Recommendation in Social Media: Utilizing Relationships among Users to Enhance Personalized Recommendation

by

Xin Wang

B.Sc., Zhejiang University, 2011

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Summer 2017

Approval

Name:	**Xin Wang**
Degree:	**Doctor of Philosophy (Computing Science)**
Title:	***Recommendation in Social Media: Utilizing Relationships among Users to Enhance Personalized Recommendation***
Examining Committee:	**Chair:** Qianping Gu Professor

Martin Ester
Senior Supervisor
Professor

Jiajun Bu
Supervisor
Professor
Zhejiang University

Jiannan Wang
Internal Examiner
Assistant Professor
School of Computing Science

Laks V.S. Lakshmanan
External Examiner
Professor
Department of Computer Science
The University of British Columbia

Date Defended: 29 June 2017

Abstract

Recommender systems are ubiquitous in our digital life in recent years. They play a significant role in numerous Internet services and applications such as electronic commerce (Amazon and eBay), on-demand video streaming (Netflix and Hulu). A key task in recommender systems is to model user preferences and to suggest, for each user, a personalized list of items that the user has not experienced, but are deemed highly relevant to her. Many of these recommendation algorithms are based on the principle of collaborative filtering, suggesting items that similar users have consumed.

With the advent of online social networks, social recommendation has become one of the most popular research topics in recommender systems, exploiting the effects of social influence and selection in social networks, where user relationships are explicit, i.e., there will be an edge connecting two users if they are friends. In addition, more information about the relationships between users in social media becomes available with the rapid development of various Internet services. For example, more and more online web services are providing mechanisms by which users can self-organize into groups with other users having similar opinions or interests, enabling us to analyze the interactions between users with others insides/outsides groups, as well as the engagement between users and groups. User relationships in these applications are usually implicit and can only be utilized indirectly for recommendation tasks.

In this thesis, we focus on utilizing user relationships (either explicit or implicit) to enhance personalized recommendation in social media. We study three problems of recommendation in social media, i.e., recommendation with strong and weak ties, social group recommendation and interactive social recommendation in an online setting. We propose to improve social recommendation by incorporating the concept of strong and weak ties which are two well documented terms in the social sciences, boost the performance of social group recommendation through modeling the temporal dynamics of engagement of users with groups, and tackle the interactive social recommendation problem via employing the exploitation-exploration strategy in an online setting. Our proposed models are all compared with state-of-the-art baselines on several real-world datasets.

Keywords: Recommendation; Personalization; Collaborative Filtering; User Behavior Modeling

Acknowledgements

First of all, I would like to express my sincere gratitude to my senior supervisor Prof. Martin Ester for the continuous support of my Ph.D study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. He has been acting as not only my supervisor but also my friend during theses years, and I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides, I would also like to thank the rest of my thesis committee: Prof. Jiannan Wang, Prof. Qianping Gu, and Prof. Laks V.S. Lakshmanan and Prof. Jiajun Bu for their insightful comments and feedback.

My sincere thanks also goes to Dr. Roger Donaldson, Dr. Peter Gorniak, and Mr. Christopher Nell, who provided me an internship position in their team at DeviantArt in summer 2014. Without their fantastic project and precious technical support, it would be impossible for the publication of my first top conference paper.

I thank my labmates at Simon Fraser University for the inspiring discussions and enjoyable days in the past years. I am particularly grateful to my friend Dr. Zhi Yu at Zhejiang University for helping me begin my initial research, and for teaching me related programming skills. Many thanks to my other friend Dr. Wei Lu in the University of British Columbia for his excellent writing skills and insightful ideas I learnt during our collaborations.

Last but not the least, I would like to thank my parents for their full support in spirit since the start of my Ph.D study, as well as their endless encouragement throughout my whole life.

Table of Contents

List of Tables

List of Figures

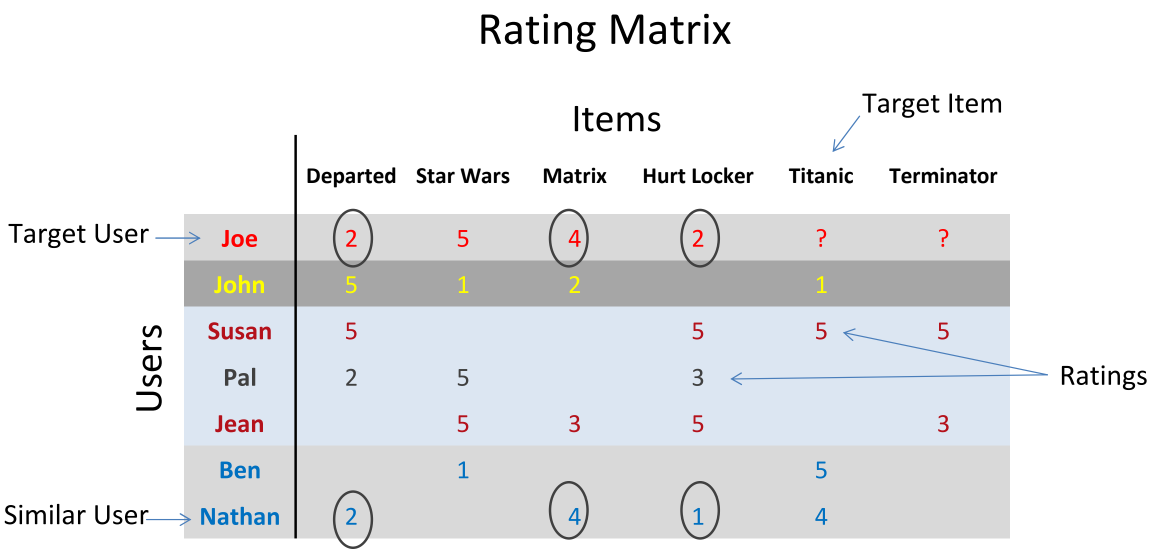
Chapter 1

Introduction

As more and more information becomes available on the Internet, people turn out to be more and more picky about the results that web services can provide with them. They want to have personalized results, but are not willing to spend a lot of time to specify their personal information needs. The advent of recommender systems enables an automatic identification of the information relevant for a given user through learning from available data. After a boosting development in the past decade, recommender systems have saturated into our daily life–we experience recommendations when we see “More Items to Consider” or “Inspired by Your Shopping Trends” on Amazon and “People You May Know” on Facebook–other popular online web services such as eBay, Netflix and LinkedIn etc. also provide users with the recommendation features. Thus algorithmic recommendation has become a necessary mechanism for many online web services which recommend items such as music, movies or books to users. These online web services normally make recommendations based on collaborative filtering, suggesting items favored by similar users. Low rank matrix factorization, as one of the most popular and representative collaborative filtering algorithms, has been widely used in many real-world recommender systems. In this chapter, we will first define the problem of recommendation in a user-item matrix view, then briefly discuss explicit and implicit feedback which are two common concepts in recommendation, followed by an overview about various applications of recommendation in social media. In the end, we would like to state our contribution in this thesis through presenting three recommendation problems in social media and how our proposed models can enhance the recommendation via utilizing user relationships in general.

1.1 Problem Definition

In recommender systems, we are given a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$, as well as a $|\mathcal{U}| \times |\mathcal{I}|$ rating matrix R whose non-empty (observed) entries R_{ui} represent the feedback (e.g., ratings, clicks etc.) of user $u \in \mathcal{U}$ for item $i \in \mathcal{I}$. The task is to predict (or rank) the missing values in R, i.e., given a user $v \in \mathcal{U}$ and an item $j \in \mathcal{I}$ for which R_{vj} is unknown, we predict (or rank) the rating of v for j using observed values in R.



	Departed	Star Wars	Matrix	Hurt Locker	Titanic	Terminator
Joe	2	5	4	2	?	?
John	5	1	2		1	
Susan	5			5	5	5
Pal	2	5		3		
Jean		5	3	5		3
Ben		1			5	
Nathan	2		4	1	4	

Figure 1.1: An example of rating matrix

Consider the example rating matrix in Figure 1.1 where lines are users ($\mathcal{U}$) and columns are items ($\mathcal{I}$). Each entry with a number (i.e., observed) in the matrix indicates the corresponding rating (R_{ui}) of a user (u) for an item (i). The task of recommender systems in this example then becomes, given the observed entries in the rating matrix, how can we predict the values for those unobserved entries (e.g., Joe's rating on Titanic and Terminator) or how can we produce a list of top-3 movies that liked most by Ben? Furthermore, the above two task options elicit the two objectives in recommendation:

- Rating prediction
  Predict the rating of target user for target item (not rated by this user).
- Top-K item recommendation
  Predict the top-K highest-rated items among the items not yet rated by target user.

Note that the values of ratings are all integers from 1 to 5, where 1 means a user dislikes an item, 5 indicates that the user likes or even favors the item. This often happens when a web service (e.g., IMDB) provides its users with the function of giving different numeric values (from a pre-selected set of values by the system) as ratings for items and higher value for an item expresses more preference for this item. We call these numeric ratings explicit ratings or explicit feedback. More often than not, we may also come across other applications in which no such explicit ratings are available. For instance, Delicious (`https://del.icio.us/`), which is a social bookmarking web service for storing, sharing, and discovering web bookmarks, allows its users to bookmark a url, resulting in a Boolean user-url bookmarking matrix whose entries are either 1 or 0 — 1 indicates that a user has bookmarked a url and 0 otherwise. We call these binary ratings implicit ratings or implicit feedback. Just as the name implies, explicit rating can express users' relative preferences over different items while implicit rating can not. Existing literature in recommendation can also be roughly divided into two categories according to which of the two it applies to. This being the case, we remark that our proposed models in this thesis fit into either implicit data or explicit data.

1.2 Explicit and Implicit Ratings/Feedback

In the above example where people have explicit ratings from 1 to 5 for different items, we can somehow infer their preferences over different items from their corresponding ratings, e.g., a user prefers the item rated 5 to the one rated 1 by himself. In such scenarios, a "good" rating prediction model will indirectly produce a good top-K raking list given the available information of explicit ratings. In other scenarios where only implicit ratings are available, we may only have a boolean value indicating whether a user has consumed an item ('rate' can be treated as a special case of 'consume' where the value of rating is 1), resulting in a boolean rating matrix whose entries are either 1 or 0. Top-K recommendation which gives a list of ranked items that each target user may like most will be more meaningful in practice and thus be more attractive for researchers when no explicit ratings are available. Actually we are surrounded by far more implicit feedback than explicit feedback in our daily lives and given the fact that inferring user preference from implicit feedback is more challenging, there is more existing work on implicit feedback than on explicit feedback.

1.2.1 Recommendation with explicit ratings

When it comes to recommender systems, collaborative filtering is one of the most popular algorithmic solutions so far, which makes recommendations based on users' past behaviors such as ratings, clicks, purchases and favorites etc. Further, low rank matrix factorization is among the most effective methods for collaborative filtering, and there is a large body of work on using matrix factorization for collaborative filtering [58, 59, 82, 98]. As a general treatment, Koren [60] gives a systematic introduction to the application of matrix factorization to recommender systems. Among the literature of matrix factorization, Salakhutdinov and Mnih [82] propose a probabilistic version of matrix factorization (PMF) which assumes a Gaussian distribution on the initialization of latent feature vectors, making the model more robust towards the problem of overfitting and linearly scalable with the number of observations at the same time.

Probabilistic Matrix Factorization in Recommendation

In recommender systems, we are given a set of users $\mathbb{U}$ and a set of items $\mathbb{I}$, as well as a $|\mathbb{U}|\times|\mathbb{I}|$ rating matrix R whose non-empty (observed) entries R_{ui} represent the feedbacks (e.g., ratings, clicks etc.) of user $u \in \mathbb{U}$ for item $i \in \mathbb{I}$. When it comes to social recommendation, another $|\mathbb{U}| \times |\mathbb{U}|$ social tie matrix T whose non-empty entries T_{uv} denote $u \in \mathbb{U}$ and $v \in \mathbb{U}$ are ties, may also be necessary. The task is to predict the missing values in R, i.e., given a user $v \in \mathbb{U}$ and an item $j \in \mathbb{I}$ for which R_{vj} is unknown, we predict the rating of v for j using observed values in R and T (if available).

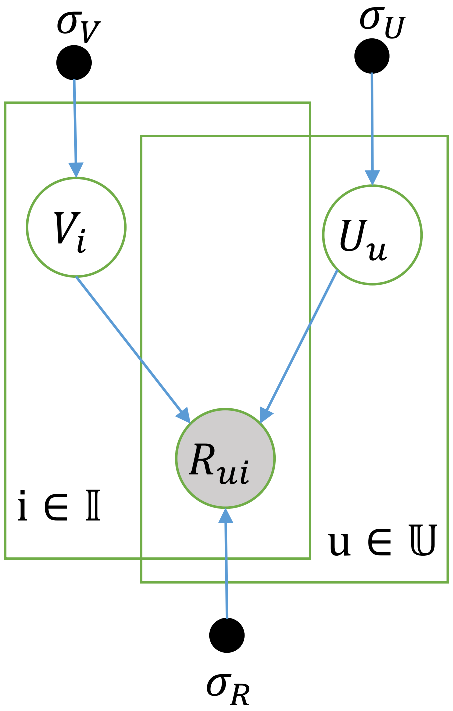


Figure 1.2: Graphical Model for Probabilistic Matrix Factorization

A matrix factorization model assumes the rating matrix R can be approximated by a multiplication of d-rank factors,

$$R \approx U^T V, \tag{1.1}$$

where $U \in \mathbb{R}^{d \times |\mathbb{U}|}$ and $V \in \mathbb{R}^{d \times |\mathbb{I}|}$. Normally d is far less than both $|\mathbb{U}|$ and $|\mathbb{I}|$. Thus given a user u and an item i, the rating R_{ui} of u for i can be approximated by the dot product of user latent feature vector U_u and item latent feature V_i,

$$R_{ui} \approx U_u^T V_i, \tag{1.2}$$

where $U_u \in \mathbb{R}^{d \times 1}$ is the u_{th} column of U and $V_i \in \mathbb{R}^{d \times 1}$ is the i_{th} column of V. For ease of notation, we let $|\mathbb{U}| = N$ and $|\mathbb{I}| = M$ in the remaining of this chapter.

Later, the probabilistic version of matrix factorization, i.e., *Probabilistic Matrix Factorization* (PMF), is introduced in [82], based on the assumption that the rating R_{ui} follows a normal distribution whose mean is some function of $U_u^T V_i$. The conditional probability of the observed ratings is:

$$p(R|U, V, \sigma_R^2) = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[\mathcal{N}\left(R_{ui} | g(U_u^T V_i), \sigma_R^2 \right) \right]^{I_{ui}^R}, \tag{1.3}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2. If u has rated i, then the indicator function I_{ui}^R equals to 1, otherwise equals to 0. $g(\cdot)$ is the sigmoid function, i.e., $g(x) = \frac{1}{1+e^{-x}}$, which bounds the range of $U_u^T V_i$ within $[0, 1]$. Moreover, U_u and V_i are both subject to a zero mean normal distribution. Thus the conditional probabilities of user and item latent feature vectors are:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{u=1}^{N} \mathcal{N}\left(U_u|0, \sigma_U^2 \mathbf{I}\right) \\ p(V|\sigma_V^2) &= \prod_{i=1}^{M} \mathcal{N}\left(V_i|0, \sigma_V^2 \mathbf{I}\right), \end{aligned} \tag{1.4}$$

where $\mathbf{I}$ is the identity matrix. Therefore, the posterior probability of the latent variables U and V can be calculated through a Bayesian inference,

$$\begin{aligned} & p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \\ \propto\ & p(R|U, V, \sigma_R^2) p(U|\sigma_U^2) p(V|\sigma_V^2) \\ =\ & \prod_{u=1}^{N} \prod_{i=1}^{M} \left[\mathcal{N}\left(R_{ui}|g(U_u^T V_i), \sigma_R^2\right)\right]^{I_{ui}^R} \\ \times & \prod_{u=1}^{N} \mathcal{N}(U_u|0, \sigma_U^2 \mathbf{I}) \times \prod_{i=1}^{M} \mathcal{N}(V_i|0, \sigma_V^2 \mathbf{I}). \end{aligned} \tag{1.5}$$

Figure 1.2 dipicts the graphical model of PMF and we refer readers to [82] for more details.

Our proposed model in Chapter 4 can be regarded as an extension of PMF and the proposed method in Chapter 6 makes use of matrix factorization technique within the Multi-armed Bandit (MAB) framework. Thus our focus will be on explicit feedback in Chapter 4 and Chapter 6.

1.2.2 Recommendation with implicit ratings

Considerable work has been done to address the problem of how to use just implicit feedback to generate high quality recommendations. Oard and Kim [83] identified several data sources to gather implicit feedback and suggested two types of recommendation strategies. The first strategy is to infer explicit (ratings) feedback that users are likely to produce and adopt available methods for explicit feedback. The second one is directly infer user preferences without converting implicit feedback to explicit ratings (feedback). Das et al. [28] presented an online recommendation algorithm for Google News where only click history of each user is available (hence implicit). They describe a linear model that combines three recommendation algorithms: collaborative filtering using MinHash clustering, probabilistic Latent Semantic Indexing (pLSI), and co-visitation counts. Hu et al. [48] proposed to transform implicit feedback into two paired quantities: preferences and confidence levels and use both of them to learn a latent factor model. Unlike matrix factorization for explicit feedback, their model takes all user-item pairs, including non-observed items, as an input and is later extended for other recommendation tasks by others [108]. Scalable learning algorithms are proposed to address the (potentially) huge amount of input. Pan et al. [87] uses weighted low-rank approximation and sampling techniques. First, different weights are assigned to the error terms of observed items and non-observed items in the objective function. Second, they sample non-observed items as negative feedback, instead of using all of them.

Hu's Model (WRMF)

Hu et al. [48] predict users' preferences for TV programs through an implicit scoring model whose factors are computed by the matrix factorization:

$$\begin{aligned} \underset{\mathbf{X}_u, \mathbf{Y}_i}{\text{minimize}} \ & \frac{1}{2} \sum_{u,i} (1 + \gamma r_{ui}) \big(p_{ui} - \mathbf{X}_u^T \mathbf{Y}_i\big)^2 \\ & + \lambda \big(\sum_u \|\mathbf{X}_u\|_2^2 + \sum_i \|\mathbf{Y}_i\|_2^2 \big). \end{aligned} \tag{1.6}$$

Vectors $\mathbf{X}_u$, $\mathbf{Y}_i \in \mathbb{R}^k$ are latent factors for user u and item i. In Hu's application, r_{ui} records the number of times user u watches TV program i. Different from ratings in the range between 1 and 5, r_{ui} can range from 0 to any positive integers (theoretically) such as 100, 1000 etc, bringing unstable fluctuations to vanila (probabilistic) matrix factorization. Therefore in WRMF, user u's preference for item (TV program) i is determined by binarizing rating $r_{ui} \geq 0$ into 0 or 1 (i.e., implicit feedback):

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0, \\ 0, & r_{ui} = 0. \end{cases} \tag{1.7}$$

In this model, the lowest rating assigned to an item a user has observed is 1, so $r_{ui} = 0$ and $p_{ui} = 0$ if u has never observed item i. This model, therefore, accounts for all user-item pairs. Parameters γ, which scales the strength of user-item ratings, λ, which regularizes matrix factors, and k, the dimension of the latent space, are chosen by experiment.

All the aforementioned methods are often referred to as *point-wise*, since they learn absolute preferences and then produce top-K recommendations by simply sorting items by their scores in descending order. Rendle et al. [95] proposed a novel *pairwise* learning method called Bayesian Personalized Ranking (BPR). Here the focus is shifted to the learning of *relative* preferences. BPR trains on pairs of items and the objective is to maximize the posterior likelihood of optimal personalized ranking, in which the assumption is that for each user, observed items are preferred over non-observed ones. Empirical results in [95] demonstrate that BPR coupled with matrix factorization or kNN indeed outperform point-wise methods proposed in [48, 87]. Recently, Rendle and Freudenthaler [94] introduced a more sophisticated sampling technique to improve the convergence rate of BPR learning.

Bayesian Personalized Ranking (BPR)

BPR [95] categorizes items into two groups:

1. **Consumed Items.** For all $u \in \mathcal{U}$, let $\mathcal{C}_u^{self} \subseteq \mathcal{I}$ denote the set of items consumed by u itself.
2. **Non-Consumed Items.** This category contains the rest of the items (not consumed by u): $\mathcal{C}_u^{none} = \{i : i \in \mathcal{I} \setminus \mathcal{C}_u^{self}\}$.

Clearly, for all $u \in \mathcal{U}$, $\mathcal{C}_u^{self} \cup \mathcal{C}_u^{none} = \mathcal{I}$.

Intuitively, each user u should prefer the consumed items to non-consumed items, i.e., the consumed items should be ranked ahead of the non-consumed ones. Mathematically,

$$i \succcurlyeq_u j, \text{ if } i \in \mathcal{C}_u^{self} \wedge j \in \mathcal{C}_u^{none}. \tag{1.8}$$

The likelihood function can be expressed as:

$$\mathcal{L}(\Theta) = \prod_{u \in \mathcal{U}} \Bigg(\prod_{i \in \mathcal{C}_u^{self}} \prod_{j \in \mathcal{C}_u^{none}} \Pr[i \succcurlyeq_u j] \Bigg), \tag{1.9}$$

where the probability that consumed items are preferred over non-consumed items is as follows:

$$\begin{aligned}
&\Pr[i \succcurlyeq_u j] \\
=&\delta(\hat{x}_{ui} - \hat{x}_{uj}) \\
=&\frac{1}{1 + \exp(-(\hat{x}_{ui} - \hat{x}_{uj}))} \\
=&\frac{1}{1 + \exp(-\langle \mathbf{P}_u, \mathbf{Q}_i \rangle + \langle \mathbf{P}_u, \mathbf{Q}_j \rangle)}.
\end{aligned} \tag{1.10}$$

For the introductory purpose of this section, we only describe the basic model idea and refer readers to the corresponding original paper for more detailed model inference.

It is fair to say that we study problems related to implicit feedback in Chapter 3 and Chapter 5 as the proposed approaches in these two chapters extend BPR and WRMF respectively.

1.3 Recommendation in Social Media

The rising of social network and rapid development of web services actuate the emergence of recommendation in social media. People not only rate movies or TV series on IMDB, but also interact with each other on Facebook and see the latest updates of their favorite idols on Twitter. This brings the idea of social recommendation which tries to utilize available information (e.g., ratings) from users' friends to infer their preferences. Lots of existing work has proved that incorporating information from social networks does help to improve the accuracy of conventional recommendation methods.

At the same time, other applications such as Wordpress, LinkedIn, Flickr, DeviantArt etc. become popular in our lives. We write blogs on Wordpress, find jobs on LinkedIn and share fantastic pictures with our friends on Flickr. Artists also upload their paintings to DeviantArt, watch other users who have submitted paintings that they are fond of and join different self-organized groups to explore art pieces collected in groups' galleries. What's more, people can even organize offline activities with their online group members on Meetup or Plancast.

Moreover, as more and more new applications appear in social media, people again are facing a huge amount of information that may be interesting to them. Thus the web service providers will have to face a similar circumstance faced by those offering traditional recommendation services in the past, that is, users want to have personalized results in social media but are not willing to spend a lot of time to specify their personal information needs. Different from traditional recommendations, items in social media can refer to almost anything. For instance, items will be groups in group recommendation that recommends groups of people to users, be traveling routes in trip recommendation and even be users themselves in friends recommendation. This being the case, when making recommendation of different kinds of items, simply applying traditional recommendation methods to the new scenarios may result in suboptimal solutions because different kinds of items may have different characteristics and different available information under different circumstances. Furthermore, different from the simple and static items such as movies or musics in conventional recommender systems, the concept of "items" in social media is extended to include more complex and dynamic ones such as groups which will change as new users join in or old members leave. The boundary between explicit and implicit feedback becomes vague in social media as well — sometimes we can have a mixture of them in one recommendation task. Extra constraints such as location and capacity should also be taken into consideration in applications such as social event organization and restaurant recommendation. Take social event organization (SEO) as an example, organizers in SEO try to recommend an event to each user (or assign sets of users to an event) in a way that maximizes the affinity between users and events and the affinity among the users in the same event, while satisfying the cardinality constraint (minimum and maximum capacity) of every event. All of the above new features make recommendation in social media a challenging and interesting topic.

A lot of research work has been done to find better recommendation strategies in social media, among which exploration of user-user relationship information is one of the most efficient ways to improve recommendation accuracy in social media, given the success of social recommendation in recent years. Social recommendation works because of social selection and social influence: people are willing to make friends with those sharing similar interets, and tend to become more and more similar with their friends as time goes by. In social networks utilized by social recommendation, user relationships are explicit, i.e, there will be an edge connecting two users if they are friends.

Existing literature on social recommendation treats all social ties (friends) equally and models them in a Boolean manner. It does not make full use of the knowledge or insights from social sciences. Additionally, as more information about the relationships between users in social media becomes available with the rapid development of various Internet services, some other social media applications gradually become popular. Take group feature as an example, more and more online web services are providing mechanisms by which users can self-organize into groups with other users having similar opinions or interests, enabling us to analyze the interactions between users with others insides/outsides groups, as well as the engagement between users and groups. Existing work on social group recommendation only utilizes Boolean user-group membership information and does not take temporal dynamics of user-group engagement relationships into consideration.

A large number of social media applications provide various types of user networks, giving birth to various kinds of user relationships. Motivated by the claim that user relationships are important information sources to improve recommendation accuracy, our aim in this thesis is to utilize various relationships among users to enhance personalized recommendation.

1.4 Our Contribution

This thesis focuses on utilizing user relationships (either explicit or implicit) to improve recommender systems. Recommendation models practically can also be categorized into offline recommendation model and online recommendation model, which can potentially leads to four research topics through combination with explicit or implicit user relationships:

- **Utilizing explicit user relationships to improve recommendation in an offline setting**.
- **Utilizing implicit user relationships to improve recommendation in an offline setting**.
- **Utilizing explicit user relationships to improve recommendation in an online setting**.
- **Utilizing implicit user relationships to improve recommendation in an online setting**.

In this thesis, we investigate the first three of the above research topics through three real-world recommendation applications in social media. Due to the lack of appropriate real-world applications and available datasets, we leave the fourth topic as future work.

The remainder of this thesis is organized as follows:

First of all, Chapter 2 discusses related work on conventional recommendation methods, social recommendation approaches, temporal recommendation strategies, group recommendation models and interactive recommendation based on online learning.

Chapter 3 and Chapter 4 tackle the problem of utilizing explicit user relationships to improve recommender systems in an offline setting (all datasets contain explicit user relationships) through incorporating the distinctions of strong and weak ties into different recommendation frameworks. Concretely, in Chapter 3 we bring the concept of strong and weak ties into Bayesian personalized ranking (BPR) through proposing the TBPR model which is capable of simultaneously classify strong and weak ties in a social network w.r.t. optimal recommendation accuracy and learning latent feature vectors for users as well as items. We again introduce the notion of strong and weak ties into probabilistic matrix factorization in Chapter 4. Our proposed PTPMF model learns a personalized tie type preference for each individual in addition to all the benefits of TBPR model introduced in Chapter 3.

In Chapter 5, the problem of how to utilize implicit user relationships to improve recommender systems in an offline setting is examined via personalized group recommendation. The groups in this problem are self-organized associations of users who have the ability collectively curate art. This collective curation provides value to artists, who benefit from the endorsement provided by well-known groups, and to individual art collectors, who can use these curated collections to discover new art. As such, artists often join groups with the intent of submitting their artwork for acceptance by the group, and collectors often join groups with the intent of discovering art that they like from the group's accepted submissions. A user who is not a member of a group can only see some (not all) of the group's collections. Thus implicit user relationships can be obtained via users consuming (such as commenting, favoring) other users' submissions and then be consequentially used to construct user-group engagement. The model proposed in this chapter tries to improve the accuracy of group recommendation by capturing the temporal dynamic of engagement between users and groups.

We turn our focus to online recommendation in Chapter 6. We investigate the topic of utilizing explicit user relationships to improve recommender systems in an online setting through solving the problem of interactive social recommendation. We adopt the exploitation-exploration strategy to study interactive social recommendation in an online setting and propose a multi-armed bandit based algorithm which is capable of not only simultaneously exploring user preferences and exploitsing the effectiveness of personalization in an interactive way, but also adaptively learning different weights for different friends. More details will be presented in the corresponding chapter.

Last but not least, we conclude the whole thesis and point out some future work for further investigation in Chapter 7.

Chapter 2

Related Work

In this chapter, we first discuss some related work on traditional recommendation without considering social information, which is the foundation of recommendation in social media. We then present some existing literature on three recommendation applications in social media mentioned in Chapter 1, i.e., social recommendation, social group recommendation and interactive social recommendation. We discuss conventional recommendation from two aspects: explicit feedback and implicit feedback. As the proposed methods in Chapter 3 and Chapter 4 tend to incorporate the concept of strong and weak ties into existing social recommendation models, we discuss related work on social ties in social media and social recommendation separately. Chapter 5 proposes to combine temporal dynamics of user group engagement with personalized social group recommendation, therefore it is necessary for the related work to contain both temporal recommendation and group recommendation. Finally, we borrow the idea from exploitation-exploration strategy in order to handle the interactive social recommendation problem. Thus we will also share some useful literature on exploitation-exploration dilemma and multi-armed bandit (which is adopted by our proposed algorithm in Chapter 6) in the end.

2.1 Conventional Recommendation

2.1.1 Recommender Systems and Collaborative Filtering

Substantial work has been done in recommender systems during the past two decades. When it comes to recommender systems, collaborative filtering (CF) is one of the most popular algorithmic solutions so far, which makes recommendations based on users' past behaviors such as ratings, clicks, purchases and favorites etc. In general, there are two groups of CF methods: memory-based and model-based. Representative memory-based methods include k-Nearest Neighbour (kNN) user-user cosine similarity and item-item cosine similarity [3]. Representative model-based methods include the widely-adopted low-rank matrix factorization (also known as latent factor models) [37, 96]. Being among the most effective methods for collaborative filtering, there is a large body of work on using matrix factorization for collaborative filtering [48, 58, 59, 82, 98, 108]. As an excellent monograph, Koren [60] gives a systematic introduction to the application of matrix factorization to recommender systems. Among the literature of matrix factorization, Salakhutdinov and Mnih [82] propose a probabilistic version of matrix factorization (PMF) which assumes a Gaussian distribution on the initializations of latent feature vectors, making the model more robust towards the problem of overfitting and linearly scalable with the number of observations at the same time. We note that all of the above literature is designed for systems with explicit feedback (numerical ratings).

2.1.2 Collaborative Filtering with Implicit Feedback

Gathering explicit feedback is often a challenging task. Thus, considerable work has been done to address the problem of how to simply use implicit feedback to generate high quality recommendations.

In one of the earliest works, Oard and Kim [83] identified several data sources to gather implicit feedback, and suggested two types of recommendation strategies. The first strategy is to infer explicit ratings that users are likely to produce and adopt available methods for explicit feedback. The second one is directly infer user preferences without converting implicit feedback to ratings.

Das et al. [28] presented an online recommendation algorithm for Google News where only click history of each user is available (hence implicit). They describe a linear model that combines three recommendation algorithms: collaborative filtering using MinHash clustering, probabilistic Latent Semantic Indexing (pLSI), and co-visitation counts.

Hu et al. [48] proposed to transform implicit feedback into two paired quantities: preferences and confidence levels and use both of them to learn a latent factor model. Unlike matrix factorization for explicit feedback, their model takes all user-item pairs as an input, including non-observed items. Scalable learning algorithms are proposed to address the (potentially) huge amount of input.

The method proposed in Pan et al. [87] uses weighted low-rank approximation and sampling techniques. First, different weights are assigned to the error terms of observed items and non-observed items in the objective function. Second, they sample non-observed items as negative feedback, instead of using all of them.

All methods described so far in this subsection are often referred to as *point-wise*, since they aim to learn *absolute* preferences and then produce top-K recommendations by simply sorting items by their scores in descending order. Rendle et al. [95] proposed a novel *pairwise* learning method called Bayesian Personalized Ranking (BPR). Here the focus is shifted to the learning of *relative* preferences. BPR trains on pairs of items and the objective is to maximize the posterior likelihood of optimal personalized ranking, in which the assumption is that for each user, observed items are preferred over non-observed ones. Empirical results in [95] demonstrate that BPR coupled with matrix factorization or kNN indeed outperform point-wise methods proposed in [48, 87]. Recently, Rendle and Freudenthaler [94] introduced a more sophisticated sampling technique to improve the convergence rate of BPR learning.

However, these matrix factorization based models still suffer from the data sparsity and cold start problems, which gives rise to *social recommendation*.

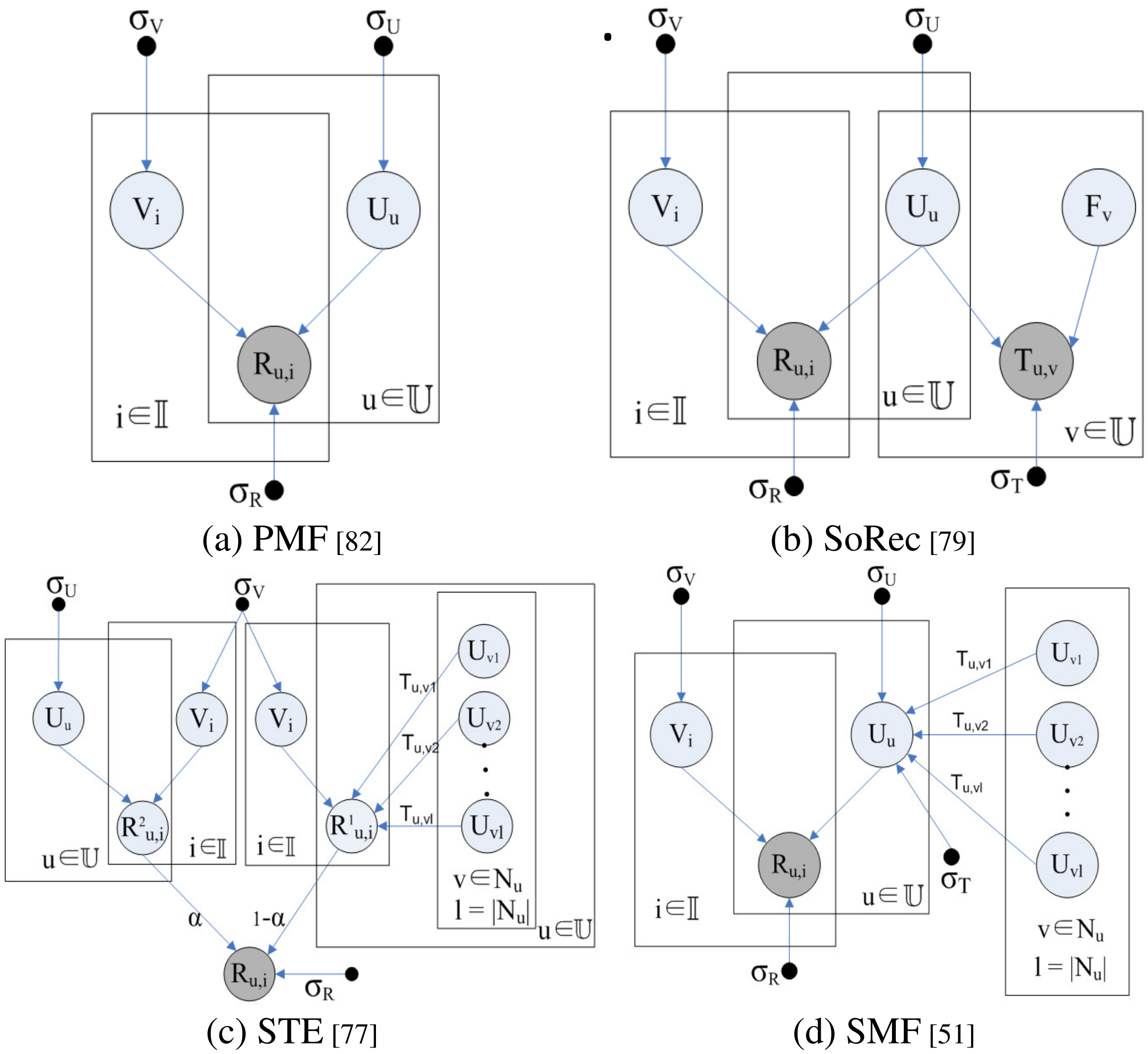
2.2 Social Recommendation and Social Ties

2.2.1 Social Recommendation

There has been no shortage of existing work on *social recommendation* whose appearance should be attributed to the advent of social networks. As the rich information on social network becomes available, social recommendation which makes use of social information from social networks to enhance recommender systems has attracted lots of attention from researchers due to the encouraging improvement (particularly for cold-start users) obtained against its non-social counterpart. The fact that cold start problem has always been an important factor to deteriorate the performance of collaborative filtering motivates the advent of work on *social recommendation*, which utilizes social information among users to improve the performances of recommender systems. Indeed, social influence tends to have strong effects in changing human behaviours [16, 56], such as adopting new opinions, technologies, and products. This has stimulated the study of *social recommendation*, which aims to leverage social network information to help mitigate the "cold-start" problem in collaborative filtering [50, 51, 77–80, 110, 117, 118, 120, 128], in the hope that the resulting recommendations will have better quality and higher relevance to users who have given little feedback to the system. In particular, Ma et al. [79] propose a probabilistic matrix factorization model which factorizes user-item rating matrix and user-user linkage matrix simultaneously. They later present another probabilistic matrix factorization model which aggregates a user's own rating and her friends' ratings to predict the target user's final rating on an item. In [51], Jamali and Ester introduce a novel probabilistic matrix factorization model based on the assumption that users' latent feature vectors are dependent on their social ties'. The graphical models for these three popular social recommendation models (i.e., SoRec [79], STE [77], SMF [51]) as well as the vanilla *Probabilistic Matrix Factorization* (PMF) [82] are shown in Figure 2.1

2.2.2 Social Ties in Social Media

Different types of social ties have attracted lots of interests from researchers in social sciences [23, 43, 44, 54], followed by some recent work which pays attention to tie strength in demographic data [93] and social media [8, 14, 40, 41, 52, 89, 90, 111, 114, 131]. In particular, Gilbert et al. [41] bridge the gap between social theory and social practice through predicting interpersonal tie strength with social media and conducting user-study based experiments over 2000 social media ties. Wu et al. [111] propose a regression analysis to discover two different types of closeness (i.e., professional and personal) for employees in an IBM enterprise social network. Panovich et al. [89] later carry out an investigation related to different roles of tie strength in question and answer online networks by taking advantage of Wu's approach.



(a) PMF [82]

(b) SoRec [79]

(c) STE [77]

(d) SMF [51]

Figure 2.1: Graphical Models of Several Social Recommendation Models

However, none of the above work leverages the theory of social ties into recommender systems. On the other hand, existing work in social recommendations do not take different types of social ties into consideration. This is no surprise, since the combination is very specific.

2.3 Recommendation with Temporal Dynamics and Group Recommendation

2.3.1 Temporal Recommendation

There has been some work on temporal recommendation of items to users. Koren [59] combines collaborative filtering and temporal dynamics together by proposing a model tracking the temporal evolution of user behavior throughout the life span of the items. Other authors ([25,57,66,86,113,115]) have examined time-dependent methods such as tensor factorization, session-based temporal graph model, dynamic matrix factorization and matrix factorization with online learning etc. This work captures past temporal patterns, but does not extrapolate future temporal dynamics to estimate future changes in users' preferences. Zhang et al. [125] incorporate a transition matrix into conventional [82] and Bayesian [98] probabilistic matrix factorization methods, modeling the evolution of user preferences under the assumption that future preferences depend only on the immediately preceding state.

2.3.2 Group Recommendation (Recommending Groups to Users)

The term *group recommendation* in this thesis refers to the problem of recommending groups of people to users. On the other hand, in literature *group recommendation* also refers to the problem of recommending items to a group of users and the PolyLens project [84], for example, is a small study in recommending content items to groups of users. We refer readers to [6,13] for more detailed information if interested.

The problem of recommending groups to users [21, 22, 104, 107, 124] has been studied through exploring a variety of probabilistic and combinatorial recommendation methods applied to Boolean user-group membership matrices, with increasing success as more side information is incorporated into the model. Specifically, [22] proposes incorporating a probabilistic model which also considers the group-word matrix derived from the textual description of the group, and [104] employ a matrix factorization based model to take an additional user-user linkage matrix derived from social network relationships into consideration. Zeng and Chen [124] incorporate both user-item ratings and user-user social relationships (called "heterogeneous resources" in their paper) into the user-group membership matrix. However, none of these authors model the temporal dynamics of user-group engagement.

2.4 Interactive Recommendation

Interactive recommendation, as its name indicates, is a recommendation schema which interacts with users. In other words, interactive recommendation can refine its model parameters upon receiving new feedback from user, which can be treated as one kind of online learning algorithm. We remark that this recommendation mechanism may be more practical for real-world recommendation scenarios where user feedback is received sequentially rather than simultaneously. At early stages of running the algorithm when only a small amount of feedback from users is gathered, we have two options to choose:

1. Make full use of the current available information to make new recommendations to users.

2. Pick some random new items which are not quite relevant to the target users for recommendations.

If collaborative filtering based algorithms are used in the first option, then with high probability items similar to those consumed by the target users may be recommended to them. This is quite a "conservative" strategy of recommendation as people will seldom be angry with an item similar to those they have consumed. However, no one can guarantee that users will not feel bored after being recommended with too many similar items, which gives significance to the second option that recommend some randomly "new" items to users instead. The challenging part about the second option is that it is possible that the recommended randomly new item may not make users happy or even annoy them in the worst case, running the risk of users ceasing to use the recommendation services. Thus the key challenge in interactive recommendation is that how can we stop users from leaving the recommendation systems due to boredom while learning their preferences as much as we can in the shortest time. Fortunately, the first option can be regarded as "exploitation" and the second one can be treated as "exploration", making the goal of interactive recommendation become balancing the trade-off between exploitation and exploration, which has been well studied in literature.

2.4.1 Exploitation-exploration (E-E) Trade-off

There always exists a trade-off between utilizing the information available so far (exploitation) and acquiring new knowledge (exploration). This kind of problems has been widely studied extensively in many fields such as *Machine Learning*, *Theoretical Computer Science*, *Operations Research* etc. This mature, yet very active, research area is known as "multi-armed bandit" in literature [5, 64, 72, 81].

2.4.2 Multi-armed Bandit (MAB) in Recommender Systems

Being first introduced by Robbins [97], multi-armed bandit is able to provide us with a clean, simple theoretical formulation for analyzing the trade-off between exploration and exploitation, and thus has been widely utilized by researchers to solve the challenges in balancing the trade-off faced by exploitation-exploration problems. We refer readers to [11, 15, 62] for a more general treatment.

Depart from the conventional stochastic multi-armed bandit [11, 42, 63], contextual bandit algorithms [9, 10, 17, 24, 39, 65, 69–71, 101–103, 112, 122, 123, 130, 132] have attracted lots of attention from researchers because they have achieved much more promising performance than their context-free counterparts. Contextual bandit settings normally assume that the expectation of the reward (also known as payoff) for an action of a user on an item is a linear function of the corresponding context, e.g., the dot product of user feature vector and item feature vector [10], which gives much flexibility for different choices of expected reward function. For instance, Chu et al. [24] and Li et al. [69] use ridge regression to calculate the expectation and confidence interval of the reward of an action. In particular, methodology for the unbiased evaluation of context bandit algorithm is introduced in [70]. Besides the linear reward, Filippi et al. [33] propose a parametric bandit algorithm for non-linear rewards. Later, a general approach to encoding prior knowledge for accelerating contextual bandit learning is introduced in [122] through employing a coarse-to-fine feature hierarchy which dramatically reduces the amount of exploration required. Bouneffouf et al. [17] investigate exploitation and exploration dilemma in mobile context-aware recommender systems and present an approach to the adaptive balance of exploitation/exploration trade-off regarding the target user's situation. By utilizing a Gaussian process kernel and taking context into consideration, Vanchinathan et al. [103] introduce a novel algorithm that can efficiently re-rank lists to reflect user preferences over the items displayed. Moreover, a contextual combinatorial bandit that plays a "super arm" at each round is proposed by Qin et al. [92] to dynamically identify diverse items which new users are very likely to be fond of. Tang et al. explore ensemble strategies of contextual bandit algorithms to obtain robust predicted click-through rate of web objects [101], and later they propose a parameter-free bandit strategy which uses online bootstrap to derive the distribution of predicting models [102]. Recently, a combination of linear bandit with cascade model is introduced in [132] to deal with the large-scale recommendation and the dynamical pattern of reward as well as the context drift in the course of time is taken into account to formulate a time varying multi-armed bandit by Zeng et al. [123].

