Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering

Tong Zhao[1,2], Julian McAuley[3], Irwin King[1,2]
[1]Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research Institute,
The Chinese University of Hong Kong, Shenzhen, China
[2]Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[3]Department of Computer Science and Engineering, UC San Diego, La Jolla, CA, USA
{tzhao, king}@cse.cuhk.edu.hk, julian.mcauley@gmail.com

ABSTRACT

Recommending products to users means estimating their preferences for certain items over others. This can be cast either as a problem of estimating the *rating* that each user will give to each item, or as a problem of estimating users' relative preferences in the form of a *ranking*. Although collaborative-filtering approaches can be used to identify users who rate and rank products similarly, another source of data that informs us about users' preferences is their set of *social connections*. Both rating- and ranking-based paradigms are important in real-world recommendation settings, though rankings are especially important in settings where explicit feedback in the form of a numerical rating may not be available. Although many existing works have studied how social connections can be used to build better models for *rating* prediction, few have used social connections as a means to derive more accurate *ranking-based* models. Using social connections to better estimate users' rankings of products is the task we consider in this paper. We develop a model, *SBPR* (Social Bayesian Personalized Ranking), based on the simple observation that users tend to assign higher ranks to items that their friends prefer. We perform experiments on four real-world recommendation data sets, and show that *SBPR* outperforms alternatives in ranking prediction both in warm- and cold-start settings.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Data Mining; J.4 [**Social and Behavioral Sciences**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Social Networks, Recommender Systems, Personalized Ranking

1. INTRODUCTION

Recommending products to users requires not only that we understand their *preferences*, but also that we understand their *interactions* with others. Just as a person's preferences can be infered from the products they rate, view, and buy, their preferences can also be infered from the products their *friends* rate, view, and buy.

In fact, feedback from a user's friends may be even *more* informative than a user's own feedback: by considering all of a user's friends, we can draw upon a much larger volume of data than we can by looking at a user in isolation. This is especially true in "cold-start" settings, i.e., when making recommendations to new users. If a user signs-on to a recommendation service that is connected to a social network, we have access to their social connections—and possibly even their friends' preferences—even before the user has performed any actions.

The goal of this paper is to leverage such social connections in order to build better models of users' preferences. Many existing works aim to use social connections in order to predict the ratings that each user will give to each item; a commonly used assumption is that each user's rating behavior should be somehow similar to that of their friends. However, in many real applications, explicit numerical ratings might not be available and one must instead try to model some form of implicit feedback, such as the media they consume, the pages they browse, the music they listen to, or whom they befriend [4, 21]. This setting is called "one-class" recommendation and a variety of solutions have been proposed to solve it by directly modeling relative preferences, or *rankings*, of items for personalized recommendation [8, 12, 18, 22].

Although a number of rating estimation methods have been proposed that leverage social information, few works have made use of social information for one-class recommendation problems. In [2], Du et al. extend Bayesian Personalized Ranking (*BPR*) [15] by adding a social regularization term. *MR-BPR* [8], a state-of-the-art method for one-class recommendation, simultaneously models users' preference on items and their social relations. Pan et al. [13] present *GBPR* ("Group-based" BPR), a method that aggregates groups of users' preferences on items to reduce modeling uncertainty and improve recommendation accuracy. Nevertheless, none of these works consider how feedback from users' friends can be used to model their preference ranking of items.

In this paper, we try to understand the underlying mechanism of how users' preferences for certain items are revealed by the items chosen by their friends. To motivate this work, we first conduct a simple analysis of preference data from four sources that we will
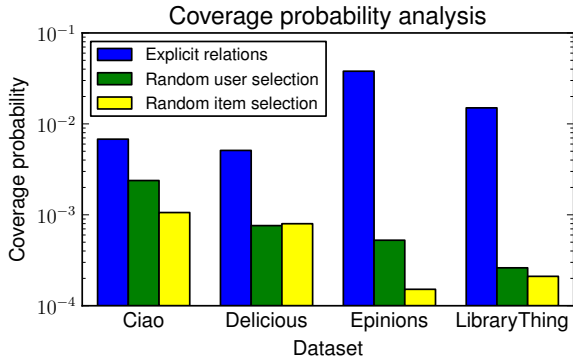
Figure 1: Coverage Analysis. The Y-axis indicates the probability that a user's friends select an item that they have selected. Users are more likely to select items selected by their friends.

