrow \boldsymbol{\theta}_{kn}^{(\ell)} - \eta_l \nabla \boldsymbol{\theta}_{kn} \mathcal{O}_{(v,\mathbf{i},r)}^{(\ell)},$$
        where $\boldsymbol{\alpha}^{(\ell)}$ is used to calculate $\nabla \boldsymbol{\theta}_{kn} \mathcal{O}_{(v,\mathbf{i},r)}^{(\ell)}$.
    end for
    UPDATE $\alpha_1, \alpha_2 \dots \alpha_V$ according to Eq. (4.4) [optional]
    UPDATE $\sigma_1^2, \sigma_2^2 \dots \sigma_K^2$ according to Eq. (4.5) [optional]
    if the stop criterion is satisfied then
        BREAK
    end if
end for
```

4.5 Bayesian Treatment

The performance of the probabilistic model is tied to the careful tuning of the hyper-parameter when model parameter Θ are estimated by Maximum a Posterior probability (MAP) [118]. When hyper-parameter are not properly tuned, such a point estimation—MAP—is often vulnerable to overfitting, especially when the data is sparse.

Like BPMF [118] to PMF [119], instead of using MAP, an alternative estimation scheme that may avoid these problems is a fully Bayesian treatment, which integrates out all model parameters and hyper-parameters, arriving at a predictive distribution of future observations given observed data. Because this predictive distribution is obtained by averaging all models in the model space specified by the priors, it is less likely to over-fit a given set of observations.

A graphical overview of our entire model is in Figure 4.7, where $R_1, \ldots, R_V$ are the observed relations. $\Theta_1, \ldots, \Theta_K$ are the latent features associated to the K entity types. $\alpha_1, \ldots, \alpha_V$ are the unobserved precisions (inverse variances) associated with the observed relations, and similarly $\mu_1, \mu_2, \ldots, \mu_k$ and $\Lambda_1, \Lambda_2, \ldots, \Lambda_k$ are the unobserved mean and variances associated with latent features. Each component is described below. Like MAP version of our model, this assumption can also be relaxed easily to model other type of generalized linear model such as Poisson or Bernoulli distribution[98] where a more sophisticated inference is necessary, such as Hybrid Monte Carlo method. Here we assume the observations follow Gaussian distribution. For each observation $(v, \mathbf{i}, r) \in \mathcal{D}$, we have

$$r|\Theta_{\mathbf{i}} \sim \mathcal{N}(\langle \mathbf{\Theta_i} \rangle, \alpha_v), \text{ where } (v, \mathbf{i}, r) \in \mathcal{D}$$

The prior distribution for hidden feature Θ is assumed to be Gaussian too, but the mean and the precision matrix (inverse of the covariance matrix) may take arbitrary value:

$$\theta_{kj} \sim \mathcal{N}(\mu_k, \Lambda_k^{-1}), \quad j = 1...N_k$$

The key ingredient of our fully Bayesian treatment is to view the hyper-parameter $\Phi_k \equiv \{\mu_k, \Lambda_k\}$ also as random variable, leading to a predictive distribution for an unobserved rating $(v, \mathbf{i}, \hat{r})$

$$p(\hat{r}|\mathcal{D}) = \int \int p(\hat{r}|\Theta_{\mathbf{i}}, \alpha_v) p(\Theta_{\mathbf{i}}, \alpha, \Phi_{\mathbf{i}}|\mathcal{D}) d\{\Theta_{\mathbf{i}}, \alpha_v\}, d\{\Phi_{\mathbf{i}}\}$$

For convenience, we also define $\Phi_{\mathbf{i}} = \{\Phi_{i_1}, \ldots \Phi_{i_{|S_v|}}\}$. We then need to choose prior distribution for the hyper-parameters. For the Gaussian parameter, we choose the conjugate distribution as priors that facilitate subsequent computation:

$$\begin{aligned} &p(\alpha_v) = \mathcal{W}(\alpha_v|W_0', v_0') \\ &p(\Phi_k) = p(\mu_k|\Lambda_k)p(\Lambda_k) = \mathcal{N}(\mu_0, (\beta_0\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_0, v_0) \end{aligned}$$

Here $\mathcal{W}$ is the wishart distribution of a $D \times D$ random matrix Λ with v_0 degrees of freedom and a $D \times D$ scale W_0:

$$\mathcal{W}(\Lambda|W_0, v_0) = \frac{|\Lambda|^{(v_0-D-1)/2}}{C} \exp\left(-\frac{\mathrm{Tr}(W_0^{-1}\Lambda)}{2}\right)$$

where C is a normalizing constant. There are several parameters in the hyper-priors: $\mu_0, \rho_0, \beta_0, W_0, v_0, W_0', v_0'$, which reflect our prior knowledge about the specific problem and can be treated as constants during training. In fact, Bayesian learning is able to adjust them according to the training data, and varying their values (within in a reasonably large range) has little impact on the final prediction, as often observed in Bayesian estimation procedures [148].

4.6 Inference

One can represent the predictive distribution of the relation value r given observation $(v, \mathbf{i}, r) \in \mathcal{D}$ by marginalizing over model parameters:

$$p(\hat{r}|\mathcal{D}) = \int\int p(\hat{r}|\Theta_{\mathbf{i}}, \alpha_v) p(\Theta_{\mathbf{i}}, \alpha, \Phi_{\mathbf{i}}|\mathcal{D}) d\{\Theta_{\mathbf{i}}, \alpha_v\}, d\{\Phi_{\mathbf{i}}\}$$

Often the exact predictive distribution is intractable, thus one relies on approximate inference such as sampling method based on Markov Chain Monte Carlo (MCMC) [96, 101]. For instance, MCMC can be used to approximate the predictive distribution of Eq. 4.6:

$$p(\hat{r}|\mathcal{D}) = \frac{1}{L}\sum_{l=1}^{L} p(\hat{r}|\Theta_{\mathbf{i}}^{(l)})$$

where the sample $\Theta_{\mathbf{i}}^{(l)}$ is generated by running a Markov chain whose stationary distribution is the posterior distribution over the model parameters and hyperparameter Θ, Φ.

One of the simplest MCMC algorithms is Gibbs sampling [43], which cycles through the latent variables, sampling each one from the conditional distribution given the current values of all other variables. Gibbs sampling is typically used when these conditional distributions can be sampled from easily. In this section we give a detailed derivation for the conditional distributions of model parameters and hyper-parameters, which are required for implementing Gibbs sampling. Note that with our model assumptions, the joint posterior distribution can be factorized as

$$\begin{aligned} p(\Theta, \alpha, \Phi|\mathcal{D}) \propto & \prod_{(v,\mathbf{i},r)\in\mathcal{D}} p(r|\theta_{S_{v1}i_1}, \ldots, \theta_{S_{v|S_v|}i_{|S_v|}}, \alpha_v) \\ & \prod_k [p(\Theta_k|\Phi_k)p(\Phi_k)] \prod_v p(\alpha_v) \end{aligned} \tag{4.6}$$

4.6.1 Hyper-parameters

We start with the derivation of the conditional distributions of the model hyper-parameters. For each v, α_v follows the Wishart distribution. By using the conjugate prior to α_v, we have the conditional distribution of α_v given R_v, Θ following the Wishart distribution:

$$p(\alpha_v|\mathcal{D}_v, \Theta) = \mathcal{W}(\alpha_v|W_0^*, v_0^*) \tag{4.7}$$

where

$$\begin{aligned} v_0^* &= v_0' + |\mathcal{D}_v|, \\ (W_0^*)^{-1} &= W_0'^{-1} + \sum_{(v,\mathbf{i},r)\in\mathcal{D}_v} (r - \langle \mathbf{\Theta_i} \rangle)^2. \end{aligned}$$

Next, we derive the conditional probability for Φ_k. Our graphical model (Fig. 4.7) assumption suggests that it is conditionally independent of all the other parameters given Θ_k. We thus integrate out all the random variables in Eq. 4.6 except Θ_k, and obtain the Gaussian-Wishart distribution:

$$p(\Phi_k|\Theta_k) = N(\mu_k|\mu_0^*, (\beta_0^*\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_0^*, v_0^*), \tag{4.8}$$

where

$$\mu_0^* = \frac{\beta_0 \mu_0 + N_k \bar{\theta}_k}{\beta_0 + N_k}\,,\ \ \beta_0^* = \beta_0 + N_k\,,\ \ v_0^* = v_0 + N_k;$$
$$(W_0^*)^{-1} = W_0^{-1} + N_k \bar{S} + \frac{\beta_0 N_k}{\beta_0 + N_k}(\mu_0 - \bar{\theta}_k)(\mu_0 - \bar{\theta}_k)^T,$$
$$\bar{\theta}_k = \frac{1}{N_k}\sum_{j=1}^{N_k} \theta_{kj}\,,\ \ \bar{S} = \frac{1}{N_k}\sum_{j=1}^{N_k}(\theta_{kj} - \bar{\theta}_k)(\theta_{kj} - \bar{\theta}_k)^T.$$

4.6.2 Model-parameters

The remaining conditional distributions are for model parameters Θ_k, and we describe the derivation of these distributions in this section. According to the graphical model (Fig. 4.7), its conditional distribution factorizes with respect to the individual entities:

$$p(\Theta_k|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k) = \prod_{j=1}^{N_k} p(\theta_{kj}|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k)$$

$$p(\theta_{kj}|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k) = \mathcal{N}(\theta_{kj}|\mu_{kj}^*, (\Lambda_{kj}^*)^{-1}) \tag{4.9}$$

where

$$\begin{aligned}
\mu_{kj}^* &= (\Lambda_{kj}^*)^{-1}(\Lambda_k \mu_k + \sum_{v \in \{v'|k \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, kj \in \mathbf{i}} r Q_{(v,\mathbf{i},r)}) \\
\Lambda_{kj}^* &= \Lambda_k + \sum_{v \in \{v'|k \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, kj \in \mathbf{i}} Q_{(v,\mathbf{i},r)} Q_{(v,\mathbf{i},r)}^T \\
Q_{(v,\mathbf{i},r)} &= \frac{\prod_{n=1}^{|S_v|} \theta_{S_{v,n}, i_n}}{\theta_{kj}}
\end{aligned}$$

4.6.3 Temporal Factor

For the temporal factor which can capture the evolution of global trends , introduced in Section 4.3, a reasonable prior belief is that they change smoothly over time. Let

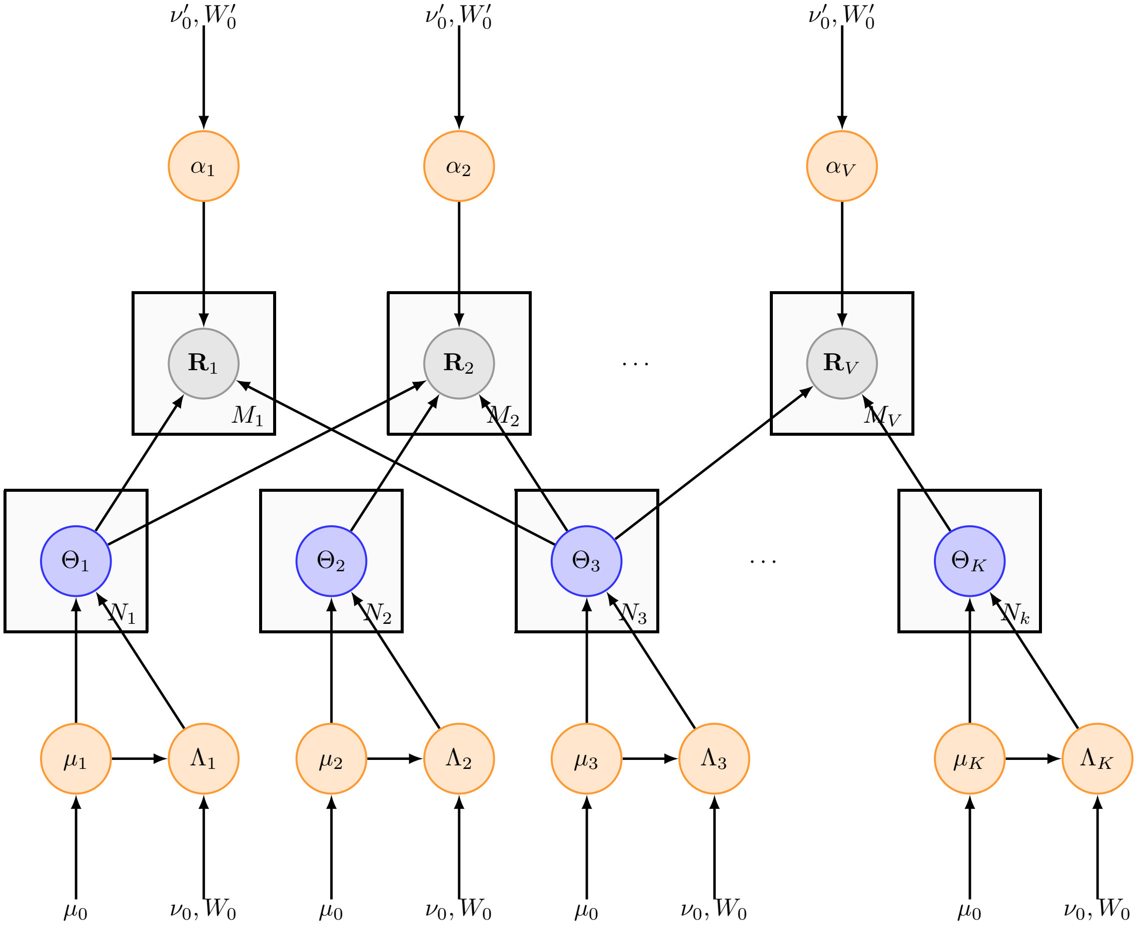


Figure 4.7: Bayesian multi-relational data graphical model.

t be the type of temporal factor among all K types, $t \in \{1, ..., K\}$, then

$$\begin{aligned} \theta_{tj} &\sim \mathcal{N}(\theta_{tj-1}, \Lambda_t^{-1}), \quad j = 2...N_t \\ \theta_{t1} &\sim \mathcal{N}(\mu_t, \Lambda_t^{-1}) \end{aligned}$$

$$p(\Phi_t) = p(\mu_t|\Lambda_t)p(\Lambda_t) = \mathcal{N}(\rho_0, (\beta_0\Lambda_t)^{-1})\mathcal{W}(\Lambda_t|W_0, v_0)$$

Next, we would like to derive the conditional probability for Φ_t.

$$p(\Theta_t|\Phi_t) = \mathcal{N}(\theta_{t1}|\mu_t, \Lambda_t^{-1}) \prod_{j=2}^{N_t} \mathcal{N}(\theta_{tj}|\theta_{tj-1}, \Lambda_t^{-1})$$

$$p(\Phi_t|\Theta_t) \propto p(\Theta_t|\Phi_t)p(\Theta_t)$$

our graphical model assumption suggests that it is conditionally independent of all the other parameters given Θ_t. We thus integrate out all the random variables in Eq. 4.6, except Θ_t. We obtain the Gaussian-Wishart distribution:

$$p(\Phi_t|\Theta_t) = N(\mu_t|\mu_0^*, (\beta_0^*\Lambda_t)^{-1})\mathcal{W}(\Lambda_t|W_0^*, v_0^*), \tag{4.10}$$

where

$$\begin{aligned}\mu_0^* &= \frac{\beta_0\rho_0 + \theta_{t1}}{\beta_0 + 1}\text{,}\ \ \beta_0^* = \beta_0 + 1\text{,}\ \ v_0^* = v_0 + N_t;\\ (W_0^*)^{-1} &= W_0^{-1} + \sum_{j=2}^{N_t}(\theta_{kj} - \theta_{kj-1})(\theta_{kj} - \theta_{kj-1})^T\\ &\qquad + \frac{\beta_0}{1+\beta_0}(\theta_{k1} - \rho_0)(\theta_{k1} - \rho_0)^T,\end{aligned}$$

We consider the temporal features Θ_t. According to the graphical model, its conditional distribution factorizes with respect to individual entities:

$$p(\Theta_t|\mathcal{D}, \Theta_{-t}, \alpha, \Phi_t) = \prod_{j=1}^{N_t} p(\theta_{tj}|\mathcal{D}, \Theta_{-t}, \alpha, \Phi_t)$$

$$p(\theta_{tj}|\mathcal{D}, \Theta_{-t}, \theta_{-j}, \alpha, \Phi_t) = \mathcal{N}(\theta_{tj}|\mu_{tj}^*, (\Lambda_{tj}^*)^{-1})$$

where

for $j = 1$

$$\begin{aligned}
\mu^*_{t1} &= (\Lambda^*_t)^{-1}(\Lambda_t(\mu_t + \theta_{t2}) \\
&\quad + \sum_{v \in \{v' | t \in S_{v'}\}} \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} \alpha_v r Q_{(v,\mathbf{i},r)}) \\
\Lambda^*_{t1} &= 2\Lambda_t + \sum_{v \in \{v' | t \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} Q_{(v,\mathbf{i},r)} Q^T_{(v,\mathbf{i},r)}
\end{aligned}$$

for $j = 2, ..., K - 1$

$$\begin{aligned}
\mu^*_{kj} &= (\Lambda^*_{tj})^{-1}(\Lambda_t(\theta_{tj-1} + \theta_{tj+1}) \\
&\quad + \sum_{v \in \{v' | t \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} r Q_{(v,\mathbf{i},r)}) \\
\Lambda^*_{kj} &= 2\Lambda_t + \sum_{v \in \{v' | t \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} Q_{(v,\mathbf{i},r)} Q^T_{(v,\mathbf{i},r)}
\end{aligned}$$

for $j = K$

$$\begin{aligned}
\mu^*_{kj} &= (\Lambda^*_{tj})^{-1}(\Lambda_t \theta_{tj-1} + \sum_{v \in \{v' | t \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} r Q_{(v,\mathbf{i},r)}) \\
\Lambda^*_{kj} &= \Lambda_t + \sum_{v \in \{v' | t \in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r) \in \mathcal{D}_v, tj \in \mathbf{i}} Q_{(v,\mathbf{i},r)} Q^T_{(v,\mathbf{i},r)} \\
Q_{(v,\mathbf{i},r)} &= \frac{\prod_{n=1}^{|S_v|} \theta_{S_{v,n}, i_n}}{\theta_{kj}}
\end{aligned}$$

Given the conditional probability for model parameters Θ and hyper-parameters Φ and α, the Gibbs sampler algorithms for BPRA are shown in Algorithm 2. For each iteration of sampling Hyper-parameters α, we have swiped the data once to calculate v^*_0 and W^*_0, and the time complexity is $O(KDM)$. For Φ, the time complexity of sampling once is $O(ND)$. Similarly, for Θ, the time complexity of sampling once is $O(KNMD^2)$, where $N = \sum_k N_k$ is the total number of entities. Assuming the maximum number of iterations is L, then the total time complexity is $O(KND^2ML)$.

Hence, since K and D (e.g. $D = 10$) are constants, the time complexity for the worst case is linear $O(NML)$. In realistic, since N is usually much less than M, the time complexity also gets linear $O(ML)$ to the number of observations.

4.7 Experiments

We conduct systematic experiments to evaluate the two versions of our proposed model, named PRA (Probabilistic Relational-data Analysis) and BPRA (Bayesian Probabilistic Relational-data Analysis) on two data sets: Flickr and Bibsonomy[4].

Among all the following three experiments, as there are different kinds of responses (such as binary, term frequency and real value) in our recommendation tasks across multi-contexts, we employ Rooted Mean Square Error (RMSE) as our main measurement for all contexts[5]. In our Bayesian probabilistic relational-data model, we simply set $\mu_0, \rho_0, \beta_0, W_0, v_0, W_0', v_0'$ all equal to one or identity vector and $D = 20$ for the dimension of latent factors, on all three data sets. Our experiments also show that the performance change affected by hyper-prior is very little.

In the following experiments, we compare our methods with the state-of-the-art latent factor methods:

- Salakhutdinov's Probabilistic Matrix Factorization (PMF) [119]: collaborative filtering using probabilistic matrix factorization which treats activities as independent.

- Bayesian Probabilistic Matrix Factorization (BPMF) proposed by Salakhutdinov et al. [118]: the Bayesian version of PMF.

[4] To facilitate replication of experiments, sourcecode and datasets are available upon request.

[5] For some contexts, more specific measurements might be preferred, but RMSE is one evaluation which could be shared by all contexts

- Rendle's Tensor Factorization (TF) [112, 114] which handles high-order relational data for tag prediction and showed prior success in the graph-based tag recommendation task.
- Bayesian Probabilistic Tensor Factorization (BPTF) proposed by Xiong et al. [148] which models temporal collaborative filtering, and whose extension is straightforward to model higher order relational data such as user-tag-comments.
- Collective Matrix Factorization (CMF) [125] handles the 2-order problem involving multiple matrix factorization tasks.

For fair comparison, we let $D = 20$ and set the hyper parameters equal one for the comparison methods.

4.7.1 Flickr: Connecting Comments and Tags

Data set

The Flickr data has been briefly described in Section 4.2. This data set includes 2,866 users, 60,339 tags, 32,752 comment terms and 46,733 items (e.g., images), leading to four relations. The relation $S_1 = (u, t, i)$ indicates that user u tags item i with tag t. The relation $S_2 = (i, f)$ characterizes item i with a 1024-dimension feature vector f extracted according to [107], which are of real numbers. The relation $S_3 = (u_1, u_2)$ encodes a partially observed adjacency matrix representing the explicitly expressed friendship relations among users. For instance, if user u_1 and u_2 are friends, then the value at (u_1, u_2) and (u_2, u_1) are both equal to 1, 0 otherwise. The relation $S_4 = (u, c, i)$ indicates that user u comments on item i using word c, and this relation can be described by term frequency (positive integers).

In the first relation, the problem of interest is tag prediction, that is, to predict tags that users will assign to items. We need to model relation S_1, for which the Flickr data set has a total of 373,125 records with time stamps. The data is partitioned into training and test sets based on the time stamp of April 1st 2010. In total, there are 2,613,388 observations for training and 205,880 observations for testing. Note that there are only positive samples of tags available for the Flickr data set, so for each record we sample 50 tags at random as negative examples for training. For the relation user-comment-item, where users could make some comments on a specific item, we try to predict the term frequency in the comments and the data also are split into training and test data set similarly, resulting in 1,366,068 training observations and 341,043 testing observations.

As mentioned above, we also have two more contexts: for image content, we characterize image i by a feature vector f of 1024-dimensional visual features according to Perronnin and Dance [107]; the social context is also comprised of binary typed observations, which contain 1,377,548 training observations and 342,576 test observations.

Analysis of relations and their co-effects

Some explanatory analysis has been presented in Section 4.2. A social tagging system is a coupled higher-order data system and multiple contexts are coupled together. Here, we will show that by using our methods together with Bayesian treatment, predictive accuracy can be mutually improved.

We first compare the two versions of the probabilistic multi-relational data model: PRA for MAP version and BPRA for Bayesian version. In Table 4.3, Context 1 for users tagging items (user-tag-item), Context 2 for item content (item-feature), Context 3 for social interaction (user-user) and Context 4 for users comments on

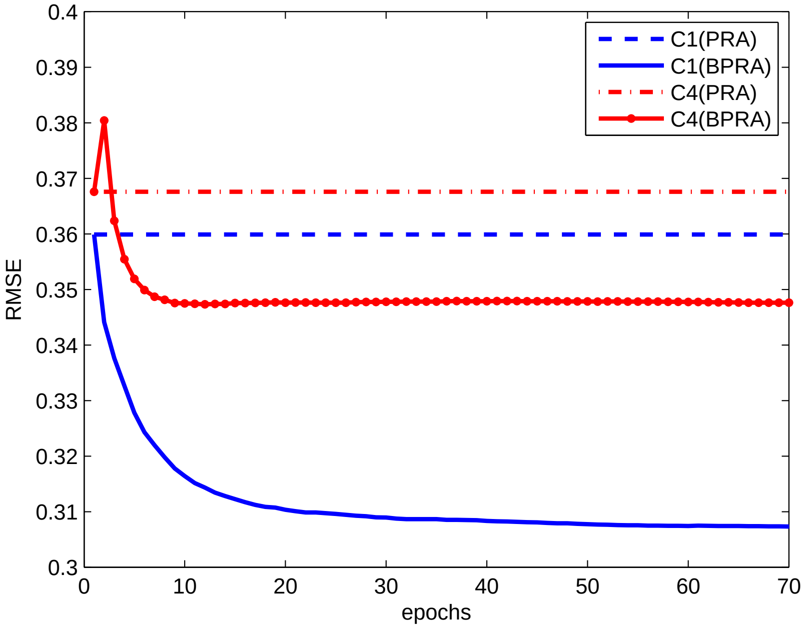


Figure 4.8: BPRA vs. PRA on Flickr.

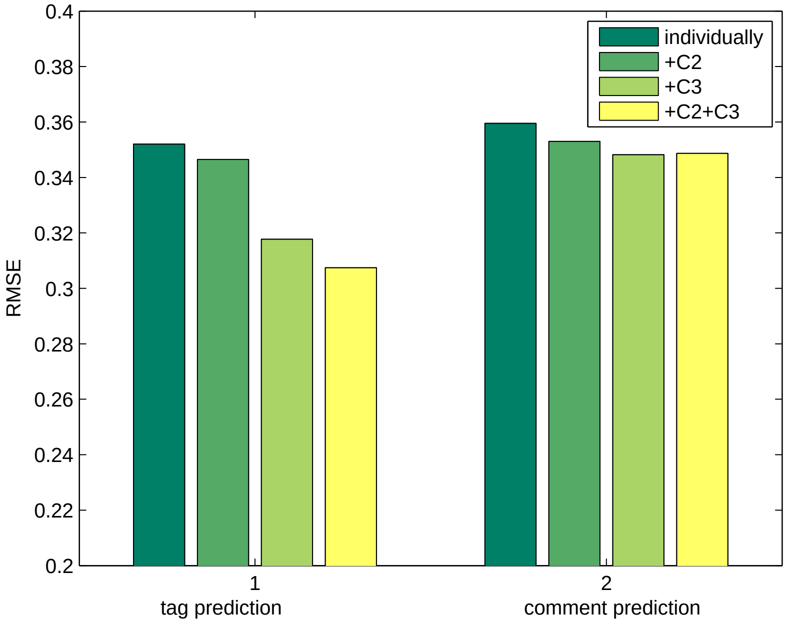


Figure 4.9: RMSE decreasing with more contexts on Flickr.

item (user-item-comments), it can be seen that the Bayesian method clearly outperforms the MAP version (in all scenarios), probably due to the high data sparsity. In Figure 4.8, we show the convergence of our Bayesian model BPRA which starts sampling with parameters based on the results of PRA. We can see that after around 50 epochs, the performance on two relations converge. In the following sections, we will use Bayesian version for analysis and comparison.