Others have also explored another variant which is designed to model dependency in the bandit setting [7, 19, 39, 71, 99, 112, 130]. In particular, authors in [19, 99] conduct investigations about contextual bandit with the probabilistic dependencies of context and actions being taken into consideration. Gentile and Li et al. [39, 71] investigate adaptive clustering algorithms based on the learnt model parameters for contextual bandit under the assumption that content is recommended to different groups (clusters) of users such that users within each group (cluster) tend to share similar interest, followed by Zhou and Brunskill who propose a contextual bandit algorithm that explores the latent structure of users through learning the distribution of users over different (fixed number) latent classes to make personalized recommendations for new users [130].

Chapter 3

Recommendation with Strong and Weak Ties

With the explosive growth of online social networks, it is now well understood that social information is highly helpful to recommender systems. Social recommendation methods are capable of battling the critical cold-start issue, and thus can greatly improve prediction accuracy. The main intuition is that through trust and influence, users are more likely to develop affinity toward items consumed by their social ties. Despite considerable work in social recommendation, little attention has been paid to the important distinctions between strong and weak ties, two well-documented notions in social sciences. In this work, we study the effects of distinguishing strong and weak ties in social recommendation. We use neighborhood overlap to approximate tie strength and extend the popular Bayesian Personalized Ranking (BPR) model to incorporate the distinction of strong and weak ties. We present an EM-based algorithm that simultaneously classifies strong and weak ties in a social network w.r.t. optimal recommendation accuracy and learns latent feature vectors for all users and all items. We conduct extensive empirical evaluation on four real-world datasets and demonstrate that our proposed method significantly outperforms state-of-the-art pairwise ranking methods in a variety of accuracy metrics.

3.1 Motivation

Recommender systems are ubiquitous in our digital life. They play a significant role in numerous Internet services and applications such as electronic commerce (Amazon and eBay), on-demand video streaming (Netflix and Hulu), as well as social networking ("People You May Know" feature of LinkedIn and Facebook). A key task is to model user preferences and to suggest, for each user, a personalized list of items that the user has not experienced, but are deemed highly relevant to her.

Lots of recommendation techniques have been proposed in the literature [3, 96]. When explicit feedback (numerical ratings) is available, model-based collaborative filtering is among the most effective methods, e.g., low-rank matrix factorization [60].

However, when explicit feedback is not readily unavailable, we may only have access to implicit feedback [48] derived from user actions such as viewing videos, clicking links, listening to songs, etc. In fact, implicit feedback is more abundant than explicit [48, 95] in practice. Although collaborative filtering approaches can be adapted [48], pairwise ranking methods have gained more traction lately [61, 88, 94, 95, 128]. This approach focuses on learning the order of items (in user preferences). The Bayesian Personalized Ranking (BPR) framework [95] is a fundamental pairwise ranking method. In a nutshell, the core idea is to learn a personalized ranking for each user based on the assumption that a user prefers an observed item over all non-observed items. Here, an observed item refers to any item that has been consumed by the user. When the context is clear, we will use "observed" or "consumed" items interchangeably. In [95], the authors further show that many scoring methods can be integrated into BPR to learn the rankings, including matrix factorization.

A critical yet common issue faced by recommender systems is *data sparsity*, because the number of items is typically huge (e.g., hundreds of thousands) but users normally only consume a very small subset of items. An even more challenging problem related to data sparsity is that when new users join in a system, they have no history records which can be utilized by the recommender systems to learn their preferences. This leads to the cold-start problem and may result in suboptimal recommendations. To mitigate this issue, many methods have been proposed to leverage social network information in recommender systems [50, 51, 75, 78–80, 118, 120, 128], bringing about the field of *social recommendation*. Specifically for BPR, Zhao et al. [128] propose the Social BPR (SBPR) model which further assumes that among all non-observed items, a user prefers those consumed by their social connections (or ties) to the rest.

Although there exists previous work that aims at predicting tie strength with social media [41] and analyzing roles of tie strength in Q&A online networks [89], to the best of our knowledge, there has been no systematic study on social tie strength and types in the context of recommender systems, and more importantly, the extent to which different social ties affect the quality of recommendations. In his influential paper [43], Granovetter introduces different types of social ties (strong, weak, and absent), and concludes that weak ties are actually the most important reason for new information or innovations to spread over social networks. In [44], through surveys and interviews, Granovetter reports that many job seekers find out useful information about new jobs through personal contacts. Perhaps surprisingly, many of those personal contacts are acquaintances (weak ties) as opposed to close friends (strong ties) [32, 44].

These insights from social sciences motivate us to study whether distinguishing between strong and weak ties would make a difference for social recommendations in terms of prediction accuracy (e.g., metrics such as precision, recall, and AUC). However, two major challenges arise. First, how to learn the label of each tie (strong or weak) in a given social network? The sociology literature [43, 44] typically assumes the *dyadic hypothesis*: the strength of a tie is determined solely by the interpersonal relationship between two individuals, irrespective of the rest of the network. For instance, Granovetter uses the frequency of interactions to classify strong and weak ties [44]. This is simple and intuitive, but it requires user activity data that is hardly available to the public in modern online social networks for security and privacy reasons[1]. Second, assuming a reliable classification algorithm for learning strong and weak ties, how can we effectively incorporate such knowledge into existing ranking methods to improve recommendation accuracy?

In this chapter, we tackle both challenges head on. We first adopt Jaccard's coefficient, a feature intrinsic to the network topology, to compute tie strength [73, 85]. Intuitively, Jaccard captures the extent to which those users' friendship circles overlap. Our choice is endorsed by the studies on a large-scale mobile call graph by Onnela et al. [85] (more details in Section 3.3). We define ties as strong if their Jaccard's coefficient is above some threshold, and as weak otherwise. Note that the optimal threshold w.r.t. recommendation accuracy will be learnt from the data.

Next, we extend the BPR model and propose a unified learning framework that simultaneously (i) classifies strong and weak ties w.r.t. optimal recommendation accuracy and (ii) learns a ranking model that effectively leverages the learned tie types. We employ the Expectation-Maximization algorithm [29] to alternatively learn types of social ties and other model parameters including the latent feature vector for each user and each item. Our experiments on four real-world datasets clearly demonstrate the superiority of our method over state-of-the-art methods.

To summarize, we make the following contributions.

- We recognize the effects of strong and weak social ties that are evident in the sociology literature, and propose to incorporate these notions into social recommendation (Section 3.3).
- We propose a more fine-grained categorization of user-item feedback for Bayesian Personalized Ranking (BPR) by leveraging the knowledge of tie strength and tie types (Section 3.4).
- We present an EM-style algorithm to simultaneously learn the optimal threshold w.r.t. recommendation accuracy for classifying strong and weak ties, as well as other parameters (Section 3.5) in our extended BPR model.
- We will carry out extensive experiments on four real-world datasets and show that our solution significantly outperforms existing methods in various accuracy metrics such as precision and recall (Section 3.6).

To the best of our knowledge, this is the first work leveraging the important distinctions between strong and weak ties in the context of social recommendation.

Before proceeding further, we now formalize the problem studied in this chapter. Consider a recommender system, and let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items, respectively. There is also a social network connecting the users, represented by an undirected graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where each node $u \in \mathcal{U}$ represents an individual user and each edge $(u, v) \in \mathcal{E}$ indicates a tie between users u and v. We know the set of items consumed by each user u, and our task is to produce a personalized ranking (a total ordering of all items), denoted $\succcurlyeq_u$, for all $u \in \mathcal{U}$.

[1] `https://en.wikipedia.org/wiki/Privacy_concerns_with_social_networking_services`

3.2 Comparison with Existing Work

In a nutshell, social recommendation aims to exploit the effects of trust and influence to address the cold-start problem, which may cause traditional CF methods to fail due to lack of feedback data from cold-start users. Jamali et al. [50] reported that in the Epinions dataset, about 50% of the users are deemed cold-start (who rated less than five items). Considerable work has been done in this domain [50, 51, 75, 78–80, 118, 120, 128]. However, the overwhelming majority of those social recommendation methods are designed for explicit feedback systems, with few exceptions.

Lu et al. [75] modeled the evolution of user interest by considering social influence and users' reactions to recommendations (attraction and aversion). They devise near-optimal recommendations using semi-definite programming techniques to maximize the total utility of all users in the steady-state of the evolution model (which is a Markov chain). Their model does not assume explicit feedback. However, they do not explicitly distinguish between strong and weak ties in social networks.

Bayesian Personalized Ranking (BPR). Rendle et al. [95] propose a novel *pairwise* learning method called Bayesian Personalized Ranking (BPR). Here the focus is shifted to the learning of *relative* preferences. BPR trains on pairs of items and the objective is to maximize the posterior likelihood of optimal personalized ranking, in which the assumption is that for each user, observed items are preferred over non-observed ones. Empirical results in [95] demonstrate that BPR coupled with matrix factorization or kNN indeed outperform point-wise methods proposed in [48, 87]. Recently, Rendle and Freudenthaler [94] introduced a more sophisticated sampling technique to improve the convergence rate of BPR learning.

Social BPR Model (SBPR). Zhao et al. [128] extended the BPR framework by further assuming that amongst all non-observed items, a user would prefer items consumed by her social ties over the rest (which we call "social items" hereafter for simplicity). In their SBPR model, for each user u, the relative preference between any self-consumed item i and any social item j is discounted by the number of u's ties who consumed j. That is, the more ties consumed j, the smaller the gap is between i and j in the eyes of u. They also discussed an alternative, opposite assumption, i.e., the social items are perceived even more negatively than "non-social" items. Their experiments showed that this alternative SBPR model is not as good as the first one.

Major Differences vs. SBPR. We depart from SBPR by making orthogonal social-aware extensions to BPR. In particular, we recognize the importance of distinguishing between strong and weak ties and extend the BPR model by incorporating such distinctions. The key difference lies in the ranking of social items. In SBPR, social items are ranked based on the number of friends who consumed the item, while in our model, the ranking is based on tie types. Our empirical results demonstrate that our new model significantly outperforms SBPR and the vanilla BPR in terms of prediction accuracy, as measured by six different metrics including precision, recall, etc (Section 3.6).

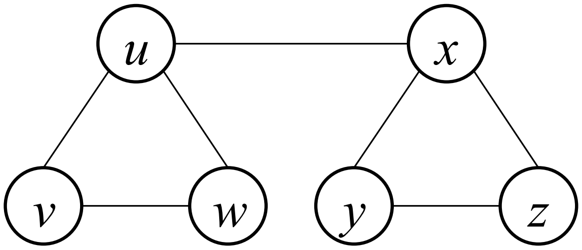


Figure 3.1: A sample social network

3.3 Strong and Weak Ties

The theory of strong and weak ties has first been formulated by Granovetter [43]. In terms of interpersonal relationship, strong ties correspond to close friends that have high frequency of interactions, while weak ties correspond to acquaintances. In terms of network structures, strong ties tend to be clustered in a dense subgraph (e.g., the triangles (u, v, w) and (x, y, z) in Figure 3.1), while weak ties tend to be "bridges" connecting two different connected components, e.g., (u, x) in Figure 3.1.

There is an elegant connection between the above two perspectives (interpersonal relationship and network structure) [32, 43]. First, we say that a node u satisfies the *Strong Triadic Closure property* if it does not violate the following condition: u has two strong ties v and w but there exists no edge between v and w. Furthermore, if a node u satisfies this property and is involved in at least two strong ties, then any local bridges[2] in which it is involved must be a weak tie.

It is well understood that since weak ties typically do not belong to the same social circle, they have access to different information sources, and thus the information exchange have more novelty [32, 43, 44]. Applying this insight to the context of social recommendation, our intuition is that the items previously consumed by weak-tie friends *might* be of more interest to the user. For example, a researcher may not be able to discover many interesting new papers from her close collaborators, as they tend to focus on the same topic and read the same set of papers. Instead, she may find papers cited by other less frequent collaborators more appealing.

To incorporate the distinction between strong and weak ties into social recommendation, we first need to be able to define and compute tie strength, and then classify ties. Several possibilities exist. First, as mentioned in Section 3.1, sociologists use dyadic measures such as frequency of interactions [44]. However, this method is not generally applicable due to lack of necessary data.

An alternative approach relies on community detection. Specifically, it first runs a community detection algorithm to partition the network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ into several subgraphs. Then, for each edge $(u, v) \in \mathcal{E}$, if u and v belong to the same subgraph, then it is classified as a strong tie; otherwise a weak tie. However, a key issue is that although numerous community detection algorithms exist [34], there is no consensual gold standard so it is unclear which one to use. Furthermore, if a "bad" partitioning (w.r.t. prediction accuracy) is produced and given to the recommender system

[2] (u, v) is a local bridge if the deletion of this edge results in u and v to have a shortest path distance of 3 or longer.

as input, it would be very difficult for the recommender system to recover. In other words, the quality of recommendation would depend on an *exogenous* community detection algorithm that the recommender system *has no control over*. Hence, this approach is undesirable.

In light of the above, we resort to node-similarity metrics that measure neighbourhood overlap of two nodes in the network. The study of Onnela et al. [85] provides empirical confirmation of this intuition: they find that (i) tie strength is in part determined by the local network structure and (ii) the stronger the tie between two users, the more their friends overlap. In addition, unlike frequency of interactions, node-similarity metrics are intrinsic to the network, requiring no additional data to compute. Also, unlike the community detection based approach, we still get to choose a tie classification method that best serves the interest of the recommender system.

More specifically, we use Jaccard's coefficient, a simple measure that effectively captures neighbourhood overlap. Let $\mathsf{strength}(u, v)$ denote the tie strength for any $(u, v) \in \mathcal{E}$. We have:

$$\mathsf{strength}(u, v) =_{\text{def}} \frac{|\mathcal{N}_u \cap \mathcal{N}_v|}{|\mathcal{N}_u \cup \mathcal{N}_v|} \quad \textit{(Jaccard)}, \tag{3.1}$$

where $\mathcal{N}_u \subseteq \mathcal{U}$ (resp. $\mathcal{N}_v \subseteq \mathcal{U}$) denotes the set of ties of u (resp. v). If $\mathcal{N}_u = \mathcal{N}_v = \emptyset$ (i.e., both u and v are singleton nodes), then simply define $\mathsf{strength}(u, v) = 0$. By definition, all strengths as defined in Equation (3.1) fall into the interval $[0, 1]$. This definition has natural probabilistic interpretations: Given two arbitrary users u and v, their Jaccard's coefficient is equal to the probability that a randomly chosen tie of u (resp. v) is also a tie of v (resp. u) [73].

Thresholding. To distinguish between strong and weak ties, we adopt a simple *thresholding* method. For a given social network graph $\mathcal{G}$, let $\theta_{\mathcal{G}} \in [0, 1)$ denote the threshold of tie strength such that

$$(u, v) \text{ is } \begin{cases} \text{strong}, & \text{if } \mathsf{strength}(u, v) > \theta_{\mathcal{G}}; \\ \text{weak}, & \text{if } \mathsf{strength}(u, v) \le \theta_{\mathcal{G}}. \end{cases} \tag{3.2}$$

Let $\mathcal{W}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) \le \theta_{\mathcal{G}}\}$ denote the set of all weak ties of u. Similarly, $\mathcal{S}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) > \theta_{\mathcal{G}}\}$ denotes the set of all strong ties of u. Clearly, $\mathcal{W}_u \cap \mathcal{S}_u = \emptyset$ and $\mathcal{W}_u \cup \mathcal{S}_u = \mathcal{N}_u$.

In our framework, the value of $\theta_{\mathcal{G}}$ is *not* hardwired, but rather is left for our model to learn (Section 3.5), such that the resulting classification of strong and weak ties in $\mathcal{G}$, together with other learned parameters of the model, leads to the best accuracy of recommendations.

Finally, we remark that other node-similarity metrics can also be used to define tie strength, e.g., Adamic-Adar [2] and Katz score [53]. However, we note that the exact choice amongst these node-similarly metrics is not the primary focus of this chapter and is orthogonal to our proposed learning framework.

3.4 The TBPR Model: BPR with Stong and Weak Ties

In this section, we present our TBPR (*BPR with Strong and Weak Ties*) model which incorporates the distinction of strong and weak ties into BPR and ranks social items based on types of ties.

3.4.1 Categorizing Items

Having defined strong and weak ties, we are now ready to present a key element in our TBPR model: For every user we categorize all items into five types using the knowledge of strong and weak ties, which we then exploit in our TBPR model. Here, we provide a fine-grained categorization of non-observed items, especially the social items, by leveraging strong and weak tie information derived from the social network graph $\mathcal{G}$. The proposed categorization is as follows.

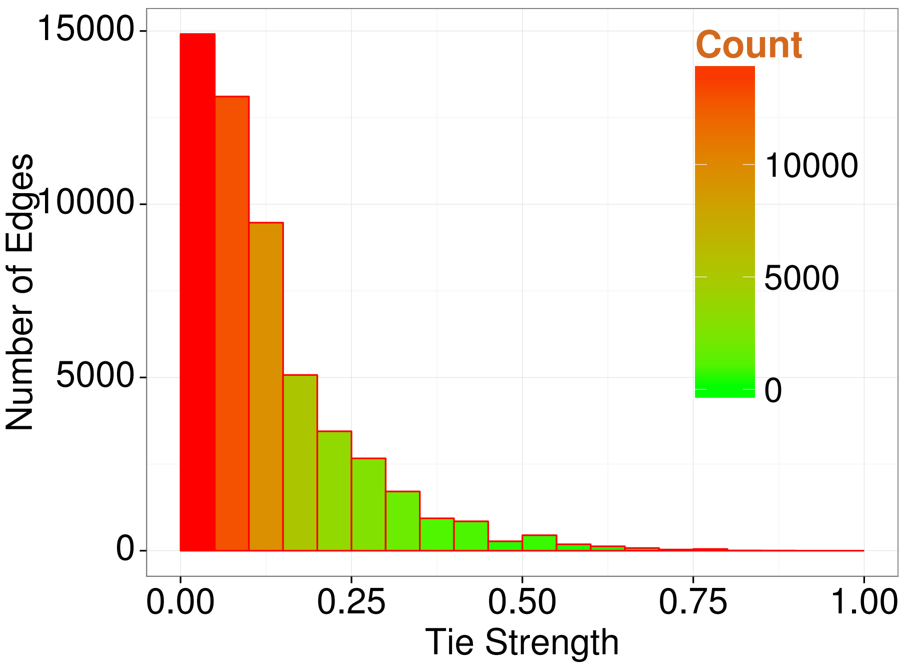
1. **Consumed Items.** For all $u \in \mathcal{U}$, let $\mathcal{C}_u^{self} \subseteq \mathcal{I}$ denote the set of items consumed by u itself.
2. **Joint-Tie-Consumed (JTC) Items.** Any item $i \in \mathcal{I} \setminus \mathcal{C}_u^{self}$ that has been consumed by at least one strong tie of u and one weak tie of u belongs to this category. We denote this set by $\mathcal{C}_u^{joint} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \exists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \ \wedge \ \exists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$
3. **Strong-Tie-Consumed (STC) Items.** If an item $i \in \mathcal{I} \setminus \mathcal{C}_u^{self}$ is consumed by at least one strong tie of u, but not by u itself or weak ties, then it belongs to this category. We denote this set by $\mathcal{C}_u^{strong} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \exists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \ \wedge \ \nexists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$.
4. **Weak-Tie-Consumed (WTC) Items.** This category can be similarly defined: $\mathcal{C}_u^{weak} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \nexists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \ \wedge \ \exists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$.
5. **Non-Consumed Items.** This category contains the rest of the items (not consumed by u or any of u's ties): $\mathcal{C}_u^{none} = \{(u, i) : \nexists x \in \mathcal{S}_u \cup \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_x^{self}\}$.

Clearly, for all $u \in \mathcal{U}$, $\mathcal{C}_u^{self} \cup \mathcal{C}_u^{joint} \cup \mathcal{C}_u^{strong} \cup \mathcal{C}_u^{weak} \cup \mathcal{C}_u^{none} = \mathcal{I}$. In addition, those five sets are pairwise disjoint. Note that the union of JTC, STC, and WTC items is the set of all social items for user u.

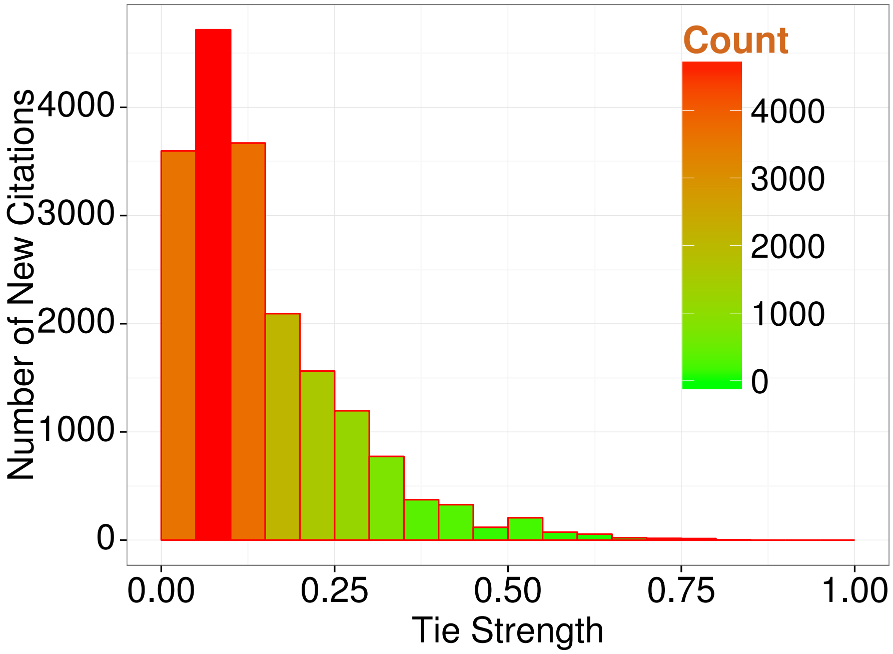
3.4.2 Ordering Item Types

We now describe our TBPR model which distinguishes between the aforementioned five types of items for every user. Same as the original BPR, we assume no particular item scoring method [95]. However, for ease of exposition and its effectiveness, we use low-rank matrix factorization [60], which is considered as a state-of-the-art collaborative filtering method in the literature.

Assume that every user and every item in the system are represented by a d-dimensional latent feature vector: let $\mathbf{P}_u \in \mathbb{R}^d$ and $\mathbf{Q}_i \in \mathbb{R}^d$ denote the feature vector for an arbitrary user u and an arbitrary item i, respectively. Here d is the number of latent features. The inner product between a user feature vector and an item feature vector measures the estimated affinity this user has toward



(a) Tie strength



(b) New citations

Figure 3.2: Histograms of tie strength and new citations for the DBLP dataset

the item (a.k.a. predicted personalized score), denoted by $\hat{r}_{ui} =_{\text{def}} \langle \mathbf{P}_u, \mathbf{Q}_i \rangle$. Since we deal with binary feedback in this work, we have $\hat{r}_{ui} \in [0, 1]$ for all $u \in \mathcal{U}$ and all $i \in \mathcal{I}$.

In this chapter, the proposed TBPR model imposes a total ordering of the five item types that specifies user preference. Indicated by the good performance of BPR and its variants [88, 128], we also assume that users prefer consumed items over others. Hence, consumed items rank at the top of the ordering. Next, it is an open question that whether users prefer WTC items to STC items, or vice versa. Although we mentioned in Section 3.1 that the sociology literature has suggested that weak ties are responsible for more novel information to spread over the social network, it does not automatically mean that WTC items are preferred.

To investigate the above question, we conduct a case study using co-authorship and citation data extracted from the DBLP Computer Science Bibliography (`http://dblp.uni-trier.de/db/`). The DBLP dataset together with three other public datasets (Epinions, Douban and Ciao) will be used to evaluate the performances of different methods later in the experiment section. Recently some work has been done on recommending papers to read or cite using the DBLP dataset [126], making it another appropriate experimental dataset for us to test the performances of different recommendation algorithms. Furthermore, the DBLP website provides researchers with an API so that they can crawl their own datum from the database for the purpose of scientific research, which enables the possibility for us to obtain the information about the evolution of co-authorship network and conduct this case study based on the assumption that co-authorship network and citations follow a similar pattern to social network and other user-item consumption behaviors.

The network graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ is constructed as follows. First, each node $v \in \mathcal{U}$ corresponds to an author satisfying both (i) she co-authored at least ten papers and (ii) at least one of her papers was published in or after 2009. If two authors u and v have co-authored at least one paper before 2009, then there is an undirected edge $(u, v) \in \mathcal{E}$. As a result, the graph contains 13.6K nodes and 107K edges.

Figure 3.2(a) shows the distribution of tie strength as computed by Equation (3.1). By definition, if two authors u and v have a strong tie, then a relatively large overlap exists amongst their collaborators. As we can see, the distribution is skewed toward weak tie strength.

Next, we analyze the citation data to see whether researchers are more likely to cite papers that were previously cited by their weak ties as opposed to strong ties, or vice versa. We are interested in the case of *follow-up citations*. For example, for any $(u, v) \in \mathcal{E}$, if there exists a paper cited by v but not by u before 2009, and u cited this paper in or after 2009, then we say that u made one follow-up citation to v. Note that, this definition eliminates all citations occurred in papers co-authored by u and v, which are not interesting to us. Figure 3.2(b) plots the number of follow-up citations against tie strength. We can see that this distribution is also heavily skewed toward weak tie strength. This suggests that from the perspective of absolute number, researchers indeed tend to cite papers that are previously cited by their weak ties.

Our two plots above are very consistent with those plots (with Y axis showing the probability of job help instead of the number of citations) presented in [38], a recent work by Gee et al. on how strong ties and weak ties relate to job finding on Facebook's social network. Gee et al. use both mutual interactions and node similarity (similar to Jaccard's coefficient) to measure tie strength and find results to be similar for both kinds of measures, which provides further support for using Jaccard's coefficient as our tie strength measure. Readers may refer to [38] for more details. A conclusion in their work is that weak ties are important collectively because of their quantity, and strong ties are important individually because of their quality. Reflected in our TBPR model, we can say that the sets of WTC items are more helpful than the sets of STC items and an individual STC item may be more helpful than an individual WTC item. Thus giving the WTC items a higher probability to be exposed (recommended) to users (i.e., ranking WTC items ahead of STC items) should help to discover potentially more interesting items. On the other hand, we also explore the opposite case of users ranking STC items ahead of WTC items. As such, we test both ranking strategies for completeness, in what follows, we present two variants of our TBPR model.

3.4.3 Two Variants of TBPR

We are now ready to define two variants of TBPR, which differ in the preference between WTC and STC items.

TBPR-W (Preferring Weak Ties). Mathematically, under the hypothesis that WTC items are preferred to STC items, the complete ordering is thus:

$$i \succcurlyeq_u j, \text{ if } \begin{cases} i \in \mathcal{C}_u^{self} \wedge j \in \mathcal{C}_u^{joint} & \text{or} \\ i \in \mathcal{C}_u^{joint} \wedge j \in \mathcal{C}_u^{weak} & \text{or} \\ i \in \mathcal{C}_u^{weak} \wedge j \in \mathcal{C}_u^{strong} & \text{or} \\ i \in \mathcal{C}_u^{strong} \wedge j \in \mathcal{C}_u^{none}. \end{cases} \tag{3.3}$$

Note that Equation (3.3) gives a total ordering of the five types due to transitivity, e.g., it also holds that $i \succcurlyeq_u j$ if $i \in \mathcal{C}_u^{self}$ and $j \in \mathcal{C}_u^{none}$.

TBPR-S (Preferring Strong Ties). Alternatively, we may also assume that users prefer STC items to WTC items, in which case the ordering can be expressed as:

$$i \succcurlyeq_u j, \text{ if } \begin{cases} i \in \mathcal{C}_u^{self} \wedge j \in \mathcal{C}_u^{joint} & \text{or} \\ i \in \mathcal{C}_u^{joint} \wedge j \in \mathcal{C}_u^{strong} & \text{or} \\ i \in \mathcal{C}_u^{strong} \wedge j \in \mathcal{C}_u^{weak} & \text{or} \\ i \in \mathcal{C}_u^{weak} \wedge j \in \mathcal{C}_u^{none}. \end{cases} \tag{3.4}$$

When it is clear from the context, we use the generic name TBPR to refer to both variants. The specific names TBPR-W and TBPR-S will be used when it is necessary to distinguish between them (e.g., comparisons in experimental results).

3.5 Parameter Learning

In this section, we present the optimization objective and an EM-style learning algorithm for our TBPR model. Without loss of generality, our presentation focuses on TBPR-W in which WTC items are preferred over STC items. The case of TBPR-S is symmetric and hence is omitted.

3.5.1 Optimization Objective

Let Θ denote the set of all parameters that consists of (i) the tie strength threshold $\theta_{\mathcal{G}}$ and (ii) the latent feature vectors: $\mathbf{P}_u$ for each user $u \in \mathcal{U}$ and $\mathbf{Q}_i$ for each item $i \in \mathcal{I}$. The likelihood function can thus be expressed as:

$$\begin{aligned}\mathcal{L}(\Theta) = \prod_{u \in \mathcal{U}} \Bigg(& \prod_{i \in \mathcal{C}_u^{self}} \prod_{j \in \mathcal{C}_u^{joint}} \Pr[i \succcurlyeq_u j] \\ & \prod_{j \in \mathcal{C}_u^{joint}} \prod_{w \in \mathcal{C}_u^{weak}} \Pr[j \succcurlyeq_u w] \\ & \prod_{w \in \mathcal{C}_u^{weak}} \prod_{s \in \mathcal{C}_u^{strong}} \Pr[w \succcurlyeq_u s] \\ & \prod_{s \in \mathcal{C}_u^{strong}} \prod_{k \in \mathcal{C}_u^{none}} \Pr[s \succcurlyeq_u k] \Bigg),\end{aligned} \tag{3.5}$$

where the probabilities are defined using the sigmoid function following common practice [95]: $\delta(x) = \frac{1}{1+\exp(-x)}$.

For instance, the probability that consumed items are preferred over JTC items can be written as follows.

$$\begin{aligned} & \Pr[i \succcurlyeq_u j] \\ = & \, \delta(\hat{x}_{ui} - \hat{x}_{uj}) \\ = & \, \frac{1}{1 + \exp(-(\hat{x}_{ui} - \hat{x}_{uj}))} \\ = & \, \frac{1}{1 + \exp(-\langle \mathbf{P}_u, \mathbf{Q}_i \rangle + \langle \mathbf{P}_u, \mathbf{Q}_j \rangle)}.\end{aligned} \tag{3.6}$$

All other probabilities except for the probability that WTC items are preferred over STC items, namely $\Pr[w \succcurlyeq_u s]$, can be defined similarly. We omit the formulas as they resemble Eq. (3.6) closely. One thing that needs to be pointed out here is that we only rank items belonging to two different categories, not those falling in the same category.

Incorporating the Tie Strength Threshold

Given a threshold $\theta_{\mathcal{G}}$, the *degree of separation* between strong ties and weak ties imposed by this threshold can be quantitatively measured using the following formula:

$$g(\theta_{\mathcal{G}}) = (\bar{t}_s - \theta_G)(\theta_G - \bar{t}_w), \tag{3.7}$$

where $\bar{t}_s$ is the average strength of all strong ties classified according to $\theta_{\mathcal{G}}$ and likewise $\bar{t}_w$ is the average strength of all weak ties.

A threshold $\theta_{\mathcal{G}}$ that gives a large degree of separation $g(\theta_{\mathcal{G}})$ is desirable. To incorporate the threshold into the objective function so that our TBPR model is able to learn it in a principled manner, we add a coefficient $1/g(\theta_{\mathcal{G}})$ into the probability that WTC items are preferred over STC items. More specifically, we define:

$$\begin{aligned} \Pr[w \succcurlyeq_u s] = \delta\left(\frac{\hat{x}_{uw} - \hat{x}_{us}}{1 + 1/g(\theta_{\mathcal{G}})}\right) \\ = \frac{1}{1 + \exp\left(-\frac{\hat{x}_{uw} - \hat{x}_{us}}{1+1/g(\theta_{\mathcal{G}})}\right)} \\ = \frac{1}{1 + \exp\left(\frac{-\langle \mathbf{P}_u, \mathbf{Q}_w \rangle + \langle \mathbf{P}_u, \mathbf{Q}_s \rangle}{1+1/g(\theta_{\mathcal{G}})}\right)}, \end{aligned} \tag{3.8}$$

where we use $1 + 1/g(\theta_{\mathcal{G}})$ to discount $(\hat{x}_{uw} - \hat{x}_{us})$, the difference between u's predicted score for w and s. The intuition is that, if the current threshold $\theta_{\mathcal{G}}$ does not separate the strong and weak ties well enough, the likelihood that user prefers w (an WTC item given the current threshold) to s (an STC items given the current threshold) should be discounted. We use the reciprocal mainly for smoothness.

Putting It All Together

Our goal is to learn the best set of parameters that maximizes the likelihood function $\mathcal{L}(\cdot)$. This amounts to maximizing the logarithm of $\mathcal{L}(\cdot)$. Regularization terms are added to avoid overfitting:

$$r(\Theta) = \lambda_p \sum_{u \in \mathcal{U}} ||\mathbf{P}_u||_2^2 + \lambda_q \sum_{i \in \mathcal{I}} ||\mathbf{Q}_u||_2^2 + \lambda_\theta \theta_{\mathcal{G}}^2.$$

Putting it all together, our final maximization objective is

$$\begin{aligned} \mathcal{J}(\Theta) = \ln \mathcal{L}(\Theta) - r(\Theta) \\ = \sum_{u \in \mathcal{U}} \Bigg(\sum_{i \in \mathcal{C}_u^{self}} \sum_{j \in \mathcal{C}_u^{joint}} \ln \delta(\hat{x}_{ui} - \hat{x}_{uj}) \end{aligned}$$

Algorithm 1: Learning Algorithm for TBPR-W

```
Input: users $\mathcal{U}$, items $\mathcal{I}$, consumed items $\mathcal{C}_u^{self}$ for each $u \in \mathcal{U}$, social network graph
       $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: $\Theta = \{\mathbf{P} \in \mathcal{R}^{|\mathcal{U}| \times d}, \mathbf{Q} \in \mathcal{R}^{|\mathcal{I}| \times d}, \theta_{\mathcal{G}}\}$
$\mathbf{P} \sim U(0,1), \mathbf{Q} \sim U(0,1)$
$t \leftarrow 0$ // iteration number
$\theta_{\mathcal{G}}^{(t)} \leftarrow$ median tie strength in $\mathcal{G}$
repeat
    for $u \leftarrow 1$ to $|\mathcal{U}|$ do
        Compute $\mathcal{C}_u^{self}$, $\mathcal{C}_u^{joint}$, $\mathcal{C}_u^{strong}$, $\mathcal{C}_u^{weak}$, $\mathcal{C}_u^{none}$ using the current tie strength threshold
            $\theta_{\mathcal{G}}^{(t)}$ // cf. Equation (3.2) and categorization rules in
            Section 3.4.1
    end
    for $r \leftarrow 1$ to $100|\mathcal{U}|$ do
        $u \leftarrow$ a random user from $\mathcal{U}$
        $i \leftarrow$ a random consumed item from $\mathcal{C}_u^{self}$
        $j \leftarrow$ a random JTC item from $\mathcal{C}_u^{joint}$
        $w \leftarrow$ a random WTC item from $\mathcal{C}_u^{weak}$
        $s \leftarrow$ a random STC item from $\mathcal{C}_u^{strong}$
        $k \leftarrow$ a random non-consumed item from $\mathcal{C}_u^{none}$
        Compute the gradients of $\mathbf{P}_u$, $\mathbf{Q}_i$, $\mathbf{Q}_j$, $\mathbf{Q}_w$, $\mathbf{Q}_s$, and $\mathbf{Q}_k$ // Equation (3.9) -
            Equation (3.14)
        Update the above feature vectors // Equation (3.16)
    end
    Compute $\frac{\partial \mathcal{J}}{\partial \theta_{\mathcal{G}}}$ // Equation (3.15)
    $\theta_{\mathcal{G}}^{(t+1)} \leftarrow$ compute according to Equation (3.16)
    $t \leftarrow t + 1$
until convergence
```

$$
\begin{aligned}
&+ \sum_{j \in \mathcal{C}_u^{joint}} \sum_{w \in \mathcal{C}_u^{weak}} \ln \delta(\hat{x}_{uj} - \hat{x}_{uw}) \\
&+ \sum_{w \in \mathcal{C}_u^{weak}} \sum_{s \in \mathcal{C}_u^{strong}} \ln \delta\left(\frac{\hat{x}_{uw} - \hat{x}_{us}}{1 + 1/g(\theta_{\mathcal{G}})}\right) \\
&+ \sum_{s \in \mathcal{C}_u^{strong}} \sum_{k \in \mathcal{C}_u^{none}} \ln \delta(\hat{x}_{us} - \hat{x}_{uk})\Bigg) \\
&- \lambda_p \sum_{u \in \mathcal{U}} ||\mathbf{P}_u||_2^2 - \lambda_q \sum_{i \in \mathcal{I}} ||\mathbf{Q}_u||_2^2 - \lambda_\theta \theta_{\mathcal{G}}^2.
\end{aligned}
$$

3.5.2 Learning Algorithm

We employ the *Expectation-Maximization* (EM) algorithm as well as *stochastic gradient descent* to learn the parameters Θ that maximize $\mathcal{J}(\cdot)$. In the EM algorithm, the tie strength threshold $\theta_{\mathcal{G}}$ is treated as a hidden parameter to be learnt from the data.

The pseudocode of the learning algorithm is presented in Algorithm 1. In the beginning, we randomly initialize the latent feature vectors for all users and all items by sampling from the uniform distribution over the interval $[0, 1]$. We initialize the tie strength threshold to be the median strength of all edges in the graph (Lines 1–3).

E-step. In each iteration t, given the current tie strength threshold $\theta_{\mathcal{G}}^{(t)}$, we first compute, for each user, their five categories of items (Line 6). Then, we take a total number of $100 \cdot |\mathcal{U}|$ samples as the training dataset to perform stochastic gradient descent. For each sample r, we first draw a user u uniformly at random from $\mathcal{U}$, and then draw one item from each category for this user: consumed ($\mathcal{C}_u^{self}$), JTC ($\mathcal{C}_u^{joint}$), WTC ($\mathcal{C}_u^{weak}$), STC ($\mathcal{C}_u^{strong}$), and non-consumed ($\mathcal{C}_u^{none}$) (Lines 9–14). All samples are drawn independently.

Notice that the pseudocode assumes all five item categories for all users are non-empty. If any of the sets $\mathcal{C}_u^{self}$, $\mathcal{C}_u^{joint}$, $\mathcal{C}_u^{weak}$, and $\mathcal{C}_u^{strong}$ is empty, we simply skip all relevant terms. The case of $\mathcal{C}_u^{none} = \emptyset$ is uninteresting as that would mean the user has consumed all items, and thus there is nothing left to rank for her.

Lastly, we compute the gradient of all corresponding feature vectors and perform updates (Lines 15–16). Gradients are computed using the following partial derivative formulas.

- The gradient of vector $\mathbf{P}_u$, for any user u:

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{P}_u} &= \delta(\hat{x}_{uj} - \hat{x}_{ui})(\mathbf{Q}_i - \mathbf{Q}_j) + \delta(\hat{x}_{uw} - \hat{x}_{uj})(\mathbf{Q}_j - \mathbf{Q}_w) + \\ &\frac{\delta(\hat{x}_{us} - \hat{x}_{uw})}{1 + 1/g(\theta_{\mathcal{G}})}(\mathbf{Q}_w - \mathbf{Q}_s) + \delta(\hat{x}_{uk} - \hat{x}_{us})(\mathbf{Q}_s - \mathbf{Q}_k) - \lambda_p \mathbf{P}_u \ . \qquad (3.9)\end{aligned}$$

- The gradient of vector $\mathbf{Q}_i$, where $i \in \mathcal{C}_u^{self}$:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_i} = \delta(x_{uj} - x_{ui})\mathbf{P}_u - \lambda_q \mathbf{Q}_i \ . \qquad (3.10)$$

- The gradient of vector $\mathbf{Q}_j$, where $j \in \mathcal{C}_u^{joint}$:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_j} = \left(\delta(x_{uw} - x_{uj}) - \delta(x_{uj} - x_{ui})\right)\mathbf{P}_u - \lambda_q \mathbf{Q}_j \ . \qquad (3.11)$$

- The gradient of vector $\mathbf{Q}_w$, where $w \in \mathcal{C}_u^{weak}$:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_w} = \left(\frac{\delta(x_{us} - x_{uw})}{1 + 1/g(\theta_\mathcal{G})} - \delta(x_{uw} - x_{uj}) \right) \mathbf{P}_u - \lambda_q \mathbf{Q}_w \; . \tag{3.12}$$

- The gradient of vector $\mathbf{Q}_s$, where $s \in \mathcal{C}_u^{strong}$:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_s} = \left(\delta(x_{uk} - x_{us}) - \frac{\delta(x_{us} - x_{uw})}{1 + 1/g(\theta_\mathcal{G})} \right) \mathbf{P}_u - \lambda_q \mathbf{Q}_s \; . \tag{3.13}$$

- The gradient of vector $\mathbf{Q}_j$, where $j \in \mathcal{C}_u^{none}$:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_k} = -\delta(x_{uk} - x_{us}) \mathbf{P}_u - \lambda_q \mathbf{Q}_k \; . \tag{3.14}$$

M-step. After updating the feature vectors associated with all $100|\mathcal{U}|$ samples, we update the tie strength threshold $\theta_\mathcal{G}$. The derivative can be computed as follows:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \theta_\mathcal{G}} = \frac{1}{100|\mathcal{U}|} \sum_{(u,w,s)} \Bigg[& - \lambda_\theta \theta_\mathcal{G} + \\ & \frac{\delta(x_{us} - x_{uw})(\langle \mathbf{P}_u, \mathbf{Q}_w \rangle - \langle \mathbf{P}_u, \mathbf{Q}_s \rangle)[(\bar{t}_w + \bar{t}_s) - 2\theta_\mathcal{G}]}{[(\theta_\mathcal{G} - \bar{t}_w)(\bar{t}_s - \theta_\mathcal{G}) + 1]^2} \Bigg], \end{aligned} \tag{3.15}$$

where (u, w, s) denotes the user, WTC item, STC item tuple sampled in one of the $100|\mathcal{U}|$ samples.

In both the E-step and M-step, the update is done using standard gradient descent:

$$x^{(t+1)} = x^{(t)} + \eta^{(t)} \cdot \frac{\partial \mathcal{J}}{\partial x}(x^{(t)}), \tag{3.16}$$

where $x \in \Theta$ denotes any model parameter. Finally, the algorithm terminates when the absolute difference between the losses in two consecutive iterations is less than 10^{-5}.

3.6 Empirical Evaluation

In this section, we conduct extensive experiments on four real-world datasets and compare the performance of our TBPR-W and TBPR-S models with different baseline methods based on various evaluation metrics.

3.6.1 Experimental Settings

Datasets.

We use the following four real-world datasets, whose basic statistics are summarized in Table 3.1.

Table 3.1: Overview of datasets (#non-zeros means the number of user-item pairs that have feedback)

	DBLP	*Ciao*	*Douban*	*Epinions*
#users	13554	1141	13492	10306
#items	51877	11640	45282	109534
#non-zeros	488368	26507	2669675	375241
#ties (edges)	106730	15059	443753	230684

- *DBLP.* This dataset contains information of author citation and co-author network between 1960 and 2010, which is extracted by us from the DBLP Computer Science Bibliography.
- *Ciao.* This dataset contains trust relationships between users and ratings on DVDs. It was crawled from the entire category of DVDs of a UK DVD community website `http://dvd.ciao.co.uk` in December, 2013, and first introduced in [45].
- *Douban.* This dataset is extracted from the famous Chinese forum social networking site `http://movie.douban.com/`. It contains user-user friendships and user-movie ratings, which is publicly available[3].
- *Epinions.* This dataset[4] is extracted from the consumer review website Epinions `http://www.epinions.com/`. The data also contains user-user trust relationships and numerical ratings.

Since ratings in *Ciao*, *Douban* and *Epinions* are all integers ranging from 1 to 5, we "binarize" them into boolean datasets: we consider items rated higher than 2 as consumed items. For DBLP, we use all citations occurring before year 2009 as the training set and leave all citations in or after 2009 for testing. For other datasets, we randomly choose 80% of each user's consumed items for training and leave the remainder for testing.

Methods Compared.

The following eight recommendation methods, including six baselines, are tested.

- **TBPR-W.** Our TBPR model with weak ties ranked above strong ties (Equation 3.3).
- **TBPR-S.** Our TBPR model with strong ties ranked above weak ties (Equation 3.4).
- **BPR.** The classic method proposed in [95], coupled with matrix factorization for item scoring.
- **SBPR.** The Social BPR method proposed in [128], using the assumption that social items are ranked higher than non-social items.

[3] `https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban`

[4] `http://www.trustlet.org/wiki/Epinions_dataset`

- **SBPR-N.** A naive version of SBPR which ranks social items without considering the number of ties that consumed the items. Comparisons between SBPR-N and TBPR is to show that TBPR's improvement over SBPR is irrespective of whether the number of ties is considered or not.