Figure 2: Users become monotonically more likely to select an item as more of their friends select it.

consider in this paper: Ciao[1], Delicious[2], LibraryThing[3] and Epinions[4]. All four sources consist of preference data in addition to explicit social connections. First, Figure 1 shows the probability that an item selected (e.g. purchased/viewed/reviewed) by a user is also selected by their friends. To make the results more clear, we compare the probability with two baseline settings: the probability that an item selected by a user is also selected by randomly sampled users and the probability that an item selected by a user belongs to a randomly sampled item set. In all cases it is clear that the first probability is an order of magnitude higher than in the other two cases. Further to this, Figure 2 shows that the probability that a user selects an item increases monotonically as a function of the number of friends who have selected the item, implying a model where each friend's selections contribute independently to the probability that a user selects an item. Based on this idea, we build a model based on a simple assumption about rankings:

$$(\text{Rank of}) \text{ items I've consumed} \; >$$
$$\text{items my friends have consumed} \; >$$
$$\text{items neither me nor my friends have consumed.}$$

From this we build a simple Bayesian model of users' ranked preferences over items.

We summarize our contributions as follows:

1. We develop a ranking algorithm, called *Social-BPR* or *SBPR*, that incorporates our model assumptions at training time by leveraging social information for training instance selection.

2. We evaluate the proposed method on four real-world datasets (two of which we introduce), and empirical results show that the proposed model can significantly improve item recommendation performance compared to state-of-the-art alternatives.

3. We report experiments on cold-start recommendation problems, and again report significant improvements over state-of-the-art alternatives in cold-start one-class recommendation problems.

---

[1] www.ciao.co.uk
[2] www.delicious.com
[3] www.librarything.com
[4] www.epinions.com
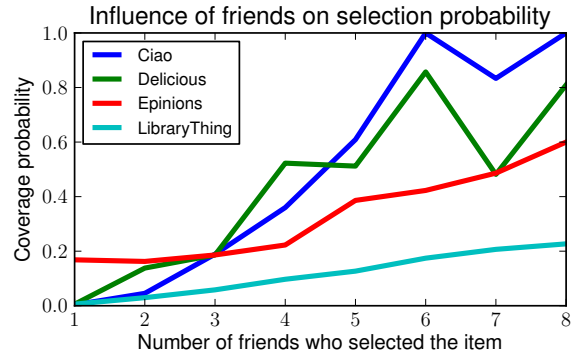
Organization. Section 2 formulates the problem; Section 3 introduces the datasets used in our study and presents some observations related to the problem we try to solve in this paper; Section 4 describes the proposed model and algorithms; Section 5 shows experimental results and verifies the contribution of our SBPR algorithm for one-class recommendation problems; Finally, Section 6 discusses related work and Section 7 concludes the paper.

2. PROBLEM DEFINITION

In this section, we first introduce several concepts and definitions used in the paper and then formally define the problem of social personalized ranking.

Let U denote the user set and I denote the item set. Assume that we observe a social network $G = (U, E)$, where $(u, v) \in E$ indicates user u and v are linked, and some "feedback" $F = \{(u, i)\}$, where $u \in U$ and $i \in I$, from $|U| = M$ users and $|I| = N$ items. The concepts that will be used in this paper are defined as:

Observed items and unobserved items : For each user u, observed items $O_u \in I$ are defined as the items for which user u shows an observed rating or preference, and unobserved items $\overline{O}_u \in I$ are the remaining items.