Another interesting question is: do coupled relations lead to mutually improved prediction performance? We conduct experiments on modeling different relations with several combinations to study this question. The tasks are described in Section 4.7.1 for different relations and the results are shown in Table 4.3. The first four rows of the table indicate that best performances are achieved for all four relations when modeling them together. The following three rows (rows 5-7) of the table indicate the performance of modeling three relations (C1, C2, C4). Similarly, the results of modeling (C1, C3, C4) and (C1, C4) are shown in the remaining rows. Taking the prediction of Context 1 (C1: user-tag-item) relation as an example: the best performance is 0.3073 in modeling all four relations, 0.3177 in modeling the three relations (C1, C3, C4), and degrades to 0.3465 when only modeling the relation (C1, C4) together. Comparable results for comment prediction are also shown in Figure 4.9.

Comparison with existing methods

We report the evaluation of our models together with comparisons to state-of-the-art approaches introduced earlier. Bayesian Probabilistic Matrix Factorization and its Bayesian treatment are popular methods and have shown success in traditional collaborative filtering. In our experiments with binary contexts, we compare our methods with PMF and BPMF. Since TF and BPTF can model the tag prediction and comment prediction tasks, we compare our methods with them in such higher-order contexts.

We summarize the results in Table 4.3. While Section 4.2 showed that in our three data sets, over 90% of real-world cases are cold start problems and the Graph-based methods (such as Hotho's Folkrank and Rendle's tensor factorization) will not work on such cases, we still compare to the state of the art method—tensor factorization. The results show that Rendle's TF performs the worst in tag prediction, because it only models a single relation without encoding external information of items. Intuitively the external information of items (e.g., comments, features) is more critical to the tag prediction task. This result agrees with [159]. For the cold start problem, the external information of items are essential for tag prediction because the items do not exist in the training data.

In comment prediction context, we see similar results; tensor factorization performs the worst because of lack of external information and data sparsity. Xiong's method—Bayesian Tensor Factorization—is much better, but our methods still achieve the best performance. In both tag and comment prediction, the experiments show that in such a real-world case, tensor factorization is insufficient and Bayesian treatment on tensor factorization can improve performance significantly because of the data sparsity. We also note that with more information in the model, the performance of our approach improves, e.g., with social relation information (C3), we can see that both tag and comment prediction improves.

Overall, we can see that for all methods, Bayesian versions always outperform the MAP version respectively, due to the sparsity of the data. Our model outperforms all four recent nontrivial methods—PMF, TF, BPMF, BPTF in the comments context, social network context and tag context. We also notice that in the item feature relation, our model is slightly worse than BPMF. That is because in our model, the parameter estimation is based on the likelihood for all relations.

Table 4.3: RMSE of 4 relations for Flickr data set.

	BPRA	PRA	PMF	BPMF	TF	BPTF
C1	**0.3073**	0.3599	N/A	N/A	0.8226	0.3520
C2	0.9215	0.9627	0.9913	**0.9004**	N/A	N/A
C3	**0.1828**	0.2053	0.1841	0.1878	N/A	N/A
C4	**0.3478**	0.3676	N/A	N/A	0.4185	0.3593
C1	0.3449	0.4450	N/A	N/A	0.8226	0.3520
C2	0.9198	0.9630	0.9913	0.9004	N/A	N/A
C4	0.3516	0.3681	N/A	N/A	0.4185	0.3593
C1	0.3177	0.3984	N/A	N/A	0.8226	0.3520
C3	0.1858	0.2298	0.1841	0.1878	N/A	N/A
C4	0.3482	0.4241	N/A	N/A	0.4185	0.3593
C1	0.3465	0.7843	N/A	N/A	0.8226	0.3520
C4	0.3530	0.3656	N/A	N/A	0.4185	0.3593

4.7.2 Bibsonomy: Incorporate Content for Tag Prediction

The second data set used to evaluate our model is Bibsonomy—the bookmark data set of the ECML-PKDD'09 Discovery Challenge. This data set involves 2,679 users, 263,004 items, 56,424 tags, 262,336 posts and 1,401,104 records. Clearly, this is also a very sparse data set, whose density is only 3.52×10^{-8}. Each post is associated with a time stamp, and each item contains textual content. In this experiment, we will show that the single graph-based model cannot work in the real world (where the data set is split by time stamp). By incorporating content into the model, prediction accuracy can be significantly improved. To generate a descriptor for the items, we first use the bag-of-words language model and then use Latent Dirichlet Allocation [15] to produce a latent factor for each item. There are only two relations for this data set: $S_1 = (u, t, i)$, where user u tag item i with tag t, and $S_2 = (i, f)$, where each item i is described by a 100-dimensional feature f. To model S_1, we use a time stamp of August 1st 2008 to distinguish training and testing sets with 7,214,426 and 1,585,179 observations respectively.

Table 4.4: RMSE on the Bibsonomy data set.

	BPRA	PRA	PMF	BPMF	TF	BPTF
C1	**.3097**	0.3484	N/A	N/A	1.0409	0.3455
C2	**1.0088**	1.0118	1.7387	1.1025	N/A	N/A

We show the results for Bibsonomy in Table 4.4 where Relation 1 is for users tagging items (user, tag, item) and Relation 2 is for item content (item, feature). At first, we compare the two versions of our model: BPRA is still clearly much better than PRA, benefiting from handling sparse data well. Similarly, in Figure 4.10, we show the convergence of our Bayesian model BPRA which starts sampling with parameters based on the results of PRA. We can see that after around 50 epochs, performance converges. The convergence in Bibsonomy experiments is consistent with our Flickr experiments. We also compare our methods with the baselines. Similarly, BPMF and BPTF outperform PMF and TF respectively. The experiments on this data set also verify the need of employing Bayesian treatment in social relational data.

TF almost fails to solve the task specified by (user-tag-item) relation without item external information, because as we have shown in Section 4.2, most items in a tagging log are new items. The results of our model are consistent with the Flickr data: our model noticeably decreases the RMSE for the tag prediction task. The performance for both relations can lead to significant improvements: 0.3097 in the (user-tag-item) relation and 1.0088 in the (item-feature) relation respectively. This also confirms that the two contexts can mutually reinforce the performance of the model. Overall, like in Flickr experiments, our Bayesian model noticeably outperforms all other methods in the Bibsonomy data set.

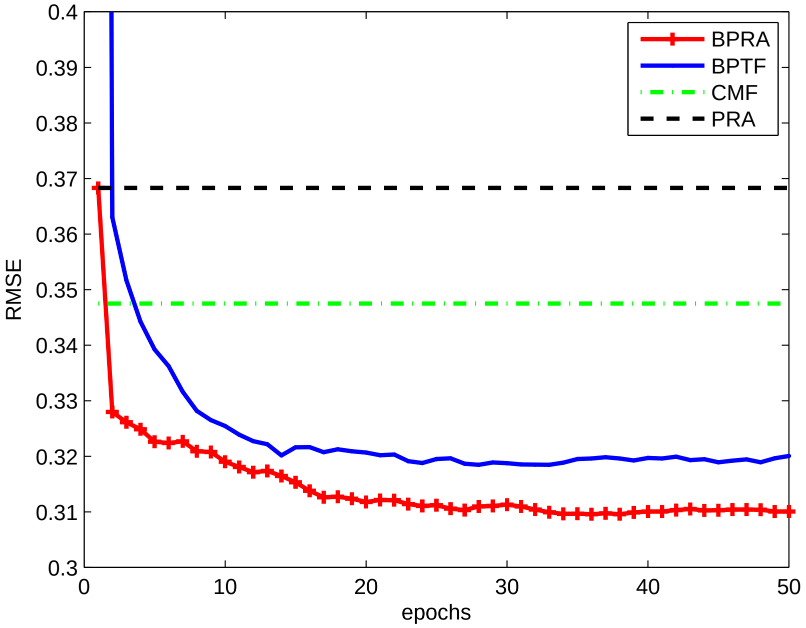


Figure 4.10: BPRA Convergence check in Bibsonomy.

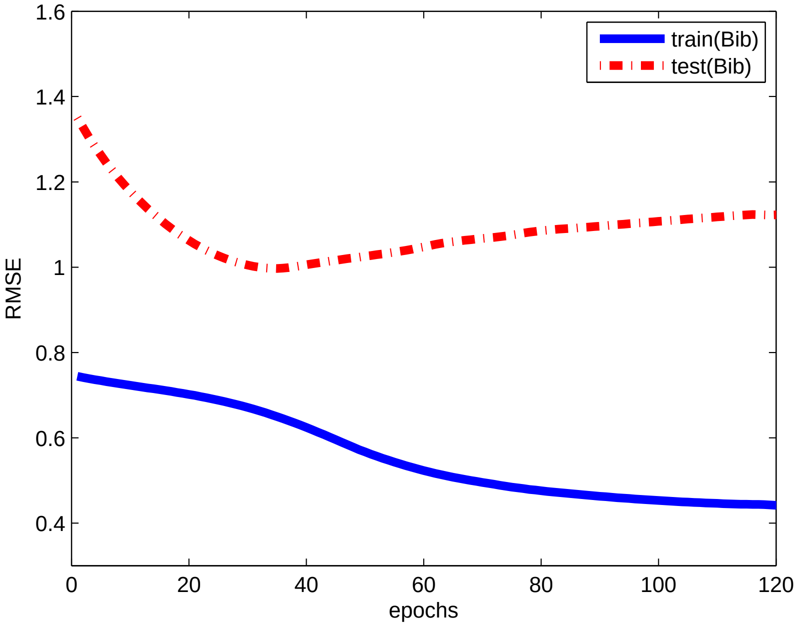


Figure 4.11: PRA Convergence check in Bibsonomy.

4.7.3 MovieLens: Temporal Factor and External Information

Data set

In traditional movie recommendation, Xiong et al. propose Bayesian probabilistic tensor factorization to model temporal information[148]. Here, we will show BPRA outperform Xiong's methods. This experiment is based on a public dataset—MovieLens[6] data set, which consists of one million ratings from 6,000 users on 4,000 movies with time stamps between April, 2000 and February, 2003. The temporal factor is based on Month, and so the ratings fall into 35 months. The ratings are integer scores ranging from 1 to 5. To these ratings is associated user demographic information (e.g., age, gender, occupation, etc.), as well as movie information (e.g., movie title, release date, genres, etc.). We model the user features as the age, the gender, and the occupation, and only consider the genre to describe movies. The ages are partitioned into 7 groups: under 18, 18 to 24, 25 to 34, 34 to 44, 45 to 49,50 to 55, and over 56. The age information is encoded by 7-dimensional binary vector with a single entry being equal to 1 to indicate which group the user belongs to. The gender, user occupation and movie genre are also represented by binary vectors.

Then we have three relations, the first is movie rating prediction for the relation (user, movie, time) and the second relation is (user, profile) and the third one is (movie, genre). For the first relation, we randomly select 100,000 ratings as training data and the rest is test data. For the second and third relation, we randomly select one out of every ten entries for testing and use the rest as training. Our objective is to predict ratings, but also to predict unobserved user features and unobserved movie genres.

[6] http://www.grouplens.org/node/73

Table 4.5: RMSE on the MovieLens data set.

	BPRA	PRA	PMF	BPMF	TF	BPTF	CMF
C1	**0.9258**	1.7234	1.399	0.9468	1.7798	0.9394	1.3451
C2	**0.1877**	0.267	0.266	0.1865	N/A	N/A	0.2673
C3	**0.2387**	0.292	0.3463	0.2405	N/A	N/A	0.2752

Experimental results

The results are shown in Table 4.5 where Relation 1 is for (user, tag, item), Relation 2 is for (user, profile) and Relation 3 is for (movie, genre). It shows that our methods clearly outperform all other methods on all three relations. We also conduct a temporal analysis: for each month's data, we test the RMSE and the results are shown in Fig 4.12 where the green bars are for monthly number of rating (left y-axis) and lines are for monthly RMSE (right y-axis). We can see that for all monthly data, our methods can get better results than BPMF. Comparing with BPTF, only at time 26 and 30, the performance of BPTF is better, but that time, the test rating is very few.

We now consider the question of when the training data become sparser, what is the change of the error for BPRA and PRA. Fig 4.13 shows the results. As the fraction of training data changes from 90% to 10%, RMSE of PRA grows significantly while the RMSE of BPRA almost stays the same. This shows that the Bayesian treatment is more robust and able to handle sparse relational data effectively. With the fact that social network high order relational data are usually very sparse, full Bayesian treatment is clearly necessary.

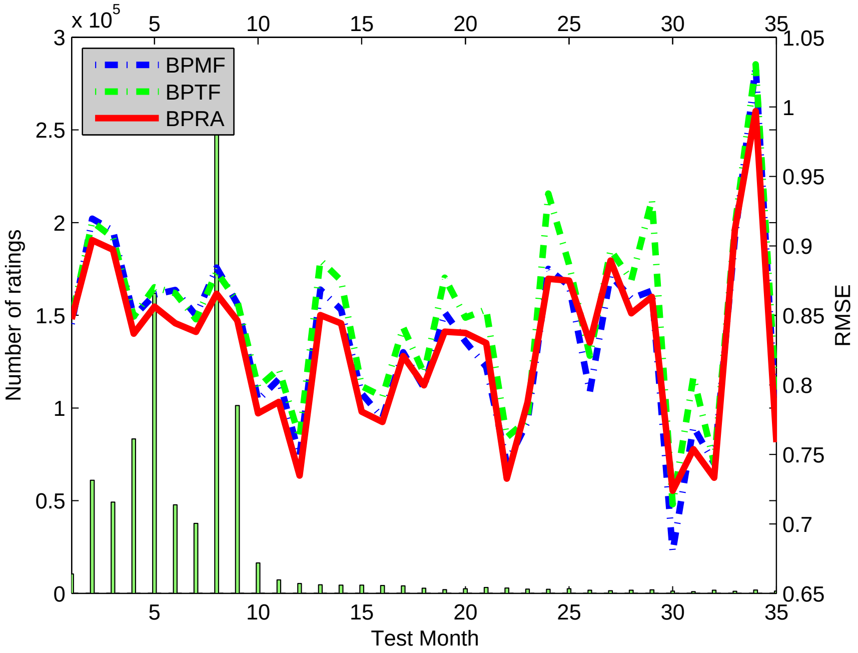


Figure 4.12: MovieLens Monthly Experiments.

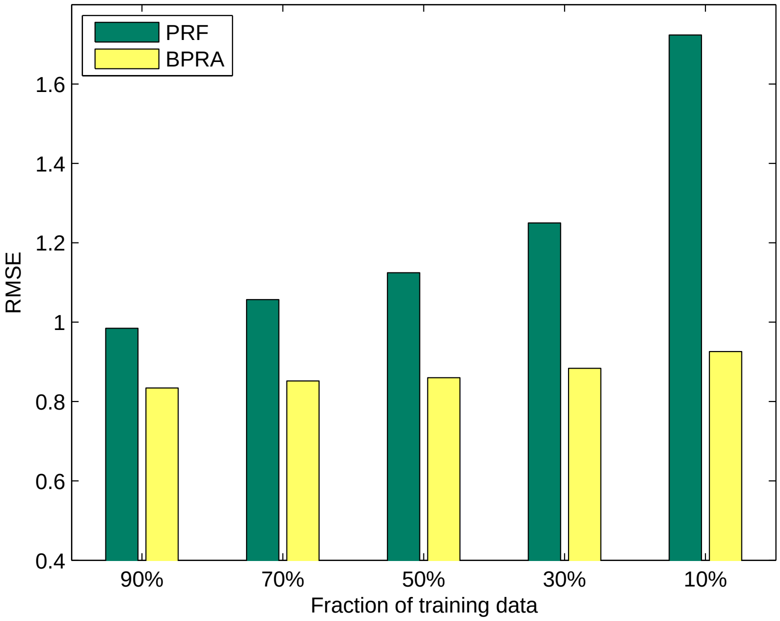


Figure 4.13: Test on different fraction of training data.

4.8 Summary

In this chapter, we studied social media relations involving high order interactions, sparsity and coupling of data across contexts. Our experiments show that in social media, there exist three problems and challenges: coupled high order interaction, data sparsity, and cold start. To make better predictions for a user in different contexts, we tackled these problems by using a generalized latent factor model and Bayesian treatment. For performance evaluation, we tested on three real-world data sets from two domains. In social tagging systems, the user-comment-item and user-tag-item can be mutually inferred based on common latent factors and thus improve prediction performance, which has not been explored previously. In traditional collaborative filtering, we investigated the combination of temporal information, external information and user-item interaction. Our novel latent factor model could handle multiple activities, such as commenting within tagging systems and could do so simultaneously and demonstrate superiority over state-of-the-art methods [119, 125, 148]. Our experiments also show the advantage of employing a fully Bayesian treatment to boost the performance of point estimation when modeling high order relations.

4.9 Bibliographic Notes

The most related work that has been published in the literature is collective matrix factorization from Singh and Gordon [125] which provides a general framework to model multi-relational data, extending many previous approaches on matrix factorization in the presence of additional features. These extensions of matrix factorization/factor analysis tend to be limited to two or three relations to take into account contextual information (such as user-specific and movie-specific features) in a recommender system [172, 1, 3, 91]. For instance, Zhu et al. [172] proposes to make use of links and content for web page classification. More recently, Agarwal and Chen [4] incorporates explicit features of users and items into latent factor models, and Ma et al. [91] proposes to improve recommendation quality based on social regularization. Other similar works [108] and [47] also incorporate side information into the collaborative filtering. However, these methods only model two factor data and cannot be directly used in a social tagging system which is naturally a higher-order system.

More generally co-factorization models [55, 163, 167] make recommendations across multiple contexts or domains. While the framework proposed by Singh and Gordon [125] is fairly general, the key weakness is that it does not enable the handling of high-order relations and it does not use Bayesian estimation to tackle the problem of data sparsity.

Higher order data in social media is often neglected in existing research. For instance, the user-comment-item context is naturally high-order, but existing opinion mining or sentiment analysis studies [34, 63, 85] focus on summarizing and classifying the comments, and discard higher-order relations for user-comment-item. Note that the framework proposed in [153] can handle high-order coupled relations, but it is restricted to problems of small scale, and it is not clear how it can generalize to large scale applications such as social tagging or recommender systems. In contrast, we focus on predicting users' opinion texts (i.e., the terms used) for an item instead of simply classifying the comment contents. Another typical problem for higher-order contexts is social tagging prediction [114, 112, 159, 158]. One possible method for this problem is tensor factorization proposed by Rendle et al. in a recommendation framework [112], which factorizes a higher-order numeric array (a.k.a. data cube) into a core tensor and one factor matrix per dimension. However, Rendle et al.'s method uses only tagging records, and does not address in the cold start problem (i.e., for new items/tags/users), which occurs often in reality [158, 159]. Realizing the necessity of processing the cold start problem, our method combines all other contexts to infer tags on new items/users.

Zheng et al. [169] propose a method on user-activity-location modeling in collaborative filtering. Their methods takes a joint tensor and matrix factorization approach. However, their only study the relationship between a core tensor (third order data) and matrices (second order data) and the mutual effects across tensors are still unknown. Moreover, their point estimation of the model can not handle the sparseness of the higher order data which is a serious problem in the social media data.

We argue that to model the typical two factor data in traditional recommender systems (e.g., as in movie/music recommendations) also requires one to develop models for high-order relations due to the importance of temporal information in analyzing dynamics. Koren [73] models temporal information in collaborative filtering and achieves better performance on movie recommendations. However, this method is based on concept drift instead of modeling the temporal information as the third factor. More recently, Xiong et al. [148] introduce an additional dimension of time to the traditional collaborative filtering problem but this method does not consider the mutual effects across different contexts (e.g the mutual effects between item content and ratings) and cannot handle the cold start problem effectively.

One of the most important points is to understand why Bayesian estimation, also known as model averaging, is beneficial for the estimation. As stated earlier, the core component of the proposed method is the Bayesian Matrix Factorization, which involves the factorization of a single binary relation [118]. In this work, the author showed that averaging over uncertainties significantly improves performances over pointwise estimation, a.k.a. MAP estimation. While the BMF solution was initially computed based on a sampling approach very close to what we proposed, it seems that deterministic approximate inference also gives very similar empirical performance, as shown by the Matchbox model implemented in the Infer.net framework [129].

Existing opinion mining or sentiment analysis studies [34, 63, 85] focus only on summarizing and classifying the comments, and discard higher-order relations for user-comment-item. In contrast, we focus on predicting users' opinion texts on an item instead of simply classifying the comment contents. Recently, Agarwal et al. [6] develop personalized comment recommendation via factor models but they do not predict the content (e.g., term frequency) of personalized comments.

Algorithm 2 Gibbs sampler for Relational Data Factorization

```
INPUT: hyper-prior parameters $\{\mu_0, \rho_0, \beta_0, W_0, v_0, W'_0, v'_0\}$
OUTPUT: model parameters $\{\Theta\}$
Initialize model parameters $\{\Theta^{(1)}\}$
for $l = 1, ..., L$, do
    Sample the hyper-parameters according to Eq. {4.7, 4.8, 4.10}, respectively:
        $\alpha_v^{(l)} \sim p(\alpha_v|\mathcal{D}, \Theta^{(1)})$ where $v = 1, \ldots, V$
        $\Phi_k^{(l)} \sim p(\Phi_k|\mathcal{D}, \Theta_k^{(1)})$ where $k = 1, \ldots, K$
    Sample the model parameters in parallel
    according to Eq. {4.9,4.11}:
    for $k = 1, \ldots, K$ do
        for regular factors
            $\theta_{kj}^{(l+1)} \sim p(\theta_{kj}|\mathcal{D}, \Theta_{1:k-1}^{(l+1)}, \Theta_{k+1:K}^{(l)}, \alpha^{(l)}, \Phi^{(l)})$
                where $j = 1, \ldots, N_k$and $k \neq t$
        for the temporal factors $k = t$
            $\theta_{t1}^{(l+1)} \sim p(\theta_{k1}|\mathcal{D}, \Theta_{1:k-1}^{(l+1)}, \Theta_{k+1:K}^{(l)}, \theta_{t2}^{(l)}, \alpha^{(l)}, \Phi^{(l)})$
            $\theta_{tj}^{(l+1)} \sim p(\theta_{k1}|\mathcal{D}, \Theta_{1:k-1}^{(l+1)}, \Theta_{k+1:K}^{(l)}, \theta_{tj-1}^{(l+1)}, \theta_{tj+1}^{(l)}\alpha^{(l)}, \Phi^{(l)})$
                where $j = 2, \ldots, N_t - 1$
            $\theta_{tj}^{(l+1)} \sim p(\theta_{k1}|\mathcal{D}, \Theta_{1:k-1}^{(l+1)}, \Theta_{k+1:K}^{(l)}, \theta_{tj-1}^{(l+1)}, \alpha^{(l)}, \Phi^{(l)})$
                where $j = N_t$
    end for
end for
```

Chapter 5

Understanding User Click Behaviors in Sponsored Search

In this chapter, we study user click behaviors in sponsored search. We start with the problem of predicting user click-through-rates (CTR) in the sponsored search and then investigate the click yields of a group of ads displayed together. Our extensive experiments on a large-scale real-world dataset from a commercial search engine show that we achieve significant improvement by solving the sponsored search problem from the new perspective.

5.1 Introduction

Sponsored search is *the* major business for today's commercial search engines. Pay-per-click (PPC) is an advertising model that has been adopted by sponsored search markets. Under the PPC model, advertisers are charged when their advertisements are clicked by search engine users. More clicks bring more revenue to search engine companies [18], which has triggered research into related issues [13, 106]. However, since showing too many ads on search result pages may hurt the user experience, search engine companies have to be conservative in the number of ads shown and try to display those that earn the most money which is a function of click-through rate (e.g., bid×CTR).

The current ad-delivery strategy is a two-step approach, which first predicts individual ad click-through-rates (CTR) when user issued a query and then selects the ads with top predicted CTRs. Therefore, estimating the click-through rate of a search ad is an important problem for commercial search engines.

Following this strategy, we first analyze several factors influencing the CTR from the perspective of context, including the number of displayed ads, the content of the ads, the relationship between the query and ads, and the mutual influences between ads. We then propose a novel Context-Aware Click Model for sponsored search. Our extensive experiments on a large-scale real-world dataset show that our methods can outperform state-of-the-art methods and make a noticeable improvement in both the Description Oriented Task and the Prediction Oriented Task.

However, this two-step strategy is suboptimal. The learning problem is focused on predicting CTR rather than click yield, which is the ultimate objective. We thus further challenge the traditional approach and propose a novel framework that can directly predict click yields for lists of ads when issued a query. Our approach addresses a number of challenges for this problem, including depth effects, interactive influence, and cold start. Moreover, to best leverage textual features and handle the sparseness in textual information, we embed a topic coding model into our framework to learn the topical information of short text for ads. Our extensive experiments on a large-scale real-world dataset from a commercial search engine show that we achieve significant improvement by solving the sponsored search problem from the new perspective. Our novel click-yield methods noticeably outperform existing approaches, including our context-aware approach.

The chapter is organized as follows: we study the contextual factors which may have influence on user click behaviors in Section 5.2. Then, we revisit the motivation and challenge the traditional approach by directly estimate the ad group performance in Section 5.3. Section 5.4 summarizes the chapter. Section 5.5 discusses related work.

5.2 Predicting Click-Through-Rate in Sponsored Search

To better understand the challenges and difficulty of estimating click-through rate in sponsored search, examine the search query "windows 8" in Figure 5.1. We see that when a user issues a query, both a list of web pages (organic search) and a list of ads (sponsored search) are displayed. Clearly, to estimate the click-through rate of the target ad (say the ad on top position in Figure 5.1), there are many visible factors: the query and the query's content (e.g., term frequency), the target ad and its content (e.g., term frequency), the adjacent ads (ads in position 2,3,4 in this example) and their content, and ad depth (the total number of the ads displayed). Behind these visible factors, we may also have some hidden interaction across the visible factors, for example, the interaction between the target ad and query, and the interaction between the target and the adjacent ads. We call all these factors the **context** of the target ad.[1]

The problem of CTR prediction in sponsored search is challenging for several reasons. As previous studies have shown, user clicks contain various biases [66, 24, 38, 31], which is both true in organic search and sponsored search [173]. But

[1] There may also be interactive influences between sponsored search and organic search results [32]. In this work, we focus on the contextual factors of sponsored search.

unlike organic search, sponsored search has additional biases and characteristics. For example, many people tend to skip ads and they are reluctant to click ads even if they are related to the query. Previous Cascade Click Models [31] assume that the top position is certainly examined by user ($P(E_1) = 1$) and is thus not suitable for modeling user click behavior in sponsored search. Intuitively, the probability of a user examining the sponsored search results is related to two factors: the depth of the ads (the number of ads displayed) and the commercial intent of a query. For the depth of the ads, the more ads that are shown at the same time, the larger the ad area is, the more attention from users is attracted. However, the search engine usually should not simply show too many ads due to the poor user experience. Moreover, with too many ads displayed together, the overall effect on the click-through rate may not be positive. For the second feature, we notice that for some queries with a high commercial intent, ads usually are more likely to be interesting to users, while for other queries, showing ads may annoy users. This indicates that the examination probability is likely to be query dependent. The effects of both factors on examination and click-through rate are still not clear and have not been systematically investigated.

Another key factor is interactive influence between the target ad and its context (e.g., query and adjacent ads). The interactive influence between ad and query has been studied by Shen et al. [123], but it only considers the query and ignores other factors in the context. Previous research into mutual relationship between similar ads [147] shows that there exists a mutually exclusive influence between similar ads. However, in general, the mutual relationship between ads is not necessarily exclusive and negative, but could also be positive [150]. Unfortunately, the relational click model [147] fails to model positive effects between the ads. Xu et al. [150] study the cases of the relationship between two ads displayed together, but the case of more ads (e.g., 3 or 4 ads) is still unknown. No previous research has investigated the interaction between the target ad and its context, including both query and adjacent ads.