- **Implicit MF (WRMF).** Weighted matrix factorization using a point-wise optimization strategy for implicit user-item feedback [48].

- **Random.** Randomly sample the non-consumed items to form a ranked list for each user.

- **Most Popular.** This is a non-personalized baseline which ranks all items based on their global popularity, i.e., the number of users that consumed an item.

All experiments are conducted on a platform with 2.3 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory. Grid search and 5-fold cross validation are used to find the best regularizer and we set $\lambda_u = \lambda_q = 0.01$ and $\lambda_\theta = 0.1$. The learning rate η of stochastic gradient descent is set to 0.1 for $\theta_{\mathcal{G}}$ and 0.01 for other parameters.

Evaluation Metrics.

The following metrics are used to measure the prediction accuracy.

- *Recall@K* (Rec@K). This metric quantifies the fraction of consumed items that are in the top-K ranking list sorted by their estimated rankings. For each user u we define $S(K;u)$ as the set of already-consumed items in the test set that appear in the top-K list and $S(u)$ as the set of all items consumed by this user in the test set. Then, we have

$$\textit{Recall@K}(u) = \frac{|S(K;u)|}{|S(u)|}.$$

- *Precision@K* (Pre@K). This measures the fraction of the top-K items that are indeed consumed by the user (test set):

$$\textit{Precision@K}(u) = \frac{|S(K;u)|}{K}.$$

- *Area Under the Curve (AUC).*

$$AUC = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{E}_u|}\sum_{(i,j)\in\mathcal{E}_u}\delta((x_{ui} - x_{uj}) > 0),$$

where $\mathcal{E}_u = \{(i,j)|i \in S(u) \wedge j \in \mathcal{I} \setminus \mathcal{C}_u^{self}\}$ and $(x_{ui} - x_{uj}) > 0$ indicates that for user u, item i is ranked ahead of item j.

Table 3.2: Performance evaluations on all users (boldface font denotes the winner in that row).

		Random	Popular	WRMF	BPR	SBPR-N	SBPR	TBPR-S	TBPR-W	Impv%
DBLP	Pre@5	0.000374	0.009642	0.029259	0.031081	0.031592	0.034522	0.033807	**0.040680**	17.8%†
	Rec@5	0.000362	0.004552	0.026533	0.029588	0.029229	0.030541	0.030146	**0.036152**	18.4%†
	AUC	0.491990	0.687722	0.823800	0.866391	0.863784	0.873853	0.867394	**0.903863**	3.43%†
	MAP	0.001235	0.006007	0.026071	0.030382	0.029903	0.031865	0.031434	**0.038393**	20.5%†
	NDCG	0.150952	0.171013	0.210249	0.224147	0.225434	0.231224	0.228612	**0.241568**	4.47%†
	MRR	0.003341	0.038732	0.088439	0.093422	0.092076	0.095209	0.094304	**0.106519**	11.9%†
Ciao	Pre@5	0.001068	0.008219	0.016689	0.015152	0.015954	0.016752	0.016614	**0.018497**	10.4%†
	Rec@5	0.000435	0.014629	0.019048	0.017522	0.017986	0.021813	0.021609	**0.023532**	7.88%†
	AUC	0.508182	0.671325	0.727143	0.770230	0.770574	0.775404	0.770964	**0.798210**	2.94%†
	MAP	0.001537	0.015193	0.017857	0.018991	0.018971	0.019704	0.019400	**0.024158**	22.6%†
	NDCG	0.125384	0.167200	0.171261	0.175962	0.179664	0.186527	0.185729	**0.199382**	6.89%†
	MRR	0.005500	0.033362	0.052541	0.050175	0.051811	0.054098	0.052668	**0.060842**	12.5%†
Douban	Pre@5	0.027144	0.137217	0.137844	0.154097	0.170087	0.170754	**0.211767**	0.170667	24.0%†
	Rec@5	0.005559	0.023895	0.040620	0.025105	0.029743	0.038584	**0.043969**	0.038549	8.24%†
	AUC	0.553510	0.839267	**0.974293**	0.971433	0.972291	0.972306	0.974195	0.972039	-0.01%‡
	MAP	0.014030	0.057165	0.072330	0.077991	0.073461	0.078055	**0.099851**	0.078199	27.9%†
	NDCG	0.299853	0.392563	0.385310	0.438571	0.434797	0.454351	**0.488024**	0.454644	7.41%†
	MRR	0.077733	0.289429	0.301047	0.300290	0.356630	0.357672	**0.437344**	0.356532	22.3%†
Epinions	Pre@5	0.000132	0.016230	0.021320	0.022658	0.023797	0.024945	0.024802	**0.026989**	8.19%†
	Rec@5	0.000040	0.014579	0.018795	0.020810	0.020552	0.021308	0.021819	**0.023960**	12.4%†
	AUC	0.514609	0.784285	0.890701	0.894476	0.894034	0.901279	0.901306	**0.918934**	1.96%†
	MAP	0.000703	0.012759	0.019503	0.021544	0.021663	0.022174	0.022452	**0.024461**	10.3%†
	NDCG	0.126838	0.174815	0.195235	0.198665	0.199031	0.209530	0.207757	**0.225629**	7.68%†
	MRR	0.001892	0.051963	0.067934	0.073059	0.073879	0.075097	0.073299	**0.086588**	15.3%†

- *Mean Average Precision (MAP).* Let $C(u)$ be the set of user u's candidate items for ranking in the test set. The average precision for u is:

  $$AP(u) = \frac{1}{|S(u)|} \sum_{K=1}^{|C(u)|} \mathit{Precision@K}(u),$$

  and the mean average precision will be:

  $$MAP = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} AP(u).$$

- *Mean Reciprocal Rank (MRR).* Let $R(u)$ be the ranking of items in $C(u)$ in descending order, then for any item i in $S(u)$, we denote its position in $R(u)$ as $rank_i^u$. Thus the mean reciprocal rank is computed as follows:

  $$MRR = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i=1}^{|S(u)|} \frac{1}{rank_i^u}.$$

- *Normalized Discounted Cumulative Gain (NDCG).* This is widely used in information retrieval and it measures the quality of ranking through discounted importance based on positions. In recommender systems, NDCG is computed as following:

  $$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG_u}{IDCG_u},$$

  where DCG and IDCG (Ideal Discounted Cumulative Gain) are in turn defined as:

  $$DCG_u = \sum_{i \in S(u)} \frac{1}{\log_2(rank_i^u + 1)},$$

  $$IDCG_u = \sum_{i=1}^{|S(u)|} \frac{1}{\log_2(i + 1)}.$$

3.6.2 Results and Analysis

Table 3.2 demonstrates the performance of all eight recommendation methods on all four datasets, measured by six different accuracy metrics. We also conduct a paired difference test (dependent t-test for paired samples) between TBPR (whichever version is better) and the best baseline over all six metrics on each dataset. In Table 3.2, † indicates that the result of a paired difference test is significant at $p < 0.05$ with degree of freedom as $\#users - 1$ on each dataset and ‡ indicates the result is not significant. Generally speaking, TBPR outperforms all six baselines in all but one cases and moreover, all the results in which TBPR outperforms the best baseline are statistically significant at $p < 0.05$.

TBPR Models vs. Baselines.

For the sake of clarity, in the last column of Table 3.2 we provide the relative improvement achieved by TBPR-W or TBPR-S (whichever is better) over the best baseline, determined on a row-by-row basis: E.g., for Pre@5 on Epinions, the best baseline is SBPR.

We observe that TBPR, with very few exceptions, outperforms the best baseline on all datasets and for all metrics. Considering the different metrics, the gap between TBPR and the baselines is typically larger for Rec@5, Pre@5, MAP, and MRR, while the smallest gaps are observed for AUC. BPR and SBPR are also quite strong in terms of AUC. This is due to a clear connection between optimizing AUC and the objective of BPR (and its extensions such as SBPR and our TBPR). We omit the details and refer the readers to [95].

In terms of datasets, the gap between TBPR and the baselines is generally larger on DBLP and Douban. For DBLP, there are four metrics (Pre@5, Rec@5, MAP, MRR) w.r.t. which TBPR's improvement is above 10%; For Douban, the advantage is more apparent: there are three metrics (Pre@5, MAP and MRR) w.r.t. which TBPR's improvement compared to the best baseline is 24.0%, 27.9% and 22.3%, respectively.

Note that although the two variants of TBPR assume reverse ordering between STC (Strong-Tie-Consumed) items and WTC (Weak-Tie-Consumed) items, they both outperform BPR. This may appear unintuitive, as one may imagine that if one particular ordering performs well, the reverse ordering should give inferior performance. To interpret these results, first recall that BPR only orders consumed items ahead of all non-consumed ones (including social and non-social), whereas both variants of TBPR order social items ahead of non-social items. The fact that both TBPR variants beat BPR actually further attests to the core intuition held by the large body of work on social recommendation: users tend to prefer social items to non-social items.

As to at least one variant of TBPR outperforming SBPR, recall that the key difference between TBPR and SBPR is the internal ordering amongst all social items of a user. SBPR "ranks" social items based on the number of ties that consumed the items, while the TBPR ordering is based on tie type. In fact, for any particular category of social items, e.g., WTC items, we do not impose any further internal ordering. This being the case, one may argue that the improvement of TBPR over SBPR might seem to lie in the fact that SBPR takes into account the number of ties and TBPR does not. Therefore we also implement a naive version of SBPR, which ranks social items without taking the number of ties that consumed the items into consideration. The comparisons demonstrate that both variants of TBPR outperform SBPR-N, suggesting that our idea of using tie type to categorize and rank social items is better.

TBPR-S vs. TBPR-W.

We observe from Table 3.2 that TBPR-W beats TBPR-S on DBLP, Ciao, Epinions, while TBPR-S performs better on Douban. This indicates that on average users in different datasets may have different preferences over strong and weak ties, which further raises a question that do different users actually have distinct inherent biases toward STC items and WTC items? In fact, this leads to an interesting direction for future work, which is to personalize the ordering of STC and WTC items and learn it for each individual user. In other words, how can we learn personalized preference of strong and weak ties for each individual user?

Recall and Precision.

Figure 3.2 depicts Recall (X-axis) vs. Precision (Y-axis) achieved by six recommendation methods. We exclude Random since it is much worse than Most Popular. Data points from left to right on each line were calculated at different values of K, ranging from 5 to 50. Clearly, the closer the line is to the top right corner (of the plot area), the better the algorithm is: which indicates that both recall and precision are high. We can see that either TBPR-W or TBPR-S dominates all baselines, consistent with the findings in Table 3.2. In addition, the trade-off between recall and precision can be clearly observed from Figure 3.2.

Table 3.3: Percentage improvement of TBPR (the better of TBRP-W and TBPR-S) over the best baseline on cold-start users

	Pre@5	Rec@5	AUC	MAP	NDCG	MRR
DBLP	8.53%	10.8%	4.27%	0.14%	2.26%	0.88%
Ciao	28.0%	16.7%	0.73%	8.19%	7.69%	26.3%
Douban	49.6%	20.0%	0.80%	33.9%	17.6%	52.3%
Epinions	6.39%	16.8%	0.30%	8.61%	2.77%	11.6%

Comparisons on Cold-Start Users.

We further investigate the performance of various recommendation methods on *cold-start* users. As is common practice, we define users that consumed less than five items as cold-start. Table 3.3 demonstrates the percentage improvement of TBPR (the better of TBPR-S and TBPR-W) over the best baseline. By comparing Tables 3.2 and 3.3, we can see that more often than not, the improvement by TBPR is larger for cold-start users. For instance, on Ciao and Douban, the improvement is larger w.r.t. five out of all six metrics.

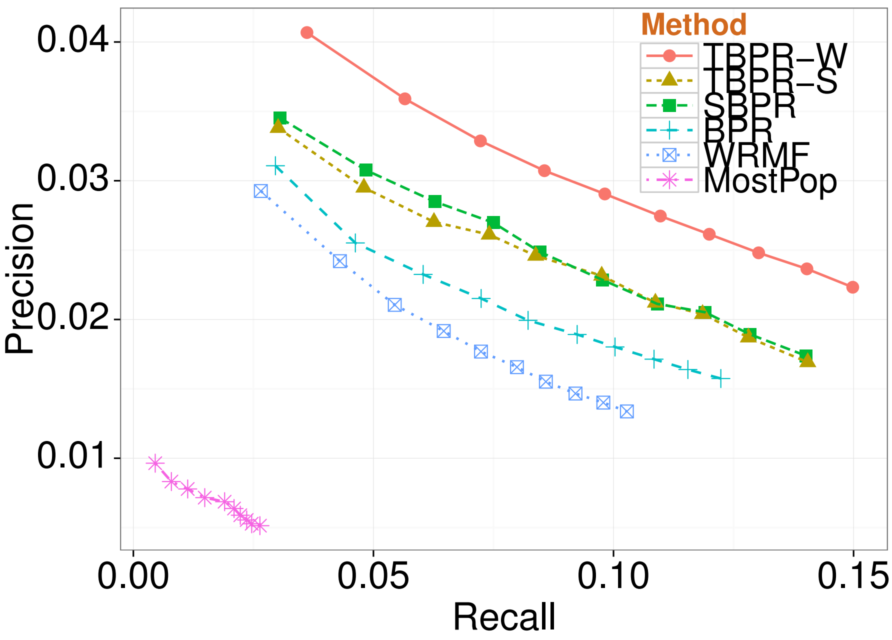
We further compare BPR, SBPR, and TBPR on all six metrics in Figures 3.2 and 3.3. SBPR outperforms BPR in all cases, which again confirms the benefit of taking social network information into consideration for recommender systems. Note that in most cases, TBPR-S is slightly better than TBPR-W. This is reasonable as cold-start users may first rely on strong ties who are more trust-worthy to them.

Finally, from our comprehensive experiments, it is fair to conclude that both TBPR-W and TBPR-S are effective social recommendation methods based on their convincing performance on not only all users, but also cold-start users.

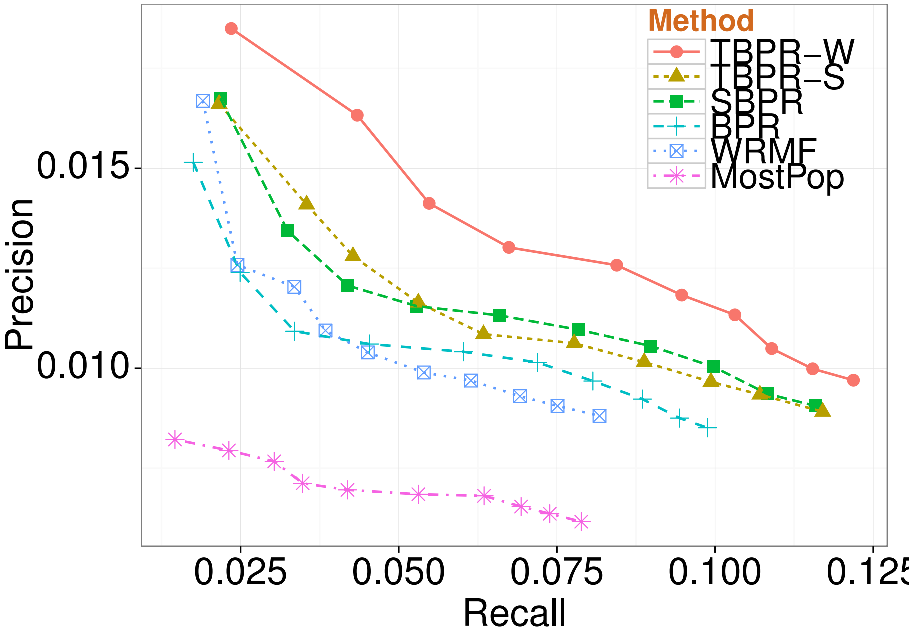
3.7 Summary

In this chapter, we present a new social recommendation method for implicit feedback data. Motivated by the seminal work in sociology by Granovetter [43, 44], we recognize the effects of strong and weak ties, in particular, the role played by weak ties in spreading novel information over social networks. Our model is a non-trivial extension to the Bayesian Personalized Ranking (BPR) model that is aware of the important distinction between strong and weak ties in social networks. We categorize "social items" (i.e., those not consumed by a user herself, but were consumed by the user's social ties) into three groups, depending on whether an item was consumed by the user's strong ties, weak ties, or both. We propose to use Jaccard's coefficient to compute tie strengths in a given social network, and then devise an EM-style algorithm that is capable of simultaneously learning the tie strength threshold and the latent feature vectors of all users and items. Our comprehensive experimental results on four real-world datasets clearly demonstrate the efficacy of our proposed methods and their superiority over existing pairwise recommendation models such as BPR [95] and SBPR [128], as well as point-wise ones such as WRMF [48].

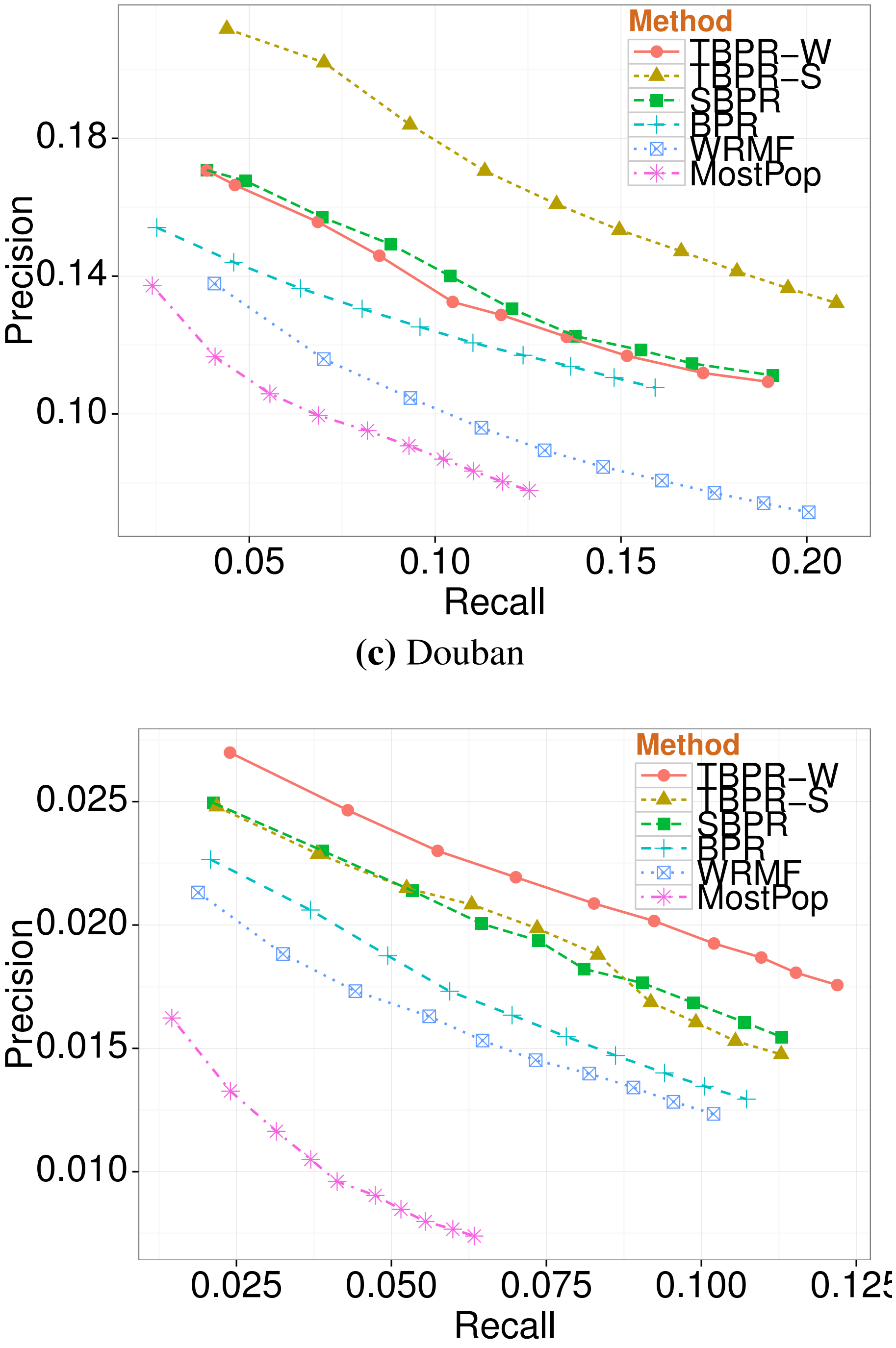
This work opens up plenty of opportunities for future research. First, as pointed out in Section 3.6.2, we conjecture that an even more personalized TBPR model warrants careful considerations, since it is plausible that while some users prefer items consumed by weak ties over those by strong ties, other users may behave in the opposite way. We will discuss this possibility in Chapter 4. Also, it is interesting to learn personalized tie strength thresholds. Namely, the model may assume each user is associated with a different threshold for classifying strong and weak ties. Last but not the least, one may couple the TBPR model with other item scorers like kNN, instead of matrix factorization.


Method
TBPR−W
TBPR−S
SBPR
BPR
WRMF
MostPop
Precision
0.04
0.03
0.02
0.01
Recall
0.00
0.05
0.10
0.15

(a) DBLP


Method
TBPR−W
TBPR−S
SBPR
BPR
WRMF
MostPop
Precision
0.015
0.010
Recall
0.025
0.050
0.075
0.100
0.125

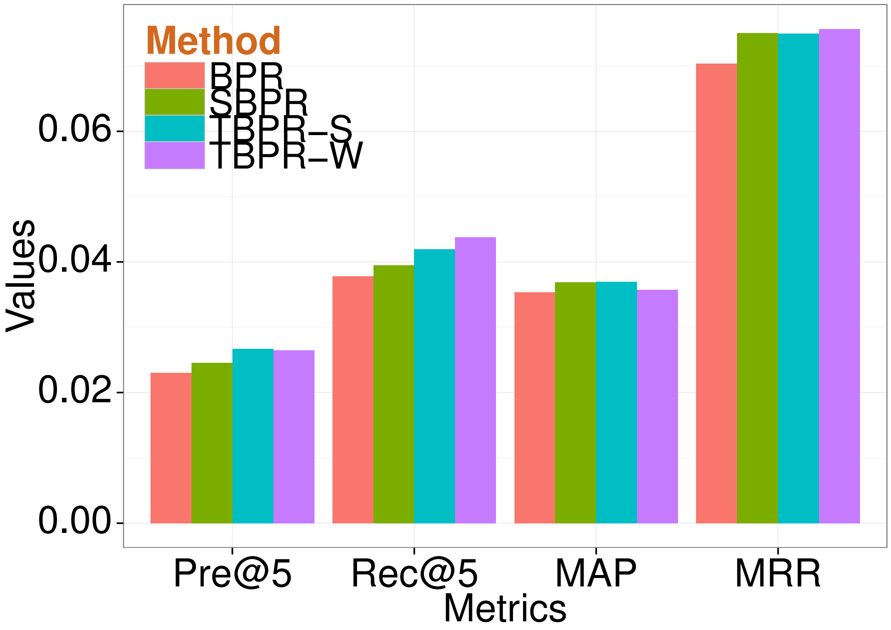
(b) Ciao



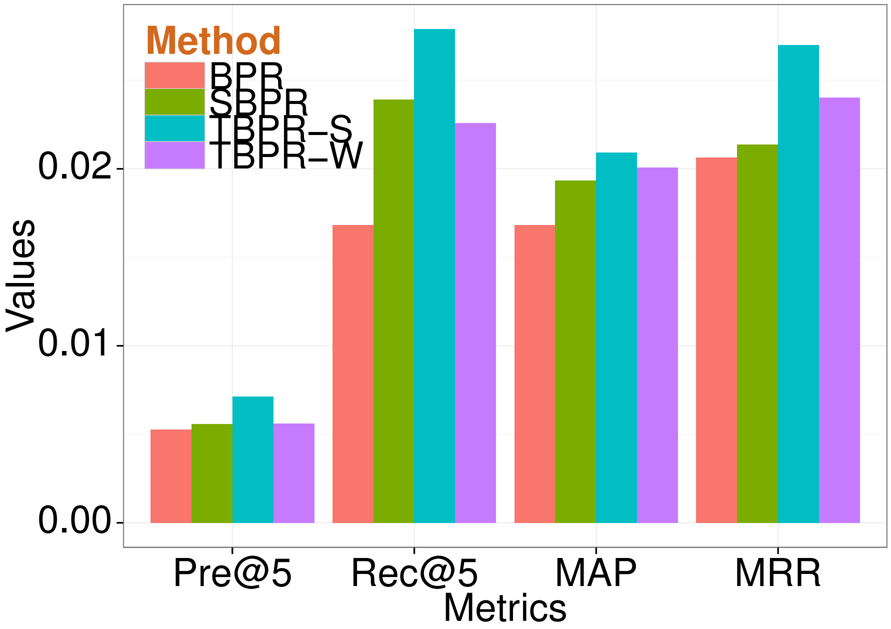
(c) Douban

(d) Epinions

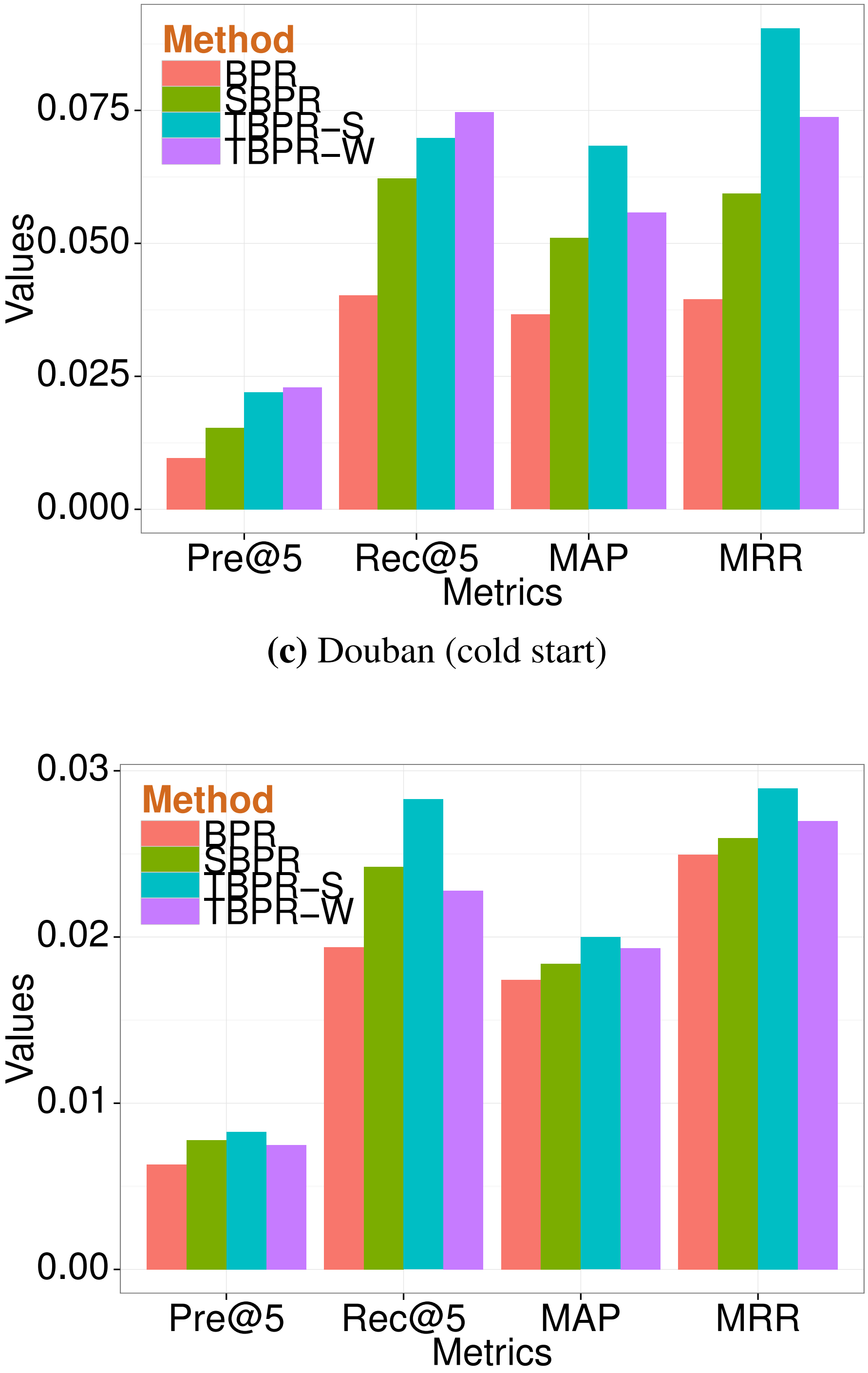
Figure 3.2: Precision@K vs Recall@K on all users, where K ranges from 5 to 50



(a) DBLP (cold start)



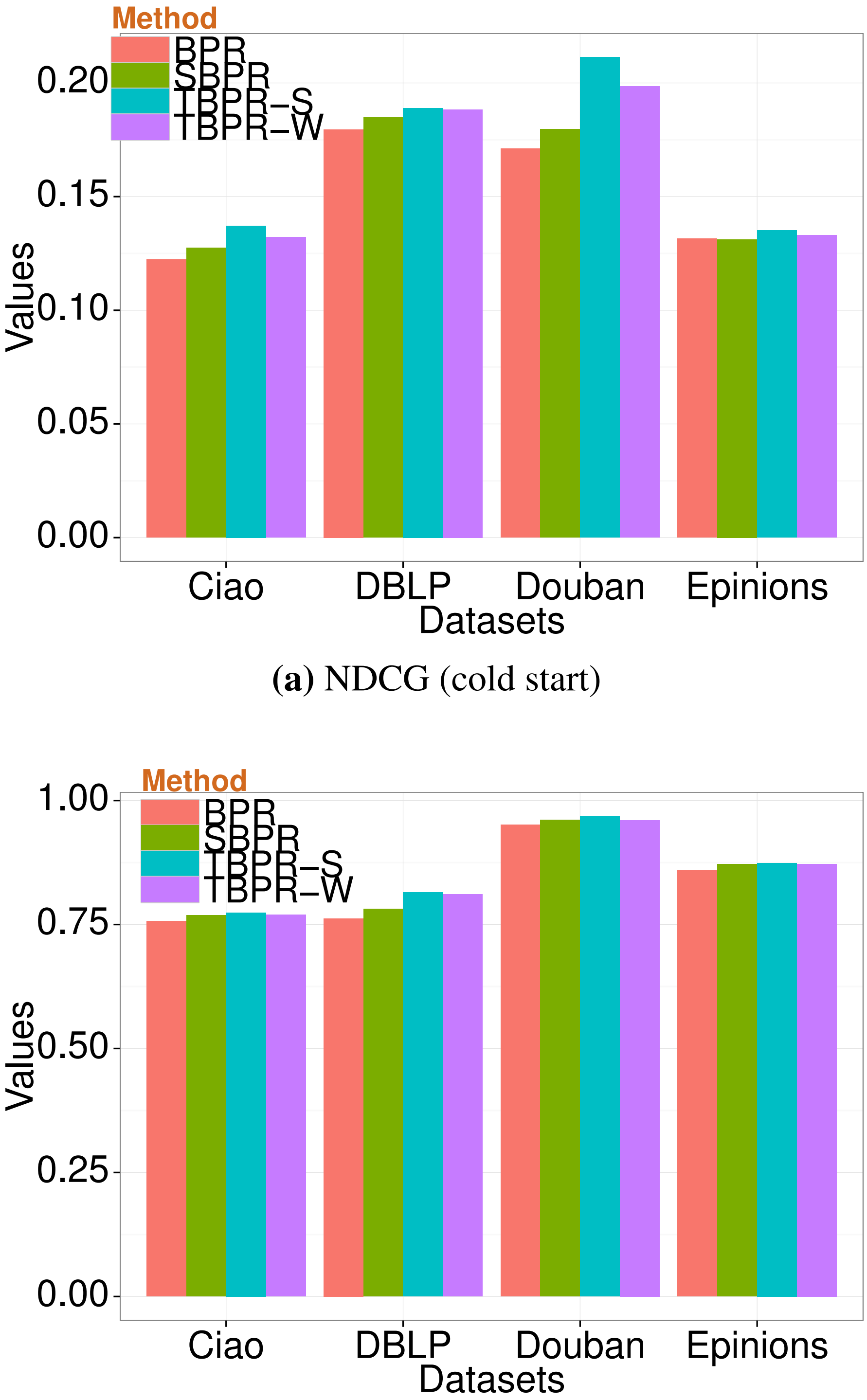
(b) Ciao (cold start)



(c) Douban (cold start)

(d) Epinions (cold start)

Figure 3.2: Performance evaluations on cold-start users (Recall, Precision, MAP, MRR)



(a) NDCG (cold start)

(b) AUC (cold start)

Figure 3.3: NDCG and AUC comparisons on cold-start users

Chapter 4

Learning Personalized Preference of Strong and Weak Ties

Recent years have seen a surge of research on social recommendation techniques for improving recommender systems due to the growing influence of social networks to our daily life. The intuition of social recommendation is that users tend to show affinities with items favored by their social ties due to social influence. Despite the extensive studies, little work has been done to distinguish between strong and weak ties in recommender systems. Although Chapter 3 in this thesis incoporates the concept of strong and weak ties, two important terms widely used in social sciences, into social recommendation, it has not attempted to learn the personalized preferences between strong and weak ties for each individual. In this chapter, we first highlight the importance of different types of ties in social relations originated from social sciences, and then propose a novel social recommendation method based on a new Probabilistic Matrix Factorization model that incorporates the distinction of strong and weak ties for improving recommendation performance. The proposed method is capable of simultaneously classifying different types of social ties in a social network w.r.t. optimal recommendation accuracy, and learning a personalized tie type preference for each user in addition to other parameters. We conduct extensive experiments on four real-world datasets by comparing our method with state-of-the-art approaches, and find encouraging results that validate the efficacy of the proposed method in exploiting the personalized preferences of strong and weak ties for social recommendation.

4.1 Motivation

Recommender systems have saturated into our daily life — we experience recommendations when we see "More Items to Consider" or "Inspired by Your Shopping Trends" on Amazon and "People You May Know" on Facebook (i.e., friend recommendation [121]) — other popular online web services such as eBay, Netflix and LinkedIn etc. also provide users with the recommendation features. Thus algorithmic recommendation [74, 106] has become a necessary mechanism for many online web services which recommend items such as music, movies or books to users. These online web services normally make recommendations based on collaborative filtering which suggests items favored by similar users. Representative collaborative filtering algorithms include low-rank matrix factorization. However, most recommender systems suffer from the *data sparsity* problem, where the number of items consumed by a user (e.g., giving a rating) is often very small compared to the total number of items (usually hundreds of thousands to millions or even billions in web-scale applications).

The *data sparsity* issue can significantly affect the performance of model-based collaborative filtering methods such as low-rank matrix factorization mainly because of two reasons: the "overfitting" problem where insufficient data is available for training models, and the "cold start" problem in which recommender systems fail to make recommendations for new users when there is no historical behavior data to be collected. To resolve the data sparsity challenge, one promising direction is resorting to *social recommendation* where the data sparsity is tackled by utilizing the rapidly growing social network information in recommender systems [50,51,77,79,80,110,118,120,128].

On the other hand, despite quite a lot of literature studies attempting to explore tie strength prediction in demographic data [93] and social media [8,14,40,41,52,68,89,90,111,114], all but one of the existing social recommendation methods fail to distinguish different types of social ties for pairs of connected users. In social sciences, Granovetter [43] introduces different types of social ties (strong, weak, and absent), and concludes that weak ties are actually the most important reason for new information or innovations to spread over social networks. Based on Granovetter's statement, the model proposed by Wang et al. [110] is the only one among those existing social recommendation approaches that pays attention to the important distinctions between strong and weak ties (Chapter3in this thesis). Nevertheless, Wang et al. simply assume every individual has the same preference for strong and weak ties — either everyone prefers strong ties to weak ties or everyone prefers weak ties to strong ties. In practice, different users may have different preferences for strong and weak ties, e.g., one may trust strong ties more than weak ties and others may behave opposite. Thus Wang's model suffers from the limitation that no personalized preferences of strong and weak ties can be learned. As such, although Wang's model addresses the concern that lacking the distinctions for different social ties may significantly limit the potential of social recommendation, we argue that ignoring the personalized tie type preference for each individual tends to result in sub-optimal solutions as well.

Therefore, inspired by the claims in social sciences and the promising results in Wang's work [110], we investigate whether distinguishing and learning the personalized tie type preference for each individual would improve the prediction accuracy of social recommendation. However, there exist several challenges for the combination of personalized tie type preferences and social recommendation. First, how to effectively identify each type of social tie ("strong" or "weak") in a given social network? Sociologists [43,44] typically assume the *dyadic hypothesis*: the strength of a tie is determined solely by the interpersonal relationship between two individuals, irrespective of the rest of the network. For example, Granovetter uses the frequency of interactions to classify strong and weak ties [44], that is, if two persons meet each other at least once a week, then their tie is deemed strong; if the frequency is more than once a year but less than twice a week, then the tie is weak. This is simple and intuitive, but requires user activity data which is not publicly available in modern online social networks because of security and privacy concerns[1]. Second, assuming there is a reliable method for differentiating between strong and weak ties, how can we efficaciously combine it with existing social recommendation approaches such as *Social Matrix Factorization* (SMF) [51] to improve the accuracy? Third, different people may have different preferences for strong and weak ties, and thus how do we learn a personalized tie type preference for each of them?

To handle these challenges, we first adopt Jaccard's coefficient [49] to compute the social tie strength [73,85]. Naturally, Jaccard's coefficient captures the extent to which those users' friendship circles overlap, making itself a feature *intrinsic* to the network topology, and requiring no additional data to compute. Our choice is supported by the studies on a large-scale mobile call graph by Onnela et al. [85], which show that (i) tie strength is partially determined by the network structure relatively local to the tie and (ii) the stronger the tie between two users, the more their friends overlap. We define ties as strong if their Jaccard's coefficient is above some threshold, and weak otherwise. We would like to point out that the optimal threshold (w.r.t. recommendation accuracy) will be learnt from the data. Furthermore, we exclude absent ties in our model because they do not play an important role as indicated in Granovetter's work. We distinguish strong and weak ties by thresholding Jaccard's coefficient between two users, while Granovetter thresholds the number of interactions between two users.

We then propose the *Personalized Social Tie Preference Matrix Factorization* (PTPMF) method, a novel probabilistic matrix factorization based model that simultaneously (i) classifies strong and weak ties w.r.t. optimal recommendation accuracy and (ii) learns a personalized preference between strong and weak ties for each user in addition to other parameters. More precisely, we employ gradient descent to learn the best (w.r.t. recommendation accuracy) threshold of tie strength (above which a tie is strong; otherwise weak) and the personalized tie type preference for each user as well as other parameters such as the latent feature vectors for users and items.

This work makes the following three contributions:

- We recognize the importance of strong and weak ties in social relations as motivated by the sociology literature, and incorporate the notion of strong and weak ties into probabilistic matrix factorization for social recommendation.

- We present a novel algorithm to simultaneously learn user-specific preferences for strong and weak ties, the optimal (w.r.t. recommendation accuracy) threshold for classifying strong and weak ties, as well as other model parameters.

[1] https://en.wikipedia.org/wiki/Privacy_concerns_with_social_networking_services

- We conduct extensive experiments on four real-world public datasets and show that our proposed method significantly outperforms the existing methods in various evaluation metrics such as RMSE, MAE etc.

The remainder of this chapter is organized as follows: Section 4.2 discusses the effects of strong and weak social ties that are evident in the sociology literature, and proposes to incorporate these notions into social recommendation. Section 4.3 gives a detailed formation of our proposed *Personalized Social Tie Preference Matrix Factorization* (PTPMF) model, followed by a description of model inferences for PTPMF in Section 4.4. Section 4.5 presents our experiments, compares our approach with baseline recommendation methods and comments on their performances for both all users and cold-start users in terms of various evaluation metrics. Finally, we summary our work in Section 4.6.

4.2 Social Ties from Offline to Online

Speaking of interpersonal ties, Granovetter may probably be the first one who comes into our mind. Granovetter, in his book *Getting a job: A study of contacts and careers* [44], conducts a survey among 282 professional, technical, and managerial workers in Newton, Massachusetts and reports that personal contact is the predominant method of finding out about jobs. The result of his survey shows that nearly 56% of his respondents used personal contacts to find a job while 18.8% used formal means and 18.8% used direct applications instead. Besides, Granovetter's research also demonstrates that most respondents prefer the use of personal contacts to other means and that using personal contacts can lead to a higher level of job satisfaction and income. Thus it will be interesting to explore the important role social influence plays in people's decision making process which does not necessarily need to be limited to an employee's decision about changing a job.

Social influence takes effect through a social network which consists of people and interpersonal ties connecting these people in the network. Granovetter, in his other work [43], introduces different types of interpersonal ties (e.g., strong tie, weak tie and absent tie) and concludes that weak ties are the most important source for new information or innovations to reach distant parts of the network. Again, different ties between the job changer and the contact person who provided the necessary information are analyzed and the strength and importance of weak ties in occupational mobility are shown in [44]. In the late 1960's and early 1970's when the Internet had not come into existence, tie strength was measured in terms of how often they saw the contact person during the period of the job transition, using the following measurement:

- **Often**: at least once a week

- **Occasionally**: more than once a year but less than twice a week

- **Rarely**: once a year or less

In the age of information, social media and online social networks are playing crucial roles in the establishment of social networks. We are able to know new friends and form new relationships/ties through the Internet without necessarily meeting them face to face. Just as Kavanaugh et al. [54] state, the appearance of the Internet has helped to strengthen weak ties and increase their numbers across social groups. Though the importance of weak ties has been exposed to us by sociologists, it is not wise to ignore the roles strong ties play in our lives because strong ties should intuitively be more trustworthy than weak ties. On the other side, different individuals may have different relative degree of trust for their strong and weak ties — one may trust his/her strong ties (or weak ties) more than one another. Thus an interesting and challenging question is that how to learn these user-specific (and perhaps different) preferences for different types of ties. This being the case, considering both strong and weak ties in social recommendation, then optimally distinguishing them w.r.t recommendation accuracy and finally learning a user-specific personalized tie type preference become three key parts of an appropriate solution to improve social recommendation.

In this section we will present how the notion of strong/weak ties and the thresholding strategy are incorporated into social recommendation. We leave the remaining two parts to section 4.3 for more concrete descriptions. In order that the distinction between strong and weak ties can be incorporated into social recommendation, we will need to be able to define and compute tie strength, and then classify ties. Several potential options seem to serve as adequate candidates. First, as mentioned in Section 4.1, sociologists use dyadic measures such as frequency of interactions [44]. However, this method is not generally applicable due to lack of necessary data. An alternative approach relies on community detection. Specifically, it first runs a community detection algorithm to partition the network $\mathcal{G} = (U, E)$ into several subgraphs. Then, for each edge $(u, v) \in \mathcal{E}$, if u and v belong to the same subgraph, then it is classified as a strong tie; otherwise a weak tie. However, a key issue is that although numerous community detection algorithms exist [34], they tend to produce (very) different clusterings, and it is unclear how to decide which one to use. Furthermore, if a "bad" partitioning (w.r.t. prediction accuracy) is produced and given to the recommender system as input, it would be very difficult for the recommender system to recover. In other words, the quality of recommendation would depend on an *exogenous* community detection algorithm that the recommender system *has no control over.* Hence, this approach is undesirable.

In light of the above, we resort to node-similarly metrics that measure neighborhood overlap of two nodes in the network. The study of Onnela et al. [85] provides empirical confirmation of this intuition: they find that (i) tie strength is in part determined by the local network structure and (ii) the stronger the tie between two users, the more their friends overlap. In addition, unlike frequency of interactions, node-similarity metrics are intrinsic to the network, requiring no additional data to compute. Also, unlike the community detection based approach, we still get to choose a tie classification method that best serves the interest of the recommender system.

More specifically, we use Jaccard's coefficient [49], a simple measure that effectively captures neighborhood overlap. Let $\mathsf{strength}(u, v)$ denote the tie strength for any $(u, v) \in \mathcal{E}$. We have:

$$\mathsf{strength}(u, v) =_{\text{def}} \frac{|\mathcal{N}_u \cap \mathcal{N}_v|}{|\mathcal{N}_u \cup \mathcal{N}_v|} \quad \textit{(Jaccard)}, \tag{4.1}$$

where $\mathcal{N}_u \subseteq \mathcal{U}$ (resp. $\mathcal{N}_v \subseteq \mathcal{U}$) denotes the set of ties of u (resp. v). If $\mathcal{N}_u = \mathcal{N}_v = \emptyset$ (i.e., both u and v are singleton nodes), then simply define $\mathsf{strength}(u, v) = 0$. By definition, all strengths as defined in Equation (4.1) fall into the interval $[0, 1]$. This definition has natural probabilistic interpretations: Given two arbitrary users u and v, their Jaccard's coefficient is equal to the probability that a randomly chosen tie of u (resp. v) is also a tie of v (resp. u) [73].