In this work, for each user, we divide the total item set into three parts: positive feedback, social feedback, and negative feedback, defined as follows:

Positive feedback : positive feedback $P_u = \{(u, i)\}$ is defined as the set of user-item pairs containing user u and his/her observed items $i \in O_u$. These could be items that u chose, purchased, rated, reviewed etc. depending on the setting and dataset in question.

Social feedback : social feedback is defined by the tuple $SP_u = \{(u, k)\}$ where u represents the user, and $k \in O_v \cap \overline{O}_u$ represents items that user u did not choose but at least one of their friends, $v, (u, v) \in E$ selected.

Negative feedback : negative feedback is defined as the set of user-item pairs $N_u = \{(u, j)\}$, where $j \in \overline{O}_u \cap \overline{O}_v \cap \ldots \cap \overline{O}_{v'}$ is the set of items that neither user u nor any of their friends, $(u, v), (u, v') \in E$, selected. Here *negative* only means no explicit feedback can be observed from the user and does not represent users' dislike of the items.

One can easily find that $P_u \cap SP_u \cap N_u = \emptyset$ and $P_u \cup SP_u \cup N_u$ contains the total item set. Given the definition of *Social feedback*, we now introduce a social coefficient s_{uk} for $(u, k) \in SP_u$ that can be used to describe the preference distance between u's positive feedback and social feedback.

Table 1: Statistics of the Datasets.

	Ciao	Delicious	Lthing	Epinions
#Users	1,705	1,670	73,882	41,554
#Item	12,252	52,613	337,561	112,991
#Observed feedback	22,839	71,105	979,053	181,394
#Social relations	47,842	13,246	12,0536	181,304
#Average P've-feedback	13	42	13.25	4.3651
#Average SP've-feedback	342	241	101.08	11.437

Table 2: Statistics of cold/cool-start users (users who have performed fewer than 5 actions).

	Ciao	Delicious	Lthing	Epinions
#Cold-start users	1,031	126	48,922	37,114
#Avg. social relations	35.95	8.14	0.8193	3.369
#Average P've-feedback	2.86	2.68	1.985	1.686
#Average SP've-feedback	195.9	105.71	33.86	14.005

Social coefficient : given a particular user u, associated with their social feedback $(u, k) \in SP_u$, s_{uk} is a parameter indicating the attitude from u's social relations towards a particular item k. The value of the social coefficient can be defined in different ways that we will discuss later, but in essence a large value of the social coefficient indicates that u's friends show a strong preferenc toward item k from which we can naturally assume u also likes the item although he has not selected it.

Unlike previous works that usually divide the item set into two classes (positive and negative), we introduce a new social feedback class by exploiting users' social information. Armed with these concepts, we can define the problem of social personalized ranking. Our goal is to recommend a (personalized) ranked list of items for each user u. Since the above concepts are defined using both user feedback and social information, the key challenge is now to learn a ranking function that incorporates all of these sources of information. More precisely:

Leveraging social connections to improve personalized ranking for collaborative filtering. Given observed feedback $F^{train} = \{U, I\}$ and a social network G from M users and N items, the goal of this paper is to learn a ranking function for each user u

$$f : (u, F^{train}, G, P_u, SP_u, N_u, \{s_{uk}\}) \rightarrow Ranked\_list(I) :$$
$$r_1(m) \succeq \ldots r_i(p) \succeq r_{i+1}(q) \ldots$$

where $r_i(p) \succeq r_{i+1}(q)$ encodes that user u shows higher preference towards item p than item q.

3. DATA DESCRIPTION

Before we introduce the proposed method, we first discuss the four data sets used in this paper and demonstrate their different features.

3.1 Data Description

The data sets used in this paper are collected from four popular web sites: Ciao, Delicious, LibraryThing and Epinions. Statistics of the four datasets are summarized in Table 1. "Feedback" in these datasets represents whether users purchased an item, or provide a bookmark (in the case of Delicious). All data sets contain such feedback in addition to explicit social network information.