Cold starts also occur frequently in the prediction of click-through rate in sponsored search [116, 140]. Search engines usually deliver at most four ads in a search engine results page (SERP) and the CTR of ads is significantly lower than organic search results. Both clicks and impressions are limited for most ads. Previous models [52, 31, 38, 87, 86] were essentially graph-based models; if the ads or queries do not exist in the training data, these methods will not work.

Due to the above challenges, it is not trivial to model user clicks in sponsored search and predict ad CTRs. Recent advances in click models have proven to be an attractive method for representing user preferences based on user clicks [123, 27]. However, most of the models were designed specifically for organic search. Although some could be used for sponsored search [173, 123], they fail to explore the specific characteristics of sponsored search (e.g., depth, interactive influence). In this section, we start with data analysis to understand the user clicks in sponsored search. We investigate a number of contextual factors including ad depth, query diversity, and ad interaction. To the best of our knowledge, this is the first work to examine the effects of the above contextual factors in sponsored search together. Our experiments show that these contextual factors play important roles in understanding and predicting user clicks. Based on our analysis, we propose a novel context-aware click model for sponsored search. Our model can handle the issues we have discussed above in a principled way. It models specific biases as well as context information in sponsored search. We conduct extensive experiments on a real world dataset. Experimental results show that our model can achieve the best performance compared to three state-of-the-art baseline methods.

We summarize our contributions as follows,

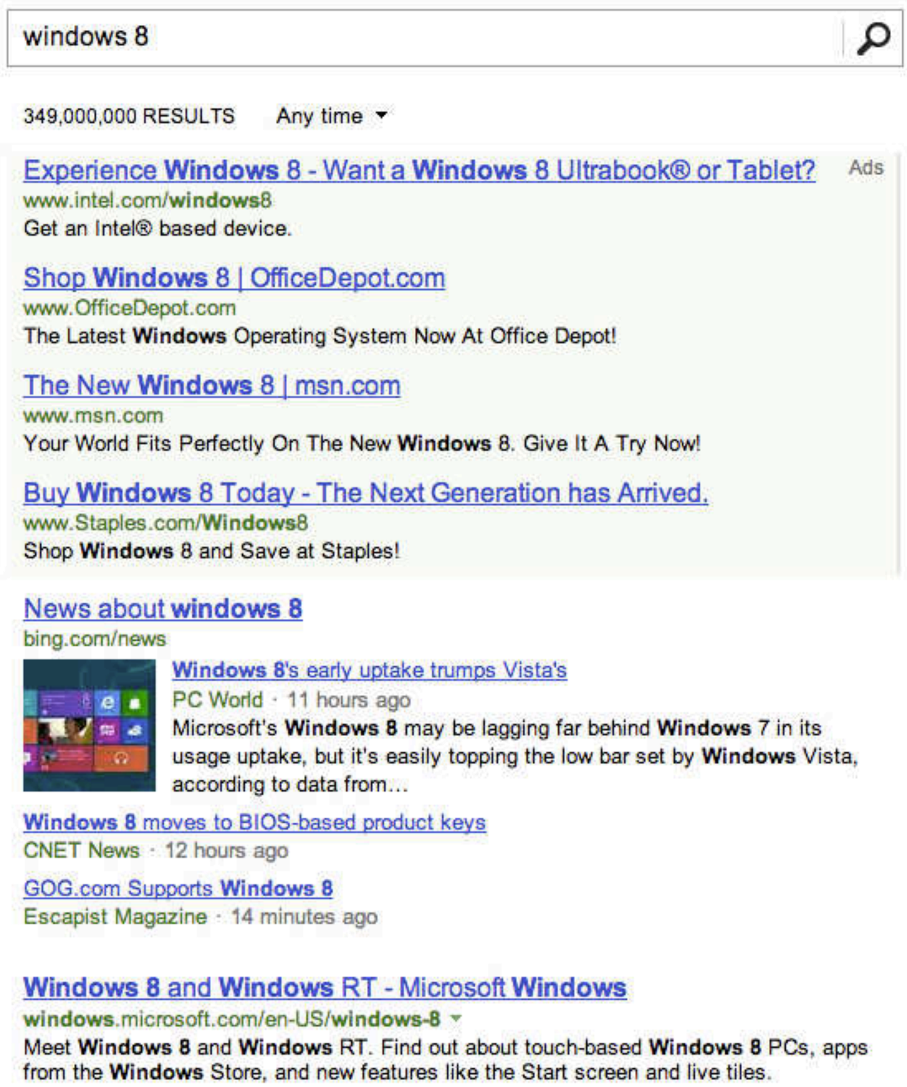
- We provide the first investigation of a combination of factors of ad context, including depth, query influence and ad interaction in sponsored search.
- We propose a Context-Aware Click Model which incorporates ad context factors, and is specifically designed for sponsored search.
- Our experiments, based on a large-scale real-world dataset, show our methods make noticeable improvements and outperform state-of-the-art methods.

5.2.1 Context Analysis

In this subsection, we introduce some explanatory experiments to verify our intuitions and conjectures in sponsored search. We collected data from a commercial search engine in the U.S. English market in April 2012. In total, there are 127,230,609 query sessions, where a session consists of an input query, a list of sponsored search results and a list of clicked position with time stamps. Since click data is noisy, we filter out the low frequency queries and ads to remove noise and use the high quality click through rate for evaluation.

Depth Analysis

We first try to answer the following question: is the number of ads related to ad CTR? Table 5.1 shows the average CTR at different positions and depth. From the bottom line, we can see that the relationship between CTR and position is consistent with previous studies [116, 31]. Ads in the top positions usually receive higher CTR. We can also make some interesting observations about the relationship between CTR and depth: the CTR at the same position increases along with the depth. That is, as Figure 5.2 shows, the CTR of position 1 and depth 4 is significantly larger than the CTR of position 1 and depth 3, and CTR of position 1 and depth 3 is also



windows 8

349,000,000 RESULTS Any time ▾

[Experience **Windows** 8 - Want a **Windows** 8 Ultrabook® or Tablet?](#) Ads
www.intel.com/**windows**8
Get an Intel® based device.

[Shop **Windows** 8 | OfficeDepot.com](#)
www.OfficeDepot.com
The Latest **Windows** Operating System Now At Office Depot!

[The New **Windows** 8 | msn.com](#)
www.msn.com
Your World Fits Perfectly On The New **Windows** 8. Give It A Try Now!

[Buy **Windows** 8 Today - The Next Generation has Arrived.](#)
www.Staples.com/**Windows**8
Shop **Windows** 8 and Save at Staples!

[News about **windows 8**](#)
bing.com/news

[**Windows 8**'s early uptake trumps Vista's](#)
PC World · 11 hours ago
Microsoft's **Windows 8** may be lagging far behind **Windows** 7 in its usage uptake, but it's easily topping the low bar set by **Windows** Vista, according to data from…

[**Windows 8** moves to BIOS-based product keys](#)
CNET News · 12 hours ago

[GOG.com Supports **Windows 8**](#)
Escapist Magazine · 14 minutes ago

[**Windows 8** and **Windows** RT - Microsoft **Windows**](#)
windows.microsoft.com/en-US/**windows-8** ▾
Meet **Windows 8** and **Windows** RT. Find out about touch-based **Windows 8** PCs, apps from the **Windows** Store, and new features like the Start screen and live tiles.

Figure 5.1: Example on query “windows 8”

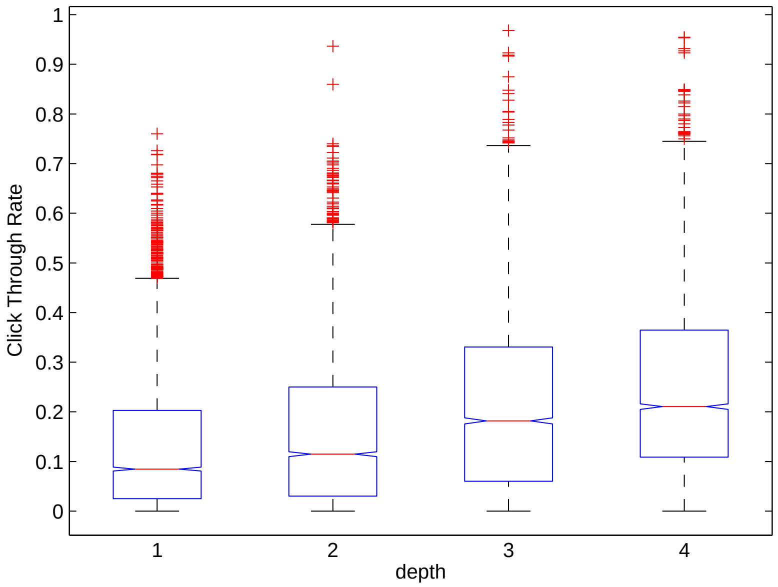


Figure 5.2: CTR on position 1 as a function of depth

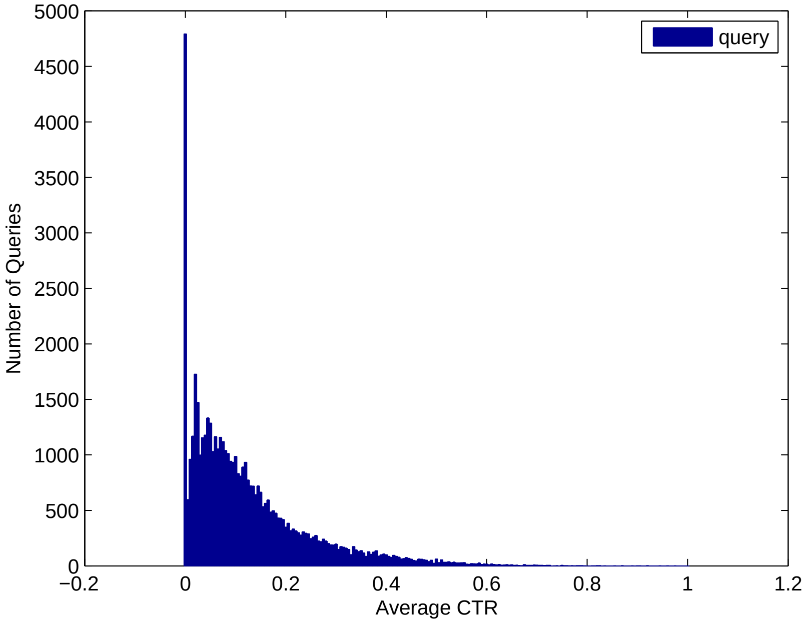


Figure 5.3: Query CTR Distribution

Table 5.1: Average ad CTR at different positions and depths

	pos. 1	pos. 2	pos. 3	pos. 4	Depth Avg.
depth 1	0.1298				0.1298
depth 2	0.1568	0.0450			0.1009
depth 3	0.2158	0.0511	0.0306		0.0992
depth 4	0.2479	0.0711	0.0430	0.0341	0.0990
Pos. Avg.	0.1586	0.0547	0.0382	0.0341	

larger than the CTR of position 1 and depth 2, etc. And these patterns also exist for position 2 and position 3.

We conjecture that these observations, shown in Table 5.1 and Figure 5.2, may be the result of four factors: first, greater ad depth typically means a larger ad area, which attracts more user attention. Therefore, ads in a larger ad area have a higher probability of user examination and higher CTRs should be expected. Second, the probability of user examining higher position might be greater than the lower position. Third, after a user examines all the ads, that user tends to click on the most relevant one. More relevant ads absorb more user clicks. The bottom ads actually play a supporting role, which has positive effects on the top ads. Finally, larger depth may imply higher commercial intent of query and higher relevance of the ads in the top position. To verify the conjectures, in Subsection 5.2.2, we incorporate these factors into the click model and the experiments in subsection 5.2.4 show their contributions on CTR prediction.

We also notice that the average CTR decreases as the depth increases (from the right column of Table 5.1). This indicates that more ads being displayed may dilute the CTR of specific ads. However, note that the total click yield (the CTR summation of all ads for a given depth) becomes larger: for instance, the click yield of depth 4 is 0.099 × 4, which is much larger than depth 1.

Query Analysis

Here we examine the relationship between query and click through rate. In Figure 5.3, we plot the distribution of click through rate for queries. The differences in click through rate across queries are quite large. Some queries have a high average CTR while for some other queries, there are no clicks. That means, in the past month, regardless of the ads shown for such a query, no users clicked on any of them. This large difference in average click-through rate of queries strongly implies that the probability of examining the ad area for specific queries may also be different. In previous methods [31, 52, 53], researchers only modeled these differences for query-ad relevance and for each impression. They assumed that for any query session, the probability of examining the top position was one, that is, $P(E_1) = 1$. Generally, this assumption may be suitable for modeling user click behavior for organic search results because the organic search results are always the examination target when a user issues a query. Unfortunately, the sponsored search results may not always be a target of examination, especially for those queries with low commercial intent. With this in mind, we conclude that the assumption that $P(E_1) = 1$ is not suitable for sponsored search and the probability of examining the ad area not only depends on the depth, but also the query.

In Subsection 5.2.1, we considered the effect of depth on click through rate: From Table 5.1, the average CTR over all positions (also averaged over ads and queries) slightly decreases when the depth increases. If we take a micro-view on queries, this may not be always true. For example, for some queries, the average CTR of the larger depth is greater than of the shorter depth. To find the properties of these queries, here we treat the average CTR of the query as the commercial intent of the query (higher average CTR means higher commercial intent), and investigate the relationship between the depth and the query's commercial intent.

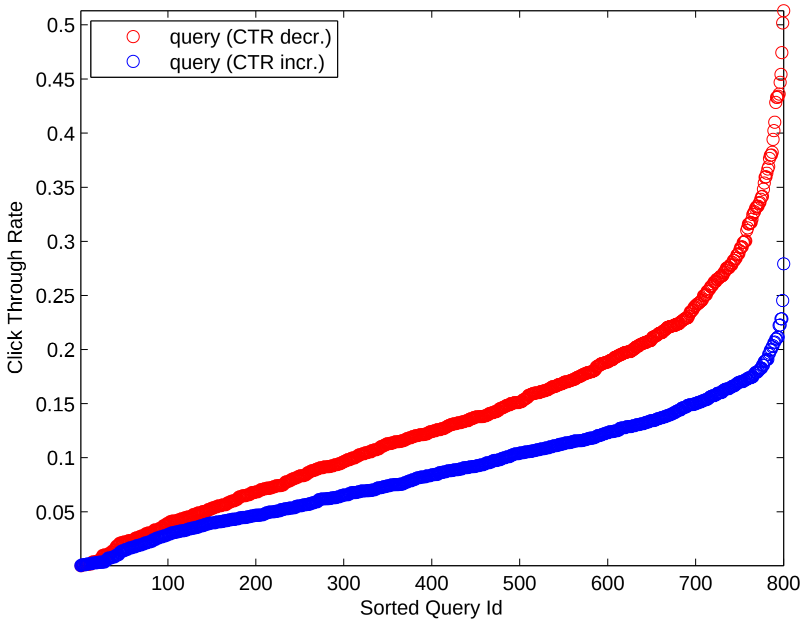


Figure 5.4: Depth changed from 1 to 3

In our experiments, we change the depth from 1 to 3, and collected queries that achieve higher CTRs and queries that achieve lower CTRs. In Figure 5.4, we plot the queries whose average CTRs increase in blue and the queries whose average CTRs decrease in red. We see that queries with an increasing CTR have lower commercial intent while queries with a decreasing CTR have higher commercial intent. We perform a two sample t-test on the two groups of CTRs, and the p-value is 3.4710e-43 which is highly significant. This indicates that for queries with higher commercial intent, a single ad has a higher probability of matching user intent, and the user will more likely click this ad. However, when we show more ads, the average CTR decreases. This is consistent with the analysis by Xiong et al. [147]. Similar ads shown together will have exclusive effects and the CTR can be diluted by adjacent ads. On the other hand, for queries with lower commercial intent, if we show more ads at the same time, we achieve higher average CTRs. We conjecture

Table 5.2: CTRs of ads appearing in different depths

No. of ads	depth 1	depth 2	depth 3	depth 4
588,619	0.0679	0.0736	0.0895	0.1161

that for these queries, the effects across ads are positive rather than exclusive or negative. In [147], the authors neglect to consider and model the positive cases.

Ad Analysis

Now we analyze the CTR from the perspective of ads. For ads that appear at different depths, we calculate their average CTR at different depths (dropping the ads that only appear in one depth). Table 5.2 shows that in our data sets, there are 588,619 ads that appear in at least two different depths. As the depth increases, the CTR (averaged over queries) also grows. This observation is consistent with the results in Subsection 5.2.1. We then further investigate the averaged CTR changes in specific positions and depths for these ads. The results are shown in Table 5.3. Take the first line as an example. There are 464,633 ads that appear at both depth 1 and depth 2. For these ads, if we show them alone, the click through rate was 0.0589. When they are shown at depth 2, their CTRs are 0.0676 and 0.0469 at position 1 and position 2, respectively. Similarly, the results of changing the depth from 1 to 3 and 4 are also shown. We can see that if the ad is in position 1, it achieves a higher CTR than when shown alone, but for position 2 and 3, a lower CTR occurs on these ads. However, we also notice that when the depth changes from 1 to 4, the ad CTR of all four positions at depth 4 are greater than the CTR at depth 1, even for the bottom position.

Next, to find ads with mutual influences, we perform a similarity analysis for ads that are shown together. Figure 5.5 presents the results. The similarity is calculated the same as Xiong et al. [147]. The X-axis indicates the similarity buckets. The

Table 5.3: CTRs of ads appearing in multiple positions

Depth	No. ads	shown alone	pos 1	pos 2	pos 3	pos 4
1→2	464,633	0.0589	0.0676	0.0469		
1→3	117,625	0.0609	0.0862	0.0558	0.0441	
1→4	80,198	0.0547	0.1056	0.0778	0.0628	0.0555

Y-axis measures the difference $CTR_{sim} - CTR_{avg}$. In this box plot, the results are similar to [147]. As the similarity becomes higher, it shows negative effects. However, we also notice that for the majority of cases (similarity between 0 and 0.4), this negative effect is almost trivial. The median shows that the positive influence was also in half of the cases. We cannot only consider the exclusive influence between the targeted ad and the adjacent ads, even for highly similar ads. In Subsection 5.2.4, we will clearly see the improvement of our interaction model, which can model both cases.

5.2.2 Context-Aware Click Model

We have seen some properties of ad context for predicting click through rate in sponsored search. In this subsection, we propose the Context-Aware Click Model for sponsored search, which is designed based on the above analysis and intuition.

Let us first define the notation. The records are presented by $\mathbf{S} = (q_m, \mathbf{a}_m, \mathbf{c}_m)_{m=1}^{M}$. For the single click log $(q, \mathbf{a}, \mathbf{c}) \in \mathbf{S}$, q means the query in the record, $\mathbf{a}$ means the list of ads $\mathbf{a} = \{a_1, a_2, \cdots, a_{|a|}\}$ and $|\mathbf{a}|$ is the ad depth of this record. i is the position of the list of ads $\mathbf{a}$, and $\mathbf{c}$ means the corresponding clicks $\mathbf{c} = \{c_1, c_2, \cdots, c_{|a|}\}$, where $c_i = 1$ for clicked position i, otherwise $c_i = 0$.

In the previous section, we have already seen some hints that for some queries, the probability of users being attracted by a specific number of ads is different. Here we introduce a new variable, $E_{(q,\mathbf{a})}$ to model the event that given a query q and the

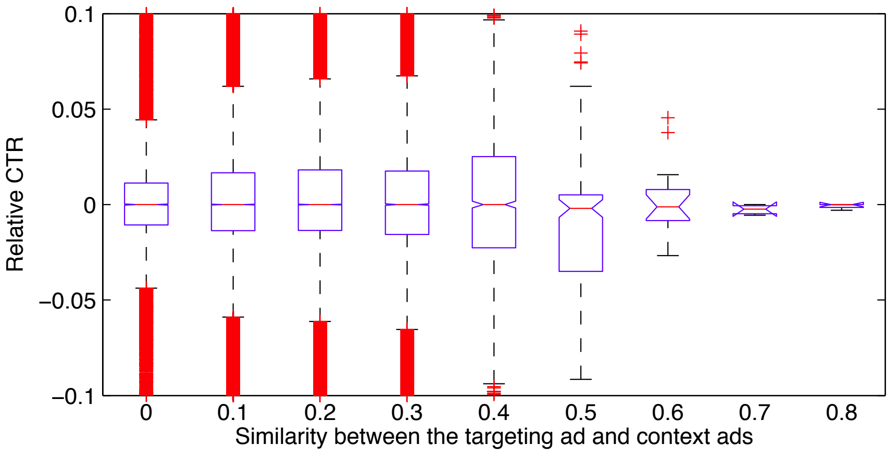


Figure 5.5: ΔCTR against Similarity

displayed ads list $\mathbf{a}$, users will examine the ad area. Then following the classical examination hypothesis, we have the model assumption: for a record $(q, \mathbf{a}, \mathbf{c}) \in \mathbf{S}$,

$$\begin{aligned} c_i = 1 \quad &\Leftrightarrow \quad E_i = 1, E_{(q,\mathbf{a})} = 1, N_{i,(q,\mathbf{a})} = 1 \\ E_{(q,\mathbf{a})} = 0 \quad &\Rightarrow \quad E_i = 0 \\ E_i = 0 \quad &\Rightarrow \quad c_i = 0 \end{aligned}$$

where E_i is the event that the user will examine the ad at position i and $N_{i,(q,\mathbf{a})}$ is the event that the user is interested in the ith ad, given the context $(q, \mathbf{a})$. $N_{i,(q,\mathbf{a})}$ can also be considered as the variable of relevance. For a record $(q, \mathbf{a}, \mathbf{c}) \in \mathbf{S}$, the click is generatoted as

$$\begin{aligned} P(c_i = 1) \quad = \quad & P(c_i = 1 | E_i = 1, E_{(q,\mathbf{a})} = 1) \\ & P(E_i = 1 | E_{(q,\mathbf{a})} = 1) P(E_{(q,\mathbf{a})} = 1) \end{aligned}$$

The probability of examining position i $P(E_i = 1 | E_{(q,\mathbf{a})} = 1)$ is well studied and can be formulized as the existing versatile position bias model, such as cascade models [52, 53, 31] and the examination hypothesis [38, 128]. In this section, we will not focus on investigating position bias and employed Position Model [128, 123]. We will however study the context-related hypotheses: the examination model for ad area $P(E_{(q,\mathbf{a})})$ and the relevance model $P(c_i = 1|E_i = 1, E_{(q,\mathbf{a})} = 1)$.

Examination Models

We now discuss several possible types of examination hypothesis in modeling sponsored search clicks.

Constant Examination (CE). A simple method for modeling the probability of examining ad area $P(E_{(q,\mathbf{a})})$ is assuming $P(E_{(q,\mathbf{a})})$ is independent and does not depend on either query or ad depth. Then we have $P(E_{(q,\mathbf{a})} = 1) = \delta$, where δ is a corpus level parameter and shared by all queries and ad lists. Under this assumption, we might notice that $P(E_i = 1|E_{(q,\mathbf{a})} = 1)P(E_{(q,\mathbf{a})} = 1) = \delta \cdot \gamma_i$. Clearly, the two parameters can be merged. That is, the effects of the new parameter δ can be absorbed by γ_i without changing prediction performance. Actually, this model is equivalent to the position model [128, 123].

$$P(E_i = 1, E_{(q,\mathbf{a})} = 1) = \beta_i \equiv \delta \cdot \gamma_i$$

Depth Dependent Examination (DDE). In Subsection 5.2.1, we analyzed the effects of ad depth on click through rate. As the analysis showed, a longer ad list displayed usually meant there was a higher probability of the ad area being examined. Following the intuition that the probability of examining the ad area is highly related to the depth of the ad list, we assume that the probability of examining the ad area $P(E_{(q,\mathbf{a})} = 1)$ is a depth dependent variable. That is $P(E_{(q,\mathbf{a})} = 1) = \delta_{|\mathbf{a}|}$, where $\delta_{|\mathbf{a}|}$ is discriminated by each different depth. Then we have the joint probability: $P(E_i = 1|E_{(q,\mathbf{a})} = 1)P(E_{(q,\mathbf{a})} = 1) = \delta_{|\mathbf{a}|} \cdot \gamma_i$. Alternatively, as in Constant Examination, we can also merge the two parameters.

$$P(E_i = 1, E_{(q,\mathbf{a})} = 1) = \beta_{i|\mathbf{a}|} \sim \delta_{|\mathbf{a}|} \cdot \gamma_i$$

The merged model, which can be considered as a depth-position bias model with parameter $\beta_{i|\mathbf{a}|}$, actually has freer properties than the original one. If d is the maximum depth, then the model with merged γ,δ will have a total of $\frac{d\times(d+1)}{2}$ parameters rather than $d \times 2$ parameters of the original depth dependent model. However, this depth dependent examination model still cannot capture the query effects on the probability of examining the ad area.

Query-Depth Dependent Examination (QDDE). As in Subsection 5.2.1, we discussed that like depth effects, for queries with different commercial intent, the ads area may also receive different examining probability. Following this intuition, we propose a more delicate model, where we assume the $P(E_{(q,\mathbf{a})} = 1)$ is related to both queries and the depth of ad list. Similarly, $P(E_i = 1|E_{(q,\mathbf{a})} = 1)P(E_{(q,\mathbf{a})} = 1) = \delta_{q|\mathbf{a}|} \cdot \gamma_i$. Alternatively, as in the above subsection, we could also merge the two parameters.

$$P(E_i = 1, E_{(q,\mathbf{a})} = 1) = \beta_{iq|\mathbf{a}|} \sim \delta_{q|\mathbf{a}|} \cdot \gamma_i$$

Similarly to the Depth Dependent Examination, the merged model with parameter $\beta_{iq|\mathbf{a}|}$ has freer properties than the original $\delta_{q|\mathbf{a}|} \cdot \gamma_i$. For every query q, we have a depth-position bias model. Let m be the total number of queries and d be the maximum depth. The model with merged γ,δ will in total have $m\frac{d\times(d+1)}{2}$ parameters rather than $m \times d \times 2$ parameters of the original Query-Depth Dependent Examination model. Although the two factors, query and depth are modeled here, one potential problem of this model is overfitting, due to the large number of parameters behind this model and sparseness of the ad data. We will test this method in the following experiments on the real data.

Relevance Models

Here we describe the possible methods for the relevance model $P(c_i = 1|E_i = 1, E_{q|\mathbf{a}|} = 1) = \alpha_{i,(q,\mathbf{a})}$.