Thresholding. To distinguish between strong and weak ties, we adopt a simple *thresholding* method. For a given social network graph $\mathcal{G}$, let $\theta_{\mathcal{G}} \in [0, 1)$ denote the threshold of tie strength such that

$$(u, v) \text{ is } \begin{cases} \text{strong}, & \text{if } \mathsf{strength}(u, v) > \theta_{\mathcal{G}}; \\ \text{weak}, & \text{if } \mathsf{strength}(u, v) \leq \theta_{\mathcal{G}}. \end{cases} \tag{4.2}$$

Let $\mathcal{W}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) \leq \theta_{\mathcal{G}}\}$ denote the set of all weak ties of u. Similarly, $\mathcal{S}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) > \theta_{\mathcal{G}}\}$ denotes the set of all strong ties of u. Clearly, $\mathcal{W}_u \cap \mathcal{S}_u = \emptyset$ and $\mathcal{W}_u \cup \mathcal{S}_u = \mathcal{N}_u$.

The value of $\theta_{\mathcal{G}}$ in our proposed approach is *not* hardwired, but rather is left for our model to learn (Section 4.3), such that the resulting classification of strong and weak ties in $\mathcal{G}$, together with other learned parameters of the model, leads to the best accuracy of recommendations. We conclude this section by pointing out that Granovetter and we both threshold strong and weak ties, we utilize Jaccard's coefficient (degree of connectivity between users) to do the thresholding while Granovetter resorts to the number of interactions between users instead.

4.3 Personalized Tie Preference Matrix Factorization for Social Recommendation

In this section, we present the proposed new model of *Personalized Tie Preference Matrix Factorization* (PTPMF) for social recommendation in detail. Before introducing PTPMF, we will first briefly explain some background knowledge of the classical *Probabilistic Matrix Factorization* (PMF) and of another popular social recommendation model known as *Social Matrix Factorization* (SMF).

4.3.1 Probabilistic Matrix Factorization

In recommender systems, we are given a set of users $\mathbb{U}$ and a set of items $\mathbb{I}$, as well as a $|\mathbb{U}| \times |\mathbb{I}|$ rating matrix R whose non-empty (observed) entries R_{ui} represent the feedbacks (e.g., ratings, clicks etc.)

of user $u \in \mathbb{U}$ for item $i \in \mathbb{I}$. When it comes to social recommendation, another $|\mathbb{U}| \times |\mathbb{U}|$ social tie matrix T whose non-empty entries T_{uv} denote $u \in \mathbb{U}$ and $v \in \mathbb{U}$ are ties, may also be necessary. The task is to predict the missing values in R, i.e., given a user $v \in \mathbb{U}$ and an item $j \in \mathbb{I}$ for which R_{vj} is unknown, we predict the rating of v for j using observed values in R and T (if available).

A matrix factorization model assumes the rating matrix R can be approximated by a multiplication of d-rank factors,

$$R \approx U^T V, \tag{4.3}$$

where $U \in \mathbb{R}^{d\times|\mathbb{U}|}$ and $V \in \mathbb{R}^{d\times|\mathbb{I}|}$. Normally d is far less than both $|\mathbb{U}|$ and $|\mathbb{I}|$. Thus given a user u and an item i, the rating R_{ui} of u for i can be approximated by the dot product of user latent feature vector U_u and item latent feature V_i,

$$R_{ui} \approx U_u^T V_i, \tag{4.4}$$

where $U_u \in \mathbb{R}^{d\times 1}$ is the u_{th} column of U and $V_i \in \mathbb{R}^{d\times 1}$ is the i_{th} column of V. For ease of notation, we let $|\mathbb{U}| = N$ and $|\mathbb{I}| = M$ in the remaining of this chapter.

Later, the probabilistic version of matrix factorization, i.e., *Probabilistic Matrix Factorization* (PMF), is introduced in [82], based on the assumption that the rating R_{ui} follows a normal distribution whose mean is some function of $U_u^T V_i$. The conditional probability of the observed ratings is:

$$p(R|U, V, \sigma_R^2) = \prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}\left(R_{ui}|g(U_u^T V_i), \sigma_R^2\right)\right]^{I_{ui}^R}, \tag{4.5}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2. If u has rated i, then the indicator function I_{ui}^R equals to 1, otherwise equals to 0. $g(\cdot)$ is the sigmoid function, i.e., $g(x) = \frac{1}{1+e^{-x}}$, which bounds the range of $U_u^T V_i$ within $[0, 1]$. Moreover, U_u and V_i are both subject to a zero mean normal distribution. Thus the conditional probabilities of user and item latent feature vectors are:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{u=1}^{N}\mathcal{N}\left(U_u|0, \sigma_U^2\mathbf{I}\right) \\ p(V|\sigma_V^2) &= \prod_{i=1}^{M}\mathcal{N}\left(V_i|0, \sigma_V^2\mathbf{I}\right), \end{aligned} \tag{4.6}$$

where $\mathbf{I}$ is the identity matrix. Therefore, the posterior probability of the latent variables U and V can be calculated through a Bayesian inference,

$$\begin{aligned} &p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \\ &\propto p(R|U, V, \sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \end{aligned}$$

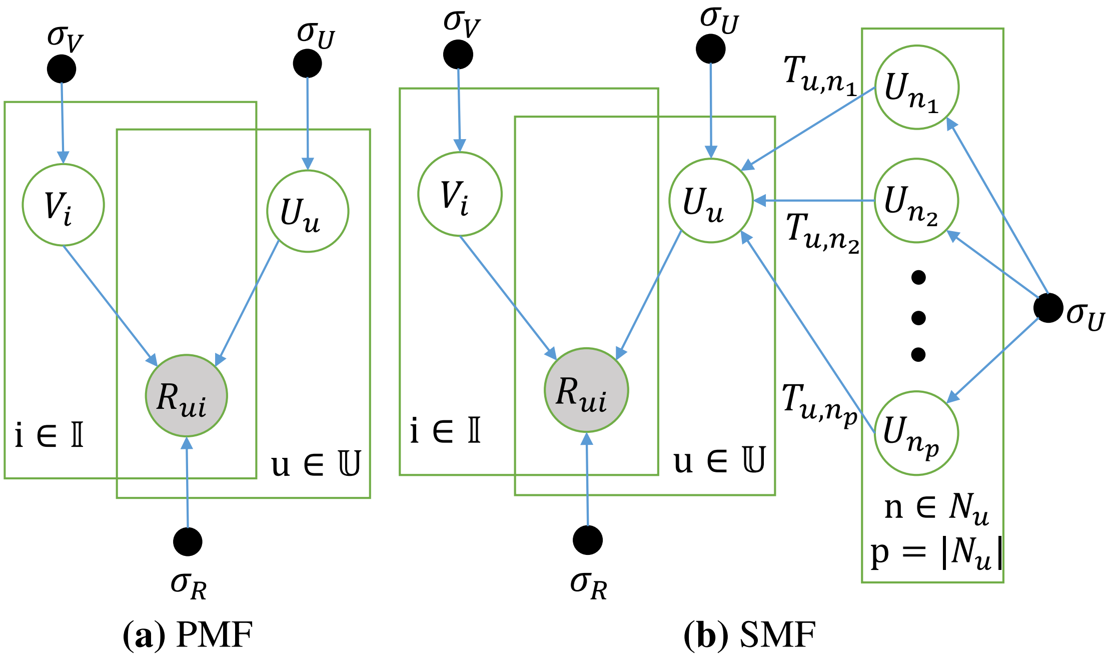


Figure 4.1: Graphical models of PMF and SMF

$$= \prod_{u=1}^{N} \prod_{i=1}^{M} \left[\mathcal{N}\left(R_{ui}|g(U_u^T V_i), \sigma_R^2\right)\right]^{I_{ui}^R}$$
$$\times \prod_{u=1}^{N} \mathcal{N}(U_u|0, \sigma_U^2 \mathbf{I}) \times \prod_{i=1}^{M} \mathcal{N}(V_i|0, \sigma_V^2 \mathbf{I}). \tag{4.7}$$

The graphical model of PMF is demonstrated in Figure 4.1a and readers may refer to [82] for more details.

4.3.2 Social Matrix Factorization

There has been some work on social recommendation, among which Jamali and Ester [51] present a well-known social recommendation model called *Social Matrix Factorization* (SMF) that incorporates trust propagation into probabilistic matrix factorization, assuming that the rating behavior of a user u will be affected by his social ties N_u through social influence. In SMF, the latent feature vector of user u depends on the latent feature vectors of u's social ties n, i.e., $n \in N_u$. As is shown by the graphical model of SMF in Figure 4.1b,

$$U_u = \frac{\sum_{n \in N_u} T_{un} U_n}{|N_u|},$$

where U_u is u's latent feature vector and N_u is the set of social ties of user u. T_{un} is either 1 or 0, indicating u and n are "ties" or "not ties".

The posterior probability of user and item latent feature vectors in SMF, given the observed ratings and social ties as well as the hyperparameters, is shown in (4.8).

$$\begin{aligned}
&p(U,V|R,T,\sigma_R^2,\sigma_T^2,\sigma_U^2,\sigma_V^2)\\
&\propto p(R|U,V,\sigma_R^2)p(U|T,\sigma_T^2,\sigma_U^2)p(V|\sigma_V^2)\\
&=\prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}\left(R_{ui}|g(U_u^T V_i),\sigma_R^2\right)\right]^{I_{ui}^R}\\
&\times\prod_{u=1}^{N}\mathcal{N}\left(U_u|\sum\nolimits_{k\in N_u}T_{uk}U_k,\sigma_T^2\mathbf{I}\right)\\
&\times\prod_{u=1}^{N}\mathcal{N}(U_u|0,\sigma_U^2\mathbf{I})\times\prod_{i=1}^{M}\mathcal{N}(V_i|0,\sigma_V^2\mathbf{I}).
\end{aligned}\tag{4.8}$$

The main idea in (4.8) and Figure 4.1b is that the latent feature vectors of users should be similar to the latent feature vectors of their social ties. We refer readers to [51] for more details.

4.3.3 The PTPMF Model

We divide social ties into two groups: strong ties and weak ties. People usually tend to share more common intrinsic properties with their strong ties while they are more likely to be exposed to new information through their weak ties. Both strong ties and weak ties are important in terms of social influence while they play different roles in affecting people. For an individual user, strong ties tend to be more similar to her, on the other hand, weak ties may provide her with more valuable information which can not be obtained from strong ties. Based on this assumption, we propose our approach, PTPMF, to utilize the different roles of strong and weak ties when making recommendations. Besides, by introducing two additional parameters, $\theta_{\mathcal{G}}$ and B_u, PTPMF is capable of learning the optimal (w.r.t. recommendation accuracy) threshold for classifying strong and weak ties, user-specific preferences between strong and weak ties as well as other parameters at the same time.

Figure 4.2 presents the graphical model of PTPMF. Same as in Chapter3, we introduce a random variable $\theta_{\mathcal{G}}$ for the threshold classifying strong and weak ties. S_u and W_u are the sets of strong and weak ties of user u respectively, classified according to (4.2). Due to different roles of strong and weak ties in affecting users' rating behaviors, we introduce two new random variables, U_u^s and U_u^w, as strong-tie and weak-tie latent feature vectors for each user u. The strong-tie (resp. weak-tie) latent feature vector of u is dependent on the latent feature vectors of all u's strong ties (resp. weak-ties). This influence is modeled as follows:

$$U_u^s=\frac{\sum_{v\in S_u}T_{uv}U_v}{\sum_{v\in S_u}T_{uv}}\quad\text{and}\quad U_u^w=\frac{\sum_{v\in W_u}T_{uv}U_v}{\sum_{v\in W_u}T_{uv}},$$

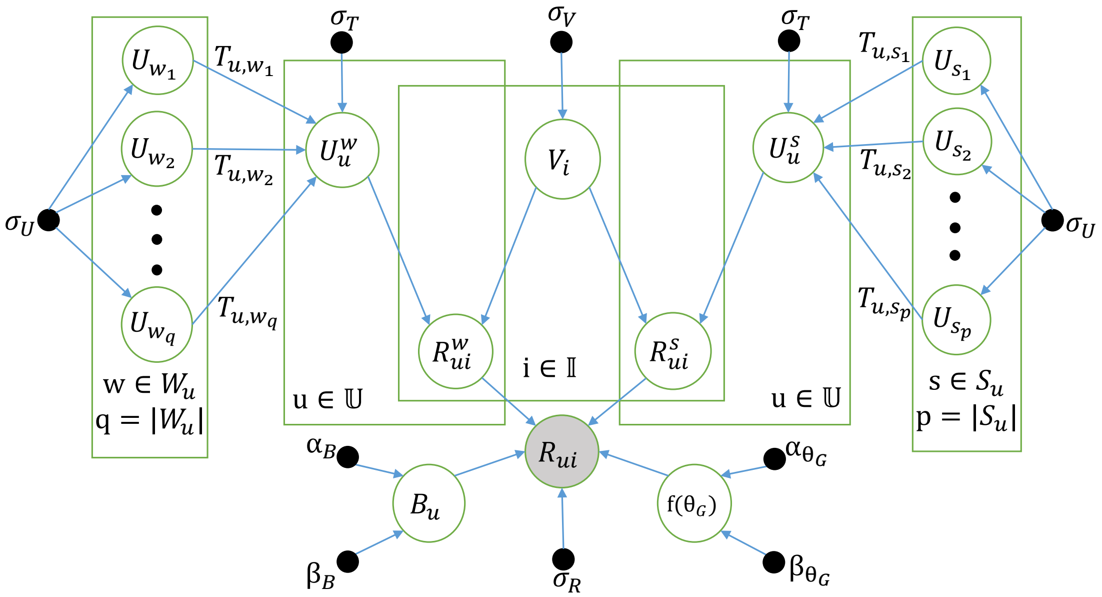


Figure 4.2: Graphical model of the proposed PTPMF

where $T_{uv} = \mathsf{strength}(\mathsf{u}, \mathsf{v})$ is the tie strength between u and v defined in (4.1), different from SMF in which T is a Boolean variable. We normalize the tie strength of u and her social ties so that $\sum_{v \in S_u} T_{uv} = 1$ and $\sum_{v \in W_u} T_{uv} = 1$. Now the conditional probability of weak-tie and strong-tie latent feature vectors, U_u^w and U_u^s, becomes:

$$
\begin{aligned}
& p(U^w, U^s | T, U, \sigma_T^2) \\
= & \prod_{u=1}^{N} \mathcal{N}(U_u^w | \sum_{k \in W_u} T_{uk} U_k, \sigma_T^2 \mathbf{I}) \\
\times & \prod_{u=1}^{N} \mathcal{N}(U_u^s | \sum_{k \in S_u} T_{uk} U_k, \sigma_T^2 \mathbf{I}).
\end{aligned} \tag{4.9}
$$

The dot product of U_u^w (resp. U_u^s) and item latent feature vector V_i then determines u's weak-tie generated rating on item i (resp. u's strong-tie generated rating on item i), denoted by R_{ui}^w (resp. R_{ui}^s). Different from SMF, PTPMF further enables the learning of a personalized preference between strong and weak ties for each user through introducing another new variable, B_u, as the probability that u prefers weak ties to strong ties. Hence, $1 - B_u$ is the probability that u prefers strong ties instead. To generate u's final rating for item i, PTPMF puts more emphasis on her weak-tie generated rating R_{ui}^w with probability B_u, and on her strong-tie generated rating R_{ui}^s with probability $1 - B_u$ (more details to be discussed below). Thus the conditional probability of the observed ratings can be expressed as:

$$
p(R | U^w, U^s, V, B, \theta_{\mathcal{G}}, T, \sigma_R^2)
$$

$$= \prod_{u=1}^{N}\prod_{i=1}^{M}\Bigg[\mathcal{N}\Bigg(R_{ui}|g\Big(B_u\Big[f(\theta_{\mathcal{G}})U_u^{wT}V_i + (1-f(\theta_{\mathcal{G}}))U_u^{sT}V_i\Big]$$

$$+ (1-B_u)\Big[(1-f(\theta_{\mathcal{G}}))U_u^{wT}V_i + f(\theta_{\mathcal{G}})U_u^{sT}V_i\Big]\Big), \sigma_R^2\Bigg)\Bigg]^{I_{ui}^R}, \tag{4.10}$$

where $g(\cdot)$ is the sigmoid function, i.e., $g(x) = \frac{1}{1+e^{-x}}$, and $f(\theta_{\mathcal{G}}) = g\big((t_s - \theta_{\mathcal{G}})(\theta_{\mathcal{G}} - t_w)\big) \geq 0.5$, given t_s, t_w as the average tie strength of strong ties and weak ties respectively. The underlying intuition is that when a threshold $\theta_{\mathcal{G}}$ gives a small degree of separation, t_s and t_w will be close to $\theta_{\mathcal{G}}$, $f(\theta_{\mathcal{G}})$ will then be close to 0.5, indicating very few distinctions between strong and weak ties. Similarly, a larger degree of separation results in more distinctions between strong and weak ties in our model. When u prefers weak ties, more weight (i.e., $f(\theta_{\mathcal{G}}) \geq 0.5$) will be given to her weak-tie generated rating (i.e., $U_u^{wT}V_i$), less weight (i.e., $1 - f(\theta_{\mathcal{G}}) \leq 0.5$) will be given to her strong-tie generated rating (i.e., $U_u^{sT}V_i$) and vice versa. Moreover, how much weight to give is dependent upon how well the current threshold, $\theta_{\mathcal{G}}$, classifies strong and weak ties – a larger degree of separation given by $\theta_{\mathcal{G}}$ will result in more weight being given to the preferred tie type.

We assume $\theta_{\mathcal{G}}$ and B follow a Beta distribution so that both of them lie in $[0, 1]$. Also, U and V follow the same zero mean normal distribution in (4.6). Through a Bayesian inference, the posterior probability of all model parameters, given the observed ratings and social ties as well as the hyperparameters, is shown in (4.11).

$$\begin{aligned}
&p(U^w, U^s, U, V, B, \theta_{\mathcal{G}}|R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2)\\
&\propto p(R|U^w, U^s, V, B, \theta_{\mathcal{G}}, T, \sigma_R^2)p(U^w, U^s|T, U, \sigma_T^2)\\
&\quad p(U|\sigma_U^2)p(V|\sigma_V^2)p(\theta_{\mathcal{G}}|\alpha_{\theta_{\mathcal{G}}}, \beta_{\theta_{\mathcal{G}}})p(B|\alpha_B, \beta_B)\\
&= \prod_{u=1}^{N}\prod_{i=1}^{M}\\
&\Bigg[\mathcal{N}\Bigg(R_{ui}|g\Big(B_u\Big[f(\theta_{\mathcal{G}})U_u^{wT}V_i + (1-f(\theta_{\mathcal{G}}))U_u^{sT}V_i\Big]\\
&+ (1-B_u)\Big[(1-f(\theta_{\mathcal{G}}))U_u^{wT}V_i + f(\theta_{\mathcal{G}})U_u^{sT}V_i\Big]\Big), \sigma_R^2\Bigg)\Bigg]^{I_{ui}^R}\\
&\times \prod_{u=1}^{N}\mathcal{N}(U_u^w|\sum_{k\in W_u}T_{uk}U_k, \sigma_T^2\mathbf{I})\\
&\times \prod_{u=1}^{N}\mathcal{N}(U_u^s|\sum_{k\in S_u}T_{uk}U_k, \sigma_T^2\mathbf{I})\\
&\times \prod_{u=1}^{N}\mathcal{N}(U_u|0, \sigma_U^2\mathbf{I}) \times \prod_{i=1}^{M}\mathcal{N}(V_i|0, \sigma_V^2\mathbf{I})\\
&\times Beta(\theta_{\mathcal{G}}|\alpha_{\theta_{\mathcal{G}}}, \beta_{\theta_{\mathcal{G}}}) \times \prod_{u=1}^{N}Beta(B_u|\alpha_B, \beta_B).
\end{aligned} \tag{4.11}$$

Compared to SMF, our PTPMF model shown in (4.11) and Figure 4.2 treats strong and weak ties separately, learns the optimal (w.r.t. recommendation accuracy) threshold for distinguishing strong and weak ties. In addition, our PTPMF is able to learn a personalized tie preference (denoted as B_u) for each user u. Our goal is to learn $U, U^w, U^s, V, B, \theta_\mathcal{G}$ which maximize the posterior probability shown in (4.11).

4.4 Parameter Learning

We learn the parameters of PTPMF using maximum a posteriori (MAP) inference. Taking the $\ln$ on both sides of (4.11), we are maximizing the following objective function:

$$
\begin{aligned}
&\ln p(U^w, U^s, U, V, B, \theta_\mathcal{G} | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) \\
&= -\frac{1}{2\sigma_R^2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{ui}^R \Big(R_{ui} - g(\mu_{R_{ui}}) \Big)^2 \\
&- \frac{1}{2\sigma_U^2} \sum_{u=1}^{N} U_u^T U_u - \frac{1}{\sigma_V^2} \sum_{i=1}^{M} V_i^T V_i \\
&- \frac{1}{\sigma_T^2} \sum_{u=1}^{N} \Big((U_u^w - \sum_{k \in W_u} T_{uk} U_k)^T (U_u^w - \sum_{k \in W_u} T_{uk} U_k) \Big) \\
&- \frac{1}{\sigma_T^2} \sum_{u=1}^{N} \Big((U_u^s - \sum_{k \in S_u} T_{uk} U_k)^T (U_u^s - \sum_{k \in S_u} T_{uk} U_k) \Big) \\
&+ \sum_{u=1}^{N} \Big((\alpha_B - 1) \ln B_u + (\beta_B - 1) \ln (1 - B_u) \Big) \\
&+ (\alpha_{\theta_\mathcal{G}} - 1) \ln \theta_\mathcal{G} + (\beta_{\theta_\mathcal{G}} - 1) \ln (1 - \theta_\mathcal{G}) \\
&- \frac{1}{2} \Big((N \cdot K) \ln \sigma_U^2 + (M \cdot K) \ln \sigma_V^2 + (2N \cdot K) \ln \sigma_T^2 \Big) \\
&- \frac{1}{2} (\sum_{u=1}^{N} \sum_{i=1}^{M} I_{ui}^R) \ln \sigma_R^2 - N \ln B(\alpha_B, \beta_B) - \ln B(\alpha_{\theta_\mathcal{G}}, \beta_{\theta_\mathcal{G}}) \\
&+ \text{Constant},
\end{aligned}
\tag{4.12}
$$

where

$$
\begin{aligned}
\mu_{R_{ui}} &= B_u \Big(f(\theta_\mathcal{G}) U_u^{wT} + (1 - f(\theta_\mathcal{G})) U_u^{sT} \Big) V_i \\
&\quad + (1 - B_u) \Big((1 - f(\theta_\mathcal{G})) U_u^{wT} + f(\theta_\mathcal{G}) U_u^{sT} \Big) V_i,
\end{aligned}
\tag{4.13}
$$

and $B(\cdot, \cdot)$ is the beta function:

$$
B(x, y) = \int_0^1 t^{x-1} (1 - t)^{y-1} dt. \tag{4.14}
$$

Fixing the Gaussian noise variance and beta shape parameters, maximizing the log-posterior in (4.12) over $U^w, U^s, U, V, B, \theta_{\mathcal{G}}$ is equivalent to minimizing the following objective function:

$$\begin{aligned}
&\mathcal{L}(R, T, U^w, U^s, U, V, B, \theta_{\mathcal{G}}) \\
&= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{ui}^{R}\Big(R_{ui} - g(\mu_{R_{ui}})\Big)^2 \\
&+ \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u^T U_u + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i^T V_i \\
&+ \frac{\lambda_T}{2}\sum_{u=1}^{N}\Big((U_u^w - \sum_{k\in W_u} T_{uk}U_k)^T(U_u^w - \sum_{k\in W_u} T_{uk}U_k)\Big) \\
&+ \frac{\lambda_T}{2}\sum_{u=1}^{N}\Big((U_u^s - \sum_{k\in S_u} T_{uk}U_k)^T(U_u^s - \sum_{k\in S_u} T_{uk}U_k)\Big) \\
&- \lambda_B\sum_{u=1}^{N}\Big((\alpha_B - 1)\ln B_u + (\beta_B - 1)\ln(1 - B_u)\Big) \\
&- \lambda_{\theta_{\mathcal{G}}}\Big((\alpha_{\theta_{\mathcal{G}}} - 1)\ln\theta_{\mathcal{G}} + (\beta_{\theta_{\mathcal{G}}} - 1)\ln(1 - \theta_{\mathcal{G}})\Big),
\end{aligned} \tag{4.15}$$

where $\lambda_U = \frac{\sigma_R^2}{\sigma_U^2}$, $\lambda_V = \frac{\sigma_R^2}{\sigma_V^2}$, $\lambda_T = \frac{\sigma_R^2}{\sigma_T^2}$ and $\lambda_B = \lambda_{\theta_{\mathcal{G}}} = \sigma_R^2$.

A local minimum of (4.15) can be found by taking the derivative and performing gradient descent on $U^w, U^s, U, V, B, \theta_{\mathcal{G}}$ separately. The corresponding partial derivative with respect to each model parameter is shown as follows:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U_u^s} &= \sum_{i=1}^{M} I_{ui}^{R}\Big(g(\mu_{R_{ui}}) - R_{ui}\Big)g^{'}(\mu_{R_{ui}}) \\
&\quad\Big(B_u + f(\theta_{\mathcal{G}}) - 2B_u f(\theta_{\mathcal{G}})\Big)V_i \\
&\quad + \lambda_T\Big(U_u^s - \sum_{k\in S_u} T_{uk}U_k\Big),
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U_u^w} &= \sum_{i=1}^{M} I_{ui}^{R}\Big(g(\mu_{R_{ui}}) - R_{ui}\Big)g^{'}(\mu_{R_{ui}}) \\
&\quad\Big(1 - f(\theta_{\mathcal{G}}) - B_u + 2B_u f(\theta_{\mathcal{G}})\Big)V_i \\
&\quad + \lambda_T\Big(U_u^w - \sum_{k\in W_u} T_{uk}U_k\Big),
\end{aligned} \tag{4.17}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U_u} &= \lambda_U U_u - \lambda_T \sum_{v|u\in W_v} T_{vu}(U_v^w - \sum_{k\in W_v} T_{vk}U_k) \\
&\quad - \lambda_T \sum_{v|u\in S_v} T_{vu}(U_v^s - \sum_{k\in S_v} T_{vk}U_k),
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial V_i} &= \sum_{u=1}^{N} I_{ui}^{R}\Big(g(\mu_{R_{ui}}) - R_{ui}\Big)g^{'}(\mu_{R_{ui}}) \\
&\quad \bigg(\Big(B_u + f(\theta_{\mathcal{G}}) - 2B_u f(\theta_{\mathcal{G}})\Big)U_u^s \\
&\quad + \Big(1 - f(\theta_{\mathcal{G}}) - B_u + 2B_u f(\theta_{\mathcal{G}})\Big)U_u^w\bigg) + \lambda_V V_i, \qquad (4.19) \\
\frac{\partial \mathcal{L}}{\partial B_u} &= \sum_{i=1}^{M} I_{ui}^{R}\Big(g(\mu_{R_{ui}}) - R_{ui}\Big)g^{'}(\mu_{R_{ui}}) \\
&\qquad \bigg(\Big(2f(\theta_{\mathcal{G}}) - 1\Big)U_u^{wT} + \Big(1 - 2f(\theta_{\mathcal{G}})\Big)U_u^{sT}\bigg)V_i \\
&\quad - \lambda_B\Big(\frac{\alpha_B - 1}{B_u} - \frac{\beta_B - 1}{1 - B_u}\Big), \qquad (4.20) \\
\frac{\partial \mathcal{L}}{\partial \theta_{\mathcal{G}}} &= (t_s + t_w - 2)g^{'}\Big((t_s - \theta_{\mathcal{G}})(\theta_{\mathcal{G}} - t_w)\Big) \\
&\quad \sum_{u=1}^{N}\sum_{i=1}^{M} I_{ui}^{R}\Big(g(\mu_{R_{ui}}) - R_{ui}\Big)g^{'}(\mu_{R_{ui}}) \\
&\quad \Big((2B_u - 1)U_u^{wT} + (1 - 2B_u)U_u^{sT}\Big)V_i \\
&\quad - \lambda_{\theta_{\mathcal{G}}}\Big(\frac{\alpha_{\theta_{\mathcal{G}}} - 1}{\theta_{\mathcal{G}}} - \frac{\beta_{\theta_{\mathcal{G}}} - 1}{1 - \theta_{\mathcal{G}}}\Big). \qquad (4.21)
\end{aligned}$$

The update is done using standard gradient descent:

$$x^{(t+1)} = x^{(t)} + \eta^{(t)} \cdot \frac{\partial \mathcal{L}}{\partial x}(x^{(t)}), \qquad (4.22)$$

where η is the learning rate and $x \in \{U^w, U^s, U, V, B, \theta_G\}$ denotes any model parameter. Finally, the algorithm terminates when the absolute difference between the losses in two consecutive iterations is less than 10^{-5}.

We note that in order to avoid overfitting, our proposed model has the standard regularization terms (L2 norm) for user latent feature vectors ($\sum U_u^T U_u$) and item latent feature vectors ($\sum V_i^T V_i$) in the third line of (4.15). Since the weak tie and strong tie latent feature vectors depend on the user latent feature vectors, these additional parameters in our model are also indirectly regularized.

4.5 Empirical Evaluation

In this section, we report the results of our experiments on four real-world public datasets and compare the performance of our PTPMF model with different baseline methods in terms of various evaluation metrics. Our experiments aim to examine if incorporating the new concepts of distinguishing strong and weak ties as well as learning a personalized tie type preference for each user is able to improve the recommendation accuracy as measured by MAE / RMSE (how close the predicted ratings are to the real ones) and Precision@K / Recall@K (accuracy for top-K recommendations), and how significant are the improvements achieved if any.

4.5.1 Experimental Settings

Datasets.

We use the following four real-world datasets.

- *Flixster.* The Flixster dataset [2] containing information of user-movie ratings and user-user friendships from Flixster, an American social movie site for discovering new movies (`http://www.flixster.com/`).
- *CiaoDVD.* This public dataset contains trust relationships among users as well as their ratings on DVDs and was crawled from the entire category of DVDs of a UK DVD community website (`http://dvd.ciao.co.uk`) in December, 2013 [45].
- *Douban.* This public dataset[3] is extracted from the Chinese Douban movie forum (`http://movie.douban.com/`), which contains user-user friendships and user-movie ratings.
- *Epinions.* This is the Epinions dataset[4] which consists of user-user trust relationships and user-item ratings from Epinions (`http://www.epinions.com/`).

The statistics of these data sets are summarized in Table 4.1.

Table 4.1: Overview of datasets (#non-zeros means the number of user-item pairs that have feedback)

	Flixster	*CiaoDVD*	*Douban*	*Epinions*
#users	76013	1881	64642	31117
#items	48516	12900	56005	139057
#non-zeros	7350235	33510	9133529	654103
#ties (edges)	1209962	15155	1390960	410570

For all the datasets, we randomly choose 80% of each user's ratings for training, leaving the remainder for testing. We split the portion of the 80% of the dataset (i.e., the training set) into five equal sub-datasets for 5-fold cross validation. During the training and validation phase, each time we use one of the five sub-datasets for validation and the remaining for training. We repeat this procedure five times so that all five sub-datasets can be used for validation. And we pick the parameter values having the best average performance. Then we evaluate different models on the 20% of the dataset left for testing (i.e., the test set).

Methods Compared.

[2] `http://www.cs.ubc.ca/~jamalim/datasets/`

[3] `https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban`

[4] `http://www.trustlet.org/wiki/Epinions_dataset`

Table 4.2: MAE and RMSE on all users (boldface font denotes the winner in that row)

		UserMean	ItemMean	PMF	SoRec	STE	SMF	SoReg	TrustMF	PTPMF
Flixster	MAE	0.840127	0.853447	0.801346	0.795724	0.770012	0.749708	0.758309	0.792434	**0.715910**
	Impv	14.7%	16.1%	10.7%	10.0%	7.03%	4.51%	5.59%	9.66%	–
	RMSE	1.061324	1.074465	1.012973	1.008995	0.974290	0.952560	0.960418	1.001670	**0.914541**
	Impv	13.8%	14.9%	9.72%	9.36%	6.13%	3.99%	4.78%	8.70%	–
CiaoDVD	MAE	0.904175	0.894703	0.876668	0.830892	0.834754	0.829287	0.865684	0.828039	**0.789901**
	Impv	12.6%	11.7%	9.90%	4.93%	5.37%	4.75%	8.75%	4.61%	–
	RMSE	1.133421	1.195009	1.106291	1.088455	1.088869	1.109867	1.124882	1.087434	**1.019105**
	Impv	10.1%	14.7%	7.88%	6.37%	6.41%	8.18%	9.40%	6.28%	–
Douban	MAE	0.685375	0.627068	0.569055	0.568788	0.554951	0.554731	0.554378	0.569364	**0.542439**
	Impv	20.9%	13.5%	4.68%	4.63%	2.25%	2.22%	2.15%	4.73%	–
	RMSE	0.852284	0.783605	0.720964	0.719435	0.716873	0.717495	0.700033	0.720403	**0.686182**
	Impv	19.5%	12.4%	4.82%	4.62%	4.28%	4.36%	1.98%	4.75%	–
Epinions	MAE	0.969965	0.988781	0.916315	0.900854	0.882172	0.870062	0.897915	0.864209	**0.822009**
	Impv	15.2%	16.9%	10.3%	8.75%	6.82%	5.52%	8.45%	4.88%	–
	RMSE	1.170197	1.189446	1.137936	1.127590	1.120252	1.119862	1.121258	1.107024	**1.060537**
	Impv	9.37%	10.8%	6.80%	5.95%	5.33%	5.30%	5.42%	4.20%	–

In order to show the performance improvement of our PTPMF method, we will compare our method with some state-of-art approaches which consist of non-personalized non-social methods, personalized non-social methods and personalized social methods. Thus, the following nine recommendation methods, including eight baselines, are tested.

- **PTPMF.** Our proposed PTPMF model, which is a personalized social recommendation approach by exploiting social ties.
- **TrustMF.** A personalized social method originally proposed by Yang et al. [117], which is capable of handling trust propagation among users.
- **SMF.** This is a personalized social approach [51] which assumes that users' latent feature vectors are dependent on those of their ties.
- **SoReg.**The individual-based regularization model with Pearson Correlation Coefficient (PCC) which outperforms its other variants, as indicated in [80]. This is a personalized social method.
- **STE.** Another personalized social method proposed by Ma et al. [77] which aggregates a user's own rating and her friends' ratings to predict the target user's final rating on an item.
- **SoRec.** The probabilistic matrix factorization model proposed by Ma et al. [79] which factorizes user-item rating matrix and user-user linkage matrix simultaneously. This is also a personalized social method.
- **PMF.** The classic personalized non-social probabilistic matrix factorization model first introduced in [82].
- **UserMean.** A non-personalized non-social baseline, which makes use of the average ratings of users to predict missing values.
- **ItemMean.** Another non-personalized non-social baseline, utilizing the average ratings of each items to make predictions.

All experiments are conducted on a platform with 2.3 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory. We use grid search and 5-fold cross validation to find the best parameters. For example, we set $\lambda_U = \lambda_V = 0.001$ after exploring each value in $(0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1)$ with cross validation and set $\lambda_B = \lambda_\theta = 0.00001$ in a similar way. The latent factor dimension is set to 10 for all models (if applicable). The learning rate of gradient descent (i.e., η) is set to 0.05 for θ_G and 0.001 for other parameters. For baselines, we adopt either the optimal parameters reported in the original paper or the best we can obtain in our experiments.

Evaluation Metrics.

We use four metrics, i.e., Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Recall and Precision, to measure the recommendation accuracy of our PTPMF model in comparison with other recommendation approaches.

- *Mean Absolute Error.*
$$\text{MAE} = \frac{\sum_{i,j} |R_{ij} - \hat{R}_{ij}|}{N}.$$

- *Root Mean Square Error.*
$$\text{RMSE} = \sqrt{\frac{\sum_{i,j} (R_{ij} - \hat{R}_{ij})^2}{N}}.$$

  where R_{ij} is the rating that user i gives to item j (original rating) and $\hat{R}_{ij}$ is the predicted rating of user i for item j. N is the number of ratings in test set.

- *Recall@K.*
  This metric quantifies the fraction of consumed items that are in the top-K ranking list sorted by their estimated rankings. For each user u we define $S(K; u)$ as the set of already-consumed items in the test set that appear in the top-K list and $S(u)$ as the set of all items consumed by this user in the test set. Then, we have
$$Recall@K(u) = \frac{|S(K; u)|}{|S(u)|}.$$

- *Precision@K.*
  This measures the fraction of the top-K items that are indeed consumed by the user in the test set:
$$Precision@K(u) = \frac{|S(K; u)|}{K}.$$

4.5.2 Experimental Results

Table 4.2 presents the performances of all nine recommendation methods on all four datasets, in terms of MAE and RMSE. We also present the percentage increase of PTPMF over each baseline right under its corresponding MAE and RMSE values and boldface font denotes the winner in each row. We would like to point out that, due to the randomness in data splitting and model initialization as well as differences in data preprocessing, our results for some baselines are slightly different from the results reported in the original papers. Among the eight baselines, UserMean and ItemMean are non-personalized methods which do not take social information into account; PMF is a personalized non-social model; the remainder are personalized approaches which also take social information into consideration. We observe from Table 4.2 that the personalized non-social method (PMF) outperforms the non-personalized non-social methods (UserMean and ItemMean), which shows the advantage of a personalized strategy. Moreover, through taking extra social network information into consideration, personalized social methods (SoRec, STE, SMF, SoReg and TrustMF) achieve a performance boost over the personalized non-social method (PMF), consistent with the assumption in the social recommendation literature that social information can help improve recommender systems. Finally, we observe that PTPMF consistently outperforms all eight baselines on all datasets for both metrics, demonstrating the benefit of the distinction and thresholding of different tie types, as well as learning a personalized tie preference for each user. Due to the randomness in data splitting, model initialization and even data preprocessing, our results for some baselines may not be exactly the same as reported in the original work, though given our best efforts to diminish the variances.

Recall and Precision.

Figure 4.2 depicts Recall (X-axis) vs. Precision (Y-axis) of the seven recommendation methods. We exclude the two naive methods (UserMean and ItemMean) for the sake of clarity of the figures. Data points from left to right on each line were calculated at different values of K, ranging from 5 to 50. Clearly, the closer the line is to the top right corner, the better the algorithm is, indicating that both recall and precision are high. We observe that PTPMF again clearly outperforms all baselines. Besides, Figure 4.2 also demonstrates the trade-off between recall and precision, i.e., as K increases, recall will go up while precision will go down.

Comparisons on Cold-Start Users.

We further drill down to the *cold-start* users. As is common practice, we define users that rated less than five items as cold-start. Figure 4.2 shows the performances of various methods on cold start users. It is well known that the social recommendation methods are superior to their non-social competitors particularly for cold-start users. The results in Figure 4.2 verify this – all social recommendation methods significantly outperform PMF in terms of both MAE and RMSE. Furthermore, our PTPMF model again beats other social recommendation baselines.

Learned threshold vs. Fixed threshold.

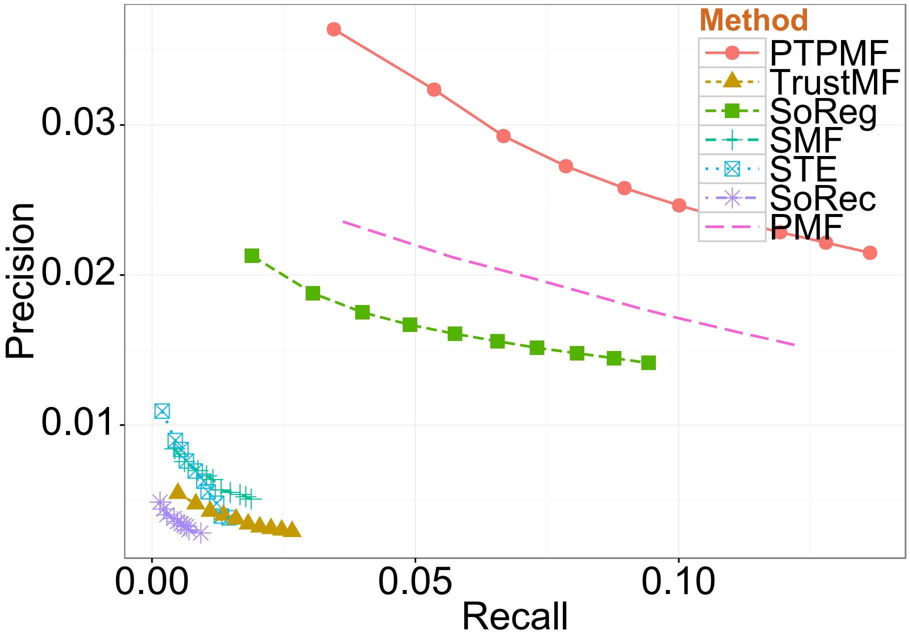
Last but not least, we compare the results from our learned thresholds with those from several pre-fixed thresholds in Figure 4.2 in order to prove that the threshold learning does contribute to the accuracy of the recommendations. For each dataset, we set θ_G to be four fixed values, i.e., 0.2, 0.4, 0.6, 0.8. We then compare the results obtained through fixing θ_G with that obtained from dynamically learning the threshold. Figure 4.2 demonstrates that the best results are achieved by the dynamically learned thresholds in terms of both MAE and RMSE. We remark that the thresholds learned from different datasets vary greatly, which is another supporting argument for learning the thresholds from the data.

In summary, we compare PTPMF with various kinds of baselines including non-personalized non-social methods, personalized non-social methods and personalized social methods in terms of both rating prediction and top-K ranking evaluation metrics. We conclude from the above extensive experiments that our proposed model, PTPMF, is an effective social recommendation method given its better performance over other baselines on both all users and cold-start users.

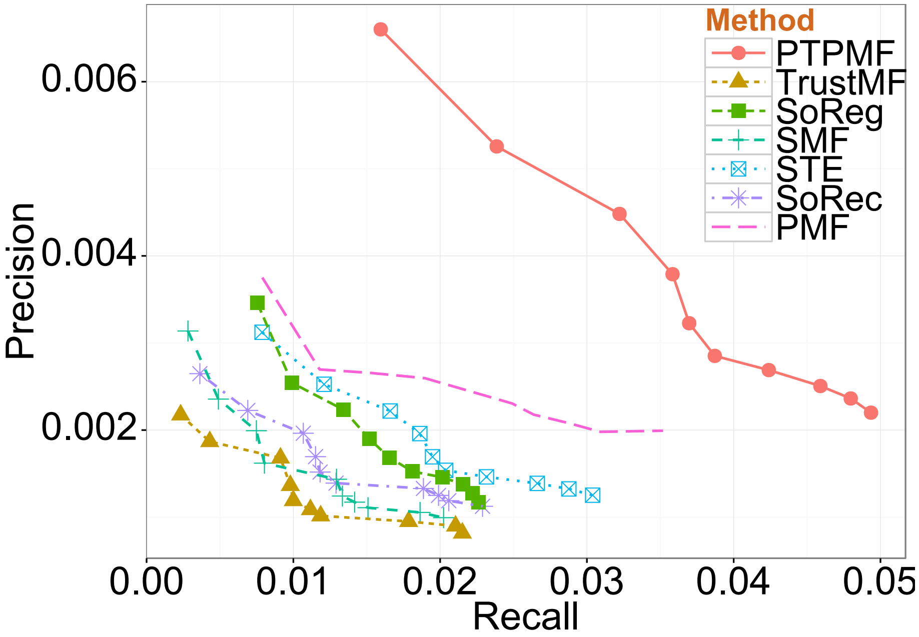
4.6 Summary

In this chapter, inspired by the seminal work in social science [43,44], we start from recognizing the important roles of different tie types in social relations and present a novel social recommendation model, a non-trivial extension to probabilistic matrix factorization, to incorporate the personalized preference of strong and weak ties into social recommendation. Our proposed method, PTPMF, is capable of simultaneously classifying strong and weak ties w.r.t. recommendation accuracy in a social network, and learning a personalized tie type preference for each individual as well as other model parameters.