The first two datasets we use are from Ciao, a product review website, and Delicious, a bookmarking website. Both datasets are available online.[5]

The remaining two datasets, LibraryThing and Epinions, we introduce in this paper. Both are "complete" datasets (they include *all* actions by *all* users of the website); both were collected from public sources and can be found on the second author's webpage[6].

LibraryThing (Lthing for brevity) is a popular book-reviewing website that allows users to create an online catalog of the books they own or have read. A user can tag and rate all the books she adds to her personal library. The dataset we crawled contains user feedback on items, including both ratings and comments. Since this paper focuses on solving the Top-N recommendation problem, we filter out explicit negative feedback (rating scores below 4 out of 5 stars) and use the remaining instances for model learning. After such preprocessing, the final version of the dataset contains 73,882 users and 337,561 items with 979,053 actions and 120,536 social relations between users.

Epinions is a popular online consumer review website. In total, we collected data from 41,554 users and 112,991 items along with their rating scores, review text and social relations (which in this case indicate trust relationships). Similar to the Lthing dataset, we also remove all negative rating feedback (less than 4 stars) and finally obtain 181,394 positive actions.

Table 1 shows the average amount of positive feedback, social feedback, and the number of social relations. From the table we observe that even when the amount of positive feedback per user is small, the amount of social feedback is relatively much larger. In other words, there is a much larger volume of data to draw upon if we leverage feedback provided by each user's friends.

3.2 Cold-start Analysis

Cold- (or cool-)start users are those with few observable historical actions in the datasets. In Table 2 we show the statistics of users with fewer than 5 explicit actions. From the table, one can see that different datasets exhibit different properties with regard to cold-start users. In the Ciao dataset, although more than 60% of users have fewer than 5 positive actions, the average number of users' social relations is quite large. In the Delicious dataset, the number of cold-start users is only 126 and they all have numerous social relations. In the Lthing dataset, we find that cold-start users rarely have social relations with other users, meaning that they are really "cold," making it difficult to model their preferences. In Epinions, we find that the average amount of social feedback is much smaller than in the other three datasets.

To conclude, we find that even if the amount of positive feedback for cold-start users is limited, there is often significantly more social feedback available when we consider their social relations. Therefore, how to design an algorithm that properly leverages both limited positive feedback in addition to social feedback becomes a critical task when making recommendations to cold-start users.

4. SOCIAL PERSONALIZED RANKING

In this section, we will detail our model assumption regarding positive, social, and negative feedback and then describe the proposed social personalized ranking algorithm.

The problem we study is usually referred to as "one-class collaborative filtering". Most work on this topic focuses on the use of positive versus negative feedback, but ignores the influence of users'

---

social connections. Moreover, when sampling positive and negative feedback instances for training, prior pairwise ranking methods assume that each sampled pair has the same weight and contribution to the model at training time.

Unlike existing methods, we incorporate feedback from a user's social network through a modeling assumption regarding a new class of items that we refer to as "social feedback." We also introduce a coefficient based on the strength between users' ties that controls how training pairs are sampled.

4.1 Model Assumption

We first introduce the basic assumption used in prior pairwise methods [8, 15], which can be represented as

$$x_{ui} \succeq x_{uj}, i \in P_u, j \in N_u, \qquad (1)$$

where x_{ui} represents the preference of user u on item i and the relationship $x_{ui} \succeq x_{uj}$ says that a user u is likely to prefer a positive item $i \in P_u$ to a negative item $j \in N_u$. This assumption reflects the differences between the basic ideas of point-wise and pairwise methods. Point-wise methods [5, 12] focus on fitting the numeric values of the data while pairwise methods [15, 16, 18] model the preference-order of the data instead.

Although the pairwise assumption generates better recommendation results than point-wise methods, it suffers from some obvious drawbacks as mentioned in the beginning of this section.