Non-Informational Relevance (NIR): As with previous methods [38, 128, 31, 173] for the click model, the relevance model can be trivially set to informational constraint on $\alpha_{i,(q,\mathbf{a})}$, that is,

$$\alpha_{i,(q,\mathbf{a})} \sim \mathcal{U}(0,1)$$

$\alpha_{i,(q,\mathbf{a})}$ can be valued from uniform distribution in the range zero to one, in order to serve as the Bernoulli parameter of $P(c_i = 1|E_i = 1, E_{q|\mathbf{a}|} = 1)$. The major drawbacks of this model are three factors: first, since there is no informational constraints on α, the model will more easily become overfitted to training data and have worse generalization properties, especially for predicting click through rate of the future data. Second, it cannot determine the hidden interactions across query, user, and ads, such as the effect of CTR between ads. Finally, it cannot handle cold starts. When new queries or ads (which do not exist in the training data) appear in the test data, the model will fail to predict CTR.

To overcome the problems in the non-information relevance model, we could add some informational constraints on the Bernoulli parameter α. Let $f_{i,(q,\mathbf{a})}$ be the informational constraints, calculated from multiple resources of the context. By placing Gaussian noise, we have

$$\alpha_{i,(q,\mathbf{a})} \sim \mathcal{N}(\Phi(f_{i,(q,\mathbf{a})}), \sigma^2)$$

where $\alpha_{i,(q,\mathbf{a})}$ follows univariate gaussian distribution with mean $\Phi(f_{i,(q,\mathbf{a})})$ and variance σ^2 and the link function $\Phi(x)$ could be Sigmoid function or Gaussian cumulative distribution function to scale the scale the value $(-\infty, \infty)$ into range $(0, 1)$. Here, we choose Gaussian cumulative distribution function (CDF) $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{x} e^{-t^2/2} dt$.

Next, we will discuss several possible models to tackle the problem of informational constraints $f_{i,(q,\mathbf{a})}$ which can be calculated ad/query features, latent bias or interaction across ads/queries.

Latent Bias (LBM): One straightforward intuition is that whether the ad is clicked or not depends on the average click rate: $f_{i,(q,\mathbf{a})} = \mu_{a_i}$ where μ_{a_i} is the click through rate of the target ad across all contexts. Certainly, this estimation is too coarse and inaccurate, because as we mentioned before, the majority of ads are not clicked and multiple factors of the contexts of the ads are not incorporated. We can extend this basic estimation by incorporating a wider bias from the context: 1) Query bias μ_q, similar to ad bias, μ_q is the click through rate of the target ad across all contexts. 2) Relevance position bias μ_i, usually, the position of the ad in the ad list is also related to the query-ad relevance, the more relevant ads will have relatively more probability of ranking higher. μ_i represents the average click through rate over all contexts. 3) Alternatively, from the context, we may also have ad list $\mu_{\mathbf{a}}$ bias and global average bias μ_0. Then we have

$$f^{(1)}_{i,(q,\mathbf{a})} = \mu_0 + \mu_{a_i} + \mu_q + \mu_i + \mu_{\mathbf{a}}$$

Note that these biases are generally unknown. We treat them as latent variables to be learned from the data set. Compared with Feature Model, which will be discussed in the following sections, whose feature vectors are sometimes difficult to calculate such as graph-based features and content-based features, this model is appealing since to no extra information besides indicators is needed for learning.

Feature Model (FM): Another relevance model is linear estimation, which predicts relevance by a linear combination of features. For a search session, we could collect the features of the ads and their contexts. Given a record $(q, \mathbf{a}, \mathbf{c}) \in \mathbf{S}$ and position i, the feature vector can be extracted from two aspects: **Ad Feature:** Let $\mathbf{x}_{a_i}$ be the feature vector of the target ad a_i, which can be extracted from the terms of the shown title and body of the ad, the history click over expected clicks (COEC) [166] or latent topic distribution of the ad content. **Context Feature:** Let $\mathbf{x}_{(q,\mathbf{a})}$ be the feature vector of the context of the target ad a_i, which can be the terms of query and the titles of the adjacent ads, user profiles, the depth of the ad list and the similarity between the target ad and the query/adjacent ads. A simple linear combination of ad features and context features can be defined as:

$$f^{(2)}_{i,(q,\mathbf{a})} = \mathbf{b}_1^T \mathbf{x}_{a_i} + \mathbf{b}_2^T \mathbf{x}_{(q,\mathbf{a})}$$

b_1 and b_2 are coefficients to be learned from the training set. The model is essentially equivalent to the one where ad features and context features are combined into a single feature vector $\mathbf{x} = [\mathbf{x}_{a_i}^T, \mathbf{x}_{(q,\mathbf{a})}^T]^T$ and a coefficient $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T]^T$. Here, we further place a zero mean Gaussian prior or Laplace prior on the coefficient $\mathbf{b}$, corresponding to the L_1 and L_2 regularization respectively. With Gaussian assumption, we get: $\mathbf{b} \sim \mathcal{N}(0, \sigma_{\mathbf{b}}^2 \mathbf{I})$

Interactive Influence (IM): Although linear models are efficient, they are usually over simplified and cannot capture interaction between the ad and the elements (e.g., query and adjacent ads) of the context. The interaction between the target ad and its adjacent ads are studied by Xiong et al. [147]. However, they only model the interaction between the target ad and adjacent ads which are shown in the same impression and fail to model the interaction between the target ad and the query. Moreover, the interaction between the target ad and adjacent ads in [147] is an exclusive interaction; that is, similar adjacent ads will decrease the CTR of the target ad through similarity weight (non-negative). Their model also fails to handle positive interaction between the target ad and its adjacent ads.

To model the interactions between the target ad and the context, we use two interaction matrices $\mathbf{Q} \in \mathbb{R}^{n \times m}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$, where m is the number of the queries and n is the number of ads. The entry $Q_{a_i q}$ of $\mathbf{Q}$ represents the interaction between the target ad a_i and query q, and the entry A_{a_i,a_j} represents the interaction between the target ad a_i and its adjacent ad a_j.

$$f^{(3)}_{i,(q,\mathbf{a})} = Q_{a_i q} + \sum_{j \neq i, j \leq |\mathbf{a}|} A_{a_i a_j} \tag{5.1}$$

The interaction matrix $\mathbf{Q}$ and $\mathbf{A}$ are unknown and their entries could be either positive or negative. We treat them as latent variables to be learned from the data set. However, the model has two serious problems: the observed pair (a_i, q) and (a_i, a_j) in the training data is extremely sparse, and the majority of entries in the interaction matrix cannot be learned effectively. The second problem is that without any constraints on the interaction matrix, this model may cause overfitting. To avoid these two problems, we place low-rank constraints on the interaction matrix $\mathbf{Q}$ and $\mathbf{A}$. The low-rank approximation is widely used in recommender systems [74, 23]:

$$\mathbf{A} \approx \mathbf{\Theta}^T \tilde{\mathbf{\Theta}} \qquad\qquad \mathbf{Q} \approx \mathbf{\Theta}^T \mathbf{\Psi} \tag{5.2}$$

Let k be the dimensionality of the latent factor vector. $\mathbf{\Theta} \in \mathbb{R}^{k \times n}$ is the latent factor matrix for the target ads, $\tilde{\mathbf{\Theta}} \in \mathbb{R}^{k \times n}$ is the latent factor matrix for the adjacent ads. Because the ads can act as both the target ad and the adjacent ad, for each ad, there will be two latent factor vectors representing the ad's two roles (the target ad and adjacent ad) respectively. Similarly, $\mathbf{\Psi} \in \mathbb{R}^{k \times m}$ is the latent factor matrix for queries. Plug Equation 5.4 back into the interaction model, and we get

$$f^{(3)}_{i,(q,\mathbf{a})} = \boldsymbol{\theta}^T_{a_i} \boldsymbol{\psi}_q + \sum_{j \neq i, j < |\mathbf{a}|} \boldsymbol{\theta}^T_{a_i} \tilde{\boldsymbol{\theta}}_{a_j}$$

As with the coefficients in the Feature Model, the latent factor vectors of ads and queries $\boldsymbol{\theta}_{a_i}$, $\boldsymbol{\psi}_q$, $\tilde{\boldsymbol{\theta}}_{a_j}$ can be assumed to be generated from Gaussian or Laplace priors, corresponding to L_2 or L_1 regularization respectively. In this section, we assume the latent factors follow zero mean multi-variate gaussian distribution.

Combined Models (CM): It is straightforward to consider combining the three models LBM, FM and IM. Thus in the combined model, different parts of the model will explain a variety of the ads and contexts. The combined model could be simply: $f_{i,(q,\mathbf{a})} = f^{(1)}_{i,(q,\mathbf{a})} + f^{(2)}_{i,(q,\mathbf{a})} + f^{(3)}_{i,(q,\mathbf{a})}$ Obviously, the combination could also be any two of the three aspects.

5.2.3 Inference

In this subsection, we will discuss the inference of this model. We take the Query-Depth Dependent Examination and Informational Relevance with Combined Model as the example. Other variations can be easily derived by pruning the models. To learn the parameters, we adopt Maximum a Posteriori (MAP). Let $\boldsymbol{\Lambda} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi}\}$ represent all model parameters. Assuming independent contexts, we get

$$\begin{aligned} p(\mathbf{S}|\boldsymbol{\Lambda}) &= \prod_{(q,\mathbf{a},\mathbf{c})\in\mathbf{S}} p(\mathbf{c}|\boldsymbol{\Lambda}) = \prod_{(q,\mathbf{a},\mathbf{c})\in\mathbf{S}} \prod_i p(c_i|\boldsymbol{\Lambda}) \\ &= \prod_{(q,\mathbf{a},\mathbf{c})\in\mathbf{S}} \prod_i \left[(\alpha_{i,(q,\mathbf{a})} \cdot \beta_{iq|\mathbf{a}|})^{c_i} \cdot (1 - \alpha_{i,(q,\mathbf{a})} \cdot \beta_{iq|\mathbf{a}|})^{1-c_i} \right] \end{aligned}$$

With the assumption of independent α, we also have

$$p(\boldsymbol{\alpha}|\boldsymbol{\mu}, \mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi}) = \prod_{i,(q,\mathbf{a})} p(\alpha_{i,(q,\mathbf{a})}|\boldsymbol{\mu}, \mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi})$$

As we describe above, we placed a zero mean Gaussian prior on the parameter $\mathbf{b}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Theta}$ and $\tilde{\boldsymbol{\Theta}}$, which are also equivalent to L2 regularization.

$$
\begin{aligned}
p(\mathbf{b}|\sigma^2_{\mathbf{b}}) &= \prod \mathcal{N}(b|0, \sigma^2_{\mathbf{b}}) \\
p(\boldsymbol{\Psi}|\sigma^2_Q) &= \prod_q \mathcal{N}(\boldsymbol{\psi}_q|0, \sigma^2_Q \mathbf{I}) \\
p(\boldsymbol{\Theta}|\sigma^2_A) &= \prod_a \mathcal{N}(\boldsymbol{\theta}_a|0, \sigma^2_A \mathbf{I}) \\
p(\tilde{\boldsymbol{\Theta}}|\sigma^2_A) &= \prod_a \mathcal{N}(\tilde{\boldsymbol{\theta}}_a|0, \sigma^2_A \mathbf{I})
\end{aligned}
$$

Then, the log-likelihood can be computed as

$$
\begin{aligned}
& \log \mathcal{L}(\boldsymbol{\Lambda}|\mathbf{S}, \sigma^2, \sigma^2_{\mathbf{b}}, \sigma^2_Q, \sigma^2_A) \\
= & \sum_{i,(q,\mathbf{a},\mathbf{c}) \in \mathbf{S}} c_i \log(\alpha_{i,(q,\mathbf{a})} \cdot \beta_{iq|\mathbf{a}|}) \\
& + \sum_{i,(q,\mathbf{a},\mathbf{c}) \in \mathbf{S}} (1 - c_i) \log(1 - \alpha_{i,(q,\mathbf{a})} \cdot \beta_{iq|\mathbf{a}|}) \\
& - \frac{1}{2\sigma^2} \sum_{i,q,\mathbf{a}} (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})}))^2 - \Omega(\mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi})
\end{aligned}
$$

where

$$
\Omega(\mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi}) = \frac{1}{2\sigma^2_{\mathbf{b}}} \|\mathbf{b}\|_2 + \frac{1}{2\sigma^2_Q} \|\boldsymbol{\Psi}\|^2_F + \frac{1}{2\sigma^2_A} \|\boldsymbol{\Theta}\|^2_F + \frac{1}{2\sigma^2_A} \|\tilde{\boldsymbol{\Theta}}\|^2_F
$$

To solve the MLE and learn the model parameters, we employ Expectation Maximization (EM) algorithm with the event of examination as the hidden variable.

EM algorithms

E-Step: Let us define E as the unobserved events of examination. The EM algorithm first finds the expected value of the log likelihood $\log P(\mathbf{S}, \mathbf{E}|\boldsymbol{\Lambda})$ with respect to hidden variable E given observed $\mathbf{S}$ and current parameter $\boldsymbol{\Lambda}$

$$\begin{aligned} Q(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{t-1}) &= E_{\mathbf{E}}[\log P(\mathbf{S}, \mathbf{E}|\Theta)|\mathbf{S}, \boldsymbol{\Lambda}^{t-1}] \\ &= \sum_{i,(q,\mathbf{a},\mathbf{c})\in\mathbf{S}} \sum_{C,E} [\log P(C, E|) P(E|C, \boldsymbol{\Lambda}^{t-1})] \end{aligned}$$

where $\boldsymbol{\Lambda}$ is the set of parameters following the tradition. We drop the indices temporarily for notational convenience. Then based on the assumption of the examination model, the probability $P(E|\mathbf{S}, \boldsymbol{\Lambda}^{t-1})$ mentioned above can be simplified to different cases. Then

$$\begin{aligned} P(E = 1|C = 1, \boldsymbol{\Lambda}^{t-1}) &= 1 \\ P(E = 0|C = 1, \boldsymbol{\Lambda}^{t-1}) &= 1 \\ P(E = 1|C = 0, \boldsymbol{\Lambda}^{t-1}) &= \frac{\beta^{t-1}(1 - \Phi(f))}{1 - \Phi(f)\beta^{t-1}} = \tilde{C}^{t-1} \\ P(E = 0|C = 0, \boldsymbol{\Lambda}^{t-1}) &= 1 - \tilde{C}^{t-1} \end{aligned}$$

where $\tilde{C}^{t-1}_{i(q,\mathbf{a})}$ can be interpreted as the probability of the ad a_i being irrelevant of the context $(q, \mathbf{a})$, judged by users. Then the $Q(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{t-1})$ can be computed as

$$\begin{aligned} & Q(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{t-1}) \\ = \ & E_{\mathbf{E}}[\log P(\mathbf{S}, \mathbf{E}|\Theta)|\mathbf{S}, \Theta^{t-1}] \\ = \ & \sum_{S^{\bullet}} \Big\{ \log(\alpha_{i,(q,\mathbf{a})}\beta_{iq|\mathbf{a}|}) + \frac{1}{2\sigma^2}(\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})}))^2 \\ & + \Omega(\mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi}) \Big\} + \sum_{S^{\circ}} \{ [\log((1 - \alpha_{i,(q,\mathbf{a})})\beta_{iq|\mathbf{a}|}) \\ & + \frac{1}{2\sigma^2}(\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})}))^2 + \Omega(\mathbf{b}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Psi})]\tilde{C}_{iq|\mathbf{a}|} \\ & + \log(1 - \beta_{iq|\mathbf{a}|})(1 - \tilde{C}_{iq|\mathbf{a}|}) \Big\} \end{aligned}$$

This competes the E-Step.

M-Step: The M-step of EM iteration tries to maximize the expectation computed above, that is, to find

$$\begin{aligned}
\boldsymbol{\Lambda}^t &= \arg\max_{\Lambda} Q(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{t-1}) \\
& \quad \text{s.t.} 0 < \alpha_{i,(q,\mathbf{a})} < 1, 0 < \beta_{iq|\mathbf{a}|} < 1
\end{aligned}$$

To solve this optimization problem, we adopt a method that mixes block co-ordinate descent and stochastic gradient descent. Taking derivatives in respect to β, produces the updating formulas.

$$\beta^t_{iqd} = \frac{S^\bullet_{iqd} + \sum_{i,q,|\mathbf{a}|=d} S^\circ_{i,(q,\mathbf{a})} \tilde{C}^{t-1}_{i,(q,\mathbf{a})}}{S_{iqd}}$$

For other parameters, we perform stochastic gradient descent. For each impression with the target ad a_i and its context $(q, \mathbf{a})$, we have the following update rule:

$$\begin{aligned}
\alpha_{i,(q,\mathbf{a})} &\leftarrow \mathcal{P}\left\{ \alpha_{i,(q,\mathbf{a})} - \eta \left[-\frac{S^\bullet_{i,(q,\mathbf{a})}}{\alpha_{i,(q,\mathbf{a})}} + \frac{\tilde{C}^{t-1}_{i,(q,\mathbf{a})} S^\circ_{i,(q,\mathbf{a})}}{1 - \alpha_{i,(q,\mathbf{a})}} \right.\right. \\
& \quad \left.\left. + \lambda W_{i,(q,\mathbf{a})} (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})})) \right] \right\} \\
b_k &\leftarrow b_{1k} - \eta W_{i,(q,\mathbf{a})} \big[-2\lambda (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})})) \\
& \quad \cdot \phi(f_{i,(q,\mathbf{a})}) x_{i,(q,\mathbf{a})k} + 2\lambda_{\mathbf{b}} b_k \big] \\
\boldsymbol{\psi}_q &\leftarrow \boldsymbol{\psi}_q - \eta W_{i,(q,\mathbf{a})} \big[-2\lambda (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})})) \\
& \quad \cdot \phi(f_{i,(q,\mathbf{a})}) \boldsymbol{\theta}_{a_i} + 2\lambda_Q \boldsymbol{\psi}_q \big] \\
\tilde{\boldsymbol{\theta}}_{a_j} &\leftarrow \tilde{\boldsymbol{\theta}}_{a_j} - \eta W_{i,(q,\mathbf{a})} \big[-2\lambda (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})})) \\
& \quad \cdot \phi(f_{i,(q,\mathbf{a})}) \boldsymbol{\theta}_{a_i} + 2\lambda_A \tilde{\boldsymbol{\theta}}_{a_j} \big] \\
\boldsymbol{\theta}_{a_i} &\leftarrow \boldsymbol{\theta}_{a_i} - \eta W_{i,(q,\mathbf{a})} \big[-2\lambda (\alpha_{i,(q,\mathbf{a})} - \Phi(f_{i,(q,\mathbf{a})})) \\
& \quad \cdot \phi(f_{i,(q,\mathbf{a})}) (\boldsymbol{\theta}_q + \sum_{j \neq i} \boldsymbol{\theta}'_{a_j}) + 2\lambda_A \boldsymbol{\theta}_{a_i}]
\end{aligned}$$

where $W_{i,(q,\mathbf{a})} = (S^\bullet_{i,(q,\mathbf{a})} + \tilde{C}^{t-1}_{i,(q,\mathbf{a})} S^\circ_{i,(q,\mathbf{a})})$ and $\mathcal{P}(x)$ is the projection which simply truncates x into range $(0, 1)$. In an efficient implementation, we first pass the data to calculate $\tilde{C}_{i,(q,\mathbf{a})}$, update β and next perform M-step in SGD style updating. While performing M-step, $\tilde{C}_{i,(q,\mathbf{a})}$ are calculated for the next iteration. In this implementation, for each iteration, the time complexity is $O(dkM)$, where d denotes max depth, k denotes the dimensions of latent factors and M is the number of records. In practice, d and k usually are very small (e.g., $d = 4$, $k = 10$), and can be treated as constants. Then, the time complexity for a single iteration will be $O(M)$. Assuming the algorithms need z iterations to converge, the total time complexity will be $O(zM)$ which allows the algorithms to handle large scale data.

5.2.4 Experiments

In this subsection, we conduct experiments on the data collected from a commercial search engine and compare our model with state-of-the-art methods.

Experiment Setting

We collected the search log data from a commercial search engine in the U.S. English market in April and the first two weeks of May 2012. The dataset from April 2012 is the one we used in Subsection 5.2.1. We filter out low frequency (less than 40) queries and ads to remove noise. In this work, we focus on head queries for three reasons: 1) Head queries hold the majority of the search traffic which is more important to the search engine. 2) Due to high traffic of head queries, user behaviors are usually consistent and we can mine the patterns of user behaviors effectively. 3) In this work, since we investigate ad list effects rather than individual ads, we do not have enough data to analyze tail queries.

In the end, the April 2012 data set contains 29,722,684 impressions and the May 2012 data set contains 13,160,289 impressions. The other data statistics are shown in Table 5.8. In our experiments, we provide two types of evaluation: Description

Table 5.4: Data sets

	April 2012	May 2012	total
Impressions	29,722,684	13,160,289	42,882,973
Context $(q, \mathbf{a})$	112,084	52,814	135,445
Ads	62,423	34,214	72,959
Queries	49,483	26,104	53,730

Oriented Evaluation (DOE) and Prediction Oriented Evaluation (POE).

Description Oriented Evaluation (DOE): this task is to understand and describe user's click behaviors, based on the click log of the existing sponsored search results. Thus, we sample one third of the click logs from April 2012 at random as test data, and the remaining two thirds of click logs are used as our training set. There are 19,829,420 impressions (query session) in the training data and 9,915,760 impressions in the test data. In this setting, the cold start problem (that is, the new query or ads of test data does not exist in the training data) will be relatively alleviated. The evaluation aims to show how well the model fits the data and understand user click behaviors on sponsored search results.

Prediction Oriented Evaluation (POE): this task is to predict the click-through rate on sponsored search, based on the trained model. We use April 2012 data set as the training data, and the May 2012 data set as the test data. Since the train/test data are split temporally, the cold start problem is serious. This evaluation is more instructive for real applications.

The major measurement for performance evaluation is log likelihood of generating the test data, which is widely used in the CTR prediction problem [27].

$$\mathrm{LL} = \frac{1}{S} \sum_{i,(q,\mathbf{a})} S^{\bullet}_{i,(q,\mathbf{a})} \log(\hat{y}_{i,(q,\mathbf{a})}) + S^{\circ}_{i,(q,\mathbf{a})} \log(1 - \hat{y}_{i,(q,\mathbf{a})})$$

Additionally, we also report Relative Information Gain (RIG) [50] and Normalized

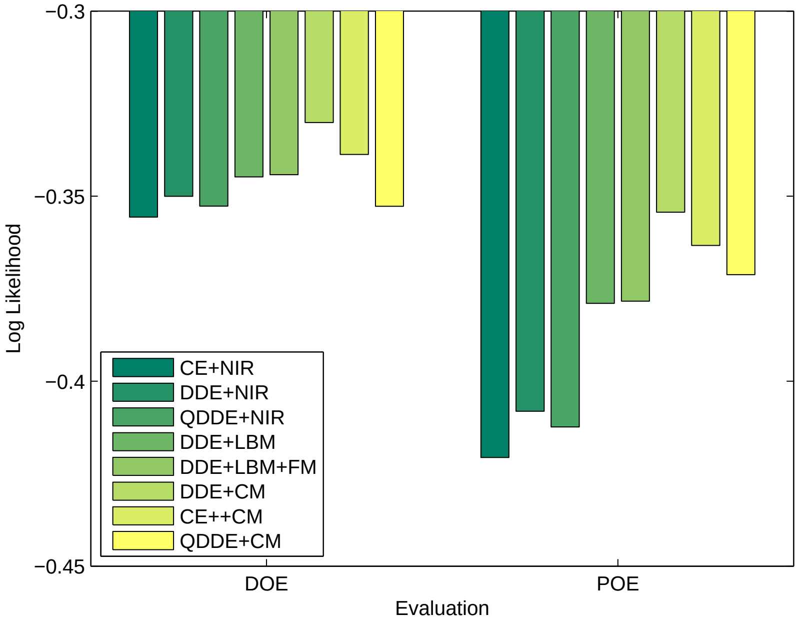


Figure 5.6: Model analysis (Log-Likelihood)

Rooted Mean Square Error (NRMSE) for references. In our experiments, the dimension of latent factors for queries and ads is set at 10. We use term frequency as the features of queries and ads. In the latent bias model, we adopt global bias, query bias, ads bias, and position bias.

We conduct experiments to analyze the performance of the variations of our context aware CTR prediction model. First, we check the examination hypotheses by setting the relevance to Non-Informational Relevance (NIR). The results are shown in Table 5.5. We can see that Depth-Dependent Examination (DDE) outperforms the Constant Examination (CE) as we expected. Figure 5.7 shows the relative improvement of DDE over CE is around 10% RMSE on both the description and prediction task. However, we also notice that the performance of the query-dependent model QDDE is slightly worse than DDE. After examining the relevant data, we find that more than half queries do not hold complete depth records, and so, for these

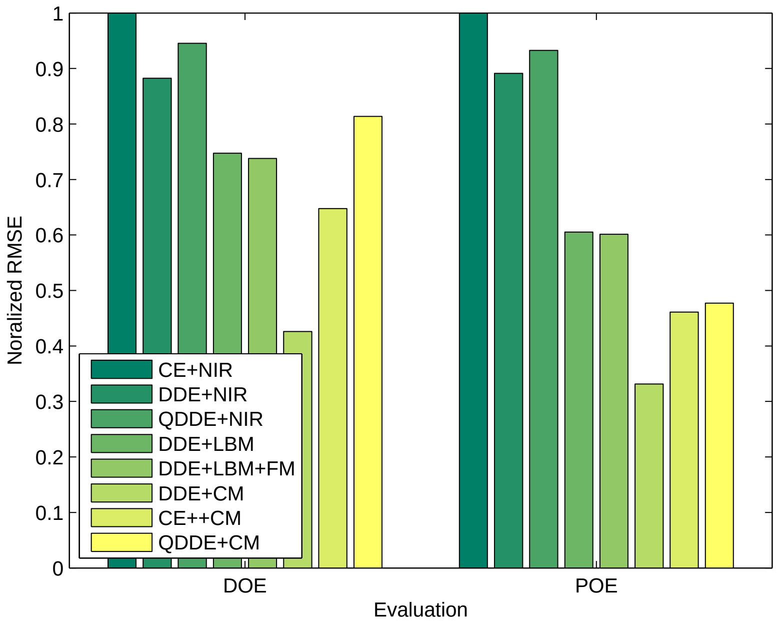


Figure 5.7: Model analysis (Normalized RMSE)

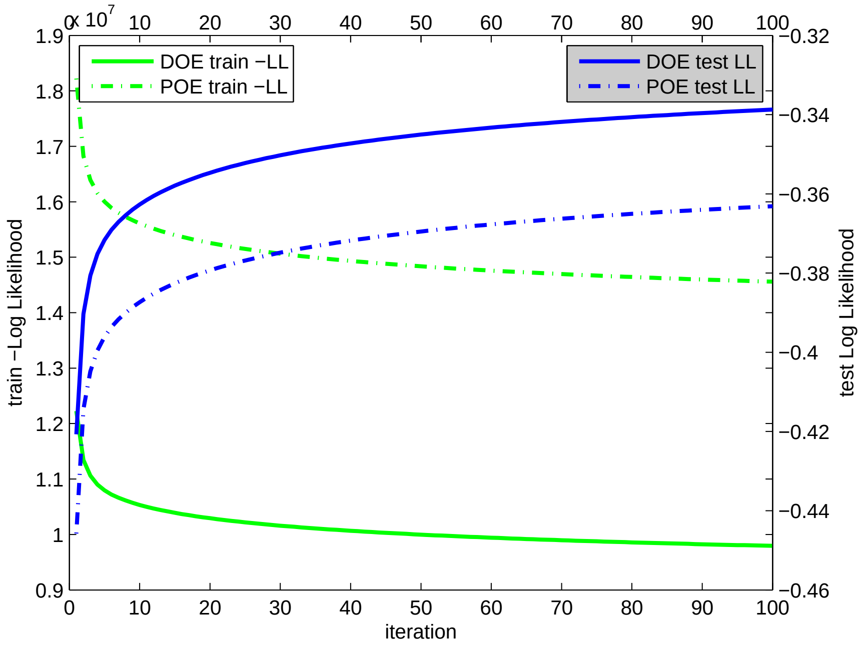


Figure 5.8: Model analysis (Convergence)

queries, some specific depths data are not applicable in the training data while they may appear in test data. In these cases, QDDE cannot make a better prediction and a more advanced model that can effectively handle query dependent parameters is necessary.