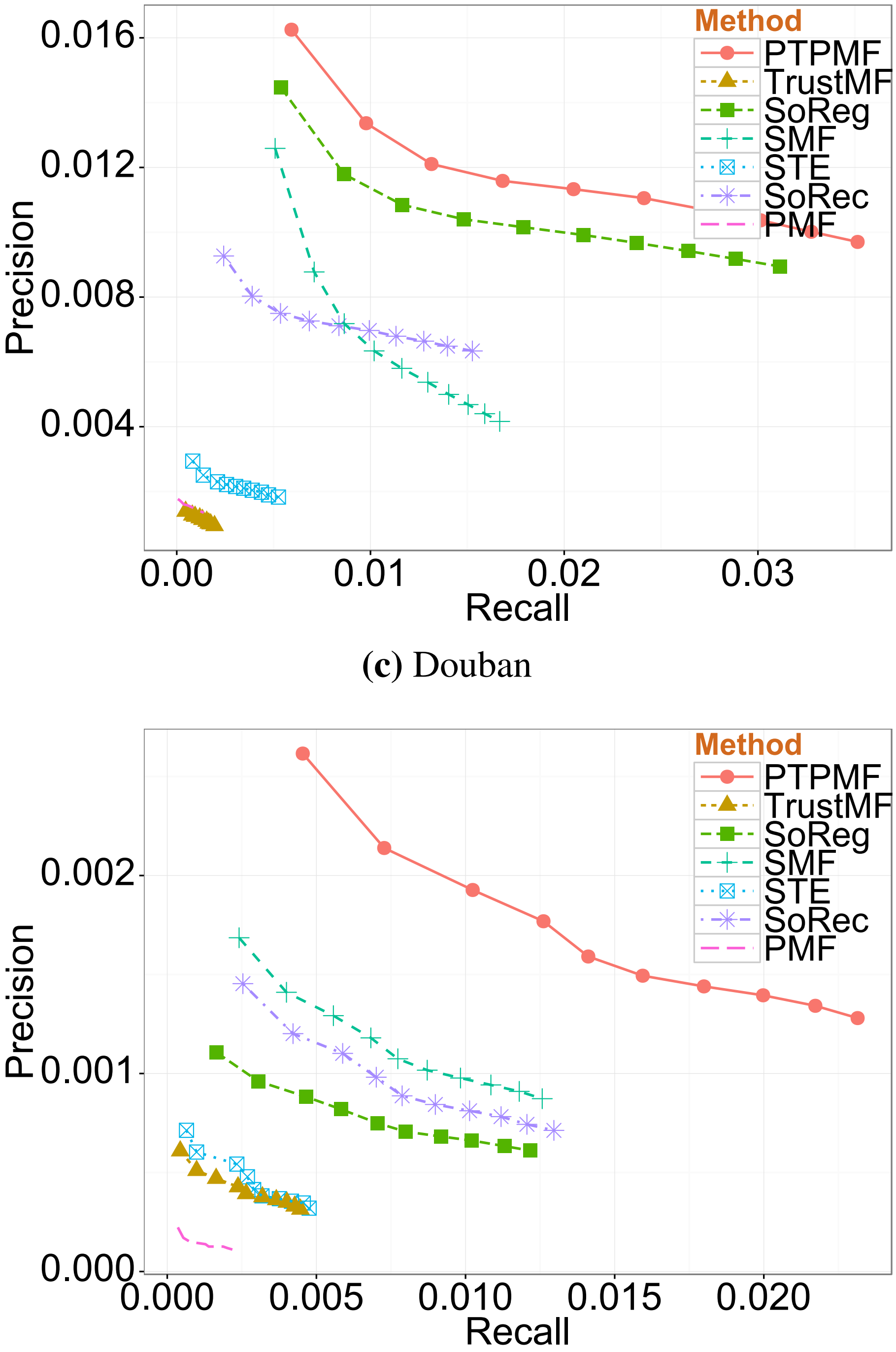
We carry out thorough experiments on four real-world datasets to demonstrate the gains of our proposed method. The experimental results show that PTPMF provides the best accuracy in various metrics, demonstrating that learning user-specific preferences for different types of ties in social recommendation does help to improve the performance.



(a) Flixster



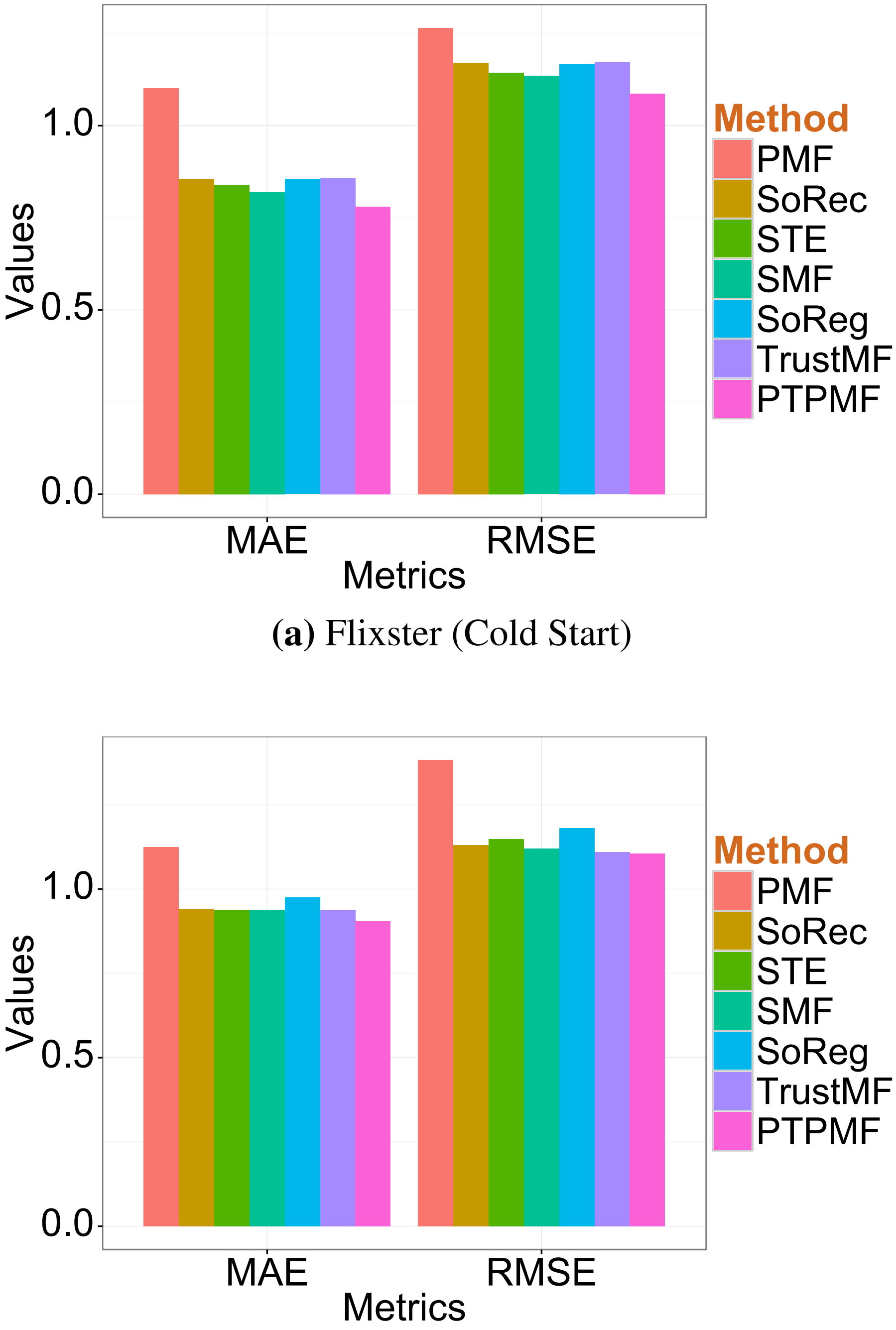
(b) CiaoDVD



(c) Douban

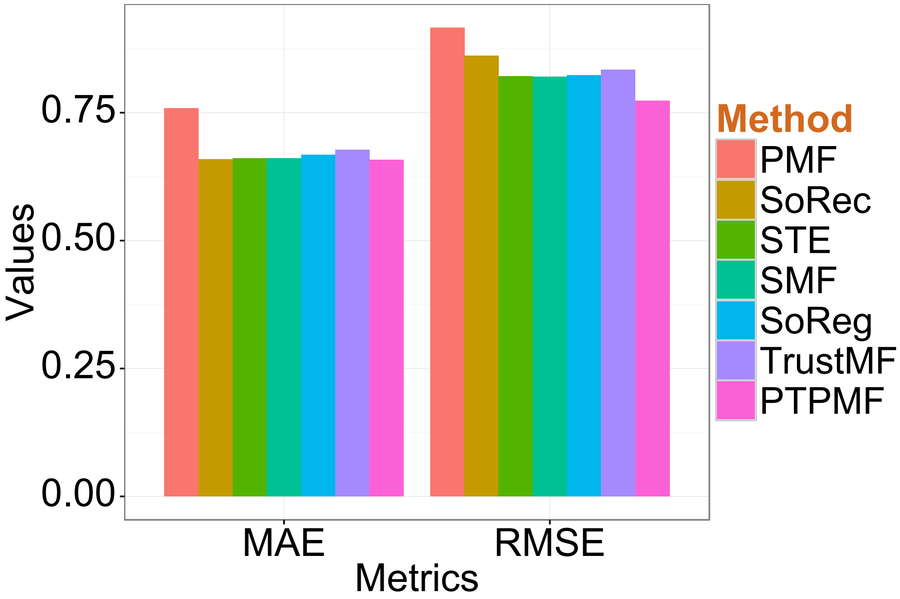
(d) Epinions

Figure 4.2: Precision@K vs Recall@K on all users, where K ranges from 5 to 50

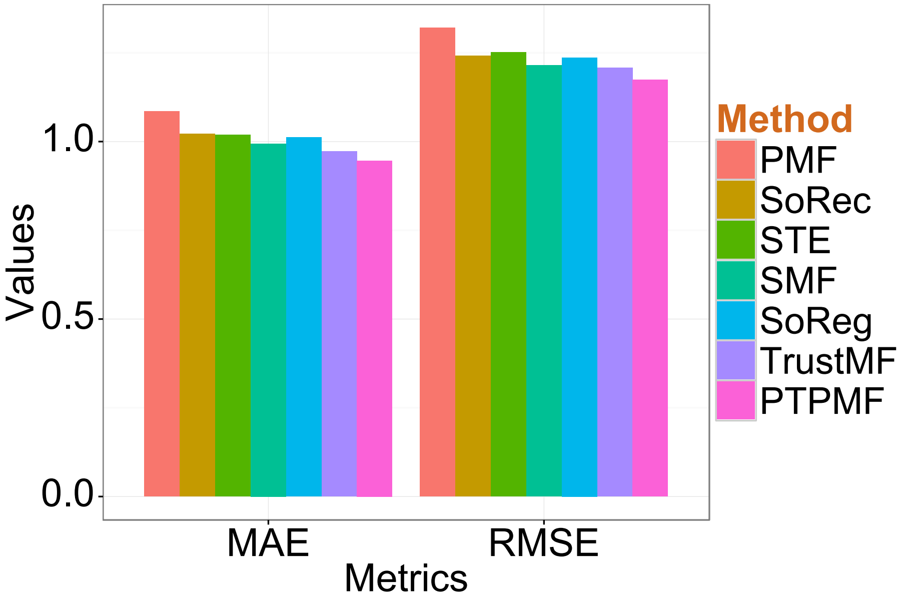


(a) Flixster (Cold Start)

(b) CiaoDVD (Cold Start)

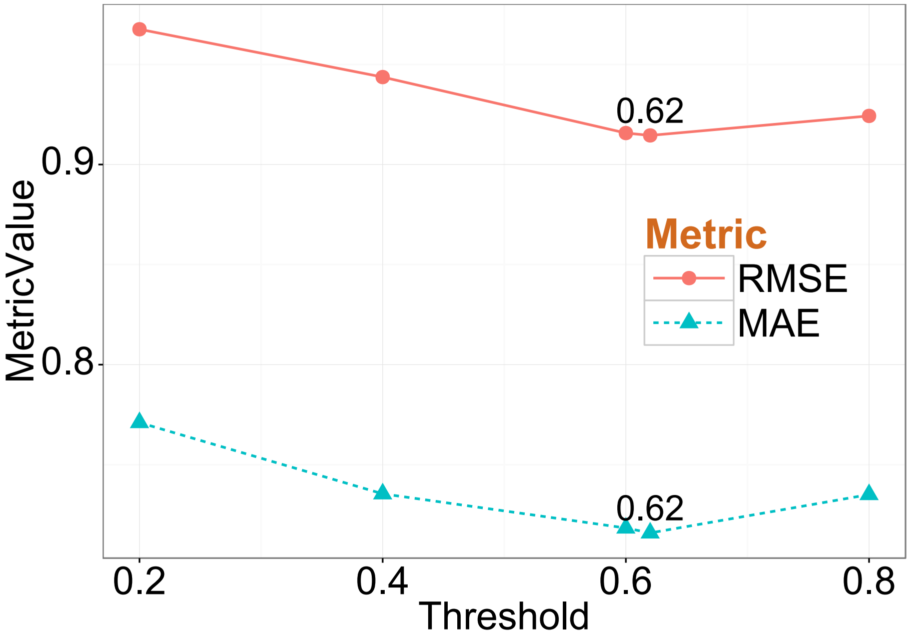


(c) Douban (Cold Start)

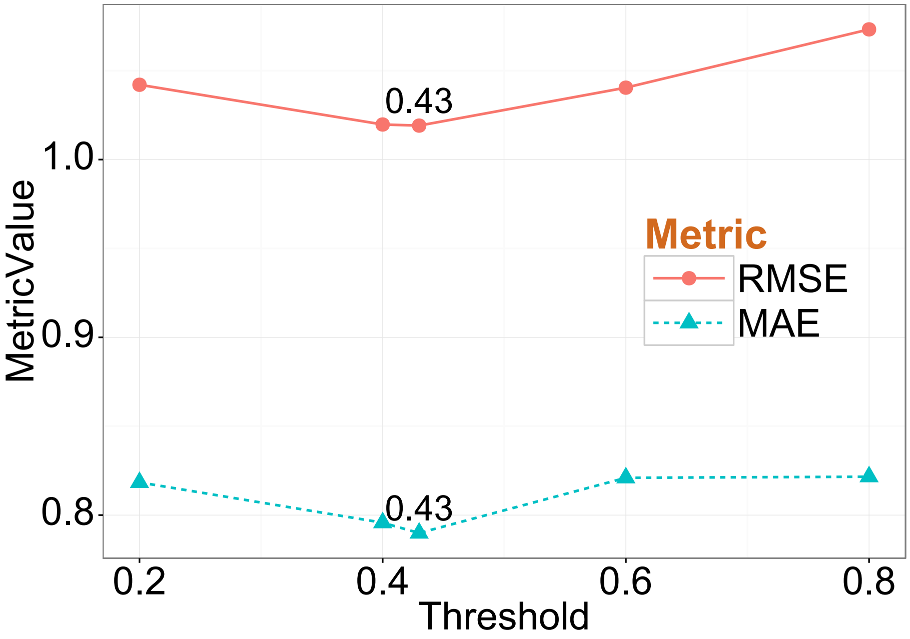


(d) Epinions (Cold Start)

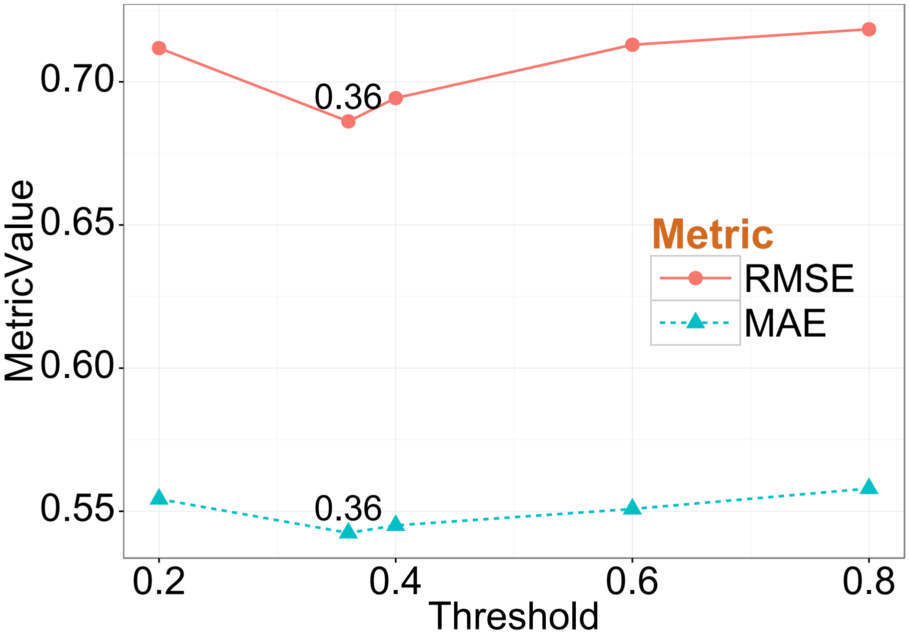
Figure 4.2: MAE and RMSE on cold-start users



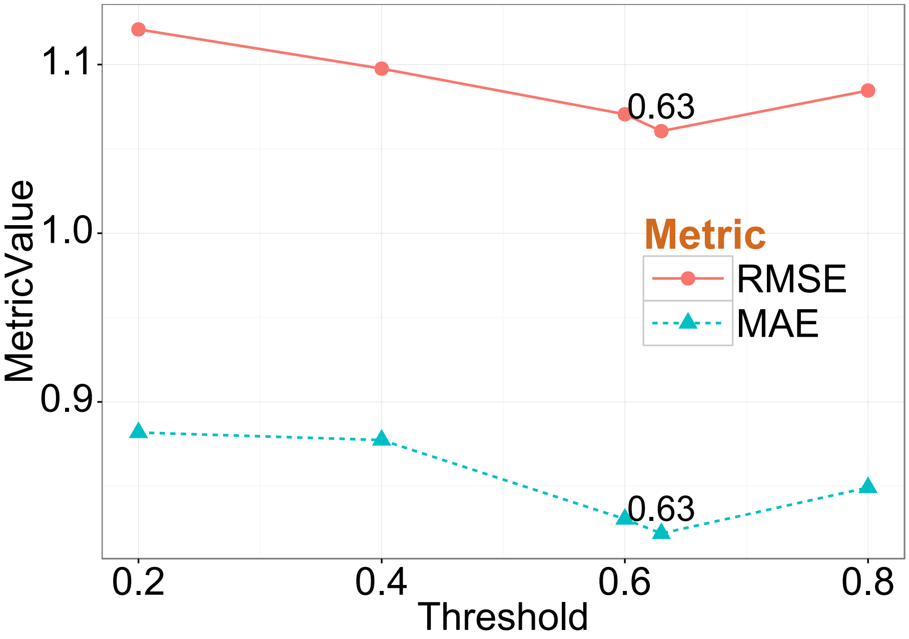
(a) Flixster



(b) CiaoDVD (Threshold)



(c) Douban



(d) Epinions

Figure 4.2: MAE and RMSE for several pre-fixed thresholds and our learned thresholds, with the numbers (and the corresponding points below them) denoting our learned threshold values

Chapter 5

Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization

Social networks often provide group features to help users with similar interests associate and consume content together. Recommending groups to users poses challenges due to their complex relationship: user-group affinity is typically measured implicitly and varies with time; similarly, group characteristics change as users join and leave. To tackle these challenges, we adapt existing matrix factorization techniques to learn user-group affinity based on two different implicit engagement metrics: (i) which group-provided content users consume; and (ii) which content users provide to groups. To capture the temporally extended nature of group engagement we implement a time-varying factorization. We test the assertion that latent preferences for groups and users are sparse in investigating elastic-net regularization. Our experiments indicate that the time-varying implicit engagement-based model provides the best top-K group recommendations, illustrating the benefit of the added model complexity.

5.1 Motivation

Online web services recommend content items such as music, movies, or books etc. to users via algorithmic recommendations. Many of these recommenders are based on the principle of collaborative filtering, suggesting items that similar users have consumed. On the other hand, more and more social media and consumer websites are providing mechanisms by which users can self-organize into groups with other users having similar opinions or interests.

The problem of recommending groups to users has been investigated in the literature. In particular, methods for factorizing the user-group membership matrix have been proposed, using group features, user-item ratings and user-user networks to improve the performance [21, 22, 104, 107, 124].

However, the existing methods fail to capture the dynamics of user group relationships: while the properties of content items typically do not change in the course of time, groups tend to evolve as users join or leave. Moreover, the preferences of users themselves tend to change over time. Thus, the existing methods are not completely adequate for recommending groups to users.

In this chapter we focus on the problem of recommending groups to users using implicit measures of user-group affinity, and model the time-varying nature of such measures.

To quantify user-group affinity, we define a user-group engagement matrix that is constructed from more data than simple Boolean user-group membership information. Rather than relying on explicit surveys of user ratings of groups which are not always available, a user's affinity for a group is measured implicitly through observing how often and in what manner the user engages with that group. Following [48], our confidence in a user's affinity for a group increases with the number of interactions between the user and the group.

Unlike many types of content where user-item engagement occurs within a relatively brief consumption period, user-group interaction occurs over extended time scales. For example, a user may engage with a particular group 10 times in one month, 20 times the next, but only 2 times in the third month, and not at all in the fourth. This kind of extended interaction offers an opportunity to explicitly model changes in both user preferences and group dynamics when producing recommendations. Our intuition is that doing so will lead to improved recommendations.

We capture the time-varying nature of the group recommendation problem in two ways. First, we propose a time-varying matrix factorization in order to capture preference changes. Second, we introduce two time-varying user biases and one time-varying group bias in order to capture changes in activity levels. We incorporate time series analysis to model the temporal evolution of these factors and biases.

Thus, this chapter makes three main contributions:

- When recommending groups to users, we consider not only group membership, but also different kinds of engagement between users and groups.
- We model evolution of the preferences and activity levels of both users and groups in order to better predict future preferences.
- We evaluate our methods using three real-world datasets from DeviantArt [30], a large social network for artists and art enthusiasts. Our experiments show that using implicit engagement measures instead of Boolean membership improves recommendation performance. Taking the temporal nature of the engagement into account produces further improvements. In addition, we adopt a sparse non-negative matrix factorization using elastic-net regression, rather than the standard unconstrained factorization using ridge regularizationand. We also see a moderate improvement from the use of non-negative factorization with elastic-net regularization.

The remainder of this chapter is organized as follows: We compare our proposed approach with some existing work in Section 5.2. Section 5.3 describes the user-group recommendation problem and defines our measures of user-group engagement. Section 5.5 presents a detailed formation of our activity level-biased model and optimization method, and our temporal model for predicting future user-group affinity. Section 5.6 presents our experiments, comparing our methods to baseline recommendations, and comments on the computational feasibility of the various approaches presented in production systems. Finally, we give a summary of our work in Section 5.7.

5.2 Comparison with Existing Literature

There is a large body of work on using matrix factorization for collaborative filtering. Our work in this chapter follows that of Hu et al. [48] (WRMF) which proposes a weighted factorization recommendation method robust enough for implicit user-item ratings. We apply Koren's method [58], which adds biases to both users and items in order to indicate their average ratings, though we employ user and group activity levels as biases in our method and do so in a time-varying manner, with different regularization, and with a non-negative factorization.

Hu's Model (WRMF).

Hu et al. [48] predict users' preferences for TV programs through an implicit scoring model whose factors are computed by the matrix factorization

$$\begin{aligned}\underset{\mathbf{X}_u,\mathbf{Y}_i}{\text{minimize}}\ \ & \frac{1}{2}\sum_{u,i}(1+\gamma r_{ui})(p_{ui}-\mathbf{X}_u^T\mathbf{Y}_i)^2 \\ & + \lambda\big(\sum_u\|\mathbf{X}_u\|_2^2+\sum_i\|\mathbf{Y}_i\|_2^2\big).\end{aligned} \tag{5.1}$$

Vectors $\mathbf{X}_u$, $\mathbf{Y}_i \in \mathbb{R}^k$ are latent factors for user u and item i. User u's preference for i is determined by binarizing rating $r_{ui} \geq 0$:

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0, \\ 0, & r_{ui} = 0. \end{cases} \tag{5.2}$$

In this model, the lowest rating assigned to an item a user has observed is 1, so $r_{ui} = 0$ and $p_{ui} = 0$ if u has never observed item i. This model, therefore, accounts for all user-item pairs. Parameters γ, which scales the strength of user-item ratings, λ, which regularizes matrix factors, and k, the dimension of the latent space, are chosen by experiment.

By extending Hu et al.'s model (also denoted as WRMF for short) [48], Zeng and Chen [124] incorporate both user-item ratings and user-user social relationships (called "heterogeneous resources" in their paper) into the user-group membership matrix. We will briefly introduce the proposed model in [124] which is so far the latest method utilizing social relationships among users to recommend groups.

Zeng's Model.

Zeng et al. extend WRMF to fit user-group membership, user-item consumption (rating) and user-user friendship, solving

$$
\begin{aligned}
\alpha \underset{\mathbf{X}_u,\mathbf{Z}_g}{\text{minimize}} \quad & \sum_{u,g} c_{ug}^m \left(p_{ug}^m - \mathbf{X}_u^T \mathbf{Z}_g\right)^2 + \lambda\Big(\sum_u \|\mathbf{X}_u\|_2^2 + \sum_g \|\mathbf{Z}_g\|_2^2\Big) \\
& + \lambda_f \Big(\|\mathbf{X}_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) \mathbf{X}_f\|_2^2\Big) \\
+(1-\alpha) \underset{\mathbf{X}_u,\mathbf{Y}_i}{\text{minimize}} \quad & \sum_{u,i} c_{ui}^r \left(p_{ui}^r - \mathbf{X}_u^T \mathbf{Y}_i\right) + \lambda\Big(\sum_u \|\mathbf{X}_u\|_2^2 + \sum_i \|\mathbf{Y}_i\|_2^2\Big), \qquad (5.3)
\end{aligned}
$$

where c_{ug}^m and c_{ui}^r are confidence levels for user-group memberships and user-item ratings similar to $(1 + \gamma r_{ui})$ in WRMF. p_{ug}^m and p_{ui}^r indicate whether user u is a member of group g and if user u has consumed item i, both of which are boolean values. $F(u)$ denotes the set of friends of user u and $\mathbf{X}_u, \mathbf{Z}_g, \mathbf{Y}_i$ are latent feature vectors for user u, group g, item i respectively. λ and λ_f are used as the regularization coefficients.

The whole objective function in Eqn 5.3 tries to minimize two parts separately. The first part approximates the user-group membership matrix through the same matrix factorization as WRMF and then restricts the user u's latent feature vector ($\mathbf{X}_u$) to be as close as the sum (weighted by the similarities between u and her friends $f \in F(U)$ and normalized by number of her friends $|F(u)|$) of her friends' latent feature vectors $\mathbf{X}_f$. The second part which approximates the user-item consumption matrix and shares the same user latent feature vector ($\mathbf{X}$) with the first part is exactly in the same form as WRMF, with c_{ui}^r replacing $(1 + \gamma r_{ui})$ in Eqn 5.1. The relative importance of these two parts is controlled by $\alpha \in [0, 1]$, larger α indicates that the model puts more emphasis on the user-group memberships and user social relationships, smaller α means more emphasis is placed on user-item consumption relationships.

The general idea of Zeng's model is to consider user-group memberships (and visiting frequencies controlled by c_{ug}^m if available), user-item consumption information and user-user social relationships simultaneously when recommending groups to users. We skip other details such as model inferences and readers may check the original paper for more references.

We make a modification to the standard factorization technique in using the non-negative sparse matrix factorization provided by elastic net regularization, rather than the usual ridge regression. Elastic net regularization was first introduced by Zou and Hastie in [133] as a new regularization and an algorithm called LARS-EN was also proposed to solve the elastic net regularization problem. Then Friedman et al. [35] explore the "one-at-a-time" coordinate-wise descent algorithms which can solve convex problems such as elastic net more efficiently. Later Friedman et al. also develop a cyclical coordinate descent algorithm which runs very fast for estimation of generalized linear models with convex penalties such as L_1 norm (the LASSO), L_2 norm (ridge regression) and mixtures of the two (the elastic net) [36]. Recently, Yang et al. propose a cocktail algorithm which contains a good mixture of coordinate decent for solving the elastic net problem in high dimensions [119]. To our knowledge, the theory [133] and coordinate-wise descent algorithm [35] for this approach have not appeared previously in the collaborative filtering literature, although the theory of non-negative sparse matrix factorizations is well developed elsewhere (see, e.g., [47]).

5.3 User-Group Engagement

To produce effective group recommendations, we believe that it is important to make measurements of user-group engagement more nuanced than Boolean user-group membership. In this section, we introduce the user-group data from DeviantArt [30] whose group features will be used in our proposed model and experiments, as well as our proposed user-group engagement measures. Although we take datasets from DeviantArt as an example, our proposed measurements of user-group engagement is general to all other social media capable of providing interactions between groups and their members.

Groups on DeviantArt

DeviantArt [30] is the world's largest online arts community, with more than 60 million unique monthly visitors and 30 million registered users. Approximately 10,000 new users register daily. DeviantArt users submit over 150,000 new artworks every day, and the site has received over 300 million submissions in total. Art appreciators can engage with art by *favoriting* artwork (2 million events/day), or by *commenting* on artwork (also 2 million events/day).

Groups on DeviantArt are self-organized associations of users who have the ability collectively curate art. This collective curation provides value to artists, who benefit from the endorsement provided by well-known groups, and to individual art collectors, who can use these curated collections to discover new art. As such, artists often join groups with the intent of submitting their artwork for acceptance by the group, and collectors often join groups with the intent of discovering art that they like from the group's accepted submissions.

Measuring user-group engagement

We consider two variants of the user-group recommendation problem: recommending groups to artists, and recommending groups to collectors. Since DeviantArt users do not explicitly identify themselves as either artists or collectors, and since DeviantArt provides no mechanism for explicit rating of groups, we instead measure artist-group and collector-group affinity implicitly, via user-group engagement. On DeviantArt, artists provide new artwork to their groups, and collectors consume artwork provided by their groups. Therefore, we define two types of user-group engagement:

- **production engagement:** the number of art submissions by user u that were accepted by group g
- **consumption engagement:** the number of art submissions accepted by group g that user u subsequently favorited

We might use production engagement when recommending groups to artists, and consumption engagement when recommending groups to collectors.

Although the details of the definitions of production and consumption engagement above are specific to DeviantArt, we argue that the high level concepts of these definitions transfer to other scenarios of user groups. For example, Douban [31], a Chinese online social network, allows its users to create content related to films, books, music, and recent events and activities in their self-organized groups (called Douban Group). Users engage with their groups through creating new content such as articles or polls (production engagement) and through reading articles or participating in a poll (consumption engagement). As another example, scientific conferences can be considered as self-organized groups of researchers. Researchers engage with conferences through submitting papers (production engagement) and through reading and reviewing papers (consumption engagement). Generally, the production engagement measures which content users provide to groups and the consumption engagement measures which group-provided content users consume.

Compared to Boolean membership or explicit item ratings data, the user-group engagement exhibits three interesting properties:

1. User-group engagement is nonnegative but otherwise unbounded; explicit ratings are usually restricted to a closed interval (e.g., integers from 1 to 5).

2. Users engage with groups gradually over extended time periods; user-item ratings are typically collected at a single time point.

3. Group characteristics change over time (for example, as users join and leave and activity levels increase or decrease); item characteristics typically do not.

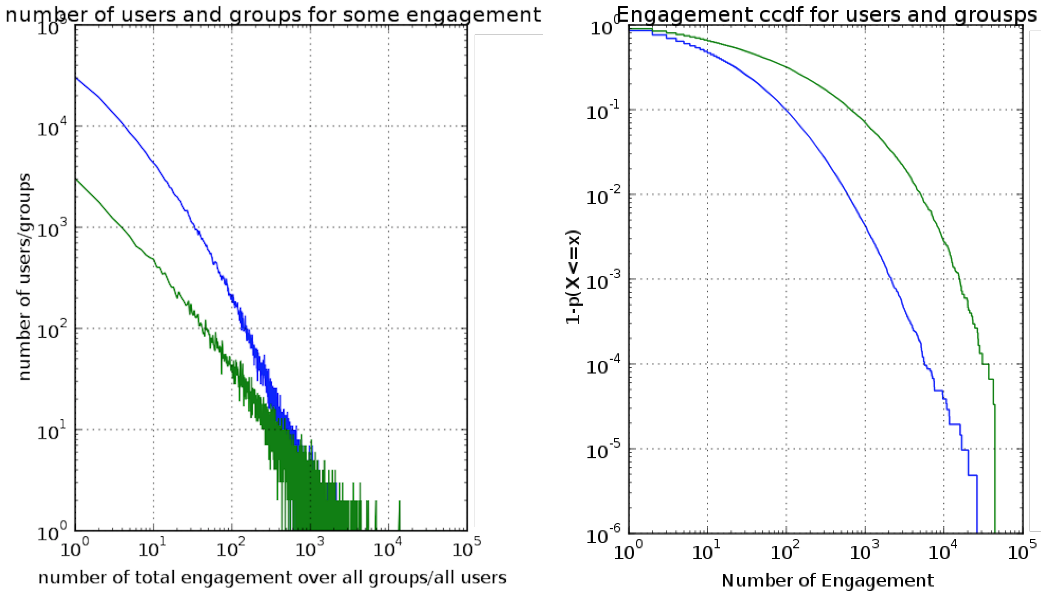
The proposed recommendation method is designed to exploit each of these three properties.

5.4 Empirical Data Analysis

In this section, we give empirical statistics of the user-group production engagement data from DeviantArt, together with the temporal features of a random user's production engagement with one of the groups he/she has been in and this user's total production engagement with all the groups that he/she belongs to. Although being slightly different from production engagement, the consumption engagement follows a similar pattern in terms of the statistical properties and temporal features. Both measures of production and consumption engagement follow the definition in Section 5.3.

Our data consists of DeviantArt users and groups from 5 May 2011 to 31 August 2014, i.e., a 40-month time span. In general, Figure 5.1a and Figure 5.1b show the power law distribution of the user-group production engagement, which is a common property shared by most of the real-world data.

Given the empirical statistics that the majority of active DeviantArt users are young people under 25, we assume that the school days should probably have a significant impact on their activities



(a) Number of users and groups for a given amount of total production engagement

(b) The complementary cumulative distribution function (CCDF) with respect to production engagement for users and groups

Figure 5.1: Some statistics about the user-group production engagement data from DeviantArt, blue (dark) line represents users and green (light) line represents groups

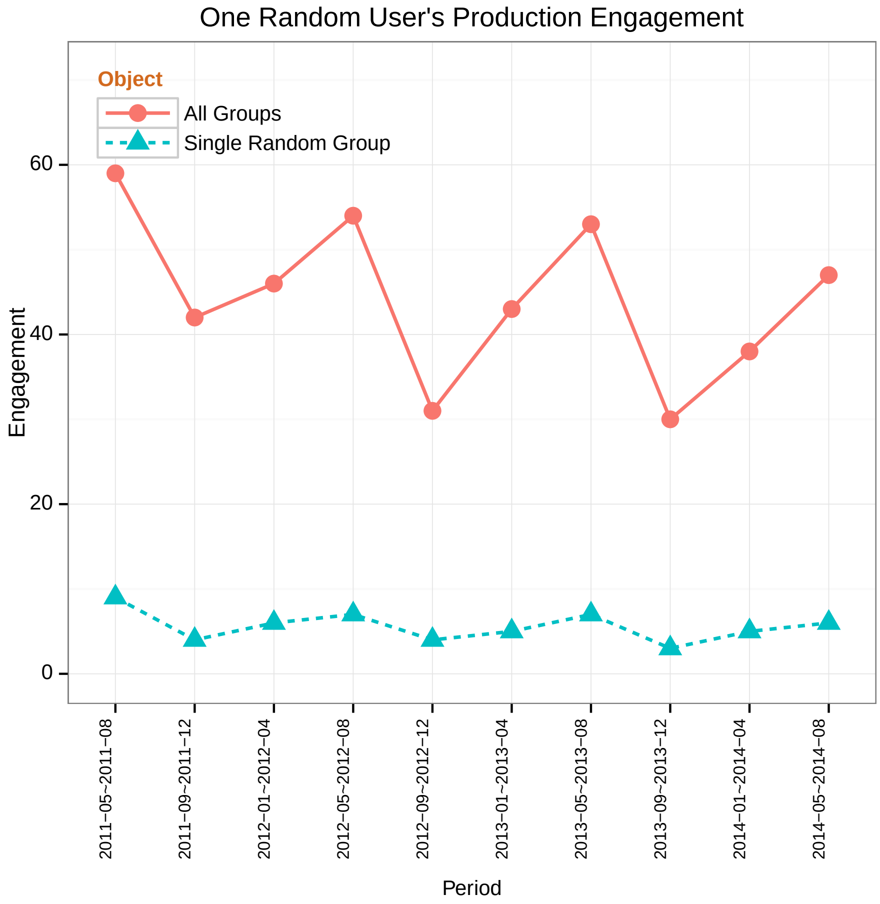


Figure 5.2: A user’s amount of production engagement over time with one single group and all groups that he/she belongs

on DeviantArt. Therefore, we then divide this 40-month time span into 10 equal 4-month intervals beginning with summer 2011 (from May 2011 to August 2011) and ending in summer 2014 (from May 2014 to August 2014), each of which is roughly one school/college semester. Figure 5.2 demonstrates the patterns of a random user's production engagement with one randomly selected group he/she has been in and this user's total production engagement with all the groups that he/she belongs to in each 4-month interval from May 2011 to August 2014. We observe that the user always has a smaller amount of engagement in fall than in spring and summer in both cases (all groups and single random group), and there is an increase in the amount of engagement from spring to summer. Further, both curves in Figure 5.2 verify the existence of consecutive correlations between the amount of engagement in one interval and that in its previous intervals. For instance, the user's production engagement in summer 2013 (2013-05$\sim$2013-08) depends mostly on summer 2012 (2012-05$\sim$2012-08), which is three intervals before, and at the same time it is less than this user's engagement in summer 2012, which may be due to the reason that the user's engagement in spring 2013 (2013-01$\sim$2013-04) and fall 2012 (2012-09$\sim$2012-12) is less than that in spring 2012 (2012-01$\sim$2012-04) and fall 2011 (2011-09$\sim$2011-12) respectively. It is these consecutive correlations between the users' engagement with groups in different time intervals that motivate us to develop a novel approach capable of capturing the long time evolution of user-group engagement for group recommendation.

As such, our analysis verifies the time-varying properties of user-group engagement mentioned in Section 5.1, demonstrating the temporal patterns and challenges not covered in prior work.

5.5 Recommending Groups to Users

To the best of our knowledge, all the existing online social websites offering group features provide no formal mechanism for users to rate groups, which motivates our use of the matrix factorization based on an implicit feedback scheme as our starting point. Our implicit scheme incorporates the strength of user-group interactions, which we measure as production and consumption engagement. As group members' tastes change, user preferences and group properties tend to subsequently change over time. Hence, after introducing a static model, we propose a temporal model intended to capture this time-dependence.

5.5.1 Static model

Hu et al. [48] predict users' preferences for TV programs through an implicit scoring model whose factors are computed by the matrix factorization

$$\begin{aligned}\underset{\mathbf{X}_u,\mathbf{Y}_i}{\text{minimize}}\ & \frac{1}{2}\sum_{u,i}(1+\gamma r_{ui})(p_{ui}-\mathbf{X}_u^T\mathbf{Y}_i)^2\\ & +\lambda\Big(\sum_u\|\mathbf{X}_u\|_2^2+\sum_i\|\mathbf{Y}_i\|_2^2\Big).\end{aligned} \tag{5.4}$$

Vectors $\mathbf{X}_u, \mathbf{Y}_i \in \mathbb{R}^k$ are latent factors for user u and item i. User u's preference for i is determined by binarizing rating $r_{ui} \geq 0$:

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0, \\ 0, & r_{ui} = 0. \end{cases} \tag{5.5}$$

In this model, the lowest rating assigned to an item a user has observed is 1, so $r_{ui} = 0$ and $p_{ui} = 0$ if u has never observed item i. This model, therefore, accounts for all user-item pairs. Parameters γ, which scales the strength of user-item ratings, λ, which regularizes matrix factors, and k, the dimension of the latent space, are chosen by experiment.

We extend this factorization to fit user-group engagement, solving

$$\begin{aligned} \underset{\mathbf{w},\mathbf{X}_u,\mathbf{Y}_g}{\text{minimize}} \quad & \frac{1}{2}\sum_{u,g}(1+\gamma r_{ug})\left(p_{ug} - \mathbf{q}_{ug}^T\mathbf{w} - \mathbf{X}_u^T\mathbf{Y}_g\right)^2 \\ & + \alpha_u \sum_u \|\mathbf{X}_u\|_1 + \frac{1}{2}\beta_u \sum_u \|\mathbf{X}_u\|_2^2 \\ & + \alpha_g \sum_g \|\mathbf{Y}_g\|_1 + \frac{1}{2}\beta_g \sum_g \|\mathbf{Y}_g\|_2^2 \\ & + \alpha_w \|\mathbf{w}\|_1 + \frac{1}{2}\beta_w \|\mathbf{w}\|_2^2 \\ \text{subject to} \quad & \mathbf{X}_u, \mathbf{Y}_g \geq 0. \end{aligned} \tag{5.6}$$

Consistent with (5.4), $\mathbf{X}_u, \mathbf{Y}_g \in \mathbb{R}^k$ are user and group latent factors. We use either producing engagement or consuming engagement defined in Section 5.3 as the implicit user-group rating r_{ug}, binarizing r_{ug} to user-group affinity p_{ug}. Parameters γ and k are rating sensitivity and latent factor dimension, and $\alpha_{\{u,g,w\}}, \beta_{\{u,g,w\}}$ are regularization parameters, all chosen by experimentation.

Variables $\mathbf{w} \in \mathbb{R}^3$ provide an optional bias for $\mathbf{q}_{ug} = [\bar{f}_u, \bar{c}_u, \bar{m}_g]^T$, measuring

- user properties:
  f_u: number of favorites user u has made
  c_u: number of comments user u has made

- group property:
  m_g: number of members in group g

These properties reflect overall levels of user and group activity. Note that we use normalized quantities

$$\bar{f}_u = \frac{f_u}{\sum_u f_u}, \quad \bar{c}_u = \frac{c_u}{\sum_u c_u}, \quad \bar{m}_g = \frac{m_g}{\sum_g m_g}, \tag{5.7}$$

to ensure that $\mathbf{q}_{ug}$ will be bounded in $(0, 1)$, reflecting the relative levels of user/group activity, and that elements of $\mathbf{w}$ are on the same scale.

Our use of biases is inspired by [58] and [4], where biases represent unpersonalized item ratings and users' personal item baselines in explicit rating systems. In contrast, our biases reflect the activity levels of users and groups, based on the assumption that users and groups with higher activity levels tend to have more engagement.

We refer to the special case where we fix $\mathbf{w} = 0$ as the unbiased model. Note that when we solve the unbiased model, we apply the optimizations reported in [48], as well as the use of a tall and skinny QR-factorization [26], to compute the matrix products of the form $Y^T Y$ which appear in the optimization procedure.

As our affinities $p_{ug} \geq 0$, we are motivated by previous work on non-negative matrix factorization (NMF) to compute $\mathbf{X}_u, \mathbf{Y}_g$ that are non-negative and sparse. As noted in the NMF literature – see [67], for example – avoiding cancellation of factors of different signs, particularly if those factors are sparse, tends to produce a factorization that is more easily interpreted. Regarding each dimension of the user and group vectors as a topic, the suggestion that $\mathbf{X}_u, \mathbf{Y}_g$ should be non-zero in only a few coordinates reflects the notion that each user and group description is dominated by a few topical preferences. Empirical experiments on real-world datasets in Section 5.6 show a better performance for non-negative constrained model over the one without non-negative constrained (i.e., Hu's model in (5.4)).

As noted elsewhere [47], the ratio between the l_1 and l_2 norms provides a measurement for vector sparsity. Rather than specify the sparsity explicitly, we manage sparsity by regularizing against both the l_1 and l_2 norms, employing elastic-net regularization [133]. This strategy reduces the model's sensitivity to the dimension of the latent factor dimension.

Having solved (5.6), we compute

$$\hat{p}_{ug} = [\bar{f}_u, \bar{c}_u, \bar{m}_g]^T \mathbf{w} + \mathbf{X}_u^T \mathbf{Y}_g, \tag{5.8}$$

as our prediction of user-group affinity. The largest predictions over user-group pairs for which $r_{ug} = 0$ are our recommendations.

Optimization procedure

Problem (5.6) is convex in each of $\mathbf{X}_u, \mathbf{Y}_g$ and $\mathbf{w}$ separately, and so we use pathwise coordinate descent [35] to optimize for each of these collections of variables iteratively. Recognizing that (5.6) is not jointly convex in all variables, we apply relaxation at each iteration, combining updated values with previous ones. This common strategy can increase the number of iterations, but prevents the algorithm from stalling at local minima.