Now we describe our proposed assumption based on two pairwise preference comparisons:

$$x_{ui} \succeq x_{uk}, x_{uk} \succeq x_{uj}, i \in P_u, k \in SP_u, j \in N_u, \qquad (2)$$

where x_{ui} represents a user u's preference on positive feedback i, x_{uk} represents the preference on social feedback k, and x_{uj} represents the preference on negative feedback j. Here, the "unobserved" feedback is divided into two parts: social feedback and negative feedback. Based on social feedback, we can assume a fine-grained preference order for the "unobserved" feedback. Our proposed assumption is more general and considers both the influence of a user's social connections as well as their explicit feedback, making it more realistic in real social-recommendation settings.

Alternative assumption : Conceivably, one could argue that "social" feedback should be treated *more* negatively than items which are not observed at all (i.e., negative feedback). Specifically, social feedback means that a user *knows about* an item (through their friends), yet has still chosen not to purchase/view/evaluate it. This might be treated as a signal that a user *dislikes* that item. Since we cannot directly observe which of these two effects is more prominent in real data, we test this hypothesis experimentally through the *Alternative assumption*:

$$x_{ui} \succeq x_{uk}, x_{ui} \succeq x_{uj}, i \in P_u, k \in SP_u, j \in N_u \qquad (3)$$

The main difference between Eq. 3 and Eq. 2 is the removal of the preference order between positive feedback and social feedback. Although the assumptions are different, they can be incorporated into the same model structure and learning method as described below. Due to the space limitations, we introduce the formulation and learning of the model with the assumption as in Eq. 2 and experimentally compare the two assumptions in Section 5.

4.2 Model Formulation

Based on our assumption, we propose an optimization criterion for each user. In particular, the inequality in Eq. 2 can be used to maximize the value of the Area Under the ROC Curve (AUC) which is widely used in classification problems. A large AUC value means that the positive feedback is more likely to be ranked higher

---

Input: Observed feedback $F = \{(u, i)\}$, where $u \in U$ and $i \in I$ and a social network $G = (U, E)$
Output: Parameters
$\quad\quad \Theta = \{W \in R^{d \times M}, V \in R^{d \times N}, b \in R^N\}$
Initialization:
Initialize **for** $u = 1; u \leq M;$ **do**
$\quad$| split N items into three parts: P_u, SP_u and N_u;
end
Training:
for *iterations* **do**
$\quad$**for** *#training sample* **do**
$\quad\quad$ Uniformly sample a user $u \in U$;
$\quad\quad$ Uniformly sample an item i from P_u;
$\quad\quad$ Uniformly sample an item k from SP_u;
$\quad\quad$ Uniformly sample an item j from N_u;
$\quad\quad$ Calculate $\frac{\partial \mathcal{O}(\Theta)}{\Theta_{u,i,k,j}}$;
$\quad\quad$ Update W_u, V_i, V_k, V_j, b according to Eq. 7;
$\quad$**end**
end

Algorithm 1: Social Personalized Ranking.

than social feedback and the social feedback is more likely to be ranked higher than negative feedback.

Thus, for each user u, the optimization criterion can be represented as follows:

$$\prod_{i \in PSP_u, k \in PSP_u} \mathcal{P}(x_{ui} \succeq x_{uk})^{\delta(u,i,k)}[1 - \mathcal{P}(x_{ui} \succeq x_{uk})]^{1-\delta(u,i,k)}$$
$$\prod_{k \in SPN_u, j \in SPN_u} \mathcal{P}(x_{uk} \succeq x_{uj})^{\zeta(u,k,j)}[1 - \mathcal{P}(x_{uk} \succeq x_{uj})]^{1-\zeta(u,k,j)}$$
$$(4)$$

where $PSP_u = P_u \cup SP_u$, $SPN_u = SP_u \cup N_u$; $\delta(u, i, k)$ is an indicator function that is equal to 1 if $i \in P_u$ and $k \in SP_u$, and 0 otherwise. Similarly, $\zeta(u, k, j)$ is an indicator function that is equal to 1 if $k \in SP_u$ and $j \in N_u$, and 0 otherwise. Eq. 4 reflects the main assumption of the paper that for a specific user, (a) her preference due to positive feedback should be larger than that of social feedback, and (b) her preference due to social feedback should be larger than that of negative feedback. Due to the totality and antisymmetry of a pairwise ordering scheme as argued in [15], the above formula can be rewritten as Eq. 5 aiming to maximize the AUC value,

$$\frac{\sum_{i \in P_u, k \in SP_u} \mathcal{P}(x_{ui} \succeq x_{uk})}{|P_u||SP_u|} + \frac{\sum_{k \in SP_u, j \in N_u} \mathcal{P}(x_{uk} \succeq x_{uj})}{|SP_u||N_u|}.$$
$$(5)$$