By using Depth Dependent Examination as our examination model, we can try different relevance models. From Table 5.5 and Figure 5.7, we can see that if we only use the Latent Bias Model (LBM), we get better results than NIR. It shows around 10% and 20% RMSE improvement in Description Oriented Evaluation and Prediction Oriented Evaluation respectively. After adding features to the model (LBM+FM), the performance gets slightly better in both tasks too, but the improvement is negligible. By using more delicate features such as Click Over Expected Clicks (COEC) [166] or latent topic distribution of the ad, a larger improvement may be possible. Next, we use Combined Model, which combines the three models: LBM, FM, and IM. As expected, the model achieve the best performance on both POE and DOE. The improvement over LBM is much larger than FM. We also test different examination models with the combined relevance model, and the experiments show similar results for both evaluation task, the Depth Dependent Model is always better than CE and QDDE.

Overall, the performance on Prediction-Oriented tasks is worse than Description-Oriented tasks as expected, due to the cold starts on queries, ads and contexts. We also check convergence of the EM algorithms on both PDE and ODE. Figure 5.8 shows that both curves converge quickly and achieve a saddle point. Next, we will employ the combined relevance model (CM) and Depth Dependent Examination model (DDE) to compare with other baselines.

Table 5.5: Model Anlysis

Model		DOE		POE	
Examin.	Relev.	LL	RIG	LL	RIG
CE	NIR	-0.3556	0.1372	-0.4206	0.0250
DDE	NIR	-0.3500	0.1508	-0.4081	0.0539
QDDE	NIR	-0.3526	0.1443	-0.4123	0.0442
DDE	LBM	-0.3448	0.1634	-0.3789	0.1215
DDE	LBM+FM	-0.3441	0.1649	-0.3783	0.1229
DDE	CM	**-0.3300**	**0.1991**	**-0.3543**	**0.1787**
CE	CM	-0.3387	0.1782	-0.3633	0.1578
DDE	CM	**-0.3300**	**0.1991**	**-0.3543**	**0.1787**
QDDE	CM	-0.3527	0.1441	-0.3712	0.1395

Comparison with Existing Methods

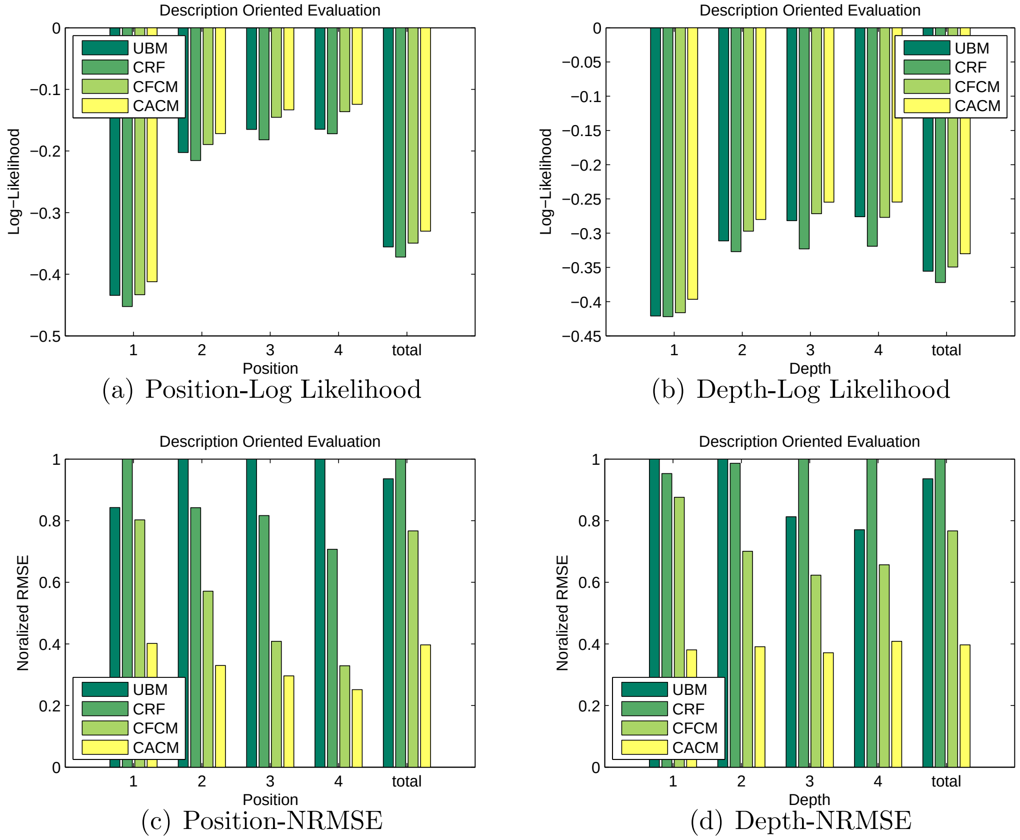
We name our method CACM for Context Aware Click Model and compare our model with three existing methods on both Description Oriented Evaluation and Prediction Oriented Evaluation:

- **User Browsing Model (UBM)** is the baseline method and a classical click model [38]. We adopt EM inference process for parameter α and β. To avoid infinite values in log-likelihood, α is truncated into the range $(0.0001, 0.9999)$.

- **Matrix Factorization Click Model (MFCM)** is proposed by Shen et al. [123].[2] The dimensionality of the latent factor vector is also set at 10 for a fair comparison.

- **Relational click model (CRF)** is proposed by Xiong et al. [147]. They adopt Conditional Random Field (CRF) to model the exclusive effects between similar ads that are shown together in the same impression.

[2] In this work, we do not focus on the personalized model. Personalization could be a factor of the ads contexts, and easily plugged into our framework.



(a) Position-Log Likelihood

(b) Depth-Log Likelihood

(c) Position-NRMSE

(d) Depth-NRMSE

Figure 5.9: Comparing with baselines in DOE

The reason we choose these three methods is that UBM is a classical click model while CFCM and CRF are very recent state-of-the-art methods. The overall results of the Log Likelihood and the RIG of the four methods on DOE are shown in Table 5.6. To accommodate the random factor in EM, we repeat the experiments many times at different random seeds, and the performances stay the same. We see that the performance of UBM is better than CRF which is a more recent method. As we mentioned earlier, we also expect additional improvement for CRF and our Feature Model by using more delicate features such as Click Over Expected Clicks (COEC) [166] or latent topic distribution of the ad, because CRF and our Feature Model are feature dependent models. Another reason UBM is better than CRF in DOE

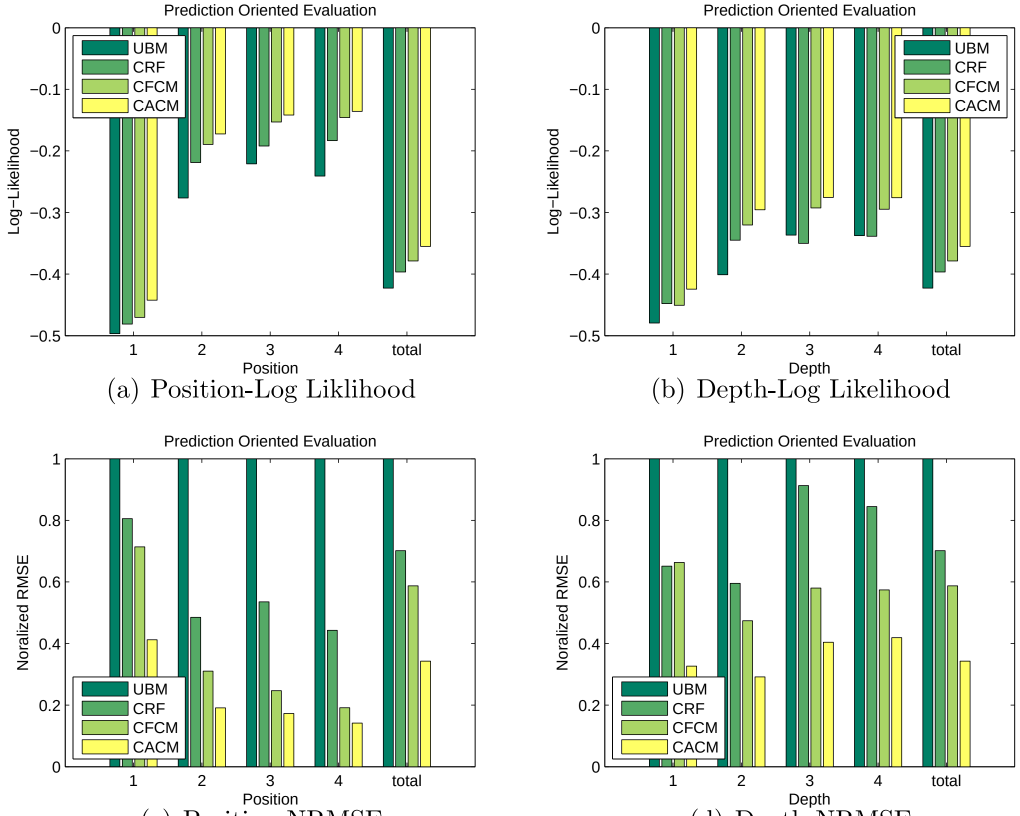
Table 5.6: Comparing with the baselines on DOE

	UBM	CRF	MFCM	CACM
Log Likelihood	-0.3556	-0.3720	-0.3494	-0.3301
RIG	0.1372	0.0973	0.1522	0.1990

is because the UBM method is designed to understand the user browsing behaviors and not targeting the prediction of the click through rate for new ads. Essentially, CFCM is an extension of UBM, and CFCM gets around a 10% improvement over UBM on RMSE. That is consistent with the results in [123]. We also notice that our methods outperform all three existing methods in DOE. To analyze the performance on different positions and depths, we plot Log Likelihood and Normalized RMSE of the four methods categorized by position and depth. The performance for different depths are shown in Figs. 5.9(b) and 5.9(d) and the results for different positions are shown in Figs. 5.9(a) and 5.9(c) respectively. The results are consistent with the overall results, except for depth 1 cases, in which the CRF is better than UBM. In all cases, our method can achieve the best performance. Next, we compare the four methods in Prediction Oriented Evaluation. The overall results are shown in Table 5.7. As expected, the results of the four methods is worse than the results in DOE. In POE, we see that CRF achieves much better results than UBM. As discussed, UBM cannot handle new query, ads or even the new pair query-ads with the existing query and ad. Thus, CRF which is designed for prediction tasks, outperforms UBM. For CFCM, we also consistently find that CFCM also outperforms

Table 5.7: Comparing with the baselines on POE

	UBM	CRF	MFCM	CACM
Log Likelihood	-0.4227	-0.39654	-0.38464	-0.3543
RIG	0.0201	0.08086	0.10844	0.1787



(a) Position-Log Liklihood

(b) Depth-Log Likelihood

(c) Position-NRMSE

(d) Depth-NRMSE

Figure 5.10: Comparing with baselines in POE

UBM significantly as shown in [123]. Just like DOE, our method outperforms all three baselines. Similarly, we also show the results on positions in Figs. 5.10(a) and 5.10(c) and the results on depths in Figs. 5.10(b) and 5.10(d). For the cases of Position 1 and depth 1, the performance of all four methods is worse than other cases. Because the number of the cases of Position 1 and depth 1 are always much larger than other cases, they have a greater variance on CTR which increases the difficulty of predicting CTR. In all cases, our methods outperform all three other methods. The improvement on RMSE over CFCM is more than 10%.

5.3 Estimating Click Yields in Sponsored Search

As we described in the beginning of this chapter, under the PPC model, advertisers are charged when their advertisements are clicked by search engine users. Therefore, user clicks is a key performance metric for sponsored search, which is directly related to revenue. In order to deliver ads with high payoff, predicting user clicks plays a critical role in sponsored search. Current ad-delivery strategies are two-step approaches, and work as follows. The system first predicts individual ad click-through rates (CTRs) for the given query. Based on the estimated CTRs, the system selects the ads as a ranking list by maximizing per search revenue subject to some conditions such as user utility and advertiser ROI [20]. The underlying assumption is that the rewards of ads are independent and clicks on the ads in a list is additive. However, this assumption does not hold in reality. For example, two similar ads in the same list could reduce the CTRs of each [147]. Moreover, this strategy is naturally suboptimal. The learning problem is focused on predicting individual performance rather than group performance which is selecting the best list of ads to obtain overall payoff. In this work, we consider click yield (the total number of clicks for the list of ads per impression) as the group performance instead of revenue for simplification.[3] The motivation behind the proposal to change the objective is based on nontrivial observations. That is, two systems that have similar precision on predicting individual performance could behave very differently on group performance. For example, we can have two models having the same CTR prediction accuracy. However, the ads selected by the two models have huge differences on the clicks obtained. Moreover, since the overall click yields are based on the delivered ad list and the CTRs are not independent, the results could be suboptimal even when

[3] The simplification assumes the bidding prices are the same for all ads. Technically it could be generalized to revenue by taking the bidding price into consideration.

the individual ad CTR prediction is perfect. In fact, it is not necessary to know accurately the CTRs of each individual ad, as long as we can select the best lists. Sometimes estimating CTRs of all ads may be an even more challenging problem due to the problem of sparsity.

Most previous research in sponsored search has focused on CTR prediction [50, 166]. In some recent work [147], context information has been take into consideration in sponsored search. However, they are still working on the CTR prediction problem, rather than the click yields prediction problem. In this section, we challenge the traditional strategy for predicting group performance based on CTR prediction. We propose a novel framework that could directly predict the click yield for a ranked list of ads. We argue this is a more "correct" problem to solve. The problem is essentially a ranking problem since we would like to judge which lists of ads are better. However, unlike the problem of learning to rank that ranks single documents, this ranks sets of ads. Additionally, we leverage a special constraint on the problem. That is, the number of ads shown on a page is usually limited to a small number, (e.g., four in most commercial search engine). Still, it is not realistic to explore all possible combination of ads. Therefore, we simplify the problem to rank historically presented ad lists only (that is, the ad lists which have been shown before).

To estimate the ad group performance in sponsored search and tackle these challenging problems, we combine heterogenous types of information into the learning model. We first investigate several effects including latent biases, contextual features, interactive influence and correlation over positions. We find that a unified model can achieve the best performance. To best leverage the text features and solve the sparseness issue in text information and cold start on ads, we embed the topic coding model into our framework to learn the topic information of short ad text. We also discuss the effects of various loss functions. We find that ranking loss is preferred for this problem. Finally, we conduct extensive experiments on a large-scale real-world dataset from a commercial search engine. Our methods achieve significant improvement by solving the sponsored search problem from this new perspective.

We summarize our contributions as follows,

- We introduce a novel perspective for sponsored search—click yields—which measures the group performance of ads displayed together.
- To solve the click yields prediction problem, we propose a Click Yields Prediction framework which incorporates multiple factors, such latent bias, textual features, interactive influences, and position correlation.
- Our experiments, based on a large-scale real-world dataset, show our methods make noticeable improvements and outperform state-of-the-art methods based on the traditional strategy.

5.3.1 Preliminaries

In this subsection, we conduct some exploratory experiments to verify our intuitions and formalize the problem definition. We use the same search log data as in Section 5.2 (collected from a commercial search engine in the U.S. market in April 2012). In total, there are 127,230,609 query sessions, where a query session consists of an input query, a list of sponsored search results and a list of clicked ads with time stamps. Since click data are noisy, we filter out low frequency queries and ads to remove some noise and use the more confident click data for analysis.

Group Performance

Compared to individual performance, (e.g., individual ad CTR), ad group performance is much closer to the real system performance. For instance, in sponsored search, given a query, we can decide which group of ads to display, according to the estimated group performance. Here, we list two examples of group performance—click yield and revenue.

Click Yield: We define the terminology of *click yield* (CY) as follows. Given a query q and an ad list $\mathbf{a} = \{a_1, a_2, \cdots, a_{|\mathbf{a}|}\}$, the click yield is the ratio of the total number of clicks on the ads in $\mathbf{a}$ over the total number of impressions:[4]

$$y_{q,\mathbf{a}} = \frac{\sum_{i=1}^{|\mathbf{a}|} \text{Click}(q, a_i)}{\text{Impr}(q, \mathbf{a})}$$

where $\text{Click}(q, a_i)$ denotes the total number of clicks on the ad a_i, with issued query q and $\text{Impr}(q, \mathbf{a})$ denotes the number of times showing the ad group $\mathbf{a}$, with issued query q. The concept of click yield differs from the click-through rate (CTR) in that it measures the overall performance of ad lists rather than single ads. We see that unlike the click-through rate of an individual ad, $y_{q,\mathbf{a}}$ might be larger than one.

Revenue: Compared with CTR, click yield is more relevant to the revenue of search engines under the PPC model. By incorporate biding price into the click yields, we will get the revenue:

$$r_{q,\mathbf{a}} = \frac{\sum_{i=1}^{|\mathbf{a}|} \text{Click}(q, a_i) \times \text{Bid}(q, a_i)}{\text{Impr}(q, \mathbf{a})}$$

where $\text{Bid}(q, a_i)$ denotes the biding price of the ad a_i, with issued query q. If we assume all biding prices equal one, we see that $y_{q,\mathbf{a}} = r_{q,\mathbf{a}}$. Because the bidding prices are dynamic, and very sensitive data, in our following experiments, we will focus on click yields, which is directly related to the revenue of search engines. Note that it is straightforward to apply the methods designed for click yields in estimating revenue.

[4] The definition of click yield is slightly different from the one used in some other papers [59], where it is defined by the total number of clicks divided by the total number of search result page views. In Hillard et al. [59], the click yield is only a global metric while in this section it is a metric for ad lists.

Problem Definition

With the definition of group performance, we formalized the problem as follows: the records are presented by $\mathcal{D} = (q_m, \mathbf{a}_m, y_m)_{m=1}^{M}$. For a single click log entry $(q, \mathbf{a}, y) \in \mathcal{D}$, q is the query in the record, $\mathbf{a}$ is the list of ads $\mathbf{a} = \{a_1, a_2, \cdots, a_{|\mathbf{a}|}\}$ and $|\mathbf{a}|$ refers to the number of ads (i.e., depth). y means the corresponding click yield. In this work, we will try to solve the following problem: given a query q and a collection of available ad lists $\mathcal{D}_q$, we predict the click yields of the ad lists or rank the ad lists and find the best ad list out of $\mathcal{D}_q$ that has the highest click yield.

Click Yields Analysis

A commercial search engine typically adopts a two-step approach. That is, the search engine first predicts the click-through rate of ads given the query. Then, it selects the ads with higher predicted CTR to display.[5] However, this approach is suboptimal even when the CTR prediction is perfect since it assumes the CTR of individual ads are independent of each other and clicks are additive. In the following analysis we will see this assumption does not hold in reality.

Fig. 5.11 shows the average click yields of specific depths (number of ads displayed together) over all queries. We can see that as the depth grows, the increase in click yield slows. For instance, the click yield of depth 1 is 0.1798, but the the click yields of depth 2 is only 27% more than depth 1. This shows that with more ads displayed the clicks on individual ads may be diluted. As we can see, the average click yield becomes larger with increasing depth. However, showing more ads does not *always* get the best click yields. We find many examples like the one in Table 5.8: for query “worldofwatches”, if we only display a single ad

[5] This is a simplified description. The real world is more complicated. It also needs to consider the bidding price, the budget of advertiser, etc., but the major idea is still the same.

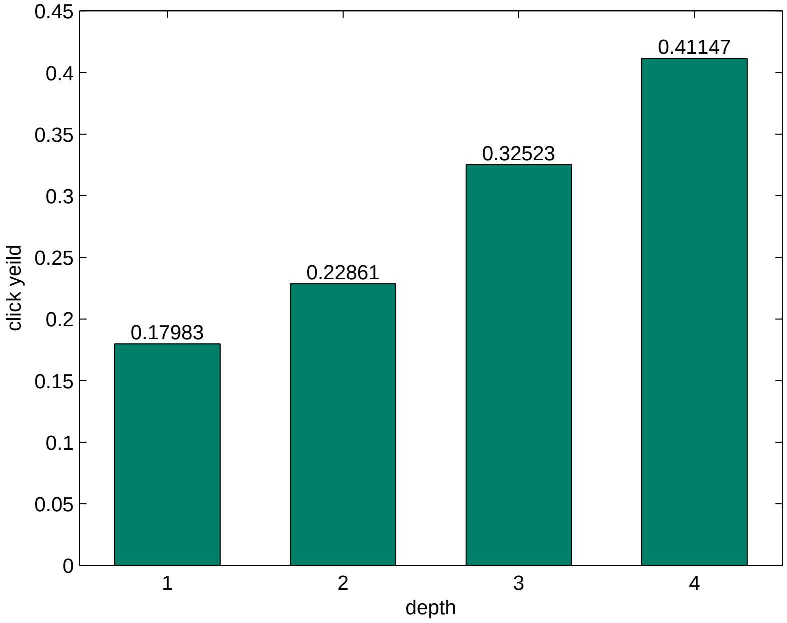


Figure 5.11: Average Click Yields

"http://WorldofWatches.com", the click yield is 0.2096, while if we display this ad with two other watch-related ads, the click yield of those ads is only 0.1428, which is even lower than the single ad. That is because with query "worldofwatches", although the users are interested in watches, the users' intent is quite clear and exactly want the website "http://WorldofWatches.com". Showing more ads may annoy users, and they are more likely to skip the ads area. In the case that some specific ad exactly matches the user's intent, it appears that less is more.

We further investigate the effects of the query's commercial intent on click yield and CTR. We treat the average CTR of the query as the measurement for commercial intent of the query (higher average CTR means higher commercial intent), and investigate the relationship between the depth and query commercial intent. In our experiments, we let the depth change from 1 to 3 and then test whether the commercial intent affects the changes of CTR or CY. We categorize the queries by

Table 5.8: An example of query "worldofwatches"

Ad list 1 (Click Yields 0.2096)
world of watches (*http://WorldofWatches.com*) Up To 80% Off On The World's Finest Watches. Free Shipping, Shop Now!

Ad list 2 (Click Yields 0.1428)
world of watches (*http://WorldofWatches.com*) Up To 80% Off On The World's Finest Watches. Free Shipping, Shop Now! **Ashford - Luxury for Less** (*http://Ashford.com/Watches_Sale*) Shop Top Brand Watches. Free Shipping. 100%Authentic w/Warranty. **Invicta up to 90% off** (*http://www.nextdaywatches.com*) Authorized Invicta Dealer. Large Selection. Free, Next Day Delivery.

two metrics, CTR and CY. Then, we obtain queries of four types: CTR increasing query, CTR decreasing query, CY increasing query, and CY decreasing query. For the majority of queries, with a larger depth, the average CTR of queries are diluted while the click yields increase, but for some special queries, the average CTR of a large depth is greater than a small depth, or the click yields of a small depth can be larger than that of a large depth. In Figures 5.12(a) and 5.12(b), we see that the majority of queries have higher commercial intent (average CTR). In our experiments, there are around 1/10 queries whose average CYs decrease when changing depth from 1 to 3. We see that the commercial intents of these queries are relatively low. Note that even if the average CY of a query increases when changing depth 1 to 3, the gain still may not be linear; that is, one cannot simply use additive CTRs to estimate CY. On the other hand, unfortunately, another type of the unusual cases—CTR increasing (CY increasing obviously) queries—also shares the same property (low commercial intent). That means, we cannot simply use average CTR of ads to decide the number of ads displayed together. A more sophisticated model is necessary to solve this problem.

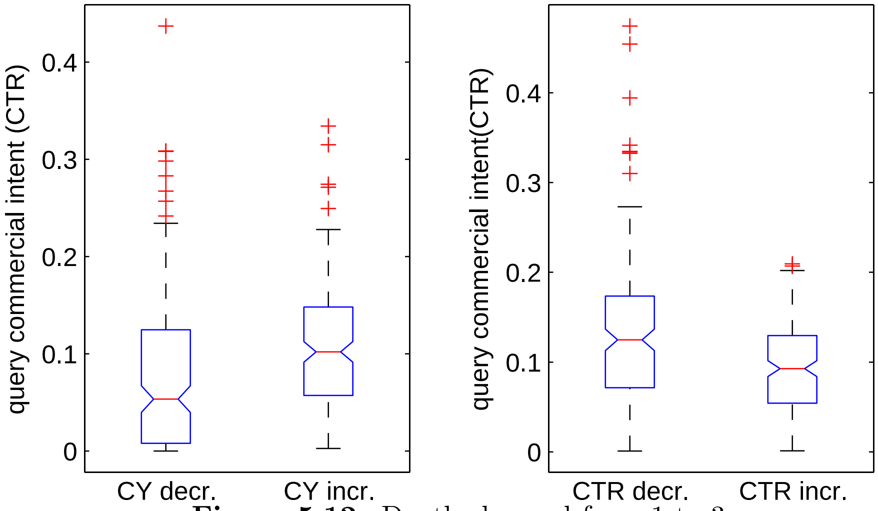


Figure 5.12: Depth changed from 1 to 3

5.3.2 Predicting Click Yields

To estimate the group performance (click yields) is an more complicated problem than individual performance, due to extra factors: the number of ads, interactive influences between query and ads, interactive influences across ads. In this subsection, we will propose a unified models for predicting the click yields of $\mathbf{a}$, given query q. $y_{q,\mathbf{a}}$ represents the click yields. $f_{q,\mathbf{a}}$ the score of relevance which can be used for estimating click yields or ranking the ads lists.