Our algorithm follows a standard pattern for coordinatewise optimization:

1. Initialize $\mathbf{Y}_g$, $\mathbf{w}$.

2. With $\mathbf{Y}_g$ and $\mathbf{w}$ fixed, solve (5.6) for $\mathbf{X}_u$.

3. With $\mathbf{X}_u$ and $\mathbf{w}$ fixed, solve (5.6) for $\mathbf{Y}_g$.

4. With $\mathbf{X}_u$ and $\mathbf{Y}_g$ fixed, solve (5.6) for $\mathbf{w}$.

5. If not converged, go to step 2; otherwise, stop.

While it appears that the order of optimizations, namely $\mathbf{X}_u$, then $\mathbf{Y}_g$, and finally for $\mathbf{w}$, is theoretically arbitrary, our observations suggest that we achieve faster convergence by optimizing with respect to the bias parameters $\mathbf{w}$ last.

Initialization

Set $\mathbf{Y}_g = (1/k)[1, 1, \ldots, 1]^T$ and $\mathbf{w} = [0.1, 0.1, 0.1]^T$.

Minimization with respect to $\mathbf{X}_u$

Let $E(u)$ be the set of groups for which $r_{ug} > 0$. With $\mathbf{Y}_g, \mathbf{w}$ fixed, for each u, minimization (5.6) becomes

$$\begin{aligned} \underset{\mathbf{X}_u}{\text{minimize}} \;\; & \frac{1}{2}\mathbf{X}_u^T\Big(\sum_g \mathbf{Y}_g\mathbf{Y}_g^T + \sum_{g\in E(u)} \gamma r_{ug}\mathbf{Y}_g\mathbf{Y}_g^T\Big)\mathbf{X}_u \\ & - \Big(\sum_g (1+\gamma r_{ug})(p_{ug} - \mathbf{q}_{ug}^T\mathbf{w})\mathbf{Y}_g\Big)^T\mathbf{X}_u \\ & + \alpha_u\|\mathbf{X}_u\|_1 + \frac{1}{2}\beta_u\|\mathbf{X}_u\|_2^2 \\ \text{subject to} \;\; & \mathbf{X}_u \geq 0. \end{aligned} \tag{5.9}$$

The $\mathbf{X}_u$ can be computed in parallel, observing that $(p_{ug} - \mathbf{q}_{ug}^T\mathbf{w})$ and $\sum_g \mathbf{Y}_g\mathbf{Y}_g^T$ can be precomputed and re-used by each $\mathbf{X}_u$ calculation. Here is where a tall and skinny QR-factorization speeds up the $\sum_g \mathbf{Y}_g\mathbf{Y}_g^T$ calculation significantly.

The calculations for the $\mathbf{X}_u$ are sign-constrained elastic-net problems of the form

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \;\; & \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x} + \alpha\|\mathbf{x}\|_1 + \frac{1}{2}\beta\|\mathbf{x}\|_2^2 \\ \text{subject to} \;\; & \mathbf{x} \geq 0, \end{aligned} \tag{5.10}$$

which can be solved by pathwise coordinate descent [35].

Writing $\mathbf{X}_u^{(k)}$ as user latent factors at iteration k, and $\tilde{\mathbf{X}}_u^{(k+1)}$ latent factors computed by (5.9), we relax the update, setting our new latent factors as

$$\mathbf{X}_u^{(k+1)} = \mathbf{X}_u^{(k)} + \theta\left(\tilde{\mathbf{X}}_u^{(k+1)} - \mathbf{X}_u^{(k)}\right). \tag{5.11}$$

Where θ is the learning rate which controls the step-size in the parameter space. Choosing parameter $\theta = 0.9$ appears to be a good compromise between fast convergence and local minima avoidance in relaxing updates for all of $\mathbf{X}_u, \mathbf{Y}_g$ and $\mathbf{w}$.

Minimization with respect to $\mathbf{Y}_g$

With the symmetry between $\mathbf{X}_u$ and $\mathbf{Y}_g$, minimization with respect to $\mathbf{Y}_g$ is the same as that of $\mathbf{X}_u$.

Minimization with respect to $\mathbf{w}$

With $\mathbf{X}_u, \mathbf{Y}_g$ fixed, optimization with respect to $\mathbf{w}$ becomes

$$\begin{aligned}
\underset{\mathbf{w}}{\text{minimize}} \ \ & \frac{1}{2}\mathbf{w}^T\Big(\sum_{ug}(1+\gamma r_{ug})\mathbf{q}_{ug}\mathbf{q}_{ug}^T\Big)\mathbf{w} \\
& - \Big(\sum_{ug}(1+\gamma r_{ug})(p_{ug} - \mathbf{X}_u^T\mathbf{Y}_g)\mathbf{q}_{ug}\Big)^T\mathbf{w} \\
& + \alpha_w\|\mathbf{w}\|_1 + \frac{1}{2}\beta_w\|\mathbf{w}\|_2^2 \\
\text{subject to} \ \ & \mathbf{w} \geq 0. \qquad (5.12)
\end{aligned}$$

This is also an elastic-net problem. Although the use of the bias term improves predicted user-group affinities, this expression shows that it comes at the cost of summing over all users and groups.

5.5.2 Temporal model

Time series analysis [18, 46] is widely used in many fields such as econometrics/finance, meteorology and bioinformatics etc. where time-dependent data analysis is very popular. Recent research tasks in data mining such as epidemic tendency prediction [116] and daily-aware personalized recommendation [127] have also utilized time series analysis to solve time-dependent problems. This inspires us to adopt auto-regression (AR) and vector auto-regression (VAR) from time series analysis [18, 46] to capture the time-varying nature of user preferences, group properties, and global biases. We collect group interaction data in discrete time intervals, forming time-dependent user-group engagement matrices. To estimate user-group affinity at time T, we solve problem (5.6) at times $T-1, T-2, \ldots, T-p$. We model the trajectories of user, group, and bias vectors using auto-regression (AR) and vector auto-regression (VAR), extrapolating parameters and solutions to time T. Prediction (5.8) combines extrapolations to compute $\hat{p}_{ug}(T)$ and hence provide recommendations.

In our application of AR and VAR to our temporal model, we extend the user and group properties and factors to a time-varying equivalents:

- time-varying user properties:
  $f_u(t)$ is the number of favourites user u made in time interval t
  $c_u(t)$ is the number of comments user u made in time interval t

- time-varying group property:
  $m_g(t)$ is the number of new members in group g in time interval t

- time-varying factors:
  $\mathbf{X}_u(t)$ is user u's latent factors in time interval t
  $\mathbf{Y}_g(t)$ is group g's latent factors in time interval t

Our assumption is that both user preferences and group characteristics change gradually as time goes by, and there are trends for the changes in user-group engagement that we can use to predict future user-group affinity.

We divide the whole time span of data into T discrete time intervals (such as weeks, months or quarters). Users and groups will have their own qualities and latent factors in different time intervals. Note that the static model treats the whole time span as one single time interval, and can be regarded as a special case of the temporal model.

In particular, we adopt p-order auto-regression (AR(p)) to extrapolate the future qualities and p-order vector auto-regression (VAR(p)) to extrapolate the future latent factors. For time-varying functions $x(t)$, auto-regression assumes

$$x(T) = \sum_{k=1}^{p} \phi(k) x(T-k) + \epsilon(T), \tag{5.13}$$

where $\phi(k), k = 1 \dots p$ are parameters we fit, and ϵ is the error in our time T estimate. Where $x(t)$ are scalar-valued, that is, where we fit $f_u(t), c_u(t)$ and $m_g(t)$ by AR(p), the $\phi(k)$ are scalar-valued; where $x(t)$ are vector-valued, that is, where we fit $\mathbf{X}_u, \mathbf{Y}_g$, and $\mathbf{w}$ by VAR(p), the $\phi(k)$ are matrix-valued.

We solve for parameters $\phi(k)$ by a least-squares minimization,

$$\underset{\phi(k)}{\text{minimize}} \sum_{t=p+1}^{T} \left[x(t) - \sum_{s=1}^{p} \phi(s) x(t-s) \right]^2. \tag{5.14}$$

This cost function represents a window of length p that passes forward over the time-varying data. We choose parameters $\phi(k)$ to best fit data at each time step $t \in p \dots T-1$ based on the previous p time steps. Where (5.14) represents VAR(p), the squared term in square brackets is understood to represent the $l2$ vector norm.

Using data from $T-1$ trial intervals, we predict the user-group affinity at time T as

$$\hat{p}_{ug}(T) = [f_u(T), c_u(T), m_g(T)]^T \mathbf{w}(T) + \mathbf{X}_u(T)^T \mathbf{Y}_g(T). \tag{5.15}$$

In assuming that our parameter trajectories are smooth, such that user, group and global properties do not change abruptly, our future predictions directly leverage a long history of behavior. This is in contrast to the Markovian assumption employed elsewhere [125].

Table 5.1: Summary of production and consumption engagement matrices

	production	consumption
number of users	8423	20328
number of groups	4579	8772
number of non-zeros	161767	670662
matrix density	0.0042	0.0038

5.6 Experiments

Our experiments consider DeviantArt users and groups from 5 May 2011 to 31 August 2014. We divide this 40-month time span into 10 equal 4-month intervals. The first 9 intervals serve as training/validation data; we withhold the last 4 months for testing. The static model aggregates the first 9 intervals into a single training-validation set, on which a 10-fold cross validation is used to select the optimal parameters, while the temporal model performs the matrix factorization on each of the first 9 intervals separately, making recommendations on the 10th interval using our temporal scheme.

5.6.1 User-group matrices

We examine two sets of engagement matrices, those for consumption engagement, geared towards art curators and viewers, and those for production engagement, geared towards artists. Production and consumption engagements are computed according to their definitions in Section 5.3.

Production engagement matrix: We filter the user-group matrix such that each user has joined at least one group and each group has at least one member by 1 May 2011, and every user has at least one production engagement with some group in each 4-month time interval. We omit groups that do not receive at least two production engagements in every time interval. After performing this filtering, the production engagement matrix contains 8423 users and 4579 groups.

Consumption engagement matrix: Just as in production engagement, we filter the matrix to ensure that each user has joined at least one group and each group has at least one member by 1 May 2011. We omit users having fewer than five consumption engagements in every time interval, and we omit groups having fewer than ten consumption engagements in every time interval. After this filtering, the consumption engagement data contains 20328 users and 8772 groups.

Boolean membership matrix: For comparison, we also produce Boolean matrices corresponding to each of the filtered production and consumption matrices, setting

$$p_{ug} = \begin{cases} 1 & r_{ug} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

We set the confidence scaling parameter $\gamma = 0$ for experiments with Boolean membership matrices.

Table 5.1 summarizes the statistics of our data.

5.6.2 Recommendation methods

To examine the contribution of each aspect of our model to the overall performance, we run several variants of our method:

- **Static**
  - **Unbiased:** Our static model of Section 5.5.1 with fixed $\mathbf{w} = 0$ in (5.6).
  - **Biased:** Our static model of Section 5.5.1 with variable bias $\mathbf{w}$
  - **Biased (L2 Norm):** Our static model with only $l2$-regularization but with variable bias $\mathbf{w}$
- **Temporal**
  - **Unbiased:** Our temporal model of Section 5.5.2 with fixed $\mathbf{w} = 0$ in (5.6)
  - **Biased:** Our temporal model of Section 5.5.2 with variable bias $\mathbf{w}$

We include four comparison methods as baselines for our approaches.

- **Popular:** A baseline to indicate the problem's difficulty, recommending the K most popular groups; every user gets the same recommendations.
- **Boolean Membership:** The same matrix factorization method as Static Unbiased but applied to the Boolean membership matrices derived from the production and consumption engagement data.
- **Hu:** Uses implicit data according to [48], but in a static manner and without any of our improvements. Compared to Static Unbiased, this method omits the $l1$-regularization and the non-negativity constraint.
- **TMF:** As a comparable temporal method, we evaluate a version of Temporal Probabilistic Matrix Factorization [125].

We should notice that the **TMF** in our comparison experiment is not exactly the same as the one in the original paper by Zhang et al. [125]. Our method is employed to solve group recommendation problem where users can have different engagements with groups in different time intervals while Zhang's method is employed to solve user-item rating recommendation problem where each user can only rate each item once in the course of time. Furthermore, our method is derived from Koren's method on implicit data while Zhang's method is derived from the standard probabilistic matrix factorization on explicit rating data. To our knowledge, directly applying models designed for explicit ratings ranging from 1 to 5 to implicit ratings without Hu's [48] modification will result in a fairly bad performance followed by an unfair comparison. Our empirical experiments also proved this to be true.

On the other hand, we observed that conceptually Zhang's method is a special case of our method in the following aspects:

1. In our method, both user factors and item factors are dynamic and we learn the relations of both user and item factors in different time intervals while in the external method only user factor is dynamic and item factor is static, which means Zhang's method only learns the relations of user factors in different time intervals and they assume that item factors in different time intervals are independent from each other. So Zhang's method is a special case of our method. (when we assume item factors in past time intervals have no influence on item factors in future time intervals in our method)

2. We take longer term temporal evolution into consideration, i.e., we think user(and group) factors in time t depends not only on factors in $t-1$ but also on factors in $t-2$, $t-3$, $\cdots$, $t-p$, while the Zhang's method assumes the factors in t only depends on factors in t-1. So Zhang's method is also a special case of our method (when $p=1$ in our temporal method).

3. Our method employs user and group activity levels as biases while Zhang's method doesn't have any biases. So again, Zhang's method is a special case of our method (when we remove biases from our method or just assume all the biases equal to 0).

Thus, in order to make fair comparison with Zhang's **TMF**, we implement Zhang's idea within our framework for implicit data and employ single time step vector autoregression rather than first order Markov chains to model the relationship between user factors in adjacent time intervals, keeping remaining settings the same as in Zhang's original work.

Regularization and other numerical parameters used in static models (including **Boolean Membership**) together with the corresponding exploratory ranges in 10-fold cross validation are given in Table 5.2. The temporal models (including **TMF**) use optimal values obtained from the static model for common parameters. Although we did not perform an extensive study of the parameter space, in exploratory trials we did observe our results to be robust to each parameter over different static models and wide ranges, i.e., **Static Unbiased**, **Static Biased**, **Static Biased(L2 Norm)** and **Boolean Membership** all have the same optimal numeric values for their common parameters. We also conduct a same 10-fold cross validation to find the optimal parameters for baseline method **Hu**, setting $k = 70$, $\gamma = 2$ and $\lambda = 0.1$ in (5.4).

All methods were implemented in Python; we rely heavily on Python packages, using `sklearn` for elastic-net optimization, `statsmodels` for VAR calculations, and `multiprocessing` for parallelization. Our experiments were conducted on a 24-core Intel(R) Xeon(R) CPU X5650 @ 2.67GHz machine with 192GB RAM. The unbiased methods took between 10 minutes and 5 hours to run. The biased methods, requiring calculation of matrix $\sum_{ug} \mathbf{q}_{ug}\mathbf{q}_{ug}^T$, are significantly more expensive, running for 2 hours to 3.5 days. While the temporal versions of methods come in at the long end of these ranges, they scale linearly in the number of time slices used.

Table 5.2: Numerical parameters used for model comparison, showing the values we use in our reported experiments, and the ranges examined in our exploratory trials.

symbol	description	optimal value	cross validation range
k	latent dimension	100	10-200
γ	confidence scaling	2	1-6
α_u	user $l1$-regularization	0.1	0.01 - 0.5
β_u	user $l2$-regularization	0.1	0.01 - 0.5
α_i	group $l1$-regularization	0.1	0.01 - 0.5
β_i	group $l2$-regularization	0.1	0.01 - 0.5
θ	learning rate	0.9	0.5-1.0
p	auto-regression history	3	1-5

5.6.3 Evaluation method

Recall and precision are two most widely used evaluation metrics in top-K recommender systems. [27] is a well known work by Cremonesi and Koren, which conducts empirical experiments on an extensive evaluation of several state-of-the-art recommendation algorithms in top-K recommendations. Therefore, we also use the *Recall@K* and *Precision@K* measures adopted in [27] to compare the performance of our methods with the baseline techniques. For every user, we consider the set of groups $\mathcal{T}_u$ with which that user has the greatest engagement in the test set, up to a maximum of 20 groups. For each such target group g, we:

1. Randomly select 1000 groups with which the user has not yet engaged; our assumption is that most of these groups will be of no interest to the user.

2. Add the target group g into these randomly selected groups to form our candidate set of size 1001.

3. Predict the affinity for every group in the candidate set using our trained model.

4. Rank all the 1001 groups in the candidate set in descending order according to their predicted affinities.

5. Form a recommendation list by picking up the first K top ranked groups from the ranked list; if the target group g is in this list, record a *hit*, otherwise record a *miss*.

For each user u, we compute

$$\textit{Recall@}K_u = \frac{\#hits}{|\mathcal{T}_u|}, \tag{5.17}$$

$$\textit{Precision@}K_u = \frac{\#hits}{K \cdot |\mathcal{T}_u|} = \frac{\textit{Recall@}K_u}{K}, \tag{5.18}$$

Table 5.3: Baselines, *Recall@K*

Production			
	Popular	Boolean Membership	Hu
K=5	0.083	0.181	0.258
K=10	0.129	0.269	0.341
K=15	0.166	0.320	0.390
K=20	0.198	0.359	0.431
Consumption			
	Popular	Boolean Membership	Hu
K=5	0.073	0.219	0.287
K=10	0.112	0.296	0.356
K=15	0.143	0.347	0.404
K=20	0.170	0.389	0.438

where $|\mathcal{T}_u|$ is the number of test groups for u. Our results report the average recall and precision values over all users.

Though the two measures *Recall@K* and *Precision@K* used here are slightly different from the conventional ones used elsewhere, it is not hard to notice that they are the same in essence.

5.6.4 Results

Table 5.3 gives the full recall results of the static baseline methods for four values of K. Similarly, Tables 5.4 and 5.5 show recall results for the static and temporal methods, respectively, including the temporal TMF baseline. These tables produce a solid evidence that our proposed time-varying implicit engagement-based model provides the best top-K group recommendations, outperforming the naive baseline by up to 354.8% (*Recall@5* on consumption engagement data over Popular) and the best state-of-art competitor by up to 13.7% (*Recall@10* on production engagement data over TMF) in both datasets.

Figure 5.3 highlights the general performance of our full Temporal Biased method relative to some of the baseline comparison methods, demonstrating that the method drastically outperforms a simple Boolean Membership encoding, and also improves substantially upon a standard static implicit approach (Hu) run on our engagement data. The visible trend of the performance across values of K largely holds across method results, so we concentrate on the $K = 5$ results in the following.

Figure 5.4 and Figure 5.6a display the results of all static methods as a percentage increase over the popular baseline. Clearly, using the more sophisticated implicit measurements of user engagement with groups provides a substantial gain in recommendation performance. The minor changes in the form of non-negative elastic net regularization between Hu and our Static Unbiased

Table 5.4: Static methods, *Recall@K*

Production			
	Unbiased	Biased (L2)	Biased
K=5	0.266	0.272	0.286
K=10	0.348	0.358	0.380
K=15	0.401	0.411	0.435
K=20	0.440	0.453	0.474
Consumption			
	Unbiased	Biased (L2)	Biased
K=5	0.288	0.307	0.308
K=10	0.360	0.376	0.380
K=15	0.407	0.419	0.428
K=20	0.442	0.452	0.463

Table 5.5: Temporal methods, *Recall@K*

Production			
	TMF	Unbiased	Biased
K=5	0.277	0.286	0.308
K=10	0.358	0.368	0.407
K=15	0.408	0.420	0.454
K=20	0.448	0.457	0.491
Consumption			
	TMF	Unbiased	Biased
K=5	0.294	0.309	0.332
K=10	0.364	0.381	0.410
K=15	0.413	0.425	0.449
K=20	0.449	0.462	0.487

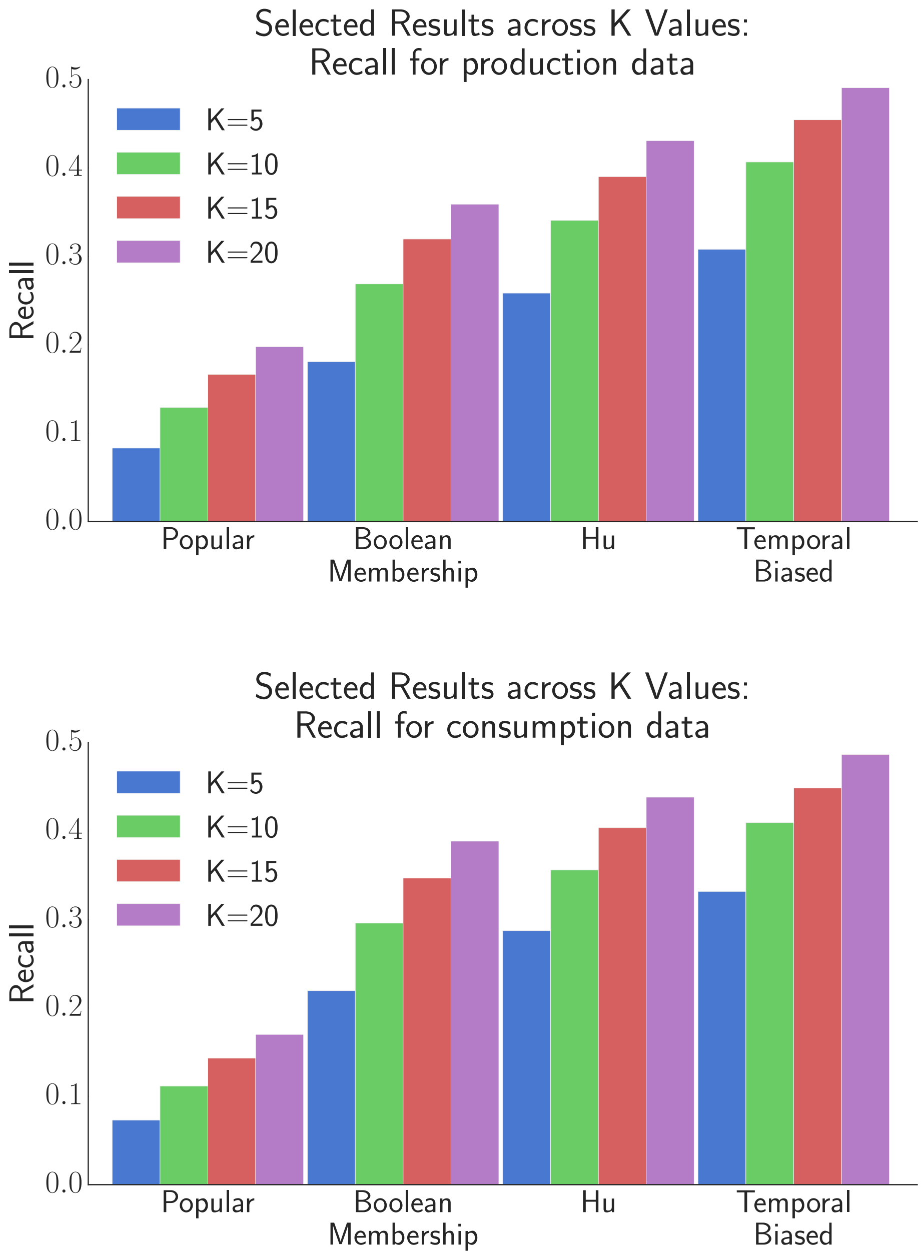


Figure 5.3: Results across a range of K values for baselines (Popular, Boolean Membership and Hu) and our full proposed method (Temporal Biased)

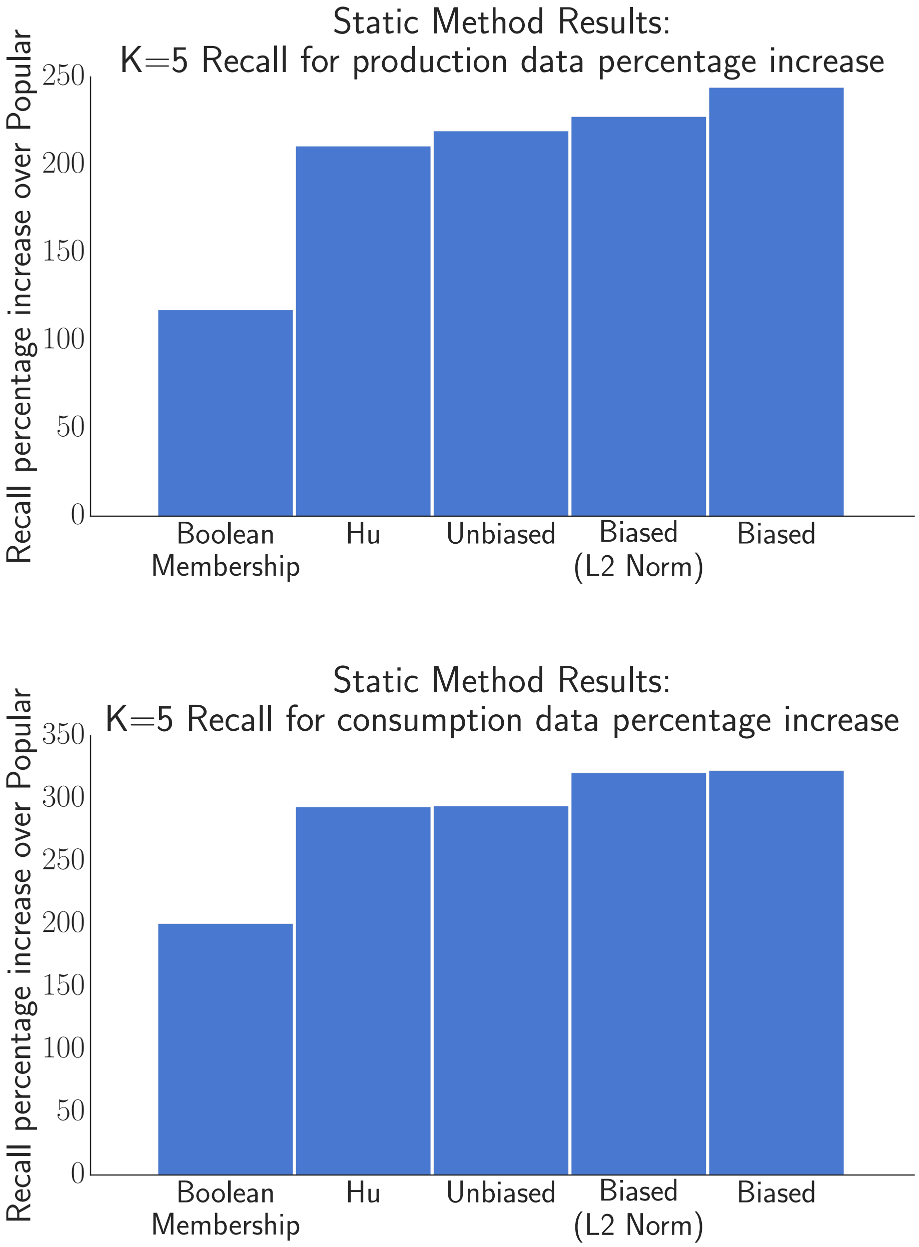


Figure 5.4: Results for the non-temporal (static) methods expressed as a percentage increase over the Popular baseline.

method yield a slight improvement on the production data. Adding the biases leads to a substantial gain in recall performance, especially together with l_1-regularization in the production case. While using a bias also improves performance on the consumption engagement data, there is almost no gain compared to using only l_2 regularization with biases in this case. This difference might be explained by the tendency of artists to have focused areas of interest and ability, and hence be well-modeled by sparse latent vectors. Conversely, art curators need not be so focused in their range of tastes, and hence may be better modeled by a less sparse latent preference factorization.

Figure 5.5 and Figure 5.6b summarize the results of the temporal baselines and methods as percentage improvements over the implicit engagement data based static Hu baseline. While TMF shows an improvement from taking the last time slice into account for user factors, modeling all the past time slices through smooth functions lets our temporal methods clearly outperform this baseline.

Figure 5.7 presents the trade-off between recall and precision. Each line in Figure 5.7 reports the precision of a method at a given recall. For instance, the precision of the Temporal Biased method on consumption data is about 0.066 when its recall is about 0.33. Recall will usually go up as K increases while precision tends to go down, which confirms that a trade-off between recall and precision is unavoidable in top-K recommendations. Additionally, Figure 5.7 demonstrates that the Temporal Biased method also obtains the best performance in terms of precision, followed by other variants of our method. We observe that the relative performances of all methods in terms of recall and precision are consistent on both datasets.

5.7 Summary

Many social media sites provide mechanisms by which users can form groups with others having similar interests. By joining groups, users are exposed to relevant content with the added benefit that this content has been curated for both topicality and quality. Recommending groups to users poses new challenges due to the complex dynamic relationship between users and groups in the course of time.

In this chapter, we propose production engagement and consumption engagement as measures that are more fine-grained than Boolean user-group membership to quantify user-group affinity more accurately. We present a time-dependent matrix factorization model to recommend groups of users, and perform experiments on two real-world user-group datasets from DeviantArt to demonstrate the improvement of our proposed method.

The experimental results show that for the complex problem of recommending social network groups to users taking into account detailed implicit engagement data rather than simply Boolean group memberships yields substantial improvements in recommendation performance. We achieve another performance boost by modeling the evolving nature of the relationships between users and groups smoothly over time rather than assuming they are static or can be predicted from the last time slice alone.

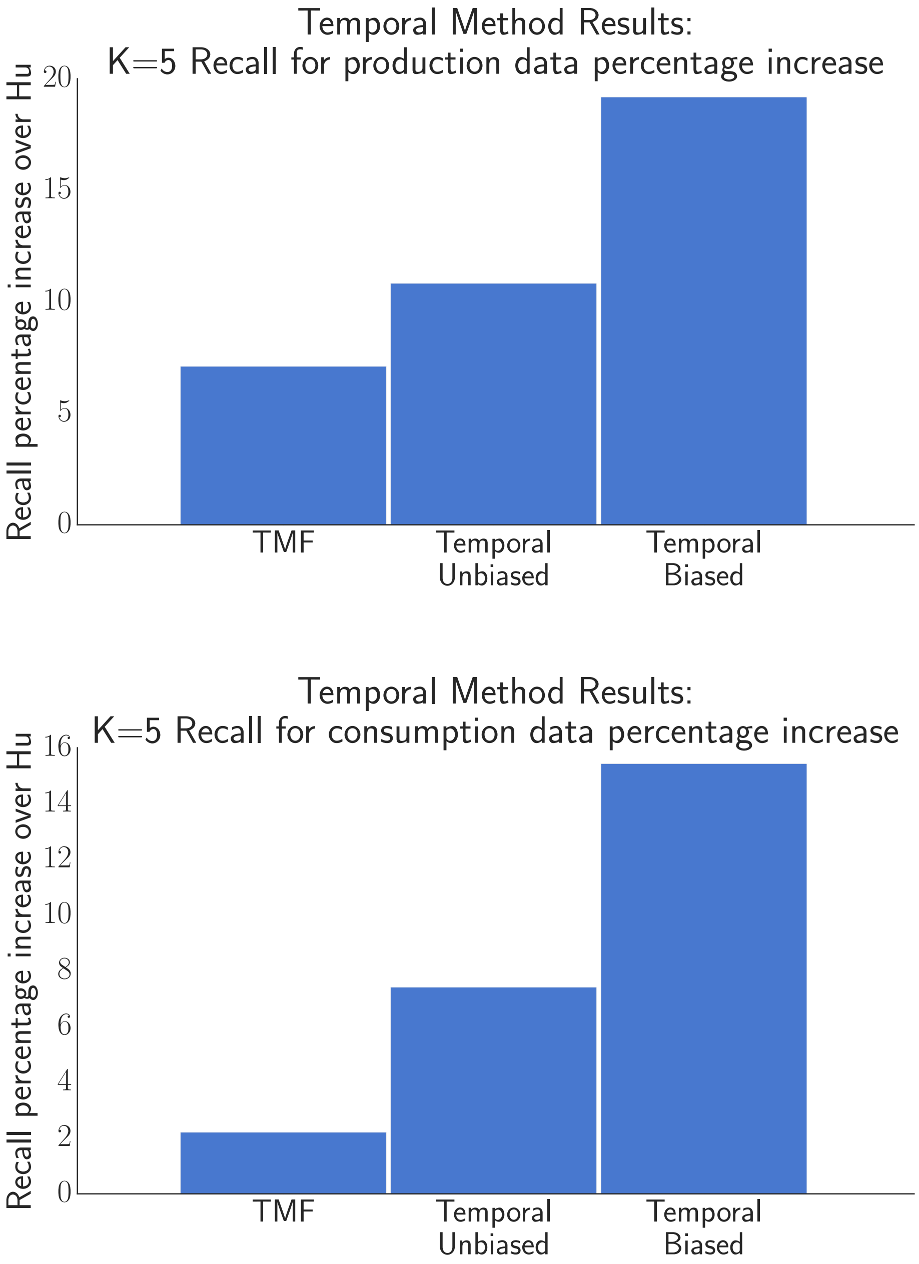
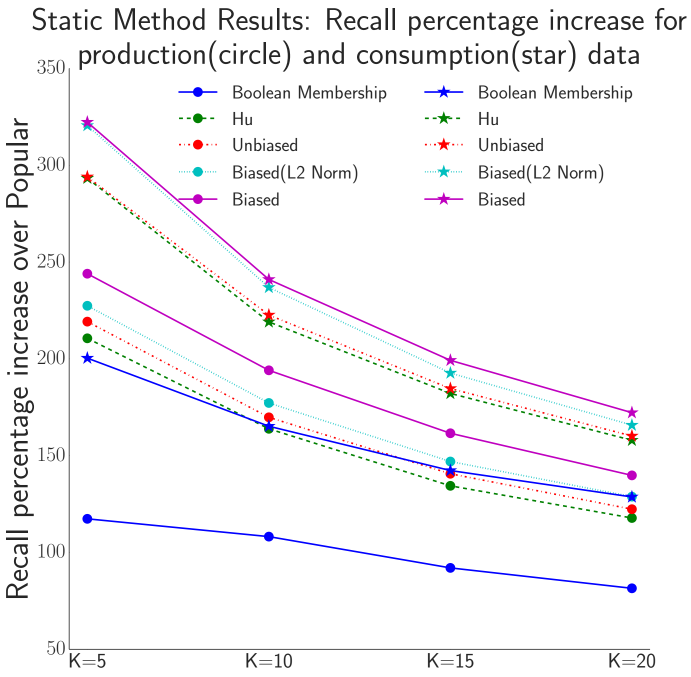
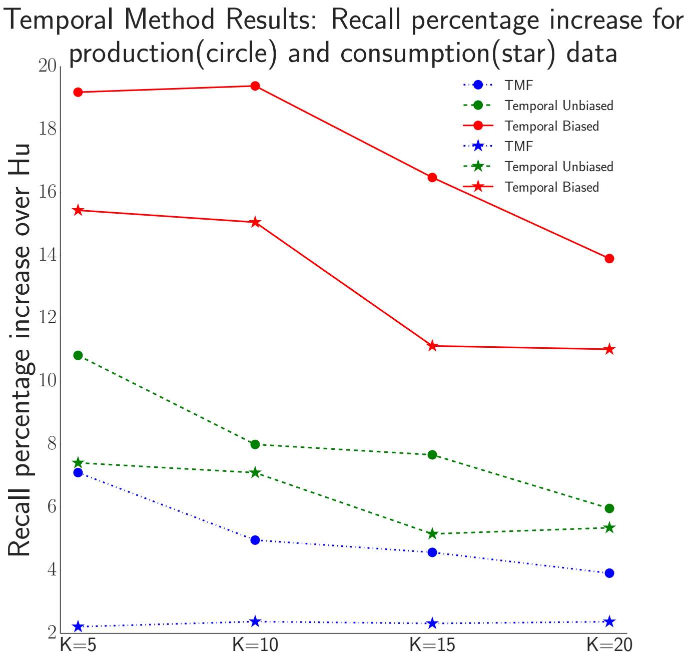


Figure 5.5: Results for the temporal methods, expressed as a percentage increase over the Hu baseline.



(a)



(b)

Figure 5.6: Recall for the static methods expressed as a percentage increase over the Popular baseline (a) and for the temporal methods expressed as a percentage increase over the Hu baseline (b)

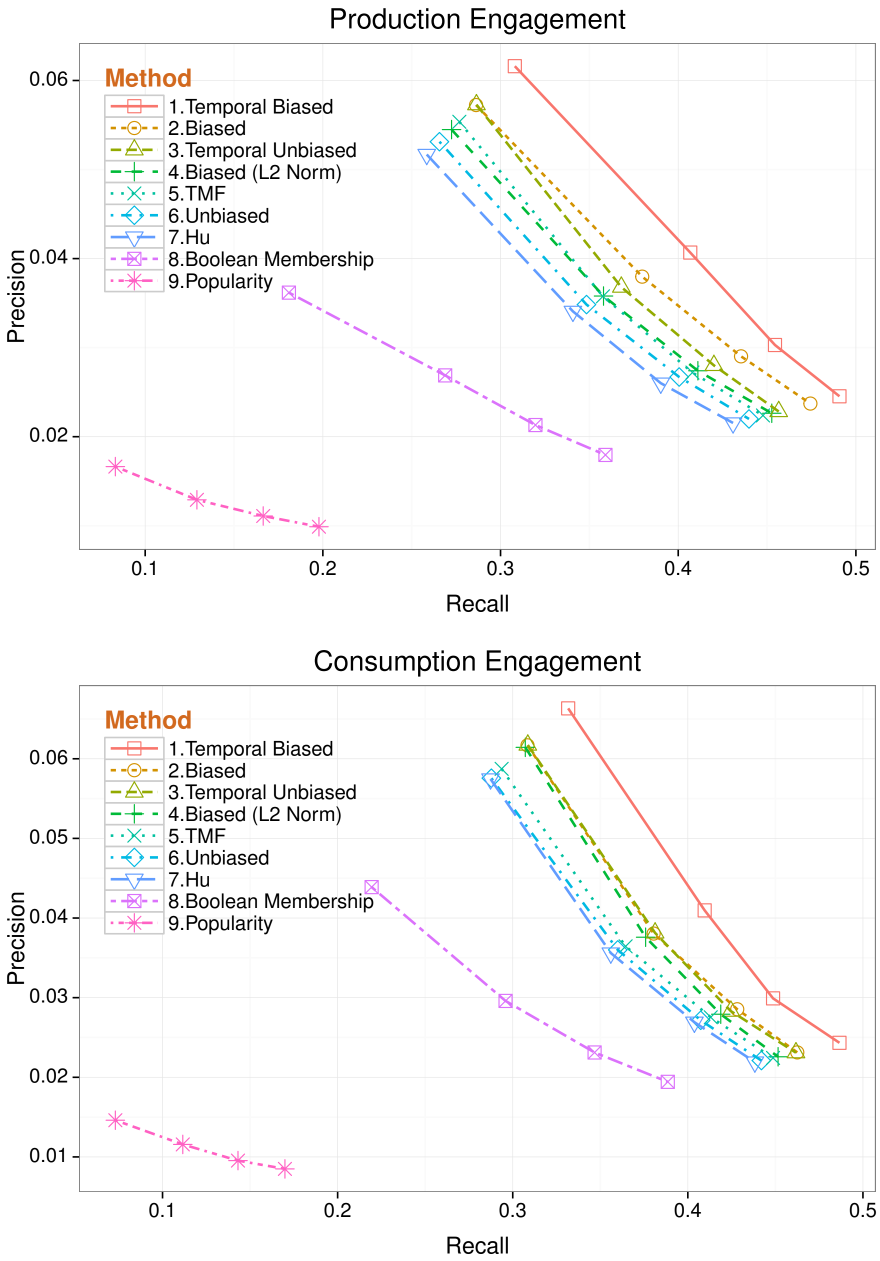


Figure 5.7: Results of precision versus recall for all methods, showing improved performance as complexity is added to the model.

Chapter 6

Interactive Social Recommendation with Online Learning

Social recommendation has been an active research topic over the last decade, based on the assumption that social information from friendship networks is beneficial for improving recommendation accuracy, especially when dealing with cold-start users who lack sufficient past behavior information for accurate recommendation. However, it is nontrivial to use such information, since some of a person's friends may share similar preferences in certain aspects, but others may be totally irrelevant for recommendations. Thus one challenge is to explore and exploit the extend to which a user trusts his/her friends when utilizing social information to improve recommendations. On the other hand, most existing social recommendation models are non-interactive in that their algorithmic strategies are based on batch learning methodology, which learns to train the model in an offline manner from a collection of training data which are accumulated from users' historical interactions with the recommender systems. In the real world, new users may leave the systems for the reason of being recommended with boring items before enough data is collected for training a good off-line model, which results in an inefficient customer retention. To tackle these challenges, we propose a novel method for interactive social recommendation, which not only simultaneously explores user preferences and exploits the effectiveness of personalization in an interactive way, but also adaptively learns different weights for different friends. In addition, we also give a rigorous analysis on the complexity and regret of the proposed model. Extensive experiments on three real-world datasets illustrate the improvement of our proposed method against the state-of-the-art algorithms.

6.1 Motivation

Recommender systems have become a hot research topic in academia and been widely adopted in industry as well. Moreover, the rising of social networks and rapid development of web services actuate the emergence of recommendation in social media. People not only rate movies or TV series on IMDB, but also interact with each other on Facebook and see the latest updates of their favorite idols on Twitter. This brings the idea of social recommendation which tries to utilize available information (e.g., ratings) from users' friends to infer their preferences. Lots of existing work [50, 51, 77–80, 91, 110, 117, 118, 120, 128] has proved that incorporating social information from social networks does help to improve the accuracy of conventional recommendation methods [60, 82, 108]. At the same time, as more and more web service providers begin incorporating social elements into their services, social recommendation has also become a well studied topic in which most of their algorithmic strategies are to learn the model off-line via batch learning without any interactions from users. The training data in batch learning is normally obtained through the accumulation of users' historical interactions with the recommender systems, which may run the risk of users in real world leaving the systems because of many boring items being recommended to them before enough data is collected for training a good off-line model, resulting in inefficient customer retention. Besides, although social information from friends has been proved to be very useful for the improvement of recommendation accuracy, some of these friends may share similar preferences with the target user while others may be totally irrelevant for recommendations because of domain differences. This poses two challenges to us: first, how can we provide good-quality recommendations as soon as possible even when the target user has little past behavior data in order to maximize user retention in social recommendation; second, how to dynamically learn different weights for different friends which can best serve the recommendation accuracy when receiving more and more feedback from users.

To handle the first challenge, multi-armed bandit (MAB) serves as a competent candidate for recommendation with user interactions given its capability of simultaneously exploiting existing information that matches user interest and exploring new information that can improve global user experience, which is known as the *exploitation-exploration* trade-off dilemma. Thus casting the mechanism of multi-armed bandit into social recommendation can help mitigate the dilemma of user retention. A significant amount of work has been done on stochastic multi-armed bandit algorithm to provide principled solutions to the exploitation-exploration dilemma [11, 12, 42, 63]. In addition to the vanilla stochastic linear bandit models, contextual bandit algorithms [10, 19, 33, 112] become promising solutions when side information like contextual content (e.g., texts, tags, etc.) about users and items is available in scenarios such as mobile recommender systems [17], news recommendation [69] and display advertising [20, 72]. In general, the multi-armed bandit based algorithms try to get a good understanding of user preferences and thus achieve a high-quality recommendation as soon as possible through collecting a small amount of interactive feedback (e.g., behaviors such as ratings, clicks and favorites etc.) from users. We will give a detailed description on how multi-armed bandit can be incorporated into social recommendation later.

As for the second challenge, it is also necessary to exploit and explore the extend to which the current user trusts her friends when utilizing social information to improve recommendations. Since our goal is to adaptively learn the weights of different friends as more and more user interactive feedback becomes available, we employ a modified multi-arm bandit schema to dynamically update these weights upon receiving new feedback from users after they interact with the systems (e.g., give feedback such as clicks or ratings).

On the other hand, all the contextual bandit models we mentioned utilize content data such as tags and texts to construct an explicit feature vector (for each user and item) which will be used to determine the expected reward of the bandit. In practice, it is not always the case that the content data used to extract user and item feature vectors can be easily obtained, which makes the contextual bandit algorithms ineligible for producing accurate recommendations. Inspired by Zhao et al.'s work [129] and Qin et al.'s work [92], we borrow the idea from matrix factorization [82] which factorizes the observed user feedback into latent feature vectors in order to address our social recommendation problem in the scenario where there is no content information to construct explicit feature vectors and only user feedback (e.g., ratings, clicks, bookmarks etc.) can be observed. We employ the factorized latent user and item feature vectors to represent content information, extend the classical matrix factorization and combine it with the contextual multi-armed bandit in social recommendation.

In summary, we make the following contributions.

- We propose a novel interactive social recommendation model (ISR) which differs from and is superior to previous work in the following aspects.
    1. Previous work on social recommendation [50, 51, 77–80, 91, 110, 117, 118, 120, 128] does not consider interactive learning.
    2. Given a user in social recommendation, some existing methods simply compute the weights (i.e., degree of trust) for his/her friends uniformly (i.e., give equal weight to every friend) [51, 77, 79], which is a suboptimal solution because of the domain differences. Some others obtain these weights by calculating the rating similarities between the given user and his/her friends [77, 78], which is in a static way as well. This being the case, our solution is novel in the sense of adaptively learning these weights.
- We give a rigorous regret analysis to show that part of our proposed interactive social recommendation model has a regret bound of $\mathcal{O}(\sqrt{T})$.
- We conduct extensive experiments on three real-world datasets and demonstrate the improvement of our proposed ISR model against the state-of-the-art methods in various accuracy metrics.