When optimizing for the AUC, it is common practice to work with a differentiable function, such as a sigmoid function which has the form $\sigma(x) = \frac{1}{1+e^{-x}}$, to approximate the function $\mathcal{P}(.)$, so that the objective function is differentiable. Based on this trick, our goal is to maximize the following objective function,

$$\sum_u \left[\sum_{i \in P_u} \sum_{k \in SP_u} \ln(\sigma(\frac{x_{ui} - x_{uk}}{1 + s_{uk}})) + \right.$$
$$\left. \sum_{k \in SP_u} \sum_{j \in N_u} \ln(\sigma(x_{uk} - x_{uj})) \right] - regularization \qquad (6)$$

where a regularization term is used to avoid overfitting in the learning process. The preference function is modeled by matrix factorization, $x_{ui} = W_u^T V_i + b_i$, $x_{uk} = W_u^T V_k + b_k$ and $x_{uj} = W_u^T V_j + b_j$ where d is the number of latent factors, $W \in R^{d \times M}, V \in$

$R^{d \times N}, b \in R^N$. We adopt ℓ_2-norm regularization terms for model parameters $\Theta = \{W, V, b\}$.

Unlike other works, we employ a coefficient s_{uk} in Eq. 6 to control the contribution of each sampled training pair to the objective function. Reviewing the definition of s_{uk}, if we assign a fixed value, say, 1 to each social relation, we find that s_{uk} counts the number of user u's friends who choose the item k when u himself does not. The objective function uses the reciprocal of $(1 + s_{uk})$ to control the preference difference between positive and social feedback. A large value of s_{uk} indicates that user u might prefer this social feedback (item) over his positive feedback since many of u's friends show their preference on the item.

4.3 Model Learning

We employ the widely used stochastic gradient descent (SGD) algorithm to optimize the objective function in Eq. 6. The main process of SGD is to randomly select a (positive, social) and (social, negative) feedback pair and iteratively update model parameters based on the sampled feedback pairs. Specifically, for each training instance, we calculate the derivative and update the corresponding parameters Θ by walking along the ascending gradient direction,

$$\Theta^{t+1} = \Theta^t + \eta \times \frac{\partial \mathcal{O}(\Theta)}{\partial \Theta}. \tag{7}$$

One can find that when users have no social relations, social feedback will vanish and the proposed preference assumption will reduce to the basic assumption of BPR [15], which does not consider the influence of social networks on users' personalized item ranking. The steps to learn the model parameters are depicted in Algorithm 1.

4.4 Training Instance Sampling Strategy

Since the learning method we use in this paper is stochastic gradient descent (SGD), the selected training instances might have great impact on the recommendation performance. Therefore, following existing works [17, 24], we try different sampling strategies to select the training instances. Specifically, for each user u, the sampling strategies used in this paper are as follows:

- Uniform sampling: all positive, social and negative feedback is drawn uniformly from P_u, SP_u and N_u.

- Static sampling: the positive feedback is sampled uniformly from P_u; the social feedback is drawn according to a Geometric distribution:

$$p(k \in SP_u | u) \propto \exp(\frac{r(k)}{\lambda}), \lambda \in \mathbb{R}^+ \tag{8}$$

where $r(k)$ is the rank of social feedback k according to the social coefficient s_{uk}. The expected rank in the distribution of Eq. 8 is identified by the parameter λ. Negative feedback instances are sampled uniformly from N_u.