Estimation Model

Latent Bias (LB): One straightforward intuition is that whether the ad is clicked or not depends on the average click rate of the ads: $f(q, \mathbf{a}) = \sum_{i=1}^{k} \mu_{a_i}$ where μ_{a_i} is the click through rate of the target ad across all other factors. Certainly, this estimation is too coarse and inaccurate, because as we mentioned before, the majority of ads are not clicked and multiple factors of the contexts of the ads are not incorporated. We can extend this basic estimation by incorporating a wider biases from the multiple factors: 1) Positional Query bias, because for each query, the average CTRs on specific positions are different. Given a query q, for each position i, one has a latent bias $\mu_q^{(i)}$. The latent bias of query q can be expressed in a vector form $\boldsymbol{\mu}_q = \left[\mu_q^{(1)}, \mu_q^{(2)}, \cdots, \mu_q^{(d)}\right]^T$ where d denotes the max depth of the ads area (the maximum number of ads shown in ads area). 2) For ad list $\mathbf{a}$, the average CTR of a specific ad depends on the both the position of the ad and the ad itself. Similarly, one has a positional bias for an ad list, $\boldsymbol{\mu}_{\mathbf{a}} = \left[\mu_{a_1}^{(1)}, \mu_{a_2}^{(2)}, \cdots, \mu_{a_d}^{(d)}\right]^T$, where $\mu_a^{(i)}$ denotes the bias of ad a on position i. To compose the final bias of a pair $(q, \mathbf{a})$, an indicator $\mathcal{I}_{\mathbf{a}}$ is also necessary. $\mathcal{I}_{\mathbf{a}}$ points out the positions which are filled by ad list $\mathbf{a}$. We define $\mathcal{I}_{\mathbf{a}} = \left[I_{\mathbf{a}}^{(1)}, I_{\mathbf{a}}^{(2)}, \cdots, I_{\mathbf{a}}^{(d)}\right]^T$ the position indicator vector of $\mathbf{a}$, where $I_{\mathbf{a}}^{(i)} = 1$ if some ad appears on position i otherwise $I_{\mathbf{a}}^{(i)} = 0$. For instance, if $d = 4$ and $|\mathbf{a}| = 2$ which means the first two positions are filled by two ads, then $\mathcal{I}_{\mathbf{a}} = [1, 1, 0, 0]^T$. Then latent bias is

$$f_{q,\mathbf{a}}^{LB} = \mu_0 + \mathcal{I}_{\mathbf{a}}^T \boldsymbol{\mu}_q + \mathcal{I}_{\mathbf{a}}^T \boldsymbol{\mu}_{\mathbf{a}}$$

where μ_0 is the global bias. Note that these biases are the latent variables to be learned from the data sets. However, this model is appealing since no extra information is needed for learning, besides requiring indicators.

Features (FM): A more powerful model than the bias model is the feature model, which predicts the relevance and harvest the information of features by a linear function of features. For a search session, we could extract the features of the ads and their contexts. Given a record $(q, \mathbf{a})$ and click yields $y_{q,\mathbf{a}}$, the feature vector can be extracted from two aspects:

- **Query Features:** Let $\mathbf{x}_q$ be the feature vector of query q, which can be extracted from the terms of the shown title and body of the query or latent topic distribution of the query.

- **Ad Features:** Let $\mathbf{x}_{a_i}$ be the feature vector of ad a_i, which can be extracted from the terms of the shown title and body of the ad, the history click over expected clicks (COEC) [166] or latent topic distribution of the ad content.

Like latent bias model, with position and depth information, we define

$$\mathbf{x}_{\mathbf{a}} = \left[I_{\mathbf{a}}^{(1)}\mathbf{x}_{a_1}^T, I_{\mathbf{a}}^{(2)}\mathbf{x}_{a_2}^T, \cdots, I_{\mathbf{a}}^{(d)}\mathbf{x}_{a_d}^T\right]^T$$

A simple linear combination of query features and ad features can be defined as:

$$f_{q,\mathbf{a}}^{FM} = \mathbf{b}_q^T\mathbf{x}_q + \mathbf{b}_{\mathbf{a}}^T\mathbf{x}_{\mathbf{a}}$$

$\mathbf{b}_q$ and $\mathbf{b}_{\mathbf{a}}$ are coefficients to be learned from the training set. The model is essentially equivalent to the one where ad feature and context features are combined into a single feature vector $\mathbf{x} = [\mathbf{x}_q^T, \mathbf{x}_{\mathbf{a}}^T]^T$ and a coefficient $\mathbf{b} = [\mathbf{b}_q^T, \mathbf{b}_{\mathbf{a}}^T]^T$. Here, we further place a zero mean Gaussian prior or Laplace prior on the values of coefficient $\mathbf{b}$, corresponding to the L_2 and L_1 regularization, respectively. With the Gaussian prior, we have: $\mathbf{b} \sim \mathcal{N}(0, \lambda_{\mathbf{b}}^{-1}\mathbf{I})$

Interactive Influence (IM): Although linear models are efficient, they are usually over-simplified and cannot capture interactions between queries and ads. The interaction between the query and ads are studied in Xiong et al. [147]. However, their methods did not consider to model the interaction across ads. Moreover, their models still are designed for CTR prediction rather than click yields prediction. Similar to Rendle [111], we model interactions of click yields of a query-ad list pair $(q, \mathbf{a})$ through an additive function of ad positions. Let $\mathbf{G}^{(i)} \in \mathbb{R}^{N_a \times N_q}$ be the interaction matrix on position i. The interaction can be modeled as follows.

$$f_{q,\mathbf{a}}^{IM} = \sum_{i=1}^{d} I_{\mathbf{a}}^{(i)} G_{a_i q}^{(i)} \tag{5.3}$$

The interaction matrix $\mathbf{G}^{(i)}$ is query-position-ad dependent, which is unknown and their entries could be either positive or negative. We treat them as the latent variables to be learned from the data set. However, the model has two issues: the observed pair a_i, q in training data are extremely sparse, meaning that the majority of entries of interaction matrix cannot be learned effectively. The second problem is that without any constraints on the interaction matrix, this model may overfit due to large number of parameters. To avoid these two problems, we place low-rank constraints on the interaction matrix G. The low-rank approximation is widely used in recommender systems [74]:

$$\mathbf{G}^{(i)} \approx \mathbf{Q}^{(i)^T}\mathbf{A} \tag{5.4}$$

Let k be the dimensionality of the latent factor vectors. $\mathbf{Q}^{(i)} \in \mathbb{R}^{k\times N_q}$ is the latent factor matrix for the queries on position i, $\mathbf{A} \in \mathbb{R}^{k\times N_a}$ is the latent factor matrix for the ads. Plugging Equation 5.4 back into the interaction model, we get

$$f_{q,\mathbf{a}}^{IM} = \sum_{i=1}^{d} I_{\mathbf{a}}^{(i)} \mathbf{Q}_{\cdot q}^{(i)^T} \mathbf{A}_{\cdot a_i} \tag{5.5}$$

Like the coefficients in Feature Model, the latent factor vectors of ads and queries $\mathbf{Q}_{\cdot q}^{(i)}$ and $\mathbf{A}_{\cdot a_i}$ could be assumed generating from a Gaussian prior,

$$\begin{aligned} \mathbf{Q}_{\cdot q}^{(i)} &\sim \mathcal{N}(0, \lambda_Q^{-1}\mathbf{I}), \quad i = 1, \cdots, d \\ \mathbf{A}_{\cdot a_i} &\sim \mathcal{N}(0, \lambda_A^{-1}\mathbf{I}) \end{aligned} \tag{5.6}$$

Correlations: However, the above model still fails to capture the connections across positions/ads. To learn the relationships between different positions/ads, we place a matrix-variate normal distribution [54] on
$\mathbf{Q} = \left[\text{vec}(\mathbf{Q}^{(1)}), \text{vec}(\mathbf{Q}^{(2)}), \cdots, \text{vec}(\mathbf{Q}^{(d)})\right]$ where vec($\cdot$) denotes the operator which converts a matrix into a vector in a column wise manner. Then,

$$p(\mathbf{Q}|\mathbf{\Omega}) = \mathcal{MN}_{N_q k\times d}(\mathbf{Q}|\mathbf{Q}', \mathbf{I}_{N_q k} \otimes \mathbf{\Omega})$$

where $\mathbf{Q}'$ is the mean matrix (e.g., zero mean $\mathbf{Q}' = \mathbf{0}_{N_q k\times d}$), $\mathcal{MN}_{u\times v}(\mathbf{X}|\mathbf{M}, \mathbf{U}, \mathbf{V})$ denotes a matrix-variate normal distribution with mean $\mathbf{M} \in \mathbb{R}^{u\times v}$, row covariance matrix $\mathbf{U} \in \mathbb{R}^{u \times u}$ and column covariance matrix $\mathbf{V} \in \mathbb{R}^{v \times v}$. The probability density function of the matrix-variate normal distribution is defined as

$$p(\mathbf{X}|\mathbf{M}, \mathbf{U}, \mathbf{V}) = \frac{\exp(-\frac{1}{2}\mathrm{tr}(\mathbf{U}^{-1}(\mathbf{X} - \mathbf{M})\mathbf{V}^{-1}(\mathbf{X} - \mathbf{M})^T))}{(2\pi)^{uv/2}|\mathbf{U}|^{v/2}|\mathbf{V}|^{u/2}}$$

where $\mathrm{tr}(\cdot)$ and $|\cdot|$ denote the trace and determinant, respectively, of a matrix. More specifically, here the row covariance matrix $\mathbf{I}_{N_q k}$ models the relationships between query latent features, and the column covariance matrix $\boldsymbol{\Omega}$ models the relationships between different $\mathbf{Q}^{(i)}$'s. In other words, $\boldsymbol{\Omega}$ models the relationships between positions. We can see that if $\mathbf{Q}' = \mathbf{0}_{N_q k \times d}$ and $\boldsymbol{\Omega} = \lambda_Q^{-1}\mathbf{I}$, the model $p(\mathbf{Q}|\boldsymbol{\Omega})$ is equivalent to Eq. 5.6.

Combined Models (CM): It is straightforward to combine the three models LB, FM and IM. The combined model could be simply:

$$f_{q,\mathbf{a}} = f_{q,\mathbf{a}}^{LB} + f_{q,\mathbf{a}}^{FM} + f_{q,\mathbf{a}}^{IM}$$

Obviously, the combined model is more expressive than any single model. With more flexibility, there is also risk of overfitting using the combined model. Therefore, regularization is more important for the combined model.

Historical CTR Regularization

Instead of using a zero-mean prior of matrix $\mathbf{Q}$, that is $\mathbf{Q}' = \mathbf{0}_{N_q k \times d}$, we can incorporate historical CTR of pair query q and ad a into the prior information. Here, all positional query latent factors $\mathbf{Q}_{\cdot q}^{(i)}$ are generated from the same corresponding query latent factor $\tilde{\mathbf{Q}}_{\cdot q}$.

$$\mathbf{Q}_{\cdot q}^{(i)} \sim \mathcal{N}(\tilde{\mathbf{Q}}_{\cdot q}, \lambda_Q^{-1}\mathbf{I}), \quad i = 1, \cdots, d$$

We call $\tilde{\mathbf{Q}}_{\cdot q}$ query latent factors and $\mathbf{Q}_{\cdot q}^{(i)}$ query positional latent factors. Then, we can place the matrix variate gaussian distribution on the $\mathbf{Q}_{\cdot q}^{(i)}$. We duplicate $\mathrm{vec}(\tilde{\mathbf{Q}})$

for d times $\mathbf{Q}' = [\mathrm{vec}(\tilde{\mathbf{Q}}), \cdots, \mathrm{vec}(\tilde{\mathbf{Q}})]$, such that it can be shaped into the same form of $\mathbf{Q}$. Similarly, we can further assume query latent factors $\tilde{\mathbf{Q}}$ is generated from a Gaussian prior. With query latent factors $\tilde{\mathbf{Q}}$ and ad latent factors $\mathbf{A}$, the click-through rate can be incorporated into optimization framework through traditional collaborative filtering techniques [75],

$$\mathrm{CTR}_{(q,a)} \sim \mathcal{N}(\tilde{\mathbf{Q}}_{\cdot q}^T \mathbf{A}_{\cdot a}, \lambda_c^{-1}\mathbf{I})$$

With the historical CTR regularization, we update the optimization problem as follows,

$$\begin{aligned} \mathrm{O}_c &= \sum_{q,\mathbf{a}} \ell(y_{q,\mathbf{a}}, f_{q,\mathbf{a}}) + \lambda_{\mathbf{b}}\|\mathbf{b}\|_2^2 + \lambda_{\tilde{\mathbf{Q}}}\|\tilde{\mathbf{Q}}\|_2^2 + \lambda_{\mathbf{A}}\|\mathbf{A}\|_2^2 \\ &\quad + kN_q \ln|\boldsymbol{\Omega}| + \mathrm{tr}((\mathbf{Q}-\mathbf{Q}')\boldsymbol{\Omega}^{-1}(\mathbf{Q}-\mathbf{Q}')^T) \\ &\quad + \lambda_c \sum_{q,a} (\mathrm{CTR}_{(q,a)} - \tilde{\mathbf{Q}}_{\cdot q}^T \mathbf{A}_{\cdot a})^2 + \mathrm{const} \end{aligned}$$

Optimization

With the proposed prediction models, we can formalize the click yield prediction problem into an optimization framework. The discrepancy between the estimation $f_{q,\mathbf{a}}$ and the true value $y_{q,\mathbf{a}}$ can be measured by a loss function. We can formalize the problem as an optimization problem as follows

$$\begin{aligned} \mathrm{O}_c &= \sum_{q,\mathbf{a}} \ell(y_{q,\mathbf{a}}, f_{q,\mathbf{a}}) + \lambda_{\mathbf{b}}\|\mathbf{b}\|_2^2 + \lambda_{\mathbf{A}}\|\mathbf{A}\|_2^2 + kN_q \ln|\boldsymbol{\Omega}| \\ &\quad + \mathrm{tr}(\mathbf{Q}\boldsymbol{\Omega}^{-1}\mathbf{Q}^T) + \mathrm{const} \end{aligned}$$

The choice of the loss function $\ell(\cdot)$ is also critical to the performance. We will discuss several possible choices for the click yield problem in this section and show the performance comparison of these loss functions on click yield.

One straightforward method is to treat the problem as a regression problem, where we could use the following pointwise loss functions.

Squared error loss (Gaussian): $\ell(y_{q,\mathbf{a}}, f_{q,\mathbf{a}}) = (y_{q,\mathbf{a}} - f_{q,\mathbf{a}})^2$, which is also known as Gaussian response in regression problems.

Huber loss (Huber):

$$\ell(y_{q,\mathbf{a}}, f_{q,\mathbf{a}}) = \begin{cases} \frac{1}{2}(y_{q,\mathbf{a}} - f_{q,\mathbf{a}})^2 & \text{if} |y_{q,\mathbf{a}} - f_{q,\mathbf{a}}| < \delta \\ \delta(|y_{q,\mathbf{a}} - f_{q,\mathbf{a}}| - \delta/2) & \text{otherwise} \end{cases}$$

This function is quadratic for small values of $|y_{q,\mathbf{a}} - f_{q,\mathbf{a}}|$, and linear for large values, with equal values and slopes of the different sections at the two points where $|y_{q,\mathbf{a}} - f_{q,\mathbf{a}}| = \delta$

ϵ-insensitive loss (SVR):

$$\ell(y_{q,\mathbf{a}}, f_{q,\mathbf{a}}) = \begin{cases} 0 & \text{if} |y_{q,\mathbf{a}} - f_{q,\mathbf{a}}| < \epsilon \\ |y_{q,\mathbf{a}} - f_{q,\mathbf{a}}| - \epsilon & \text{otherwise} \end{cases}$$

This loss function is used by support vector regression [36]. It has no penalty on any training data whose prediction is close enough to the ground truth (within a threshold ϵ).

The problem of click yields prediction essentially is to find the $\mathbf{a}$ which can generate maximum click yields given a query q. From this perspective, learning to rank methods are much preferable to the regression method.

Margin ranking criterion(MRC):

$$\ell^M(\mathbf{a}_1, \mathbf{a}_2) = \sum_{y_{q,\mathbf{a}_1} > y_{q,\mathbf{a}_2}} \max[0, 1 - f_{q,\mathbf{a}_1} + f_{q,\mathbf{a}_2}]$$

MRC considers all pairs of $y_{q,\mathbf{a}_1} > y_{q,\mathbf{a}_2}$, and assign each a cost if the negative label $f_{q,\mathbf{a}_2}$ is larger or within a "margin" of 1. This loss function has a problem that all pairwise violations are considered equally if they have the same margin violation, independent of their position in the list. For this reason the margin ranking loss might not optimize precision at k very accurately.

Weighted Approximately Ranked Pairwise loss (WARP): This loss, proposed in Usunier et al. [138], has been successfully applied in image retrieval tasks [142] and IR tasks [143]. The idea of WARP is to focus more on the top of the ranked list where the top k positions are those we care about, comparing to MRC where no notion of ranked list is introduced. By using the precision at k measure, one can weigh the pairwise violations depending on their position in the ranked list. WARP is defined as an error function as follows:

$$\text{WARP} = \sum_{q,\mathbf{a}} error(f_{q,\mathbf{a}}, y_{q,\mathbf{a}}) = \sum_{q,\mathbf{a}} L(rank(f_{q,\mathbf{a}})) \tag{5.7}$$

where $rank(f_{q,\mathbf{a}})$ is the rank of the ads list $\mathbf{a}$, given by $rank(f(q,\mathbf{a})) = \sum_{a' \in \mathcal{S}^{-}_{q,\mathbf{a}}} \mathbb{I}[f_{q,\mathbf{a}'} > f_{q,\mathbf{a}}]$, where $\mathbb{I}$ is the indicator function, $\mathcal{S}^{-}_{q,\mathbf{a}} = \{\mathbf{a}' | y_{q,\mathbf{a}'} < y_{q,\mathbf{a}}\}$. Here, rank measure $L(\cdot)$ is the function which transforms the rank to a loss:

$$L(k) = \sum_{j=1}^{k} \alpha_j, \text{with } \alpha_1 \geq \alpha_2 \geq \cdots \geq 0.$$

The idea of the rank function is to compute the violations where negative instances are ranked higher than the positive ones and the L function is to transform the violations into loss. Different choices of α define different importance of the relevance position: for $\alpha_j = 1, \forall i$, we have the same AUC optimization as margin ranking criterion. For $\alpha_j = 1$ and $\alpha_{j>1} = 0$, the precision at 1 is optimized and similarly for $\alpha_{j<N} = 1$ and $\alpha_{j \geq N} = 0$, the precision at N is optimized. For $\alpha_i = 1/i$, a smooth weighting over position is given, where most weight is given to the top position. It is also shown to be superior to other schemes of α and yields state-of-the-art performance [138]. In this work, we will also adopt this setting $\alpha_i = 1/i$.

From the Eq. 5.7, we notice that it is difficult to directly optimize WARP due to the discrete nature of indicator functions. In addition, since the number of negative instances is significantly larger than positive instances, the rank function is inefficient to be calculated. As in [142], Eq. 5.7 can be readily re-written as

$$\text{WARP} = \sum_{q,\mathbf{a}} \frac{L(rank(f_{q,\mathbf{a}})) \sum_{a' \in \mathcal{S}^{-}_{q,\mathbf{a}}} \mathbb{I}[f_{q,\mathbf{a}'} > f_{q,\mathbf{a}}]}{rank(f_{q,\mathbf{a}})}$$

with the convention 0/0=0 when the correct label y is top-labeled. The WARP can be approximated by the hinge loss $\max(0, 1 - f(q, \mathbf{a}) + f(q, \mathbf{a}'))$ instead of the indicator function, to make the loss function continuous [142]. To perform stochastic gradient descent (SGD) updates, another difficulty is that the rank function is still unknown without computing $f_{q,\mathbf{a}}$ for all $q, \mathbf{a}$. In order to approximate the rank function, for a given pair $q, \mathbf{a}$, one draws negative instances until one which violates the indicator function. Thus, the approximate $rank(f_{q,\mathbf{a}})$ by using $\lfloor \frac{D^{-}-1}{N} \rfloor$ where $\lfloor \cdot \rfloor$ is the flow function, D^{-} is the number of items in $\mathcal{S}^{-}_{q,\mathbf{a}}$ and N is the number of trials of sampling until a violating pair is found. This approximation only requires local knowledge of negative instances, making it possible for SGD style updating rule.

A Efficient Implementation

To solve the problem, we adopt a hybrid optimization approach which mixes stochastic gradient descent and co-ordinate descent. In each iteration, we first perform stochastic gradient descent on $\mathbf{b}, \mathbf{Q}, \mathbf{A}, \tilde{\mathbf{Q}}$. Then, we update $\mathbf{\Omega}$ by the rule $\mathbf{\Omega} \leftarrow \frac{1}{N_q k}(\mathbf{Q} - \mathbf{Q}')^T(\mathbf{Q} - \mathbf{Q}')$ which is obtained by taking the derivative $\mathcal{O}_c$ with respect to $\mathbf{\Omega}$ and setting it to 0. Thus, for each iteration, the time complexity is $O(dkM)$, where d denotes max depth, k denotes the dimensions of latent factors and M is the number of records. In practice, d and k usually are very small (e.g. $d = 4$, $k = 10$), and can be treated as constants. Then, the time complexity for a single iteration will be $O(M)$. Assuming the algorithms need z iterations to get converged, the total time complexity will be $O(zM)$ which allows the algorithms to handle large-scale data.

5.3.3 Harvesting the Textual Information of Ads

Although the click yield optimization framework proposed in the previous subsection could utilize many features for the prediction task, there are still issues with leveraging the content features, because the words representation of ad content is extremely sparse. Usually, an ad body only contains around 10 to 20 words. Bag-of-words features (e.g., term frequency or TFIDF) cannot effectively capture the underlying semantics. On the other hand, a latent factor model does not explicitly incorporate ad content, such that it cannot handle cold start ads (the ads do not exist in the training data), which frequently occur in real systems. Topic models [15, 171, 61] have been developed to learn the latent semantics of texts. A number of approaches, such as LDA [15] and sparse topical coding [171], could be used in our problem. In this work, we adopt the topical coding model [171] to incorporate ad content into our click yield optimization framework. The reasons for using topical coding are as follows: 1) the topical coding model is a non-probabilistic formulation of topic models for discovering latent representations of large collections of data. It can easily fit into our optimization framework. 2) it can directly control the sparsity of inferred representation by using an appropriate regularizer. 3) the integrated model can be solved efficiently. We treat the content of each ad as a document. For simplicity, we use a as the index of an ad. Let $V = \{1, ..., N_v\}$ be the vocabulary with N_v words. Then w_{an} represents the raw word count of term n in ad a. Let $\theta_a \in \mathbb{R}^k$ represent the document code, playing a similar role as $P(z|\theta)$ in traditional topic models like probabilistic latent semantic analysis (PLSA) [61] or latent Dirichlet allocation (LDA) [15] . Similarly, $\beta \in \mathbb{R}^{k \times N}$ be a dictionary with k bases, which can be treated as $P(z|w)$ in LDA or PLSA.

For θ, due to the connection with the click yield model, we can place either Gaussian prior $p(\theta) \propto \exp(-\lambda_\theta \|\theta\|_2^2)$ or Laplace prior $p(\theta) \propto \exp(-\lambda_\theta \|\theta\|_1)$ for

L_2 and L_1 regularization respectively rather than Dirichlet prior. Then let $\mathbf{\Theta} = \{\theta_a\}_{a=1}^{N_a}$ denote the codes for a collection of ads $\{\mathbf{w}_a\}_{a=1}^{N_a}$. We learn the parameter by maximimum a posteriori (MAP) estimation. Unlike [171], since we only use ad level topical coding in our methods, we remove word level coding. Then we have

$$\begin{aligned} \mathcal{O}_t = \min_{\boldsymbol{\theta},\beta} \sum_{a,n} - \log Poiss(w_{an}; \theta_a^T \beta_{.n} + \delta \cdot \beta^b_n) + \lambda_\theta \sum_d \|\boldsymbol{\theta}_d\|_2^2 \\ \text{s.t. :} \theta_a \geq 0, \forall a; \beta_k \in \mathcal{P}, \forall k, \delta > 0, \beta^b \in \mathcal{P}, \end{aligned}$$

where $\ell(\theta_a, \beta)$ is Poisson loss, and alternatively we can place a background topic β^b_n.

$$\begin{aligned} \ell(\theta_a, \beta) &= - \log Poiss(w_{an}; \theta_a^T \beta_{.n} + \delta \cdot \beta^b_n) \\ &= -w_{an} \log(\theta_a^T \beta_{.n} + \delta \cdot \beta^b_n) + \theta_a^T \beta_{.n} + \delta \cdot \beta^b_n \end{aligned}$$

The optimization problem can be solved efficiently due to three facts: 1) the property of multilinearity [111], which means that the model is linear with respect to each model parameter when others are fixed, 2) Proposition 1 in [171] states that the optimal value of a single parameter when others are fixed is the maximum between zero and the value obtained by a non-constrained version of the same problem. 3) efficient methods [37] exist to project real-valued vectors onto the simplex.

To incorporate the topical coding into our click yields model, we explore two methods, one through features and the other through latent factors.

Topical Coding Through Features

In this form, we put the topical coding of ads into ad features and the new ad features become $\tilde{\mathbf{x}}_a = [\mathbf{x}_a^T, \theta_a^T]^T$. With the shared model parameter θ, we connect the two optimization problem $\mathcal{O}_c$ and $\mathcal{O}_t$. This form is similar to the supervised topic model [14] and MedLDA [170]. The final optimization problem is

$$\mathcal{O}_c + \lambda \cdot \mathcal{O}_t$$

where λ is a weight parameter for $\mathcal{O}_t$. Then, we perform stochastic gradient descent to update θ, referring to the total optimization.

When passing a sample of click $y_{q,\mathbf{a}}$ and $a \in \mathbf{a}$, we have the following updates

$$\theta_a \quad \leftarrow \quad \mathcal{P}\left\{\theta_a - \eta \frac{\partial \mathcal{O}_c(y_{q,\mathbf{a}})}{\partial \theta_a}\right\}$$

Similarly, when passing a sample of word count w_{an}, we have

$$\theta_a \quad \leftarrow \quad \mathcal{P}\left\{\theta_a - \eta \frac{\partial \mathcal{O}_t(w_{an})}{\partial \theta_a}\right\}$$

where $\mathcal{P}\{\cdot\}$ is the projection function which truncates θ_a and guarantees $\theta_a \geq 0$

Topical Coding Through Latent Factors

An alternative method that can incorporate topical coding of ads is through latent factors. Without constraints, the latent representations of an ad learned from $\mathcal{O}_c$ and $\mathcal{O}_t$ are different. To connect the two problems, we can place constraints on the two latent representations of ads, such that the information of ad content can be incorporated into click yields optimization. One natural approach to require the two latent factors be the same:

$$\theta_a = \mathbf{A}_{.a}, \forall a$$

In this method, the objective function will be $\mathcal{O}_c + \lambda \cdot \mathcal{O}_t$ that is the same as the one in Sec. 5.3.3 and similar optimization methods (e.g., stochastic gradient descent) can be used to solve the problem. Another more flexible approach is to use a regularizer to keep the two latent factors similar rather than identical, as:

$$\lambda_{\theta\mathbf{A}} \|\theta_a - \mathbf{A}_{.a}\|_2^2, \forall a$$

The objective function will be $\mathcal{O}_c + \lambda_{\theta\mathbf{A}} \sum_a \|\theta_a - \mathbf{A}_{.a}\|_2^2 + \lambda \cdot \mathcal{O}_t$. We can also view one latent factor as a sample drawing from a multivariate normal distribution with the mean of the other latent factor: $\mathbf{A}_{.a} \sim \mathcal{N}(\theta_a, \lambda_{\theta\mathbf{A}}^{-1}\mathbf{I})$. This will produce the formalism in Wang and Blei's work [139]. Also, the SGD-style optimization can be used to solve the problem. In our studies, we find the performances of the two methods are comparable and we will use the first one in the following experiments.