The remainder of this chapter is organized as follows: we present comparisons between our proposed model and existing methods in Section 6.2. Section 6.3 illustrates the general idea of multi-armed bandit methodology together with several popular bandit models in the context of recommendation. Section 6.4 gives a detailed formation of our proposed *Interactive Social Recommendation* (ISR) model, followed by complexity and regret analyses on the scalability and regret bound of ISR model. Section 6.5 presents our experiments, compares our approach to several state-of-art recommendation methods and comments on their performances in terms of various evaluation metrics. We finally summarize our work in Section 6.6.

6.2 Comparison with Existing Approaches

Cesa-Bianchi et al. [19], Wu et al. [112] and Wang et al. [105] study the bandit setting where information from social networks is taken into account. Specifically, Cesa-Bianchi et al.'s model utilizes a graph Laplacian to regularize the model so that users and their friends have similar bandit parameters and Wu et al.'s model, on the other hand, assumes the reward in bandit is generated through an additive model, indicating that friends' feedback (reward) on their recommendations can be passed via the network to explain the target user's feedback (reward). Wang et al. examine the bandit setting from another view through combining it with matrix completion. However, all the proposed models in [19, 105, 112] assume the weights for different friends to be fixed, without learning these weights adaptively to best serve the recommendation accuracy. Besides, their focus is orthogonal to ours in this work as they are based on explicit features whose model formulations and experimental settings are different from those based on latent features.

There is also some work that incorporates matrix factorization into the bandit setting [55, 129], among which Kawale et al. [55] employ Thompson sampling to perform online recommendation and Zhao et al. [129] propose an interactive collaborative filtering method based on probabilistic matrix factorization. We remark that neither of these models takes social information into consideration.

6.3 Multi-armed Bandit Methodology in Recommendation

In practice, we often face many situations where it is necessary to find a balance between exploiting our current knowledge and obtaining new knowledge through searching unknown space. Take recommender systems as an example, ultimately we would like to recommend "good" items to users with the best knowledge we have so far as well as explore users' other interests which we have no idea about through exposing some "random" items to them and observing their corresponding reactions to these random recommendations. As is discussed previously, multi-armed bandit is adequate as an appropriate solution for this exploitation-exploration dilemma. In this section, we will give a mathematical description of the general idea for multi-armed bandit (MAB) strategy in the context of recommender systems, as well as several existing multi-armed bandit models which are to be used as baselines for comparison with our proposed model in the experiments.

Formally, a K-armed bandit consists of K arms, representing K candidate items to be recommended to a user and pulling an arm means recommending an item to a user. In a general stochastic formation, for each user u, these K arms can also be treated as K probability distributions $[D_{u,1}, D_{u,2}, \cdots, D_{u,K}]$ with associated expected values (i.e., means) $[\mu_{u,1}, \mu_{u,2}, \cdots, \mu_{u,K}]$

and variances $[\sigma_{u,1}, \sigma_{u,2}, \cdots, \sigma_{u,K}]$ where the distribution $D_{u,i}$ is initially unknown. A bandit algorithm proceeds in discrete trials (rounds) $t = 1, 2, 3, \cdots$, and given a user u, it chooses one item i out of the K candidates(through pulling one of the K arms) and recommends it to the user u in each trial (round). After each recommendation, the algorithm receives a reward $r_{u,i}(t) \sim D_{u,i}(t)$ for picking item i as the recommendation for user u. The *total expected regret* is used to measure the performance of bandit algorithms. For a bandit algorithm running totally T trials (rounds), the *total expected regret* R_T is defined as follows:

$$R_T = \sum_{u \in U} \left[\mathbb{E}\Big[\sum_{t=1}^{T} \mu_*\Big] - \mathbb{E}\Big[\sum_{t=1}^{T} \mu_{u,i}(t)\Big] \right] , \tag{6.1}$$

where U is the set of users for evaluation and $\mu^* = max_{j=1,2,\cdots,K}\mu_j$ is the expected reward from the best arm (i.e., best candidate item) in each round. Our objective is to find an optimal set of items, minimizing the total expected regret R_T, as the recommendation for each user, which equals to maximizing the cumulative expected reward during T rounds for every user:

$$I_u(T) = \bigcup_{t=1}^{T} \arg\max_i \mathbb{E}[r_{u,i}(t)] \; = \bigcup_{t=1}^{T} i_u(t) \; . \tag{6.2}$$

Most bandit strategies maintain empirical average rewards which will be updated in every round for each arm chosen. We denote $\hat{r}_{u,i}(t)$ as the empirical average reward of arm (i.e., item) i after t rounds for user u, and $p_{u,i}(t)$ as the probability of picking arm i for user u (i.e., recommending item i to user u) in round t.

ϵ-greedy.

The ϵ-greedy algorithm is widely used because of its simplicity, and obvious generalizations for sequential decision problems. In each round $t = 1, 2, \cdots$ the algorithm selects the item with the highest empirical average reward from the K candidate items with probability $1 - \epsilon$, and selects a random item with probability ϵ. In other words, given initial empirical average rewards $\hat{r}_{u,1}(0), \hat{r}_{u,2}(0), \cdots, \hat{r}_{u,K}(0)$ for user u,

$$p_{u,i}(t+1) = \begin{cases} 1 - \epsilon + \epsilon/K, & \text{if } i = \arg\max_{j=1,\cdots,K} \hat{r}_{u,j}(t) \\ \epsilon/K, & \text{otherwise.} \end{cases} \tag{6.3}$$

Boltzmann Exploration (Softmax).

Softmax methods are based on Luce's axiom of choice [76] and pick each item for recommendation with a probability that is proportional to its average reward. Therefore items with greater empirical average rewards should be picked with higher probabilities. In the following we will describe Boltzmann Exploration [100], a Softmax method which selects an item using a Boltzmann distribution. Given the initial empirical average rewards of the K candidate items for user u (denoted as $\hat{r}_{u,1}(0), \hat{r}_{u,2}(0), \cdots, \hat{r}_{u,K}(0)$), the probability of picking item i as recommendation for user u in round $t+1$ is:

$$p_{u,i}(t+1) = \frac{e^{\hat{r}_{u,i}(t)/\tau}}{\sum_{j=1}^{K} e^{\hat{r}_{u,j}(t)/\tau}} \ , \tag{6.4}$$

where τ is a temperature parameter controlling the randomness of the choice. We would like to point out that Boltzmann Exploration acts like pure greedy when τ tends to 0, and selects items for recommendations uniformly at random as τ tends to ∞.

Upper Confidence Bounds (UCB).

Lai and Robins are the first to introduce the technique of upper confidence bounds for the asymptotic analysis of regret in stochastic bandit models [63]. Later Auer employs the UCB based algorithm to show how confidence bounds can be applied to elegantly deal with the trade-off between exploitation and exploration in online learning [9]. Then the family of UCB algorithms are proposed in [11] as a simple and elegant implementation of the idea for optimism under uncertainty. In addition to the empirical average reward, UCB maintains the number of times that each item is picked for recommendation up to round t as well. Initially all the items are assumed to be chosen once and afterwards the algorithm greedily selects item i in round t as follows:

$$i(t) = \underset{j=1,\cdots,K}{\arg\max} \left(\hat{r}_{u,j}(t) + \sqrt{\frac{2 \log t}{n_j(t)}} \right) \ , \tag{6.5}$$

where $n_j(t)$ represents the number of times item j has been selected for recommendations so far. We note that $\hat{r}_{u,j}(t)$ is the empirical mean estimate of $r_{u,j}(t)$ in round t given previous observations in the past $t-1$ rounds and $\sqrt{\frac{2\log t}{n_j(t)}}$ is an upper confidence bound. This can be interpreted as a good trade-off between exploitation, i.e., $\hat{r}_{u,j}(t)$, and exploration, i.e., $\sqrt{\frac{2\log t}{n_j(t)}}$.

Linear UCB (LinUCB).

Li et al. propose a linear model under the UCB framework (called LinUCB) through combining linear bandit and contextual bandit together to focus on the problem of personalized news article recommendation [69]. LinUCB assumes that the mean of $r_{u,i}(t)$ can be obtained through the dot product of an item-dependent coefficient with the concatenation of user u's and item i's feature vectors in round t, which is linear with respect to the item-dependent coefficient given that the user and item feature vectors are known to us.

However, explicit feature vector may not be always available in practice. Take movie recommendation as an example, most of the state-of-the-art methods are based on collaborative filtering where user and item latent feature vectors are learnt through low rank matrix factorization. Therefore, given the success of collaborative filtering in recommender systems, we formulate LinUCB through employing the latent feature vectors learnt by low rank matrix factorization instead of explicit feature vectors extracted directly from texts or labels in this chapter, which is similar to algorithm 2 in [129].

As such, a common strategy widely adopted by many matrix factorization based collaborative filtering algorithms is to approximate the feedback (e.g., ratings, clicks etc.) through the inner product of user and item latent feature vectors ($\mathbf{p}_u$ and $\mathbf{q}_i$):

$$r_{u,i} = \mathbf{p}_u^\intercal \mathbf{q}_i \ . \tag{6.6}$$

To incorporate the low rank matrix factorization into LinUCB, we reformulate the bandit strategy for item selection in the same way as [129]:

$$\begin{aligned} i(t) &= \underset{j=1,\cdots,K}{\arg\max}\, \mathbb{E}\big[r_{u,j}(t)\big] \\ &= \underset{j=1,\cdots,K}{\arg\max}\, \mathbb{E}_{\mathbf{p}_u}\big[\mathbf{p}_u^\intercal | t\big]\mathbf{q}_j \\ &= \underset{j=1,\cdots,K}{\arg\max}\Big(\hat{\mathbf{p}}_{u,t}^\intercal \mathbf{q}_j + c\sqrt{\mathbf{q}_j^\intercal \mathbf{\Sigma}_{u,t}^{-1}\mathbf{q}_j}\Big) \ . \end{aligned} \tag{6.7}$$

And we treat the user feedback for an item as the reward of picking this item for recommendation.

We conclude this section by pointing out that all of these existing models handle users' preferences over items without considering the influences from their friends on social networks, nor do they adaptively learn the different weights for different friends to best serve the recommendation accuracy. This motivates us to develop a novel multi-armed bandit (MAB) model that is capable of taking not only user-item interactions but also social information from social networks into consideration and learning the weights between the target user and her social ties dynamically so that a boost in terms of recommendation quality can be achieved.

6.4 Interactive Social Recommendation

In this section, we propose our interactive social recommendation model (ISR) which is capable of refining itself to best serve the customers after each interaction with a user.

Let $\mathcal{U}$ be the set of users for evaluation and $\mathcal{I}$ be the set of candidate items, given a user $u \in \mathcal{U}$, N_u denotes the set of her friends, i.e., her directly connected users, and $w_{u,f}$ is the weight for the edges (connections) between user u and her friend $f \in N_u$. Recall that the vanilla matrix factorization presented in (6.6) has been widely adopted by collaborative filtering in both academia and industry [60]. Thus given the great success of matrix factorization in recommendation during the past years, lots of social recommendation models [51, 77, 79, 80, 117] actually are extensions based on the vanilla matrix factorization, among which Ma et al. propose the STE (Recommendation with Social Trust Ensemble) model that uses a weighted aggregation of a user's own preferences and her friends' preferences to predict the target user's final feedback (e.g., rating) on an item:

$$r_{u,i} = \alpha \mathbf{p}_u^\intercal \mathbf{q}_i + (1-\alpha)\sum_{f \in N_u} w_{u,f}\mathbf{p}_f^\intercal \mathbf{q}_i \ , \tag{6.8}$$

where α is a pre-set parameter controlling the relative importance of the target user's own preferences and her friends' influences, which naturally simulates the real-world scenario in which people's final decisions depend on both own preferences and friends' influences. Although this idea is elegant and effective in reducing the inaccuracy of traditional matrix factorization, it has some limitations:

1. It is an offline method depending on batch learning and not applicable for real-world recommender systems which serve in an online and interactive manner.

2. It assumes a pre-calculated and fixed weight for each friend, which may not always hold as the degree of trust between users and their friends tends to change when new user feedback is observed. Our proposed ISR model, on the other hand, is capable of addressing these limitations.

6.4.1 The ISR Model

A modified version of linUCB is proposed in [129] via replacing the dot product of contextual feature vectors and coefficients with probabilistic matrix factorization, where the reward of recommending an item i to a user u in round t is regarded as the feedback (such as ratings, clicks etc.) of user u on item i:

$$r_{u,i}(t) = \mathbf{p}_u^\intercal(t)\mathbf{q}_i \ . \tag{6.9}$$

The ISR model extends this formula (6.9) by incorporating the social part:

$$r_{u,i}(t) = \alpha\mathbf{p}_u^\intercal(t)\mathbf{q}_i + (1-\alpha)\sum_{f\in N_u} w_{u,f}^\dagger(t)\mathbf{p}_f^\intercal\mathbf{q}_i \ , \tag{6.10}$$

where same as in (6.8), α is the importance controlling parameter in range $[0,1]$ and $w_{u,f}^\dagger = \frac{w_{u,f}}{\sum_{v\in N_u} w_{u,v}}$ is the normalized edge weight between u and f. Then the item that has the largest weighted sum of expected rewards from u and all her friends $f \in N_u$ is selected in round t:

$$\begin{aligned} i(t) &= \arg\max_{j=1,\cdots,K} \mathbb{E}\Big[\alpha\hat{r}_{u,j}(t) + (1-\alpha)\sum_{f\in N_u}\hat{w}_{u,f}^\dagger r_{f,j}(t)\Big] \\ &= \arg\max_{j=1,\cdots,K}\left(\alpha\hat{\mathbf{p}}_u^\intercal(t)\mathbf{q}_j + (1-\alpha)\sum_{f\in N_u}\hat{w}_{u,f}^\dagger(t)\mathbf{p}_f^\intercal\mathbf{q}_j\right) \ , \end{aligned} \tag{6.11}$$

where K is the number of candidate items for u in round t. For convenience, we construct a social weight coefficient vector for each user u (denoted as $\mathbf{w}_u$) that consists of all the edge weights for her friends : $\mathbf{w}_u = [w_{u,f_1}^\dagger, w_{u,f_2}^\dagger, w_{u,f_3}^\dagger, \cdots w_{u,f_{|N_u|}}^\dagger]^\intercal$, and by further denoting $\mathbf{s}_{u,i} =$

```
Algorithm 2: Interactive Social Recommendation
Input: $c_1, c_2 \in \mathbb{R}_+$, $\alpha \in [0,1]$, $\lambda_p$, $\lambda_w$
        Graph $G(U,E)$, where $\mathcal{U}$ is the set of users, $\mathcal{E}$ is the set of edges indicating the
  connected linkage graph.
        MAP solutions for item latent feature vectors:
        $\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_3}, \cdots \mathbf{q}_{|\mathcal{I}|}$
Initialization:
$\mathbf{\Sigma}_{u,1} \leftarrow \lambda_p I$, $\mathbf{h}_{u,1} \leftarrow \mathbf{0}$
$\mathbf{\Delta}_{u,1} \leftarrow \lambda_w I$, $\mathbf{z}_{u,1} \leftarrow \mathbf{0}$
for $t \leftarrow 1$ to $T$ do
    $\mathbf{p}_{u,t} \leftarrow \mathbf{\Sigma}_{u,t}^{-1}\mathbf{h}_{u,t}$
    $\mathbf{w}_{u,t} \leftarrow \mathbf{\Delta}_{u,t}^{-1}\mathbf{z}_{u,t}$
    where $\mathbf{w}_u = [w_{u,f_1}^{\dagger}, w_{u,f_2}^{\dagger}, w_{u,f_3}^{\dagger}, \cdots w_{u,f_{|N_u|}}^{\dagger}]^{\mathsf{T}}$,
    and $f_1, f_2, \cdots, f_3 \in N_u$.
    foreach $i \in \mathcal{I}$ do
        foreach $f \in N_u$ do
            $s_{f,i} = \mathbf{p}_f^{\mathsf{T}}\mathbf{q}_i$
        end
        $\mathbf{s}_{u,i} = [s_{f_1,i}, s_{f_2,i}, s_{f_3,i}, \cdots, s_{f_{|N_u|},i}]^{\mathsf{T}}$,
        where $f_1, f_2, \cdots, f_3 \in N_u$.

        $$g_{u,i}(t) \leftarrow \alpha\Big(\mathbf{p}_{u,t}^{\mathsf{T}}\mathbf{q}_i + c_1\sqrt{\mathbf{q}_i^{\mathsf{T}}\mathbf{\Sigma}_{u,t}^{-1}\mathbf{q}_i}\Big) + (1-\alpha)\Big(\mathbf{w}_{u,t}^{\mathsf{T}}\mathbf{s}_{u,i} + c_2\sqrt{\mathbf{s}_{u,i}^{\mathsf{T}}\mathbf{\Delta}_{u,t}^{-1}\mathbf{s}_{u,i}}\Big)$$
    end
    Choose the item $i = \arg\max g_{u,j}(t)$ where $j = 1, \cdots, K$, with ties broken arbitrarily.
    Receive a real-value reward $r_{u,i}(t)$.

    Update:

        $$\mathbf{\Sigma}_{u,t+1} \leftarrow \mathbf{\Sigma}_{u,t} + \mathbf{q}_i\mathbf{q}_i^{\mathsf{T}}$$
        $$\mathbf{\Delta}_{u,t+1} \leftarrow \mathbf{\Delta}_{u,t} + \mathbf{s}_{u,i}\mathbf{s}_{u,i}^{\mathsf{T}}$$
        $$\mathbf{h}_{u,t+1} \leftarrow \mathbf{h}_{u,t} + \frac{\big(r_{u,i}(t) - (1-\alpha)\mathbf{w}_{u,t}^{\mathsf{T}}\mathbf{s}_{u,i}\big)\mathbf{q}_i}{\alpha}$$
        $$\mathbf{z}_{u,t+1} \leftarrow \mathbf{z}_{u,t} + \frac{\big(r_{u,i}(t) - \alpha\mathbf{p}_{u,t}^{\mathsf{T}}\mathbf{q}_i\big)\mathbf{s}_{u,i}}{1-\alpha}$$
end
Output: P = {$\mathbf{p}_u$: $u \in \mathcal{U}$},W = {$\mathbf{w}_u$: $u \in \mathcal{U}$}
```

$[s_{f_1,i}, s_{f_2,i}, s_{f_3,i}, \cdots, s_{f_{|N_u|},i}]^{\mathsf{T}}$ where $s_{f,i} = \mathbf{p}_f^{\mathsf{T}}\mathbf{q}_i$, we can rewrite (6.11) as follows:

$$i(t) = \arg\max_{j=1,\cdots,K}\Big(\alpha\hat{\mathbf{p}}_{u,t}^{\mathsf{T}}\mathbf{q}_j + (1-\alpha)\hat{\mathbf{w}}_{u,t}^{\mathsf{T}}\mathbf{s}_{u,j}\Big)\,. \tag{6.12}$$

In plain English, our ISR model aims to find an *optimal* set of items as recommendations for different users, such that the accumulated expected reward of the recommendations over all users will be maximized.

Given the fact that user preferences tend to change in the course of time while item characteristics normally remain static, it is natural for our ISR model to place more focus on the knowledge obtained in the last round for the target user rather than for the item, especially when a sufficient amount of feedback has been collected to infer the latent feature spaces for items. Therefore, we will assume that the item latent feature vectors have already been pre-learnt through the *maximum a posterior* (MAP) estimate under matrix factorization. We will talk more about this experimental setting later in Section 6.5.

If the item latent feature vectors remain fixed, then the reward in (6.10) becomes linear with respect to the user latent feature vectors with the social weight coefficient vectors treated as constants, and also linear with respect to the social weight coefficient vectors with the user latent feature vectors treated as constants. Our goal is to find the best user latent feature vectors and the optimal edge weights for their friends.

The uncertainty of the reward comes from two parts: self-reward ($\mathbf{p}_u^\intercal \mathbf{q}_j$) and social-reward ($\mathbf{w}_u^\intercal \mathbf{s}_{u,j}$), whose uncertainty derives from the estimation for user latent feature vector $\mathbf{p}_u$ and social weight coefficient vector $\mathbf{w}_u$ respectively. According to ridge regression, the uncertainty of estimation for $\mathbf{p}_u$ is:

$$||\mathbf{q}_i||_{\Sigma_{u,t}^{-1}} = \sqrt{\mathbf{q}_i^\intercal \Sigma_{u,t}^{-1} \mathbf{q}_i} \,, \tag{6.13}$$

where $\Sigma_{u,t}^{-1}$ is the inverse covariance matrix for u's self-reward in round t. And similarly, the uncertainty in the estimation of $\mathbf{w}_u$ can be formulated as follows:

$$||\mathbf{s}_{u,i}||_{\Delta_{u,t}^{-1}} = \sqrt{\mathbf{s}_i^\intercal \Delta_{u,t}^{-1} \mathbf{s}_i} \,, \tag{6.14}$$

where $\Delta_{u,t}^{-1}$ is the inverse covariance matrix for u's social-reward in round t. ISR chooses the item with the highest upper confidence bound in each round:

$$\begin{aligned} i(t) = \underset{j=1,\cdots,K}{\arg\max} \Bigg[& \alpha\Big(\mathbf{p}_{u,t}^\intercal \mathbf{q}_j + c_1\sqrt{\mathbf{q}_j^\intercal \boldsymbol{\Sigma}_{u,t}^{-1} \mathbf{q}_j}\Big) \\ & + (1-\alpha)\Big(\mathbf{w}_{u,t}^\intercal \mathbf{s}_{u,j} + c_2\sqrt{\mathbf{s}_{u,j}^\intercal \boldsymbol{\Delta}_{u,t}^{-1} \mathbf{s}_{u,j}}\Big)\Bigg] \,, \end{aligned} \tag{6.15}$$

where c_1 and c_2 are two parameters used to determine the confidence. The details of our proposed ISR model are given in Algorithm 2.

6.4.2 Complexity Analysis

Exploitation-exploration is essentially all about the parameter space for exploration. Existing multi-armed bandit (MAB) based recommendation methods normally treat each item as an arm, which results in $|\mathcal{I}|$ (i.e., total number of candidate items) parameters for each user. LinUCB [69] reduces the number of parameters for each user to $O(d)$ (i.e., the sum of the length of user and item feature vectors) by a linear model, so does the modified LinUCB under matrix factorization introduced in [129] (whose number of parameters is exactly d, the length of item latent feature vector, for every user). As for our ISR model, given a user u, there is one more parameter w_{uf} for each friend $f \in N_u$ of user u, thus we will have $|N_u|$ (number of u's friends) parameters added to the social part of our ISR model. Therefore, ISR requires $d + |N_u|$ parameters for each user u. If $|N_u|$ is larger than the number of latent factors, it would introduce more parameters to estimate. Then there is a potential problem that we might need more data/rounds to give good recommendation. The only hope is that friends' latent factors are helpful enough to reduce the risk.

6.4.3 Regret Analysis

We remark that the self-reward part of our proposed ISR model has a regret bound of $\mathcal{O}(\sqrt{T})$ under certain assumptions and will provide the regret analysis in detail as follows.

Recall that for UCB based algorithm, take (6.5) and (6.7) for instance, the choice of item in each round is:

$$i(t) = \underset{j=1,\cdots,K}{\arg\max}\Big(\hat{r}_j(t) + \hat{c}_j(t)\Big) \ , \tag{6.16}$$

where for each item $j = 1, \cdots, K$, the true mean reward $r_j(t)$ in round t lies in a confidence interval:

$$C_j(t) : \Big[\hat{r}_j(t) - \hat{c}_j(t) \quad , \quad \hat{r}_j(t) + \hat{c}_j(t)\Big] \ . \tag{6.17}$$

To be brief, the estimation of $r_j(t)$ is supposed to be as optimistic as possible and then the item with the best optimistic estimate will be chosen.

As such, we formulate the regret in the vanilla stochastic multi-arm bandit setting as a simpler version of that indicated in (6.1):

$$R_T = \sum_{t=1}^{T} \Big(\mu_* - r_i(t)\Big) \ , \tag{6.18}$$

where u_* denotes the expected reward of the best item. Then [11] shows that after running the UCB based algorithms, with high probability:

$$R_T = \sum_{t=1}^{T} \Big(\mu_* - r_i(t)\Big) \leq \sum_{t=1}^{T} \Big(\hat{r}_i(t) + \hat{c}_i(t) - r_i(t)\Big)$$

$$\leq \sum_{t=1}^{T} \left(\hat{r}_i(t) + \hat{c}_i(t) - \Big(\hat{r}_i(t) - \hat{c}_i(t) \Big) \right) = 2 \sum_{t=1}^{T} \hat{c}_i(t) \ . \tag{6.19}$$

Confidence Intervals.

It is easy to show that through concatenating all feature vectors into a single "larger" one, the self-reward part of ISR can be treated as a special case of general linear stochastic bandit [1], which in each round chooses the item such that:

$$i(t) = \underset{j=1,\cdots,K}{\arg\max} \Big(\hat{\mathbf{p}}_t^{\intercal} \mathbf{q}_{t,j} + c \sqrt{\mathbf{q}_{t,j}^{\intercal} \mathbf{\Sigma}_t^{-1} \mathbf{q}_{t,j}} \Big) \ . \tag{6.20}$$

And the ellipsoid confidence interval for $\mathbf{p}$ is:

$$C_t = \{ \mathbf{p} \mid ||\mathbf{p} - \hat{\mathbf{p}}||_{\mathbf{\Sigma_t^{-1}}} \leq c \} \ , \tag{6.21}$$

where $||x||_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^{\intercal} \mathbf{\Sigma} \mathbf{x}}$. Given that $\mathbf{\Sigma}_t$ is a symmetric positive definite matrix and:

$$||\mathbf{p} - \hat{\mathbf{p}}_t||_{\mathbf{\Sigma}_t^{-1}} = \sqrt{(\mathbf{p} - \hat{\mathbf{p}}_t)^{\intercal} \mathbf{\Sigma}_t^{-1} (\mathbf{p} - \hat{\mathbf{p}}_t)} \ , \tag{6.22}$$

if we set $\mathbf{\Sigma}_t$ to be identity matrix, resulting in a norm-2 regularization on $\mathbf{p} - \hat{\mathbf{p}}_t$, then $\hat{\mathbf{p}}_t$ can be estimated through the standard ridge regression:

$$\hat{\mathbf{p}}_t = \underset{\mathbf{p}}{\text{argmin}} \sum_{t'=1}^{t-1} \Big(\hat{r}_i(t') - \mathbf{p}^{\intercal} \mathbf{q}_{t',i} \Big) + \lambda ||\mathbf{p}||^2 \ . \tag{6.23}$$

The corresponding regret is then measured as follows:

$$R_T = \sum_{t=1}^{T} \Big(\mathbf{p}_t^{\intercal} \mathbf{q}_{t,j^*} - \mathbf{p}_t^{\intercal} \mathbf{q}_{t,j} \Big) \ , \tag{6.24}$$

where $j^* = \underset{j=1,\cdots,K}{\arg\max}\ \mathbf{p}_t^{\intercal} \mathbf{q}_{t,j}$.

As a common setting, we follow the assumption that everything is Gaussian, e.g., the distribution D described in Section 6.3 follows a Gaussian distribution with μ and σ as mean and variance respectively. Thus from the solution of ridge regression, we have:

$$\Sigma_t = \lambda_p I + \sum_{t'=1}^{t} \mathbf{q}_{t',i} \mathbf{q}_{t',i}^{\intercal} \ , \tag{6.25}$$

making C_t in (6.21) a valid ellipsoid confidence set containing the true $\mathbf{p}$ with a very high probability controlled by c. Abbasi-Yadkori et al. [1] give a general condition on the use of valid confidence ellipsoid, which says if the linearity of true model and the independence of the rewards with R-sub-

Gaussian (with $R \geq 0$) hold, and $\mathbf{p}$ as well as $\mathbf{q}$ are bounded by some constants, i.e., $||\mathbf{p}|| \leq S$ and $||\mathbf{q}|| \leq L$, then for any $0 \leq \delta \leq 1$ and all $t \geq 0$, with probability at least $1 - \delta$, the true optimal value $\mathbf{p}_*$ lies in the following ellipsoid confidence set C_t:

$$\mathbf{p} \in \mathbb{R}^d : ||\mathbf{p} - \hat{\mathbf{p}}_t||_{\Sigma_t^{-1}} \leq R\sqrt{d \log\left(\frac{1 + tL^2/\lambda}{\delta}\right)} + \lambda^{\frac{1}{2}} S \, . \tag{6.26}$$

We refer readers to Theorem 2 in [1] for more details.

Therefore, applying (6.26) with R-sub-Gaussian tails on the noise, $\mathbf{p}$ and $\mathbf{q}$ upper bounded by S and L, C_t in (6.21) will be at most:

$$\mathcal{O}\Big(R\sqrt{d|\mathcal{I}| \log \frac{t}{\delta}} + \lambda^{\frac{1}{2}} S\Big) \, , \tag{6.27}$$

where d is the latent feature dimension and $|\mathcal{I}|$ is the number of candidate items.

Regret Bound.

Under the assumption that $\lambda \geq \max_{\mathbf{q}} ||\mathbf{q}||^2$ and based on the proof of Theorem 3 in [1], we can further write (6.19) as follows:

$$R_T \leq 2\sum_{t=1}^{T} c_i(t) = 2\sum_{t=1}^{T} c_t ||\mathbf{q}_{t,i}||_{\Sigma_t^{-1}} \leq 2\sqrt{\sum_{t=1}^{T} c_t^2 ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} \tag{6.28}$$

$$\leq 2\sqrt{c_T^2 \sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} = 2c_T \sqrt{\sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} \, , \tag{6.29}$$

where (6.28) is obtained by applying Cauchy-Schwarz inequality [1] and (6.29) is obtained based on the fact that c_t is monotonically increasing. Again, Abbasi-Yadkori et al. [1] prove that if $\lambda \geq \max_{\mathbf{q}} ||\mathbf{q}||^2$ holds, then:

$$\sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}} \leq 2 \log \det(\mathbf{\Sigma}_T) \leq \mathcal{O}\Big(d|\mathcal{I}| \log T\Big) \, . \tag{6.30}$$

Last, by putting (6.28) and (6.30) together, we have:

$$R_T \leq \mathcal{O}\Big(dRS|\mathcal{I}|\lambda^{\frac{1}{2}} \log\left(\frac{T}{\delta}\right)\sqrt{T}\Big) \, , \tag{6.31}$$

and if we further ignore the logarithmic factors and regards the latent feature dimension parameter d as a constant, then the regret of the self-reward part of ISR is at most $\mathcal{O}(\sqrt{T})$.

[1] https://en.wikipedia.org/wiki/Cauchy-Schwarz_inequality

6.5 Empirical Evaluation

In this section, we report the results of our experiments on three real-world public datasets and compare the performance of the proposed *Interactive Social Recommendation* (ISR) model with various baselines including bandit based interactive methods and non-bandit based offline methods in terms of different evaluation metrics.

6.5.1 Experimental Setup

Although an online experimental setting with real time user-system interactions is most appropriate for evaluations of different algorithms in this chapter, it is typically impossible to have such an environment in academic research [69]. Therefore, we follow the unbiased offline evaluation strategy for bandit alorithms proposed in [70] under the assumption that the user-system interactions (ratings) recorded in our experimental datasets are not biased by the recommender systems and these records can be regarded as unbiased user feedback in our experimental setting.

Table 6.1: Overview of datasets

	Flixster	*Douban*	*Epinions*
#users	76013	64642	10702
#items	48516	56005	39737
#ratings	7350235	9133529	482492
#ratings per user	96.70	141.29	45.08
#ratings per item	151.50	163.08	12.14
#social connections	1209962	1390960	219374

Datasets.

We use the following three real-world datasets, whose basic statistics are summarized in Table 6.1.

- *Flixster.* The Flixster dataset[2] containing information of user-movie ratings and user-user friendships from Flixster, an American social movie site for discovering new movies (`http://www.flixster.com/`).
- *Douban.* This public dataset[3] is extracted from the Chinese Douban movie forum (`http://movie.douban.com/`), which contains user-user friendships and user-movie ratings.
- *Epinions.* This is the popular consumer review dataset, Epinions[4], which consists of user-user trust relationships and user-item ratings from Epinions (`http://www.epinions.com/`).

[2] `http://www.cs.ubc.ca/~jamalim/datasets/`

[3] `https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban`

[4] `http://www.trustlet.org/wiki/Epinions_dataset`

For all datasets, we split the data into two user-disjoint sets : training set and test set. The test set is constructed by randomly choosing 200 users who have at least 120 ratings and 20 social connections, leaving the remaining users and their ratings in the training set.

Methods for Comparisons.

We compare ISR with several state-of-the-art approaches including three exploitation-exploration (i.e., MAB based) interactive methods (**ϵ-greedy, Softmax, LinUCB**), one non-interactive personalized social recommendation method (**STE**), one non-interactive personalized non-social recommendation method (**PMF**) and one non-interactive non-personalized non-social recommendation method (**Random**). Thus, the following seven recommendation methods, including six baselines, are tested.

- **ISR.** Our proposed ISR model, which is an interactive personalized social recommendation approach.

- **ϵ-greedy.** As is presented in (6.3), it is one of the most popular exploitation-exploration strategies in literature. In our problem setting, the expected reward of item i for user u at round t, $\hat{r}_{u,i}(t)$, is assumed to be estimated by the dot product of user latent feature vector at round t ($\mathbf{p}_{u,t}$) and item latent feature vector ($\mathbf{q}_j$). Thus the ϵ-**greedy** algorithm picks the item with the largest estimated reward based on the current knowledge with probability $1-\epsilon$ at round t:

$$i(t) = \underset{j=1,\cdots,K}{\arg\max}\ \hat{\mathbf{p}}_{u,t}^{\mathsf{T}}\mathbf{q}_j\ , \tag{6.32}$$

  and randomly picks an item with probability ϵ.

- **Softmax.** Another well-studied exploitation-exploration strategy described in (6.4), which is fitted into our problem setting through substituting $\hat{r}_{u,i}(t)$ with $\hat{\mathbf{p}}_{u,t}^{\mathsf{T}}\mathbf{q}_j$ (i.e., $\hat{r}_{u,i}(t) = \hat{\mathbf{p}}_{u,t}^{\mathsf{T}}\mathbf{q}_j$), in a similar way to ϵ-greedy.

- **Linear UCB (LinUCB).** Algorithm 2 in [129] where c is a tuning parameter, see equation (6.7) in section 6.3.

- **STE.** This is a personalized social recommendation method proposed by Ma et al. [77] which aggregates a user's own rating and her friends' ratings to predict the target user's final rating on an item.

- **PMF.** The classic personalized non-social probabilistic matrix factorization model first introduced in [82].

- **Random.** Randomly recommend unrated items to each user.

As is pointed out in section 6.2 that the three models proposed in [19, 105, 112] are designed for explicit features rather than latent features, resulting in different model formulations and exper-

Table 6.2: Cumulative precision and recall on test users (bold font highlights the winner).

	Flixster				**Douban**				**Epinions**			
	Cumulative Precision				*Cumulative Precision*				*Cumulative Precision*			
Round T	20	40	80	120	20	40	80	120	20	40	80	120
ϵ-greedy	0.6065	0.5358	0.4346	0.3689	0.6362	0.5588	0.4496	0.3804	0.6943	0.5749	0.4337	0.3537
Softmax	0.6138	0.5427	0.4385	0.3719	0.6380	0.5616	0.4510	0.3814	0.6967	0.5776	0.4351	0.3547
LinUCB	0.7798	0.6393	0.4792	0.3897	0.8073	0.6582	0.4918	0.3989	0.6884	0.5763	0.4399	0.3616
ISR	**0.8442**	**0.6824**	**0.5059**	**0.4088**	**0.8790**	**0.7094**	**0.5221**	**0.4203**	**0.7790**	**0.6379**	**0.4786**	**0.3899**
Imprv	8.26%*	6.74%*	5.57%*	4.90%	8.88%*	7.78%*	6.16%*	5.37%*	11.81%*	10.44%*	8.80%*	7.83%*
	Cumulative Recall				*Cumulative Recall*				*Cumulative Recall*			
Round T	20	40	80	120	20	40	80	120	20	40	80	120
ϵ-greedy	0.0960	0.1698	0.2745	0.3464	0.0792	0.1283	0.1957	0.2338	0.1030	0.1629	0.2437	0.2811
Softmax	0.0975	0.1720	0.2772	0.3494	0.0797	0.1287	0.1960	0.2345	0.1037	0.1634	0.2443	0.2818
LinUCB	0.1229	0.2019	0.3022	0.3658	0.1007	0.1516	0.2138	0.2451	0.1021	0.1632	0.2472	0.2869
ISR	**0.1333**	**0.2161**	**0.3195**	**0.3845**	**0.1097**	**0.1629**	**0.2270**	**0.2582**	**0.1150**	**0.1811**	**0.2694**	**0.3094**
Imprv	8.46%*	7.03%*	5.73%*	5.11%	8.94%*	7.45%*	6.17%*	5.35%*	10.90%*	10.83%*	8.98%*	7.84%*

imental settings from ours. This being the case, their work is orthogonal to ours and we are unable to compare ISR with these three models.

Evaluation Metrics.

We evaluate different models in two aspects: $1)$ recommending one single item in each round and $2)$ recommending multiple items in each round. If we only recommend a single item in each round, one straightforward measure is to count the number of hit (i.e., recommendation in which the recommended item has a rating that is no smaller than 4) after T rounds and average it by the number of users. Thus based on this methodology, we adopt two metrics, cumulative $Precision@T$ and cumulative $Recall@T$, for the evaluation in the scenario of single item recommendation per round.

- Cumulative $Precision@T$ ($Pre@T$).

$$Precision@T = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{T} \sum_{t=1}^{T} \theta_{hit} \ ,$$

  where $\theta_{hit} = 1$ if the rating of the target user u on the recommended item i in round t is equal to or higher than 4 and $\theta_{hit} = 0$ otherwise. $\mathcal{U}_{test}$ denotes those users in the test set.

- Cumulative $Recall@T$ ($Rec@T$).

$$Recall@T = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \sum_{t=1}^{T} \frac{\theta_{hit}}{|\mathcal{R}_u|} \ ,$$

  where $\mathcal{R}_u$ is the set of items that have been rated no less than 4 by user u in the test set.

When recommending multiple items in each round, the relative rankings of these candidate items become fairly important for the evaluation. Normalized Discounted Cumulative Gain ($NDCG$) is such a top-n recommendation measure suitable for this purpose. Let $S(u)$ be the set of all items rated by user u in the test set and $C(u)$ be the set of candidate items to be ranked in the test set for user u. We denote $R(u)$ as the ranking of items in $C(u)$ in a descending order, then for any item i in $S(u)$, its position in $R(u)$ is noted as $rank_i^u$.

- $NDCG$. In the context of recommender systems, $NDCG$ is defined as follows:

$$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG_u}{IDCG_u} \ ,$$

  where DCG and $IDCG$ (Ideal Discounted Cumulative Gain) are in turn defined as:

$$DCG_u = \sum_{i \in S(u)} \frac{1}{\log_2(rank_i^u + 1)} \ , \text{ and } \ IDCG_u = \sum_{i=1}^{|S(u)|} \frac{1}{\log_2(i+1)} \ .$$

Thus the $NDCG$ value for exploitation-exploration (MAB based) interactive methods will take the summation over all T rounds and then average on the number of total rounds. In our experiments, we test $NDCG@n$ (where $n = 3, 5$), indicating that $C(u)$ only contains items with top-n largest rating values from u.

6.5.2 Experimental Results

For exploitation-exploration (MAB based) algorithms including ϵ-greedy, Softmax, LinUCB and ISR, probabilistic matrix factorization is first used to train all the item latent feature vectors which will remain unchanged thereafter and be utilized to learn the user latent feature vectors (and the social weight coefficients for ISR) later. The winner in each column in Table 6.2 and Table 6.3 is highlighted in bold font, with * indicating that the corresponding result is significant by Wilcoxon signed-rank test at $p < 0.05$. We compare ISR with three interactive baselines in the scenario of recommending a single item in each round, then include three non-interactive baselines (i.e., STE, PMF, Randoms) and their variants as comparative partners in the scenario of recommending multiple items in each round.

Recommending a single item in each round.

In this evaluation scenario, up to 120 rounds of interactions are studied for each exploitation-exploration algorithm, given that each user in the test sets has at least 120 ratings. We compare the performance of our proposed ISR model with other three exploitation-exploration methods: ϵ-greedy, Softmax and LinUCB, in term of cumulative precision and recall. Table 6.2 presents the performances of all four approaches on all three datasets for $T =$ 20, 40, 80 and 120, with the last row showing the improvement of ISR over the best baseline. Clearly, the proposed ISR model outperforms all three exploitation-exploration baselines, with a trend towards a decreasing improvement as T becomes larger. Take cumulative precision as an example, as T increases from 20 to 120, the improvement of ISR over the best baseline decreases from 8.26% to 4.90% on Flixster, from 8.88% to 5.37% on Douban and from 11.81% to 7.83% on Epinions. One possible reason is that during the first several runs of the model when very little feedback is available, ISR model is capable of making much better recommendations than the baselines due to the benefit of taking social influences into consideration. On the other hand, these models will receive more and more feedback, which may increase their recommendation accuracy (especially for non-social exploitation-exploration baselines) as T increases, resulting in a less improvement for ISR against the baselines.

Recommending multiple items in each round.

In the scenario of recommending m ($m > 1$) items per round, we study up to $T = \frac{120}{m}$ rounds of interactions when evaluating each algorithm. In our experiments, we test the performance of different algorithms by setting $m = 3$ and $m = 5$ and study up to $T = 40$ and $T = 24$ rounds of interactions. Moreover, each of the two non-MAB based baselines (i.e., PMF and STE) is designed to have three variants: *-os* (short for out of sample), *-half* and *-all*. For variant *-os*, we train the model on the training set and test its performance on the test set. Note that as the training set and

Table 6.3: NDCG@n for ϵ-greedy, Softmax, LinUCB and ISR on three datasets (bold font highlights the winner).

	Flixster				**Douban**				**Epinions**			
	NDCG@3		*NDCG@5*		*NDCG@3*		*NDCG@5*		*NDCG@3*		*NDCG@5*	
Round T	20	40	12	24	20	40	12	24	20	40	12	24
ϵ-greedy	0.2398	0.2793	0.2267	0.2894	0.2646	0.3260	0.2932	0.3668	0.1503	0.1769	0.1559	0.1918
Softmax	0.2421	0.2859	0.2197	0.2739	0.2588	0.3202	0.2906	0.3617	0.1454	0.1733	0.1479	0.1888
LinUCB	0.2537	0.2838	0.2403	0.2865	0.3325	0.3692	0.3269	0.3780	0.1516	0.1762	0.1546	0.1943
ISR	**0.2802**	**0.3197**	**0.2657**	**0.3250**	**0.3510**	**0.3949**	**0.3490**	**0.4060**	**0.1640**	**0.1999**	**0.1629**	**0.2098**
Imprv	10.45%*	11.82%*	10.57%*	12.30%*	5.56%*	6.96%*	6.76%*	7.41%*	8.18%	13.00%*	4.49%	7.98%*

test set are user-disjointed, users in the test set will never appear in the training set (i.e., out of sample), which may result in very poor performance for non-MAB based models. As for the other two variants, we randomly select η ratings to train the user latent feature vector for each user u in the test set. We set η to be the number of observable ratings during the first $\frac{T}{2}$ rounds in the test set for the *-half* variant and be the number of all available ratings in the test set for the *-all* variant. In other words, the - *all* variant is trained on all available observations in the test set, indicating the best solution we can obtain and the performance of *-half* should intuitively lie between *-all* and *-os*. In Figure 6.1, Figure 6.2 and Figure 6.3, we can see seven straight horizontal lines (they are straight because these non-MAB based off-line models do batch trainings and have nothing to do with the rounds of interactions) in each of the six sub-figures, representing the Random baseline (the lowest one) as well as the three variants for each of PMF and STE: PMF-all, PMF-half, PMF-os and STE-all, STE-half, STE-os. It is easy to observe that PMF-half lies between PMF-all and PMF-os, and similarly STE-half lies between STE-all and STE-os, which verifies our assumptions above. On the other hand, LinUCB and ISR which can be regarded as the exploitation-exploration (MAB based) versions of PMF and STE to some extent, start with very poor performance , gradually get improved when receiving more and more feedback in rounds of interactions and closely approach PMF-all and STE-all respectively in round 120. For both $NDCG@3$ and $NDCG@5$ on all three datasets, the *-half* baselines outperform their MAB based algorithms (LinUCB and ISR) in early rounds before being surpassed by their exploitation-exploration counterparts soon after. This is reasonable since the *-half* variant can get access to a portion of the observations in the test set to learn the user preferences, but when more user feedback is available the MAB based algorithm gets improved through dynamically adapting to user feedback and finally reaches a comparable performance with the *-all* variant. Besides, our proposed ISR outperforms LinUCB which does not utilize social information, through the benefit of taking social influences from friends into account and adaptively learning weights for these friends. In addition to LinUCB, we also compare ISR with other exploitation-exploration baselines including ϵ-greedy and Softmax, whose results are list in Table 6.3. With no surprise, we observe that ISR beats both of them in all cases .