5.3.4 Experiments

In this subsection, we analyze variations of the proposed model and compare our model with the state-of-the-art methods.

Experiment Setting

We use the same search log data as in Section 5.2. (collected from a commercial search engine in the U.S. market in April 2012 and the first two weeks of May 2012). We used April 2012 dataset as training set, and the May 2012 dataset as the test set. Similarly as in Section 5.2, the train/test data are split temporally rather than randomly, and we avoid the problem of predicting the past using future data. This evaluation is more consistent with a real application scenario. At the same time, the cold start problem becomes serious.

To evaluate the performance of click yields prediction, we report the standard ranking measurements—Mean Average Precision (MAP) and Precision@N for references.

By using our click yields prediction framework, with an issued query, we can reselect the group of ads to display with the higher predicted click yields, and it will improve the performance of current search engine. To evaluate this relative improvement, we present a novel evaluation metric, referred to as Relative Gain of Click Yields (RGCY). The overall click yields of the current system is $\underline{\mathrm{CY}} = \frac{1}{\mathrm{Imp}} \sum_q \sum_{\mathbf{a}} y_{q,\mathbf{a}} \cdot \mathrm{Imp}(q, \mathbf{a})$. Given a query q, ideally, if we know the optimal $\mathbf{a}$, which can generate the maximum click yield, that is for all impressions of q, one can always show this optimal result $\mathbf{a}_{opt(q)}$ which is defined $\mathbf{a}_{opt(q)} \in \{\mathbf{a}'|\forall\mathbf{a}, y_{q,\mathbf{a}'} \geq y_{q,\mathbf{a}}\}$. The maximum click yield which the system could have is $\overline{\mathrm{CY}} = \frac{1}{\mathrm{Imp}} \sum_q \tilde{y}_q \sum_{\mathbf{a}} \mathrm{Imp}(q, \mathbf{a})$ where $\tilde{y}_q = y_{q,\mathbf{a}_{opt(q)}}$ denotes the optimal click yield for query q.

Given a query q and its ranked ad lists $l_q = \{\mathbf{a}_{(1)}, \mathbf{a}_{(2)}, \cdots, \mathbf{a}_{(n)}\}$ where $f_{q,\mathbf{a}_{(1)}} \geq f_{q,\mathbf{a}_{(2)}} \cdots \geq f_{q,\mathbf{a}_{(n)}}$, $\mathrm{CY@}N = \frac{1}{\mathrm{Imp}} \sum_q \hat{y}_q \sum_{\mathbf{a}} \mathrm{Imp}(q, \mathbf{a})$ where $\hat{y}_q$ is the average click yields of top N ad lists for query q, $\hat{y}_q = \frac{1}{N} \sum_{i=0}^{N} y_{q,\mathbf{a}_{(i)}}$. We define the relative click yield gain (RCYG) to be

$$\mathrm{RGCY@}N = \frac{\mathrm{CY@}N - \underline{\mathrm{CY}}}{\overline{\mathrm{CY}} - \underline{\mathrm{CY}}}$$

It is trivial to see that $\overline{\mathrm{CY}} \geq \mathrm{CY@}N \geq \underline{\mathrm{CY}}$. The RGCY@$N$ actually measures the gap with the maximum click yields. More specifically, RGCY@1 is more critical to the system since it can generate the maximum click yields.

Model Analysis

In this subsection, we detail the anatomy of the proposed model and systematically analyze the contributions and effects of each part.

Estimation Model Analysis: we first test the scoring model. The results are shown in the Figure 5.13. In this experiment, the performance is evaluated by PREC@1 and MAP, and the loss function is WARP. We can see that the performances shown are consistent in terms of PREC@1 and MAP. The bias model performs the worst on the left. With the feature model added, the model performs better. When we further incorporate the interaction model into the system, the model achieve the best performance in both PREC@1 and MAP.

Convergence Check: We also show the learning curve of the combined in Figure 5.14, where two loss functions are tested: Gaussian response and WARP. The convergence of the two loss functions can be achieved after 100 iterations. We

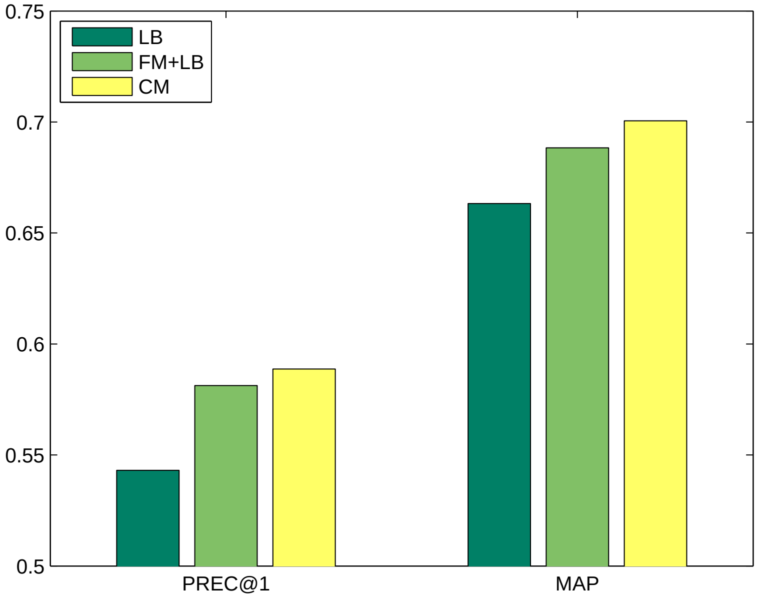


Figure 5.13: Model Analysis (Click Yields)

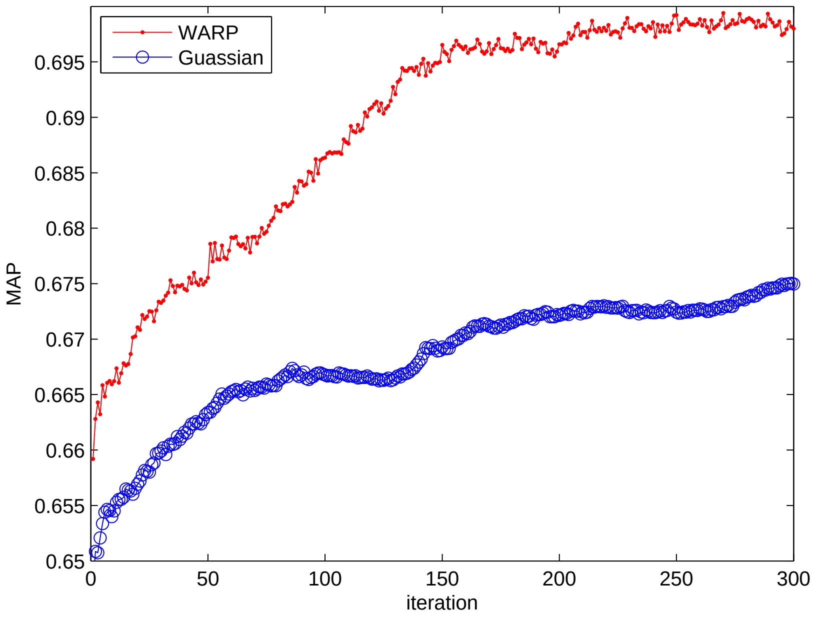


Figure 5.14: WARP vs. Gaussian

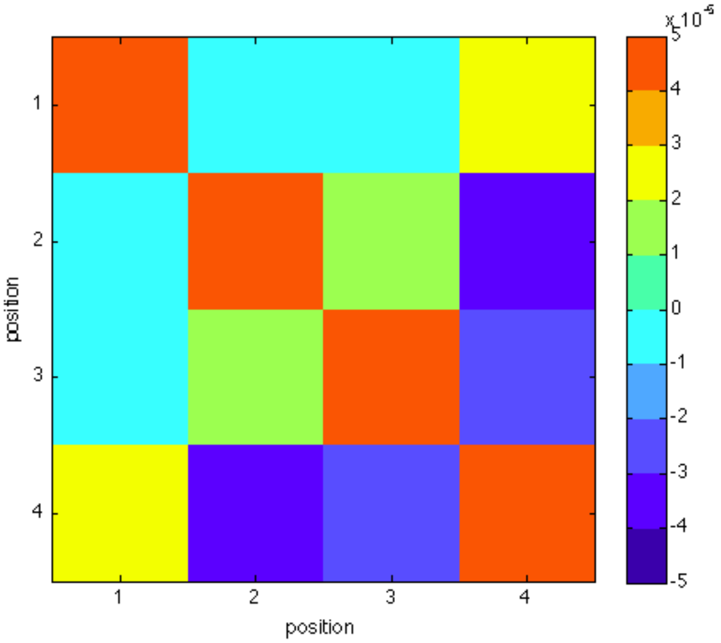


Figure 5.15: Co-relation across positions

can also see that the performance of the ranking loss (WARP) is much better than the regression loss (Gaussian response).

Position Correlation Analysis: Matrix $\mathbf{\Omega}$ is shown in Figure 5.15. We can see from the figure that position 1 is positively correlated to the position 4, while position 1 is negatively correlated to position 2 and position 4. This results are consistent with our intuitions: if the ads in position 2 and position 3 are similar to the ads in the top position, users are more likely to skip the ads due to information redundancy, which may decrease click yields. On the other hand, the ads at the bottom position (position 4) are positively correlated with position 1. One possible reason is that position 4 is far apart from the top position, and similarity between them is insensitive such that the click yields might increase.

Loss Function: From Figure 5.14, we have already seen the performance differences between different loss functions (WARP and Gaussian). In this experiment, all five loss functions in Section 5.3.2 are systematically compared. The results are shown in Table 5.9. For pointwise regression loss, we can see that the least square loss and ϵ-insensitive loss (SVR) can generate comparable performance. Huber loss, which is quadratic for small errors and linear for large errors, can achieve slightly better performance than the two other pointwise regression losses in this problem. On the other hand, for pairwise ranking loss, WARP can be treated as a weighted version of MRC, which emphasizes the top positions weight. From Table 5.9, we can see that the overall performance (MAP) of WARP is similar to MRC, while the performance of WARP on the RGCY@1 which only evaluates the accuracy of top position (RGCY@1 can be considered as the weighted version of MRC) is much better than MRC. Overall, comparing with the pointwise regression losses, the two pairwise rank losses are preferable in click yields prediction problem in all three evaluations (PREC@1, MAP and RGCY@1).

Side Information

For the topical coding model, we explore how topics are learned. The dictionary $\boldsymbol{\beta}$ can be interpreted as a topic matrix as in standard topic models. We can describe topics as in other topic models by ranking terms in probabilities. We show some example topics in Table 5.10. We can see that these topics can be easily recognized. Moreover, modeling ad content is not only useful for explanatory analysis, it indeed improves the prediction tasks. From Table 5.9, we can see that by incorporating the topical coding of ads, we get further improvement on RGCY@1.

Next, we examine the effects of two side information: historical CTR regularization and topical modeling of ad content. The results are shown in Table 5.9, where CTR represents historical CTR regularization, TCF represents topical model incorporated through features, TCL represents topical model incorporated through latent factors. We see that the performances get slightly improved across all loss

Table 5.9: Predictive results of Click Prediction

Side	loss	PREC@1	MAP	RGCY@1
N/A	Gaussian	0.5683	0.6792	0.6475
CTR	Gaussian	0.5689	0.6796	0.6510
CTR,TCF	Gaussian	0.5789	0.6894	0.6592
CTR,TCL	Gaussian	0.5771	0.6887	0.6701
N/A	Huber	0.5694	0.6811	0.6617
CTR	Huber	0.5695	0.6811	0.6618
CTR,TCF	Huber	0.5714	0.6834	0.6627
CTR,TCL	Huber	0.5730	0.6854	0.6679
N/A	ϵ-Insensitive	0.5785	0.6886	0.6365
CTR	ϵ-Insensitive	0.5787	0.6889	0.6401
CTR,TCF	ϵ-Insensitive	0.5824	0.6925	0.6402
CTR,TCL	ϵ-Insensitive	0.5833	0.6941	0.6416
N/A	MRC	0.5839	0.6930	0.6641
CTR	MRC	0.5848	0.6932	0.6661
CTR,TCF	MRC	0.5849	0.6936	0.6697
CTR,TCL	MRC	0.5859	0.6946	0.6732
N/A	WARP	0.5841	0.6923	0.6685
CTR	WARP	0.5845	0.6924	0.6782
CTR,TCF	WARP	0.5851	0.6936	0.6829
CTR,TCL	WARP	0.5859	0.6943	0.6882

Table 5.10: Examples of topics are shown.

Shopping	**Video**	**Travel**	**Finance**	**Game**
low	tv	book	online	games
free	order	hotels	credit	play
order	free	rates	apply	store
shop	time	price	card	online
shipping	watch	insurance	get	free
quality	college	car	university	today
today	live	parts	gift	guarantee
orders	24	hotel	degree	official
high	take	great	website	fun
prices	like	best	back	enjoy
supplies	sears	auto	earn	favorite
site	movies	low	college	kids
state	community	quotes	magazine	better
official	episodes	deals	cash	toys

functions when adding historical CTR regularization into system. When adding topical information, we see that the performance becomes much better than the original model, due to alleviating the cold-start and sparseness problems. We notice that WARP loss with CTR regularization and incorporated topical coding through latent factors achieve relative better performance than other combinations (although MRC is slightly better than WARP for MAP under CTR,TCL). In the following sections, we will use it to compare with the other existing methods.

Comparison with existing methods

In this subsection, we name our methods CYP for Click Yield Prediction. Although there is no previous work on the click yield prediction problem, some models could be easily adapted to solve the problem. We compare our model with four existing methods:

- **Probabilistic Matrix Factorization (PMF)** is a popular method [119] in collaborative filtering. Here, we treat queries as the users and ads as the items to be recommended for users. CTR is the response in this framework.
- **User Browsing Model (UBM)** is the baseline method and a classical click model [38]. We adopt the EM inference process for parameter α and β. To avoid infinite values in log-likelihood, α is truncated into the range $(0.0001, 0.9999)$.
- **Matrix Factorization Click Model (MFCM)** is proposed by Shen et al. [123]. The model incorporates collaborative filtering techniques into the click model.
- **Relational click model (CRF)** proposed by Xiong et al. [147]. They adopt Conditional Random Fields (CRF) to model the exclusive effects between similar ads which are shown together in the same impression.

We choose these four methods to compare to various alternative methods to solve this problem. PMF is a classical recommender method, UBM is a classical click model while CFCM and CRF are very recent methods and the state-of-the-arts in modeling clicks. We also compare it with Context-Aware Click Model (CACM) presented in Section 5.2. All five methods follows the classical two-step approach, that is, predicting CTR of individual ad at first and then compose the click yields for ad lists and rank them. The comparison results (on precision, RGCY and MAP) are shown in Table 5.11. We notice that the performances across five evaluations are consistent: the two classical methods—PMF and UBM—shown similar performance on this problem, and they are relatively worse than the two recent click models—MFCM, CRF. MFCM which incorporates examination hypothesis into matrix factorization, can get slightly better results than matrix factorization. CRF is the first click model which tries to model the relational ad clicks on the ads displayed

Table 5.11: Comparison with existing methods

Model	PREC@1	PREC@5	MAP	RGCY@1	RGCY@5
CYP	**0.5859**	**0.7428**	**0.6943**	**0.6882**	**0.7438**
PMF	0.5251	0.6828	0.6450	0.4991	0.5597
UBM	0.5338	0.6845	0.6647	0.4519	0.5015
MFCM	0.5553	0.6985	0.6754	0.5830	0.6293
CRF	0.5760	0.7320	0.6887	0.5975	0.7175
CACM	0.5676	0.7321	0.6858	0.6508	0.7051

together. We see that CRF and CACM achieve similar performance overall, while CACM is better than CACM in RGCY@1. The performances of CRF and CACM are much better than other three baselines which only focus on modeling clicks on single ad. However, for the click yield prediction problem, because all these five methods are two-step approaches and not designed to optimize click yields directly, our method outperforms all these methods noticeably. Especially, on RGCY@1, our methods can achieve 0.6882 which almost improves the click yields 10% absolutely, comparing to 0.5975—the value of the best comparison method, CRF.

5.4 Summary

In this chapter, we have studied contextual factors of the problem of predicting click-through rates for sponsored search. From our data analysis, we have found that these factors play important roles in understanding and predicting user clicks on ads and include factors such as ad depth and the interaction between the target ad and its context. A novel Context Aware Click Model for sponsored search was proposed, based on this analysis. We incorporated the context factors into the click model and conducted extensive experiments on a large-scale real-world dataset. Experiments in which different combinations of our model have been tested also verified the findings and conjectures in our data analysis. By adopting the Depth

Dependent Examination model and combining a latent bias model, feature model and interaction model into an informational relevance model, we produced significant improvement in CTR estimation. By comparing our methods with three strong baseline methods in multiple metrics, we showed that our approaches can outperform all three methods, which include two recent state-of-the-art methods in both the Description Oriented Evaluation and the Prediction Oriented Evaluation tasks.

Subsequently, we further studied advertising click yield prediction which is also a critical problem for modern search engines and provides a new perspective to measure group performance of ads displayed together. We then systematically explored different aspects: bias model, features, interactive influence, depth, and correlation across the position. Additionally, to best leverage the text features and solve the sparseness issue in textual information and cold starts on ads, we incorporate a topic coding model into our framework to learn the topic information of short texts for ads in two ways—through features and through latent factors. Finally, various loss functions are also studied. We find that ranking loss is preferred for this problem. We collected a large-scale real world dataset from a commercial search engine to conduct experiments. Our experiments show that our methods which directly predict the click yields achieve significant improvement, comparing with the existing two-step approaches. Our methods noticeably outperform the existing state-of-the-art approaches.

5.5 Bibliographic Notes

Related work is primarily in three topics: the users' click model, sponsored search and learning to rank techniques.

The bias in user click behavior during search was first studied through eye-tracking [66]. After that, many click models have been proposed to correct the biases and thus better model the users' click behaviors. There are two major assumptions in click models: the Examination Hypothesis and the Cascade Model.

The examination hypothesis assumes that if a displayed URL is clicked, it must be both examined and relevant [116]. Following the examination hypothesis, there are three basic models: the click over expected clicks (COEC) model [166], the examination model [24] and the logistic model [31]. They have been compared by O'Chapelle [24] and experimentally found to be outperformed by the cascade model. An important extension of the examination hypothesis is the user browsing model proposed by Dupret et al. [38, 87, 86]. It assumes the examination depends not only on the position, but also on the most recently clicked position in the same query session. Following this idea, Srikant et al. [128] propose user browsing models which attribute CTR changes to both changes in relevance and examination.

Another branch of the click model is the Cascade Model proposed by Craswell et al. [31], which assumes that the user views search results from top to bottom and decides whether to click each url. Once a click is issued, documents below the clicked result are not examined regardless of the position. Then the dependent click model [53] generalizes the cascade model to allow multiple clicks within a single session. The click chain model [52] and dynamic Bayesian network [24] (which is inferred through Infer.NET [97]) provide a further generalization by allowing the user to abandon examination of more results. The general click model [173] treats all relevance and examination effects in the model as random variables. In addition to the above methods, there have been several click models following the above two assumptions but apply them in different contexts, such as federated search [25], the task centric click model [168], and the position normalized click model [26].

However, the above methods are mainly designed for understanding user click behavior in organic search. They fail to explore the specific properties in sponsored search as we discussed before. Moreover, they cannot handle the cold start problem.

For modeling sponsored search and predicting ad click through rate, Graepel et al. [50] describe a Bayesian model that is based on a probit regression model. Richardson et al. [116] present a method based on logistic regression, where they extract features in four aspects: term CTR, ad quality, order specificity and external sources. In [166], Zhang et al. introduce the Clicks Over Expected Clicks (COEC) model. Cheng et al. [27] develop user-specific and demographic-based features that reflect the click behavior of individuals and groups to improve accuracy over the baseline model. Menon et al. [95] view the CTR prediction problems as a problem of matrix completion. Shen et al. [123] propose a personalized model, where they extend the matrix factorization to tensor factorization by involving the user factor. Although some of these methods adopt content-based features, they neglect the context information, including examination bias on different depths and ad mutual influence.

Recently, Xu et al. investigated the relational click behavior [150], but their method can only be applied in cases with two ads. They fail to model the more general cases (e.g., 3,4 ads) or handle new ads and queries. Xiong et al. [147] showed the mutual exclusive influence between similar ads, but their model fails to include positive effects between the ads. Also, they do not take examination bias into account and are essentially two step-approaches and do not directly estimate ad group performance.

Another area of related work is learning to rank techniques. In IR, a generic task is to construct a ranked list of documents relevant to a query issued by a user. Although ranking is a fundamental problem in IR and has been studied for decades, it still remains challenging. Recent methods are summarized in [88]. In the standard Learning to Rank setting, a typical training set consists of queries with their associated documents represented by feature vectors as well as corresponding relevance judgements. A machine learning algorithm is employed to learn the ranking model, which can predict the ground truth label in the training set as accurately as possible in terms of a loss function. In the test phase, when a new query comes in, the learned model is applied to sort the documents according to their relevance to the query, and return the corresponding ranked list to the user as the response to the query. Depending on different hypotheses, input spaces, output spaces and loss functions, approaches to LtoR can be loosely grouped into three categories [88]: point-wise, pairwise, and list-wise [99].

Chapter 6

Conclusion and Future Work

In this chapter, we summarize our research findings and contributions. We also discuss future research directions.

6.1 Summary

Online media has gained popularity in recent years. People get deeply involved in online media, and user behaviors in real life (e.g., reading news, sharing resources and shopping) have been able to be performed through online services. It attracts a great amount of research on understanding and predicting user behaviors in online media. In this thesis, we mainly study the problem—understanding and prediction of user online behaviors—in different contexts, such as social tagging prediction, link prediction and online advertising.

In terms of methodologies, we have designed specific models for different applications. Our contributions are summarized as follow:

- We proposed a novel probabilistic model for personalized tag prediction, which is a Bayesian approach, and integrates three factors—an ego-centric effect, environmental effects and web page content. Two methods—both intuitive calculation and learning optimization—are provided for parameter estimation.

- To model temporal dynamics in social tagging system, we proposed a user-tag-specific temporal interests model for tracking users' interests over time. The model stands on techniques introduced to address "concept drift" which imposes a continuous smoothing scheme over the timeline. Our model can benefit from the integration of topic switch detection and that temporal characteristics of social tagging systems are different from traditional concept drift problems.

- A novel personalized structure-based link prediction model is proposed in Chapter 3, based on a latent factor model. With structural regularization, it can incorporate structural information into the model.

- To make better predictions for a user in different contexts, we tackle these tasks by using a generalized latent factor model and Bayesian treatment. This model is able to predict users' different behaviors in specific contexts simultaneously and capture mutual effects across different contexts while the Bayesian treatment can handle the data sparsity which is a serious problem in social media data.

- For click behavior prediction in sponsored search, we propose a novel probabilistic model which adopts the Depth Dependent Examination model and combines a latent bias model, feature model and interaction model into an informational relevance model.

- In order to estimate ad group performance in sponsored search, we design a novel framework that directly predicts group performance for lists of ads. To best leverage the text features and solve the sparseness issue in textual information, we embed a topic coding model into our framework to learn the topical information of short text for ads.

In this dissertation, besides the methodological contributions, we have also made contributions in understanding behaviors in different contexts. Through data analysis, and experiments, we have the following contributions in four domains:

- **Social Tagging System**

  We suggest that social tagging by nature is an incremental process, and perform a time-sensitive sampling on an existing public dataset. Our analysis shows that in the real world, the problem of tag prediction is dominated by the need to predict tags for existing users when they tag new items. We proposed a novel probabilistic model for personalized tag prediction. Our online experiments and 5-fold cross validation experiments indicate that our model achieves over 30% improvement on F-measure compared to a leading method, in a "real-world" test scenario. Moreover, we investigated the temporal dynamics of user interests in tagging systems, and proposed a user-tag-specific temporal interests model for tracking users' interests. Using three public datasets we showed the impact of personalization and user-tag specification. Based on our experiments, we are able to conclude that our temporal user interests model, generated only from the temporal tag sequence, can outperform the state-of-the-art by more than 10% in F-measure for Bibsonomy data. Combining with either our probabilistic personalized tag prediction model or LHKM, performance further improved to 0.357 and 0.369, respectively. All three methods incorporating TIM can outperform the state-of-the-art as well as a leading algorithm addressing concept drift.

- **Microblogging System**

We examined the link structure and link prediction task within the Twitter microblogging network. In daily monitoring experiments, we analyzed properties of new links and saw from where in the network those links come and compared three sampling methods for the link prediction task. We proposed a novel personalized structure-based link prediction model and compared its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

- **Modeling Simultaneous Contexts in Online Social Media**

  We study social media relations involving high order interactions, sparsity and coupling of data across contexts. Our experiments show that in social media, there exist three problems and challenges: coupled high order interaction, data sparsity, and cold start. To make better predictions for a user in different contexts, we tackle these problem by using a generalized latent factor model and Bayesian treatment. For performance evaluation, we test on three real-world data sets from two domains. In social tagging systems, the user-comment-item and user-tag-item can be mutually inferred based on common latent factors and thus improve prediction performance, which has not been explored previously. In traditional collaborative filtering, we investigate the combination of temporal information, external information and user-item interaction. Our novel latent factor model can handle multiple activities, such as commenting within tagging systems and can do so simultaneously and demonstrate superiority over state-of-the-art methods [119, 125, 148]. Our experiments also show the advantage of employing a fully Bayesian treatment to boost the performance of point estimation when modeling high order relations.

- **Sponsored Search and Online Advertising**

  We studied contextual factors of the problem—predicting click-through rates for sponsored search. From our data analysis, we have found that these factors (such as ad depth and the interaction between the target ad and its context) play important roles in understanding and predicting user clicks on ads. A novel Context Aware Click Model for sponsored search was proposed, based on this analysis. Experiments in which different combinations of our model have been tested also verified the findings and conjectures in our data analysis. By comparing our methods with three strong baseline methods in multiple metrics, we showed that our approaches can outperform all three methods, which include two recent state-of-the-art methods in both the Description Oriented Evaluation and the Prediction Oriented Evaluation tasks.

  Moreover, we provide a new perspective to measure group performance of ads displayed together. We then systematically explored different aspects: bias model, features, interactive influence, depth, and correlation across the position. Various loss functions are also studied. We find that ranking loss is preferred for this problem. We collect a large-scale real world dataset from a commercial search engine to conduct experiments. Our experiments show that our methods which directly predict the click yields achieve significant improvement, comparing with the existing two-step approaches. Our methods noticeably outperform the existing state-of-the-art approaches.

Through this dissertation, readers learned characteristics, properties, challenges in predictive online media, especially in social tagging system, micro-blogging system and online advertising. We also expect that readers could acquire knowledge of most popular methods such as probabilistic model and latent factor model. In addition, some methodologies of modeling online media could be learned, which could handle some specific challenges in online media and may provide some hints of potential solutions for new problems.