Impact of controlling parameter α.

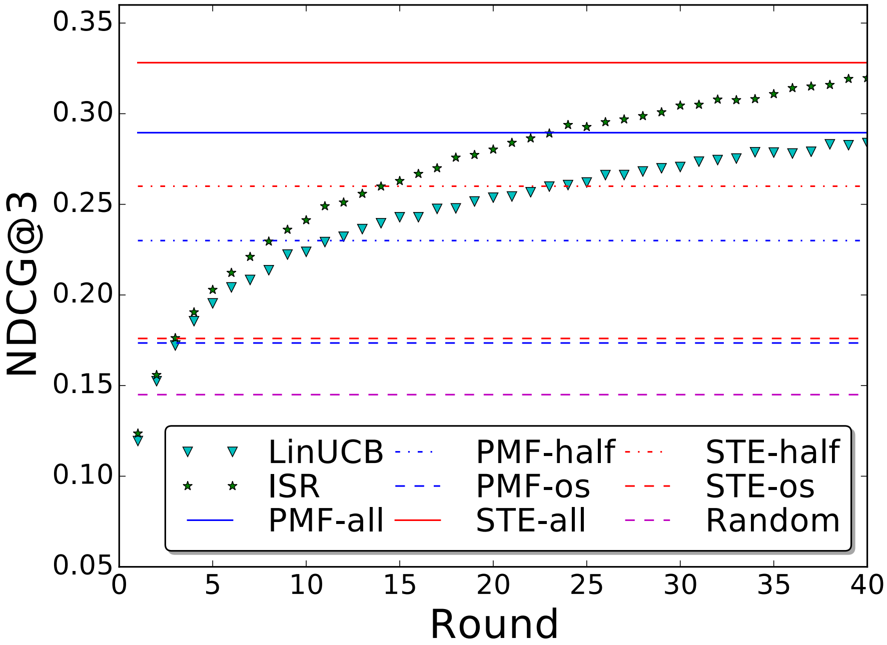
As a controlling parameter, α balances the target user's own preferences and the tastes of her friends. It controls the extent to which ISR should trust the target user' own interests and how much the model should emphasis on the tastes of her friends. In two extreme cases, ISR will only consider the target user's own preferences without any social influences when α is set to 1 and merely take the preferences of the target user's friends into account when α is set to 0. With α being set to other real values between 1 and 0, ISR will take both the target user's and her friends' interests into consideration when making recommendations. Figure 6.4, Figure 6.5 and Figure 6.6 show the impact of α on both cumulative precision and recall for all three datasets. We observe that the optimal α equals to 0.4 on Flixster and Epinions, and equals to 0.5 on Douban, which confirms the efficacy of fusing favors of the target user and her friends together in improving the recommendation accuracy. Moreover, each of the plots in Figure 6.4, Figure 6.5 and Figure 6.6 looks analogous to a parabolic shape for both cumulative precision and recall on all datasets, indicating that α with either a larger or smaller value than the optimal one may cause a decline in the performance of the algorithm. In other words, it is necessary to find a good balance between the tastes of the target users and their friends — leaning too much against either of them may result in suboptimal recommendations.

Learning the edge weights.

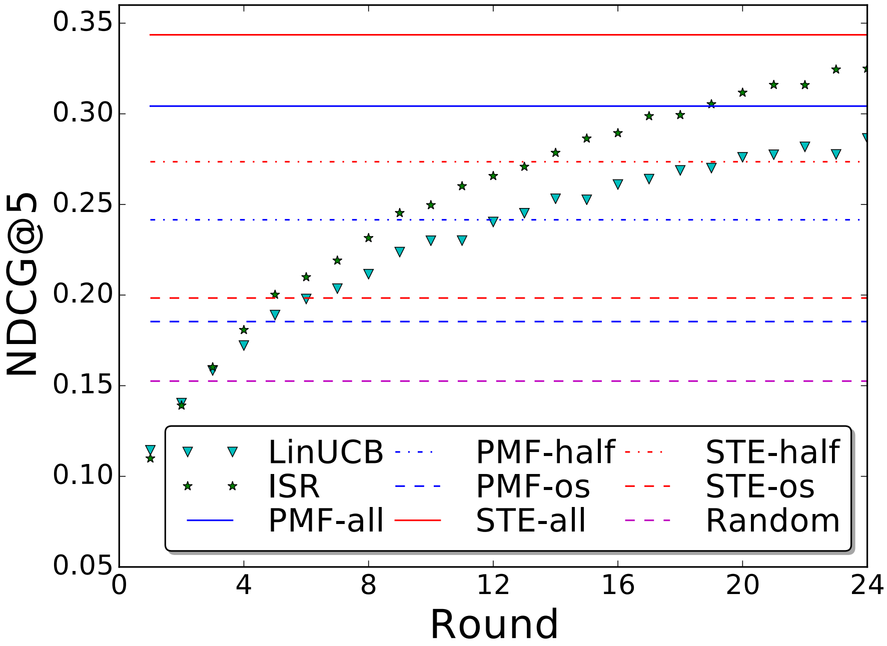
Last but not least, we also present some statistics on the learned edge weights by ISR. As discussed in section 6.4, we adopt the normalized edge weights so that the initial edge weights depend on the number of friends for each user (i.e., initial weights are equally set to $\frac{1}{|N_u|}$ for all edges of user u). Thus we show the relative changes in edge weights with respect to their initial values after 120 rounds of ISR in Figure 6.7, where positive bin values on X axis indicate relative increases and negative ones indicate relative decreases. We observe that weights of 1829 edges in Flixster, 2117 edges in Douban and 2075 edges in Epinions are updated during the 120 rounds where most of them have a relative change (either increase or decrease) between -80% and 80% of their initial values, demonstrating the necessity of learning the edge weights.

6.6 Summary

In this chapter, we propose a novel interactive social recommendation model (ISR) with online learning, which can not only dynamically adapt itself based on user feedback but also adaptively learn different weights for different friends in social networks. We employ the similar idea of multi-armed bandit (MAB) strategy for the interactive learning procedure and analyze the regret bound of our proposed ISR model. We evaluate the performance of the proposed ISR model and compare with various baselines including MAB based algorithms and non-MAB based ones in terms of cumulative precision, cumulative recall and $NDCG@n$ on three real-world datasets, demonstrating the advantages of ISR against these state-of-the-art approaches.

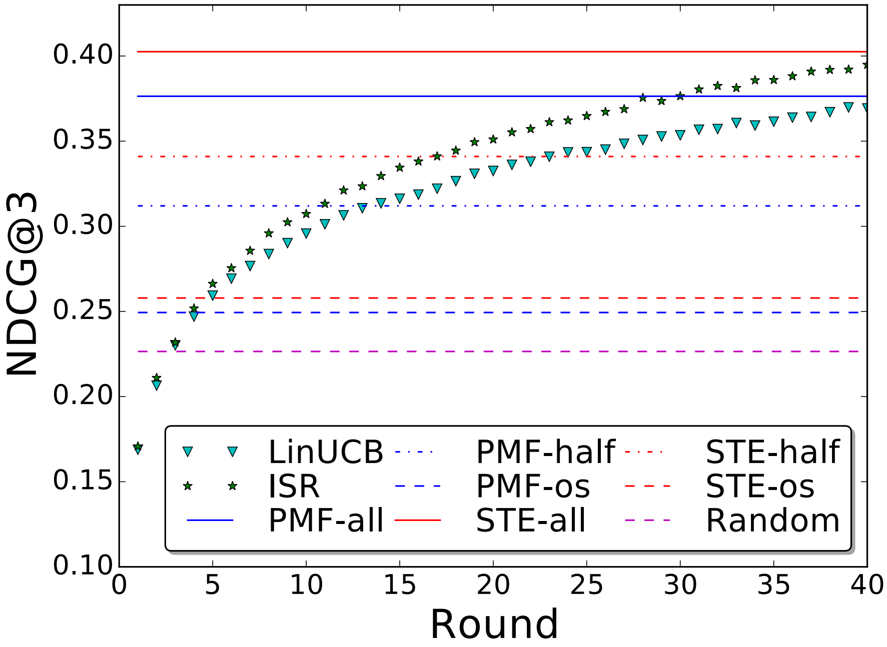


(a) NDCG@3

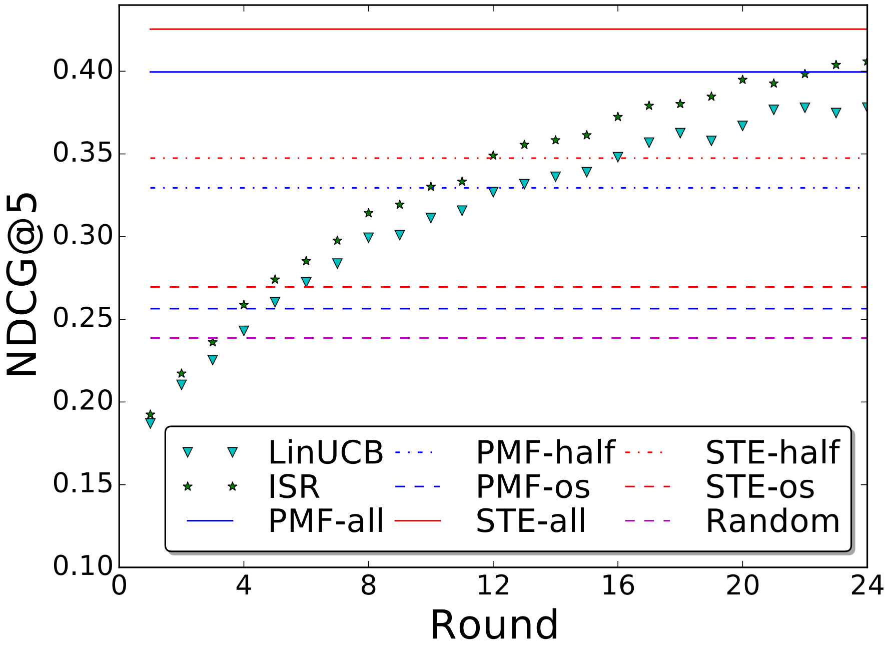


(b) NDCG@5

Figure 6.1: NDCG@3 and NDCG@5 for Random, LinUCB, ISR, PMF and STE as well as their variants on Flixster

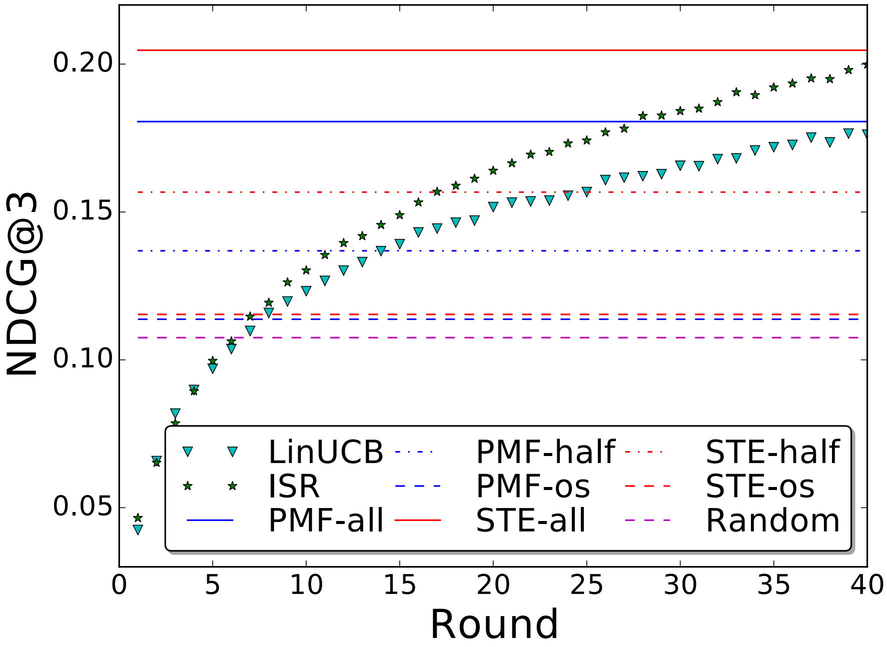


(a) NDCG@3

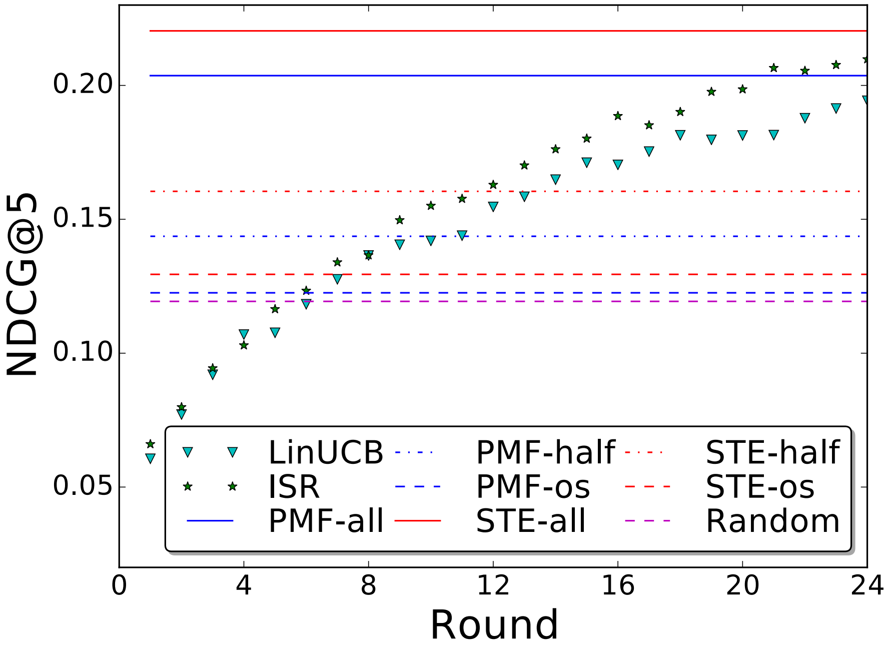


(b) NDCG@5

Figure 6.2: NDCG@3 and NDCG@5 for Random, LinUCB, ISR, PMF and STE as well as their variants on Douban

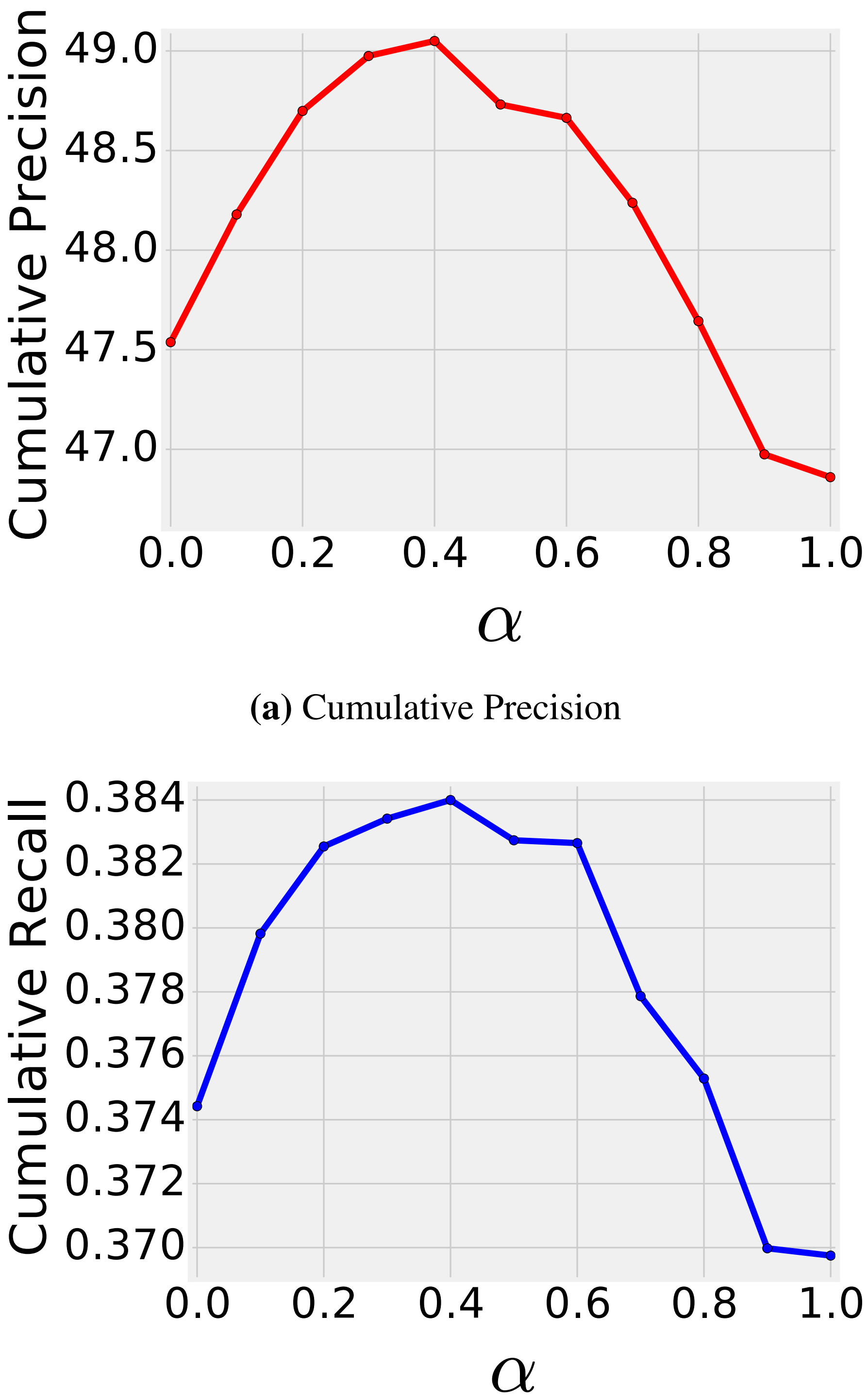


(a) NDCG@3



(b) NDCG@5

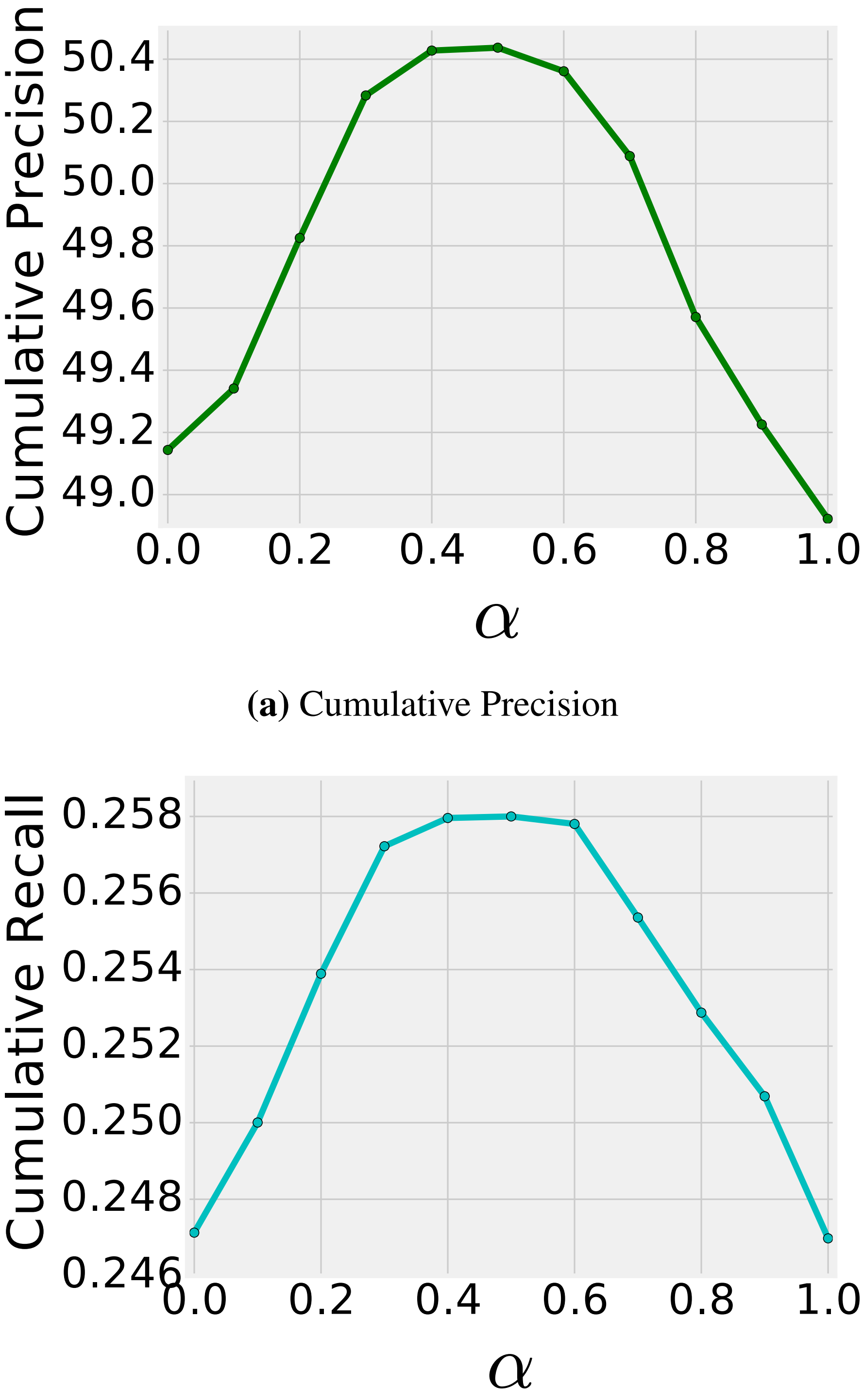
Figure 6.3: NDCG@3 and NDCG@5 for Random, LinUCB, ISR, PMF and STE as well as their variants on Epinions



(a) Cumulative Precision

(b) Cumulative Recall

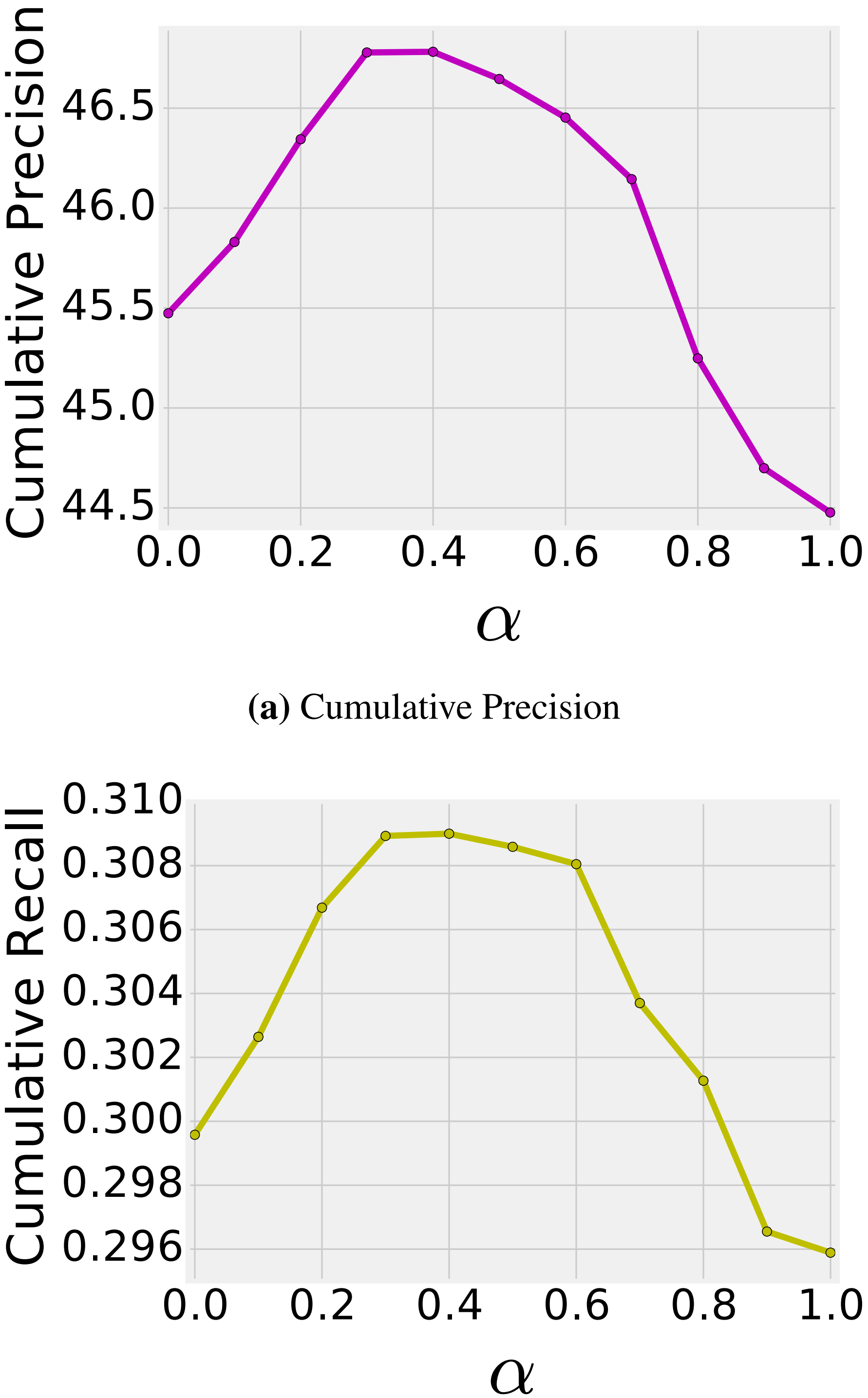
Figure 6.4: Impact of different α values in ISR on cumulative precision and recall for Flixster in round 120



(a) Cumulative Precision

(b) Cumulative Recall

Figure 6.5: Impact of different α values in ISR on cumulative precision and recall for Douban in round 120



(a) Cumulative Precision

(b) Cumulative Recall

Figure 6.6: Impact of different α values in ISR on cumulative precision and recall for Epinions in round 120

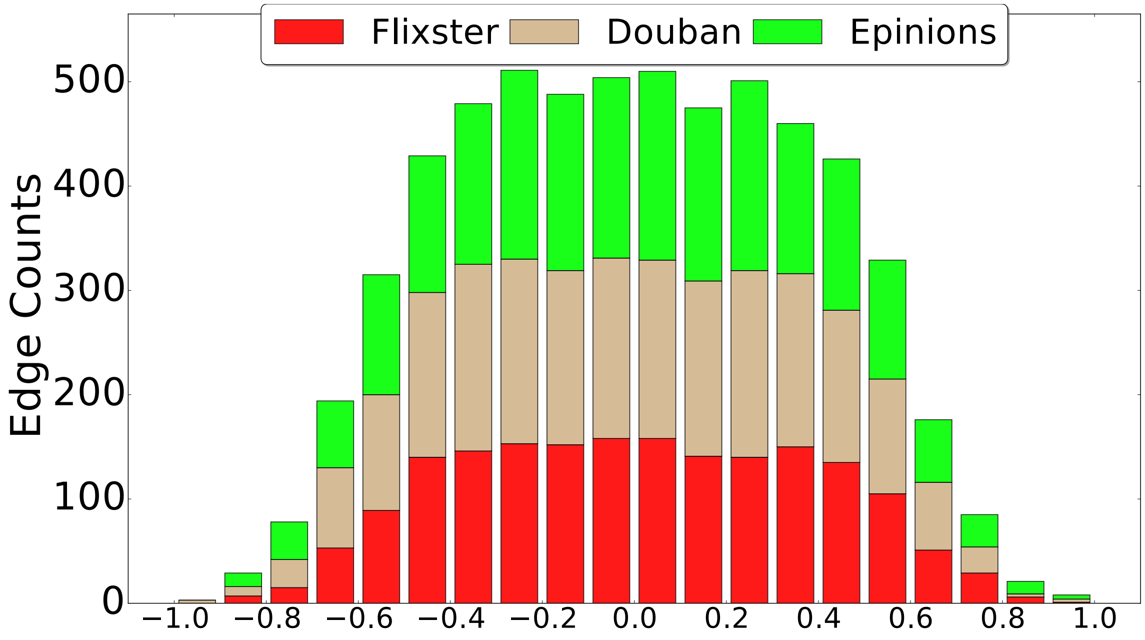


Figure 6.7: Relative edge weight changes after round 120

Chapter 7

Conclusion

Data mining in recommender systems has many practical applications in our real lives, making it a very attractive research topic in both industry and academia. With the advent of various web services and applications on the Internet, a large amount of social media data becomes available. Due to the data driven essence of recommendation, it is the boost in available social media data that leads to the promising progress in a variety of recommendation research in social media. Different from conventional recommendation without social media which suffers from data sparsity, recommendation in social media is able to benefit from utilizing extra information such as social connections (friendships), user-group engagement and user-item interactions in social media to enhance the performance of recommender systems.

In this thesis, we first give a general problem definition for recommender systems, then present some background knowledge on explicit and implicit feedback in recommendation followed by an overview of new challenges for applications of recommendation in social media brought by the rapid development of online web services in Chapter 1. We point out that exploring various user relationships can be one of the most convenient and effective ways to help improve recommendation accuracy for applications in social media, which is verified by the success of existing social recommendation methods. We pick three representative applications of recommendation in social media for investigation, i.e., strong and weak ties in recommendation, social group recommendation and interactive social recommendation, We discuss the existing literature related to each of the three topics in Chapter 2.

In Chapter 3, we propose to study the effects of distinguishing strong and weak ties in social recommendation, given that little attention has been paid to the important distinctions between strong and weak ties, two well-documented notions in social sciences [43, 44]. We incorporate the notions of strong and weak ties into the Bayesian Personalized Ranking (BPR) framework and learn the optimal threshold w.r.t. recommendation accuracy for distinctions of strong and weak ties [110]. In Chapter 4, we further bring the concepts of strong and weak ties to Probabilistic Matrix Factorization (PMF) framework through proposing the novel PTPMF model [109]. Besides learning the optimal (w.r.t. recommendation accuracy) threshold and other model parameters simultaneously, our proposed PTPMF model is also capable of obtaining a personalized tie type preference for each individual at the same time. In short, we introduce the distinctions of strong and weak ties to BPR and PMF, two popular state-of-the-art recommendation framework, and demonstrate the efficacy of our proposed methods against several baselines in Chapter 3 and Chapter 4.

In Chapter 5, we focus on the problem of recommending groups of people to users. Since groups in social media consist of different users, the characteristics of groups will change as new users join in or old users leave, making group properties dynamic in the course of time. On the other hand, user preference also tends to change as time goes by, making discovering relevant groups for users become time dependent, which poses new challenges to the problem of social group recommendation. Thus we present a time dependent matrix factorization approach [108] to model the temporal dynamics between users and groups in the course time so that the recommendation of social groups to users can become more accurate.

We switch our research attention from offline setting to online setting in Chapter 6. The motivation is that most existing social recommendation models utilize a batch learning based strategy which trains the model in an offline manner from a collection of historical user interaction data with the recommender systems. We argue that this in practice may run the risk of making new users leave the systems for being recommended with items they don not like before a sufficient amount of data can be collected to train a good off-line model, which results in an inefficient customer retention. Therefore, we borrow the ideas from the exploitation-exploration (E-E) methodology and employ the multi-armed bandit (MAB) technique to handle social recommendation problem within online setting. The proposed interactive social recommendation (ISR) model can not only simultaneously explore user preferences and exploit the effectiveness of personalization in an interactive way, but adaptively learn different weights for different friends as well.

Last but not least, we end up this thesis by pointing out several promising directions for future investigations:

Future work directly related to topics in this thesis

- **Strong and weak ties in recommendation with a personalized threshold for classifying strong and weak ties**
  One interesting direction for future work is to find a personalized threshold for classifying strong and weak ties for each user, though it can be challenging due to the sparsity of data. Further, we did not examine other node similarity metrics such as Adamic-Adar [2] or Katz [53] in this thesis and it is also quite interesting to explore different node similarity metrics.

- **Recommendation with complex long term relationships**
  In addition to the problem of social group recommendation in this thesis, it is worth trying to investigate methods for other recommendation problems (e.g., job recommendation)

which involve more complex long term relationships between users and the items being recommended.

- **Incremental social information and popular users for interactive social recommendation**
  Despite the promising results obtained, some open issues remain unsolved in Chapter6. First of all, some users might get new friends during the interactions, which will lead to the problem of incremental social information. Second, there always exist popular users who have lots of friends, making the exploration space considerably huge. As such, it will be quite interesting and challenging to investigate these two problems.

Other open problems for future research

- **Domain aware social recommendation**
  In spite of the huge success of social recommendation in alleviating the data sparsity in recommender systems, there still exist some open questions that deserve our serious ponders. Current social recommendation methods tend to improve traditional recommender systems via taking social influence and social selection among friends in to account when making recommendations. They treat all social ties (friends) of an individual equally in every knowledge domain, which can not be true in reality because we may have different degrees of trust for our friends in different knowledge domains. For instance, let us assume Alice has two friends, Tom and David. Tom is an expert in computer programming while David is a classical music enthusiast. It is obvious that Alice should trust Tom more when she wants to get some suggestions in C++ programming and trust David more when picking some pieces of symphony for listening. We believe that developing a domain aware social recommendation approach will certainly boost the performance of existing state-of-the-art social recommendation methods.

- **Reverse engineering for recommendation**
  Another interesting application is the reverse recommendation problem. Take Netflix as an example, Netflix has planned to produce its own TV series (such as House of Cards) and movies. Then a natural question is that what types of TV series and movies can become popular and attract lots of attentions from its users. Moreover, if we regard each TV series and movies as a combination of different features consisting of genres, actors, directors etc., the problem becomes that TV series or movies containing which combinations of features are able to achieve high audience ratings. In the problem of reverse recommendation, we want to create new items with a combination of known features such that these new items can become popular among the potential consumers. This is a meaningful practical problem as solving it can not only increase a company's revenue but also present enjoyable products such as films and television programs to the public.

Bibliography

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[4] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 19–28. ACM, 2009.

[5] Deepak Agarwal, Bee-Chung Chen, and Pradheep Ela. Explore/exploit schemes for web content optimization. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 1–10. IEEE, 2009.

[6] Irfan Ali and Sang-Wook Kim. Group recommendations: approaches and evaluation. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, page 105. ACM, 2015.

[7] Kareem Amin, Michael Kearns, and Umar Syed. Graphical models for bandit problems. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

[8] Valerio Arnaboldi, Andrea Guazzini, and Andrea Passarella. Egocentric online social networks: Analysis of key features and prediction of tie strength in facebook. *Computer Communications*, 36(10):1130–1144, 2013.

[9] Peter Auer. Using upper confidence bounds for online learning. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 270–279. IEEE, 2000.

[10] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[11] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[12] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

[13] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010.

[14] Shlomo Berkovsky, Jill Freyne, and Gregory Smith. Personalized network updates: increasing social interactions and contributions in social networks. In *User Modeling, Adaptation, and Personalization*, pages 1–13. Springer, 2012.

[15] Donald A Berry and Bert Fristedt. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer, 1985.

[16] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.

[17] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *International Conference on Neural Information Processing*, pages 324–331. Springer, 2012.

[18] Peter J Brockwell and Richard A Davis. *Introduction to Time Series and Forecasting*, volume 1. Taylor & Francis, 2002.

[19] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

[20] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[21] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th International Conference on World Wide Web*, pages 681–690. ACM, 2009.

[22] Wen-Yen Chen, Dong Zhang, and Edward Y Chang. Combinational collaborative filtering for personalized community recommendation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 115–123. ACM, 2008.

[23] Nicholas A Christakis and James H Fowler. Connected: how your friends' friends' friends affect everything you feel, think, and do. *New York, NY: Little, Brown, and Company*, 2009.

[24] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

[25] Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. Modeling temporal adoptions using dynamic matrix factorization. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 91–100. IEEE, 2013.

[26] Paul G Constantine and David F Gleich. Tall and skinny QR factorizations in MapReduce architectures. In *Proceedings of the Second International Workshop on MapReduce and its Applications*, pages 43–50. ACM, 2011.

[27] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 39–46. ACM, 2010.

[28] Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW*, pages 271–280, 2007.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[30] DeviantArt. `http://www.deviantart.com/`.

[31] Douban. `http://www.douban.com/`.

[32] David Easley and Jon M. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, 2010.

[33] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

[34] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[35] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[36] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[37] Simon Funk. Netflix update: Try this at home.

[38] Laura K Gee, Jason Jones, Moira Burke, et al. Social networks and labor markets: How strong ties relate to job finding on facebookâĂŹs social network. *Journal of Labor Economics*, 35(2):485–518, 2017.

[39] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *ICML*, pages 757–765, 2014.

[40] Eric Gilbert. Predicting tie strength in a new medium. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1047–1056. ACM, 2012.

[41] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 211–220. ACM, 2009.

[42] John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979.

[43] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[44] Mark S Granovetter. *Getting a job: A study of contacts and careers.* University of Chicago Press, 1974.

[45] G. Guo, J. Zhang, and N. Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2619–2625, 2013.

[46] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton University Press, 1994.

[47] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[48] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

[49] Paul Jaccard. *Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines*. Rouge, 1901.

[50] Mohsen Jamali and Martin Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[51] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[52] Indika Kahanda and Jennifer Neville. Using transactional information to predict link strength in online social networks. *ICWSM*, 9:74–81, 2009.

[53] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[54] Andrea L Kavanaugh, Debbie Denise Reese, John M Carroll, and Mary Beth Rosson. Weak ties in networked communities. *The Information Society*, 21(2):119–131, 2005.

[55] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, pages 1297–1305, 2015.

[56] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[57] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.

[58] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[59] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[60] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[61] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, pages 173–182. ACM, 2012.

[62] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*, 2014.

[63] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[64] John Langford, Alexander Strehl, and Jennifer Wortman. Exploration scavenging. In *Proceedings of the 25th international conference on Machine learning*, pages 528–535. ACM, 2008.

[65] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[66] Neal Lathia, Stephen Hailes, and Licia Capra. Temporal collaborative filtering with adaptive neighbourhoods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 796–797. ACM, 2009.

[67] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[68] Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. Unsupervised streaming feature selection in social media. In *Proceedings of the 24th ACM International Conference on Conference on Information and Knowledge Management*, pages 1041–1050. ACM, 2015.

[69] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[70] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

[71] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *The 39th International ACM SIGIR Conference on Information Retrieval (SIGIR)*, 2016.

[72] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. Exploitation and exploration in a performance based contextual advertising system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 27–36. ACM, 2010.

[73] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[74] Chenghao Liu, Tao Jin, Steven CH Hoi, Peilin Zhao, and Jianling Sun. Collaborative topic regression for online recommender systems: an online and bayesian approach. *Machine Learning*, 2017.

[75] Wei Lu, Stratis Ioannidis, Smriti Bhagat, and Laks VS Lakshmanan. Optimal recommendations under attraction, aversion, and social influence. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 811–820. ACM, 2014.

[76] R Duncan Luce. Individual choice behavior, a theoretical analysis. *Bull. Amer. Math. Soc. 66 (1960), 259-260 DOI: http://dx. doi. org/10.1090/S0002-9904-1960-10452-1 PII*, pages 0002–9904, 1960.

[77] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[78] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29, 2011.

[79] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[80] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[81] Aditya Mahajan and Demosthenis Teneketzis. Multi-armed bandit problems. In *Foundations and Applications of Sensor Management*, pages 121–151. Springer, 2008.

[82] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[83] Douglas Oard and Jinmook Kim. Implicit feedback for recommender systems. In *AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[84] Mark O'Connor, Dan Cosley, Joseph A Konstan, and John Riedl. Polylens: a recommender system for groups of users. In *ECSCW 2001*, pages 199–218. Springer, 2001.

[85] J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

[86] Róbert Pálovics, András A Benczúr, Levente Kocsis, Tamás Kiss, and Erzsébet Frigó. Exploiting temporal influence in online recommendation. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 273–280. ACM, 2014.

[87] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 502–511. IEEE, 2008.

[88] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, pages 2691–2697. AAAI Press, 2013.

[89] Katrina Panovich, Rob Miller, and David Karger. Tie strength in question & answer on social network sites. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 1057–1066. ACM, 2012.

[90] Andrea Petróczi, Tamás Nepusz, and Fülöp Bazsó. Measuring tie-strength in virtual social networks. *Connections*, 27(2):39–52, 2007.

[91] Sanjay Purushotham, Yan Liu, and C-C Jay Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. 2012.

[92] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *SDM*, pages 461–469. SIAM, 2014.

[93] Ray Reagans. Preferences, identity, and competition: Predicting tie strength from demographic data. *Management Science*, 51(9):1374–1383, 2005.

[94] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282. ACM, 2014.

[95] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[96] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[97] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[98] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[99] Aleksandrs Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[100] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 2011.

[101] Liang Tang, Yexi Jiang, Lei Li, and Tao Li. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 73–80. ACM, 2014.

[102] Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, and Tao Li. Personalized recommendation via parameter-free contextual bandits. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 323–332. ACM, 2015.

[103] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232. ACM, 2014.

[104] Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, and Inderjit S Dhillon. Affiliation recommendation using auxiliary networks. In *Proceedings of the fourth ACM conference on Recommender Systems*, pages 103–110. ACM, 2010.

[105] Huazheng Wang, Qingyun Wu, and Hongning Wang. Factorization bandits for interactive recommendation. 2017.

[106] Jialei Wang, Steven CH Hoi, Peilin Zhao, and Zhi-Yong Liu. Online multi-task collaborative filtering for on-the-fly recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 237–244. ACM, 2013.

[107] Jingdong Wang, Zhe Zhao, Jiazhen Zhou, Hao Wang, Bin Cui, and Guojun Qi. Recommending flickr groups with social topic model. *Information retrieval*, 15(3-4):278–295, 2012.

[108] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[109] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International World Wide Web Conference*. WWW, 2017.

[110] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 5–14. ACM, 2016.

[111] Anna Wu, Joan M DiMicco, and David R Millen. Detecting professional versus personal closeness using an enterprise social network site. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1955–1964. ACM, 2010.

[112] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538. ACM, 2016.

[113] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng 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, 2010.

[114] Rongjing Xiang, Jennifer Neville, and Monica Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990. ACM, 2010.

[115] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G Schneider, and Jaime G Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *SDM*, volume 10, pages 211–222. SIAM, 2010.

[116] Danqing Xu, Yiqun Liu, Min Zhang, Shaoping Ma, Anqi Cui, and Liyun Ru. Predicting epidemic tendency through search behavior analysis. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2361. Citeseer, 2011.

[117] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. Social collaborative filtering by trust. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2747–2753. AAAI Press, 2013.

[118] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.

[119] Yi Yang and Hui Zou. A cocktail algorithm for solving the elastic net penalized Cox's regression in high dimensions. *Statistics and its Interface*, 6(2):167–173, 2012.

[120] Mao Ye, Xingjie Liu, and Wang-Chien Lee. Exploring social influence for recommendation: a generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 671–680. ACM, 2012.

[121] Zhi Yu, Can Wang, Jiajun Bu, Xin Wang, Yue Wu, and Chun Chen. Friend recommendation with content spread enhancement in social networks. *Information Sciences*, 309:102–118, 2015.

[122] Yisong Yue, Sue Ann Hong, and Carlos Guestrin. Hierarchical exploration for accelerating contextual bandits. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[123] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. Online context-aware recommendation with time varying multi-arm bandit. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[124] Wei Zeng and Li Chen. Recommending interest groups to social media users by incorporating heterogeneous resources. In *Recent Trends in Applied Artificial Intelligence*, pages 361–371. Springer, 2013.

[125] Chenyi Zhang, Ke Wang, Hongkun Yu, Jianling Sun, and Ee-Peng Lim. Latent factor transition for dynamic collaborative filtering. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 2014.

[126] Chenyi Zhang, Xueyi Zhao, Ke Wang, and Jianling Sun. Content+ attributes: A latent factor model for recommending scientific papers in heterogeneous academic networks. In *Advances in Information Retrieval*, pages 39–50. Springer, 2014.

[127] Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1373–1383. International World Wide Web Conferences Steering Committee, 2015.

[128] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 261–270. ACM, 2014.

[129] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1411–1420. ACM, 2013.

[130] Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.

[131] Jinfeng Zhuang, Tao Mei, Steven CH Hoi, Xian-Sheng Hua, and Shipeng Li. Modeling social strength in social media community via kernel-based learning. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 113–122. ACM, 2011.

[132] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32th Conference on Uncertainty in Artificial Intelligence*, 2016.

[133] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.