6.2 Future Work

From the micro-view, based on this thesis, there are many possible extensions of the current approaches, either in terms of experiments or in terms of modeling in the future:

For social tagging, although manually tuned parameters can achieve a high performance, all the users share the same ego weight. We believe that different users should have different user profiles—personalized weights of ego-centric effect and environmental effects. In the future, a probabilistic analysis on the effects of neighboring users may be needed to make further improvements. About temporal dynamics, from the experiments, we also find that the personal topic switch is an important problem. The experimental results imply that further research and analysis are necessary on modeling personal topic switches.

In microblogging, while this work has focused on link prediction as a function of link structure, we also expect that content analysis and user profiles are likely to be important for link prediction in hybrid networks. Future work should investigate the value of user attributes and capturing user interests in a hybrid network.

In modeling simultaneous relations across multiple contexts, while the proposed algorithm can scale to hundreds of thousands of observations, it requires several hours to converge. It is necessary to explore deterministic approximate inference techniques such as variational Bayes to further improve the convergence speed and enable the possibility of using gradient descent algorithms instead of Gibbs sampling. Another possible extension to this work is also the use of more advanced factorization techniques such as Tucker decomposition [133]. The model could also be modified to include non-negative latent features to improve the interpretability or the results; our approach can be easily extended to this framework by using an exponential prior on the latent dimension, as done in the case of matrices [122] and tensors [11].

For sponsored search, several interesting points remain to be explored in the future: first, a more advanced query-dependent examination model would be helpful for handling the data sparseness and cold start problems. Secondly, research beyond the prediction of group performance in sponsored search would be valuable. A more interesting problem is, given a query, finding the best depth and the ad list that can generate the best click yields for search engines. This is a much harder problem, similar to a combinatorial optimization problem. Finally, although we mainly study top queries in this work, studying tail queries is also interesting and necessary as future work.

From the macro-view, there are multiple potential directions for prediction and recommendation in online media, based on the current research. Here, we present three directions:

6.2.1 Additional Resources

Online media has been popular for a couple of years. In the early stage, user behaviors are mainly focused on desktop internet. More recently, more new characteristics for online behaviors are shown in online media. The population of mobile internet is dramatically growing: one may not have a desktop but must have a cellphone. In mobile internet, users are involved into the online media in a much deeper level than desktop internet. User's geographical information, gesture, even call logs and contacts, etc., can be easily collected and modeled for better prediction; meanwhile these various information bring more challenges to recommender systems. Preliminary study on this direction has shown its value [94], but more efforts should be made in this area. For instance, when a user walks into a store, one can analyze users' interests based on users' past behaviors. In general, compared to traditional recommender system, new online media allow us to collect and model more various information such as time, location, store information to make better recommendation to users.

6.2.2 Privacy

On the other hand, the over exposure of user information may cause another problem—a reduction in privacy. One can imagine that it is very dangerous that if a model can predict users' home addresses by analyzing user personal data (e.g., users' past geographical information). On this aspect, the fact that users may not be willing to open their privacy related data will raise a problem—how to design a privacy-preserving recommender system? This could be always a balance between benefits of better prediction and user privacy [70]. Here, we would like to make a good prediction and recommendation for users while using fewer data, especially user privacy related data. The possible solutions could be in two directions: 1) by effectively selecting a subset of training, a model even is able to achieve better performance, such as in active learning. In this scheme, one can incorporate data privacy into the selection procedure to balance performance and privacy. 2) combine local model and central model. In this scheme, we can design a local model on user/client end, which is light weight model, running on user end and will utilize users' local privacy data. These data will be only kept in user end instead of uploading to server end. Meanwhile, we can also have a central model, based on cloud/server data. This model is more like traditional recommender system, however, it only model non-privacy data. When user triggers a request of recommendation/prediction, the central model and local will work together to achieve a better performance.

6.2.3 Large Scale Data

Another research direction is scalable learning algorithms. As the data in online media is dramatically growing today, scalable learning algorithms still remain a tremendous challenge. Several possible directions may provide solutions to this problem: 1) approximate methods could speed up the learning process without losing too much performance. For example, some NP-hard problems could be solved approximately in polynomial time. 2) parallel paradigm provides a potential direction of research is to exploit large computing clusters of commodity machines, adapting existing algorithms in such contexts [103, 126, 7]. However, most existing methods cannot directly fit into the current standard parallel paradigm such as MapReduce.

6.3 Conclusion

In this dissertation, we provide a comprehensive study on understanding and predicting user behaviors in online media. Each online service has its specific characteristics, and user behaviors are different across these online services. We analyze user online behaviors—in different domains, such as social tagging system, microblogging system, and online advertising system. Based on the analysis, we report our findings on different online media and propose specific models for users online behaviors. We conduct extensive experiments on large-scale real-world datasets. The experimental results show that we advance the state-of-the-art and the prediction and recommendation in online media can be tackled at a large scale. Future work may arise from three directions: modeling more resources, taking privacy into account and scalability of models.

Bibliography

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In *Proceedings of Mining and Learning with Graphs*, 2011.

[2] L. A. Adamic and E. Adar. Friends and neighbors on the web. *SOCIAL NETWORKS*, 25:211–230, 2001.

[3] R. Adams, G. Dahl, and I. Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In *Proceedings of the Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 1–9, Corvallis, Oregon, 2010. AUAI Press.

[4] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 19–28, New York, NY, USA, 2009. ACM.

[5] D. Agarwal, B.-C. Chen, and B. Long. Localized factor models for multi-context recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 609–617, New York, NY, USA, 2011. ACM.

[6] D. Agarwal, B.-C. Chen, and B. Pang. Personalized recommendation of user comments via factor models. In *EMNLP*, pages 571–582. ACL, 2011.

[7] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. Scalable inference in latent variable models. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 123–132, New York, NY, USA, 2012. ACM.

[8] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 33–42. ACM, 2012.

[9] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[10] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 501–510, New York, NY, USA, 2007. ACM.

[11] A. Bhattacharya and D. B. Dunson. Simplex factor models for multivariate unordered categorical data. *Journal of the American Statistical Association*, 107(497):362–377, 2012.

[12] C. Biancalana, A. Micarelli, and C. Squarcella. Nereau: a social approach to query expansion. In *Proceedings of the 10th ACM workshop on Web information and data management*, WIDM '08, pages 95–102, New York, NY, USA, 2008. ACM.

[13] M. Bilenko and M. Richardson. Predictive client-side profiles for personalized advertising. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 413–421, 2011.

[14] D. Blei and J. McAuliffe. Supervised topic models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128, Cambridge, MA, 2008. MIT Press.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[16] A. Blum. On-line algorithms in machine learning, 1996.

[17] L. Bottou and O. Bousquet. The trade-offs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 1161–168. MIT Press, 2008.

[18] R. Briggs and N. Hollis. Advertising on the web: Is there response before click-through? In *Journal Of Advertising Reserach*, 1997.

[19] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April 1998.

[20] A. Broder and V. Josifovski. Introduction to computational advertising, 2011. Retrieved from http://www.stanford.edu/class/msande239/.

[21] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1010–1015, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[22] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 238–245, New York, NY, USA, 2002. ACM.

[23] B. Cao, D. Shen, K. Wang, and Q. Yang. Clickthrough log analysis by collaborative ranking. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[24] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1–10, New York, NY, USA, 2009. ACM.

[25] D. Chen, W. Chen, H. Wang, Z. Chen, and Q. Yang. Beyond ten blue links: enabling user click modeling in federated web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 463–472, New York, NY, USA, 2012. ACM.

[26] Y. Chen and T. W. Yan. Position-normalized click prediction in search advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 795–803, New York, NY, USA, 2012. ACM.

[27] H. Cheng and E. Cantú-Paz. Personalized click prediction in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 351–360, New York, NY, USA, 2010. ACM.

[28] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, May 2008.

[29] M. Collins, S. DasGupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 617–624. MIT Press, 2001.

[30] D. Conry, Y. Koren, and N. Ramakrishnan. Recommender systems for the conference paper assignment problem. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 357–360, New York, NY, USA, 2009. ACM.

[31] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 87–94, New York, NY, USA, 2008. ACM.

[32] C. Danescu-Niculescu-Mizil, A. Z. Broder, E. Gabrilovich, V. Josifovski, and B. Pang. Competing for users' attention: on the interplay between organic and sponsored search results. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 291–300, New York, NY, USA, 2010. ACM.

[33] M. De Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher. How Does the Data Sampling Strategy Impact the Discovery of Information Diffusion in Social Media? In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, May 2010.

[34] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search*

and Data Mining, WSDM '08, pages 231–240, New York, NY, USA, 2008. ACM.

[35] Y. Ding and X. Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 485–492. ACM Press, 2005.

[36] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *NIPS*, pages 155–161, 1996.

[37] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of ICML '08*, 2008.

[38] G. E. Dupret and B. Piwowarski. 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*, SIGIR '08, pages 331–338, New York, NY, USA, 2008. ACM.

[39] F. Eisterlehner, A. Hotho, and R. Jäschke, editors. *ECML/PKDD Discovery Challenge Workshop (DC09)*, volume 497 of *CEUR Workshop Proceedings*, Sept. 2009.

[40] Facebook. Facebook home page. `http://www.facebook.com/`, 2011.

[41] Flickr. Flickr home page. `https://www.flickr.com/`, 2011.

[42] N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys)*, pages 67–74. ACM Press, 2008.

[43] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[44] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35:61–70, December 1992.

[45] S. Golder and S. Yardi. Structural predictors of tie formation in Twitter: Transitivity and mutuality. In *Proc. 2nd IEEE Int'l Conf. on Social Computing (SocialCom)*, Aug. 2010.

[46] S. Golder, S. Yardi, M. Marwick, and d. boyd. A structural approach to contact recommendations in online social networks. In *Proc. 2nd Workshop on Search in Social Media (SSM)*, June 2009.

[47] N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song. Predicting links and inferring attributes using a social-attribute network (san). *CoRR*, abs/1112.3265, 2011.

[48] Google. Google plus home page. `https://plus.google.com/`, 2011.

[49] O. Görlitz, S. Sizov, and S. Staab. Pints: Peer-to-peer infrastructure for tagging systems. In *Proceedings of the Seventh International Workshop on Peer-to-Peer Systems (IPTPS)*, Tampa Bay, USA, 2 2008.

[50] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft s bing search engine. In J. Frnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 13–20. Omnipress, 2010.

[51] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 540–547, 2009.

[52] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 11–20, New York, NY, USA, 2009. ACM.

[53] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 124–131, New York, NY, USA, 2009. ACM.

[54] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall, 2000.

[55] S. K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh. Nonnegative shared subspace learning and its application to social media retrieval. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1169–1178, New York, NY, USA, 2010. ACM.

[56] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th International Conference on World Wide Web*, pages 211–220. ACM Press, 2007.

[57] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[58] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 531–538, 2008.

[59] D. Hillard, S. Schroedl, E. Manavoglu, H. Raghavan, and C. Leggetter. Improving ad relevance in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 361–370, New York, NY, USA, 2010. ACM.

[60] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 289–296, San Francisco, CA, 1999. Morgan Kaufmann.

[61] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, January-February 2001.

[62] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: search and ranking. In *Proceedings of the 3rd European conference on The Semantic Web: research and applications*, ESWC'06, pages 411–426, Berlin, Heidelberg, 2006. Springer-Verlag.

[63] M. Hu and B. Liu. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177, 2004.

[64] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Proceedings of the 11th European*

Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pages 506–514, 2007.

[65] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proc. 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 538–543, 2002.

[66] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting click through data as implicit feedback. In *Proceedings of SIGIR '05*, 2005.

[67] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proc. 6th Int'l Conf. on Data Mining (ICDM)*, pages 340–349, 2006.

[68] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, March 1953.

[69] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[70] B. P. Knijnenburg, A. Kobsa, and H. Jin. Preference-based location sharing: are more privacy options really better? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2667–2676, New York, NY, USA, 2013. ACM.

[71] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[72] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.

[73] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456, 2009.

[74] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data*, 4(1):1–24, January 2010.

[75] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[76] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International World Wide Web Conference*, Apr. 2010.

[77] G. Lebanon and Y. Zhao. Local likelihood modeling of temporal text streams. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 552–559. ACM Press, 2008.

[78] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562, The MIT Press, 55 Hayward Street Cambridge, MA 02142-1493 USA, April 2001. MIT Press.

[79] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 631–636, New York, NY, USA, 2006. ACM.

[80] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. 12th Int'l Conf. on Information and Knowledge Management (CIKM)*, pages 556–559, 2003.

[81] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007.

[82] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proc. 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2010.

[83] Linkedin. Linkedin home page. `http://www.linkedin.com/`, 2011.

[84] M. Lipczak, Y. Hu, Y. Kollet, and E. Milios. Tag sources for recommendation in collaborative tagging systems. In *ECML PKDD Discovery Challenge*, 2009.

[85] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW*, pages 342–351, 2005.

[86] C. Liu, F. Guo, and C. Faloutsos. Bbm: bayesian browsing model from petabyte-scale data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 537–546, New York, NY, USA, 2009. ACM.

[87] C. Liu, F. Guo, and C. Faloutsos. Bayesian browsing model: Exact inference of document relevancfe from petabyte-scale data. *ACM Trans. Knowl. Discov. Data*, 4(4):19:1–19:26, Oct. 2010.

[88] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.

[89] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306. ACM Press, 2009.

[90] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 931–940, New York, NY, USA, 2008. ACM.

[91] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, New York, NY, USA, 2011. ACM.

[92] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems*, volume 16, pages 627–634. MIT Press, 2003.

[93] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 17–24, New York, NY, USA, 2007. ACM.

[94] C. Matyas and C. Schlieder. A spatial user similarity measure for geographic recommender systems. In *Proceedings of the 3rd International Conference on GeoSpatial Semantics*, GeoS '09, pages 122–139, Berlin, Heidelberg, 2009. Springer-Verlag.

[95] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 141–149, New York, NY, USA, 2011. ACM.

[96] N. Metropolis and S. Ulam. The Monte Carlo methods. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.

[97] T. Minka, J. Winn, J. Guiver, and A. Kannan. A click through model - sample code. `http://research.microsoft .com/en-us/um/cambridge/projects/infernet/docs/ Click%20through%20model%20sample.aspx` , 2009.

[98] S. Mohamed, K. A. Heller, and Z. Ghahramani. Bayesian exponential family pca. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1089–1096. MIT Press, 2008.

[99] T. Moon, A. Smola, Y. Chang, and Z. Zheng. Intervalrank: isotonic regression with listwise and pairwise constraints. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 151–160, New York, NY, USA, 2010. ACM.

[100] T. Murata and S. Moriyasu. Link prediction of social networks based on weighted proximity measures. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 85–88, 2007.

[101] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. In *(Technical Report CRG-TR-93-1). Department of Computer Science, University of Toronto.*, 1993.

[102] Netflix. Netflix home page. `https://www.netflix.com/`, 2011.

[103] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10:1801–1828, Dec. 2009.

[104] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, July 1980.

[105] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[106] P. Papadimitriou, H. Garcia-Molina, P. Krishnamurthy, R. A. Lewis, and D. H. Reiley. Display advertising impact: search lift and social influence. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1019–1027, 2011.

[107] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[108] I. Porteous, A. Asuncion, and M. Welling. Bayesian matrix factorization with side information and Dirichlet process mixtures. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 563–568, Atlanta, Georgia, USA, July 2010. AAAI Press, Menlo Park, California.

[109] M. Rajih, P. Comon, and R. A. Harshman. Enhanced line search: A novel method to accelerate PARAFAC. *SIAM J. Matrix Anal. Appl.*, 30:1128–1147, 2008.

[110] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 54–63, New York, NY, USA, 2009. ACM.

[111] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[112] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736, New York, NY, USA, 2009. ACM.

[113] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[114] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM 2010: Proceedings of the Second International Conference on Web Search and Web Data Mining*, New York, NY, USA, 2010. ACM.

[115] Renren. Renren home page. `http://www.renren.com/`, 2011.

[116] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 521–530, New York, NY, USA, 2007. ACM.

[117] D. Romero and J. Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010.

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

[119] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[120] J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*, pages 502–507, 1986.

[121] K.-U. Schmidt, T. Sarnow, and L. Stojanovic. Socially filtered web search: an approach using social bookmarking tags to personalize web search. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 670–674, New York, NY, USA, 2009. ACM.

[122] M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, ICA '09, pages 540–547, Berlin, Heidelberg, 2009. Springer-Verlag.

[123] S. Shen, B. Hu, W. Chen, and Q. Yang. Personalized click model through collaborative filtering. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 323–332, New York, NY, USA, 2012. ACM.

[124] Sina. Weibo home page. `https://www.weibo.com/`, 2011.

[125] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on*

Knowledge discovery and data mining, KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.

[126] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3(1-2):703–710, Sept. 2010.

[127] Y. Song, L. Zhang, and C. L. Giles. A sparse gaussian processes classification framework for fast tag suggestions. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 93–102, New York, NY, USA, 2008. ACM.

[128] R. Srikant, S. Basu, N. Wang, and D. Pregibon. User browsing models: relevance versus examination. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 223–232, New York, NY, USA, 2010. ACM.

[129] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online Bayesian recommendations. In *Proceedings of the 18th International Conference on World Wide Web*, pages 111–120. ACM, April 2009.

[130] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50, New York, NY, USA, 2008. ACM.

[131] R. Tomioka, K. Hayashi, and H. Kashima. On the extension of trace norm to tensors. In *Proceedings of NIPS Workshop on Tensors, Kernels, and Machine Learning*, 2010.

[132] J. Travers, S. Milgram, J. Travers, and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[133] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. 10.1007/BF02289464.

[134] Tumblr. Tumblr home page. `https://www.Tumblr.com/`, 2011.

[135] Twitter. Twitter home page. `http://www.twitter.com/`, 2011.

[136] T. Tylenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. In *SNA-KDD '09: Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, pages 1–10, 2009.

[137] University of Kassel, Germany. Bibsonomy. `http://www.bibsonomy.org/`, 2010.

[138] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1057–1064, New York, NY, USA, 2009. ACM.

[139] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. ACM.

[140] X. Wang and et al. Click-through prediction for sponsored search advertising with hybrid models. In *KDDCUP 2012*, 2012.

[141] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: finding topic-sensitive influential twitterers. In *Proc. 3rd ACM Int'l Conf. on Web Search and Data Mining (WSDM)*, pages 261–270, 2010.

[142] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Mach. Learn.*, 81(1):21–35, Oct. 2010.

[143] J. Weston, C. Wang, R. Weiss, and A. Berenzweig. Latent collaborative retrieval. In *Proceedings of the 29th Annual International Conference on Machine Learning*, 2012.

[144] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. In *Machine Learning*, pages 69–101, 1996.

[145] Wordpress. Wordpress home page. `https://www.wordpress.com/`, 2011.

[146] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 723–732. ACM Press, 2010.

[147] C. Xiong, T. Wang, W. Ding, Y. Shen, and T.-Y. Liu. Relational click prediction for sponsored search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 493–502, New York, NY, USA, 2012. ACM.

[148] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of SIAM Data Mining*, 2010.

[149] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 155–162, New York, NY, USA, 2008. ACM.

[150] W. Xu, E. Manavoglu, and E. Cantu-Paz. Temporal click model for sponsored search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 106–113, New York, NY, USA, 2010. ACM.

[151] Yahoo! Delicious home page. `http://delicious.com/`, 2011.

[152] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *WWW*, pages 537–546, 2011.

[153] Y. K. Yilmaz, A. T. Cemgil, and Umutşimşekli. Generalised coupled tensor factorisation. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2151–2159, 2011.

[154] D. Yin, S. Guo, B. Chidlovskii, B. D. Davison, C. Archambeau, and G. Bouchard. Connecting comments and tags: improved modeling of social tagging systems. In *Proceedings of the sixth ACM international conference on Web search and data mining*, WSDM '13, pages 547–556, New York, NY, USA, 2013. ACM.

[155] D. Yin, L. Hong, and B. D. Davison. Exploiting session-like behaviors in tag prediction. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 167–168, New York, NY, USA, 2011. ACM.

[156] D. Yin, L. Hong, and B. D. Davison. Structural link analysis and prediction in microblogs. In *Proceedings of 20th ACM Conference on Information and Knowledge Management (CIKM 2011)*, October 2011.

[157] D. Yin, L. Hong, X. Xiong, and B. D. Davison. Link formation analysis in microblogs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1235–1236, New York, NY, USA, 2011. ACM.

[158] D. Yin, L. Hong, Z. Xue, and B. D. Davison. Temporal dynamics of user interests in tagging systems. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 1279–1285, August 2011.

[159] D. Yin, Z. Xue, L. Hong, and B. D. Davison. A probabilistic model for personalized tag prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 959–968, New York, NY, USA, 2010. ACM.

[160] Z. Yin, M. Gupta, T. Weninger, and J. Han. LINKREC: a unified framework for link recommendation with user attributes and graph structure. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1211–1212, 2010.

[161] Z. Yin, M. Gupta, T. Weninger, and J. Han. A unified framework for link recommendation using random walks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2010.

[162] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 957–966, New York, NY, USA, 2009. ACM.

[163] J. Yoo and S. Choi. Bayesian matrix co-factorization: Variational algorithm and cramer-rao bound. In *Proceedings of the ECML/PKDD 2011*, 2011.

[164] Youtube. Youtube home page. `https://www.youtube.com/`, 2011.

[165] D. Zhang, R. Mao, and W. Li. The recurrence dynamics of social tagging. In *Proceedings of the 18th International Conference on World Wide Web*, pages 1205–1206. ACM Press, 2009.

[166] W. V. Zhang and R. Jones. Comparing click logs and editorial labels for training query rewriting. In *Query Log Analysis: Social And Technological Challenges. A workshop at WWW 2007*, 2007.

[167] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 725–732, Corvallis, Oregon, 2010. AUAI Press.

[168] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1388–1396, New York, NY, USA, 2011. ACM.

[169] V. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the 24rd AAAI Conference on Artificial Intelligence*, 2010.

[170] J. Zhu, A. Ahmed, and E. P. Xing. Medlda: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1257–1264, New York, NY, USA, 2009. ACM.

[171] J. Zhu and E. Xing. Sparse topical coding. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 831–838, Corvallis, Oregon, 2011. AUAI Press.

[172] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 487–494, New York, NY, USA, 2007. ACM.

[173] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 321–330, New York, NY, USA, 2010. ACM.

Vita

1984 Born in Changle, Shandong Province, China.

2002 Graduated from the Changle No. 2 high school, Changle, China.

2006 B.E. in Computer Science, Shandong University.

2007 - 2008 Graduate study in Department of Computer Science, The University of Hong Kong.

2008 - 2013 Graduate study in Department of Computer Science and Engineering, Lehigh University.

2010 M.S. in Computer Science, Lehigh University.

2011 Research internship at Xerox Research Center Europe.

2012 Research internship at Microsoft Research Asia.

2013 Joined Yahoo! Labs as a Scientist.

2009 Xin Han, He Guo, Dawei Yin and Yong Zhang. **A note on on-line broadcast scheduling with deadlines**. *Information Processing Letters*, Volume 109 Issue 3, January 2009

2009 Dawei Yin, Zhenzhen Xue, Xiaoguang Qi and Brian D. Davison. **Diversifying Search Results with Popular Subtopics**. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*, Gaithersburg, MD: NIST, November 2009.

2009 Dawei Yin, Zhenzhen Xue, Liangjie Hong, B. D. Davison, A. Kontostathis, and L. Edwards. **Detection of Harassment on Web 2.0**. In *Proceedings of the Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009*, Madrid, Spain, April 2009.

2010 Dawei Yin, Henry S. Baird and Chang An. **Time and Space Optimization of Document Content Classifiers**. In *Proceedings of IS&T/SPIE Document Recognition and Retrieval Conf. (DR&R XVII)*, San Jose, CA, January 17-21, 2010.

2010 Dawei Yin, Chang An and Henry S. Baird. **Safely Selecting Subsets of Training Data**. In *Proceedings of IAPR 9th International Workshop on Document Analysis Systems (DAS 2010)*, Boston, MA, June 2010.

2010 Chang An, Dawei Yin, Henry S. Baird. **Document Segmentation using Pixel-Accurate Ground Truth**. In *Proceedings of IAPR 20th International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, August 2010.

2010 Dawei Yin, Chang An, Henry S. Baird. **Imbalance and Concentration in k-NN Classification**. In *Proceedings of IAPR 20th International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, August 2010.

2010 Dawei Yin, Zhenzhen Xue, Liangjie Hong and Brian D. Davison. **A Probabilistic Model for Personalized Tag Prediction**. In *Proceedings of the 16th Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*, Washington, DC, July 2010.

2010 Xiaoguang Qi, Dawei Yin, Zhenzhen Xue and Brian D. Davison. **Choosing Your Own Adventure: Automatic Taxonomy Expansion to Permit Many Paths**. In *Proceedings of 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*, Toronto, Canada, October 2010.

2011 Dawei Yin, Liangjie Hong and Brian D. Davison. **Exploiting Session-like Behavior in Tag Prediction**. In *Proceedings of the 20th International World Wide Web Conference (WWW 2011)*, Hyderabad, India, March 2011.

2011 Zaihan Yang, Dawei Yin and Brian Davison. **Award Prediction with Temporal Citation Network Analysis**. In *Proceedings of the 34th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, Beijing, China, July, 2011.

2011 Dawei Yin, Liangjie Hong, Xiong Xiong and Brian D. Davison. **Link Formation Analysis in MicroBlogs**. In *Proceedings of the 34th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, Beijing, China, July, 2011.

2011 Zhenzhen Xue, Dawei Yin and Brian D. Davison. **Normalizing Microtext**. In *Proceedings of the AAAI-11 Workshop on Analyzing Microtext*, San Francisco, USA, August, 2011.

2011 Dawei Yin, Liangjie Hong, Zhenzhen Xue and Brian D. Davison. **Temporal Dynamics of User Interests in Tagging Systems**. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, San Francisco, USA, August, 2011.

2011 Liangjie Hong, Dawei Yin, Jian Guo and Brian D. Davison. **Tracking Trends: Incorporate Volume into Temporal Topic Models**. In *Proceedings of the 17th Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, San Diego, CA, 2011.

2011 Dawei Yin, Liangjie Hong and Brian D. Davison. **Structural Link Analysis and Prediction in MicroBlogs**. In *Proceedings of 20th ACM Conference on Information and Knowledge Management (CIKM 2011)*, Glasgow, Scotland, UK, October 2011.

2013 Dawei Yin, Shenbo Guo, Boris Chidlovskii, Brian D. Davison, Cedric Archambeau and Guillaume Bouchard. **Connecting Comments and Tags: Improved Modeling of Social Tagging Systems**. In *Proceedings of 6th ACM Conference on Web Search and Data Mining (WSDM 2013)*, Rome, Italy, Feb 2013.

2013 Guillaume Bouchard, Shengbo Guo and Dawei Yin. **Convex Collective Matrix Factorization**. In *Proceedings of Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2013)*, Scottsdale, AZ, USA, April 2013.