- Adaptive sampling: instead of using a static parameter, e.g. a social coefficient or popularity, an adaptive sampling strategy uses a preference scoring function to define the sampling distribution. For each user u, positive instances are sampled uniformly from P_u. Both social and negative instances are sampled through following distribution:

$$p(k \in SP_u | u) \propto \exp(\frac{x_{uk}}{\lambda}), \lambda \in \mathbb{R}^+$$
$$p(j \in N_u | u) \propto \exp(\frac{x_{uj}}{\lambda}), \lambda \in \mathbb{R}^+. \tag{9}$$

- Dynamic negative sampling: negative feedback instances are sampled according to the preference scoring function. Unlike the adaptive sampling strategy, in [24], Zhang et al. design a rejection sampling procedure. Here we follow a linear weight function. For each user u, positive and social instances are drawn uniformly from P_u; then we uniformly draw $j, l \in N_u$ and calculate x_{uj} and x_{ul}; if $x_{uj} > x_{ul}$, return j as selected negative feedback with probability $\frac{1}{1+\beta}$ or return l otherwise; if $x_{ul} > x_{uj}$, return l as selected negative feedback with probability $\frac{1}{1+\beta}$ or return j otherwise. Here β is a constant and we fix it to 0 as in [24].

5. EXPERIMENTS

In this section, we conduct experiments on the four real-world datasets to evaluate the effectiveness of the proposed method.

5.1 Experimental Setup

In order to demonstrate the performance of our approach, we use four datasets and several metrics to evaluate all compared algorithms. Specifically, we split the data into a training part, used for model training, and a test part, used for model evaluation (there are no model hyperparameters so a validation set is not required). For each user, we randomly select 90% of their observed feedback as P_u and leave the remainder as T_u for testing. Then SP_u and N_u are as defined in Section 2. Grid search is applied to find regularization parameters, and we set the values of parameters α_u, α_v and α_b as 0.015, 0.025, and 0.01 respectively.

Our experiments are intended to address the following questions:

1. How does our approach compare with related personalized ranking methods for item recommendation?

2. Can the proposed method solve the cold-start recommendation problem?

3. How quickly does our proposed method converge?

4. How do different definitions of social feedback alter recommendation performance?

Comparison methods. In order to demonstrate the benefits of our approach, we compare our model with the following methods for item recommendation. Since the problem we solve in this paper is one-class recommendation (without rating scores), it is unsuitable to compare our methods with rating estimation methods. Instead, we consider some state-of-the-art social one-class recommendation methods [8, 13] as baselines.

- **Random (Rand):** This method randomly orders the items for recommendation.

- **Most Popular (MP):** This method presents a non-personalized ranked item list based on how often items are chosen among all users.

- **MMMF:** This method [20] is an extension of Maximum Margin Matrix Factorization for item recommendation optimized by a soft hinge ranking loss.

- **WRMF:** The weighted matrix factorization method is proposed by [12], which uses a point-wise strategy for solving one-class recommendation problems.

- **BPR-MF:** This method [15] proposes a pairwise assumption for item ranking. Here we employ a uniform sampling strategy for training instance selection.

Table 3: Recommendation performance of different methods on four real-world datasets. The last column shows the improvement of the proposed method compared with the best baseline method.

Dataset	Metrics	Rand	MP	MMMF	WRMF	BPR	MR-BPR	GBPR	SBPR-1	SBPR-2	Improv.
Ciao	R@5	0.0075	0.0350	0.0220	**0.0355**	0.0243	0.0236	0.0306	0.025	0.0264	-25.6%
	R@10	0.0097	0.0487	0.0457	0.0594	0.0417	0.0429	0.0494	0.0605	**0.0634**	6.7%
	NDCG	0.1445	0.1687	0.1692	0.1881	0.1672	0.1804	0.1683	0.1852	**0.1890**	0.4%
	AUC	0.5054	0.5822	0.6079	0.6948	0.5894	0.7068	0.5909	0.7171	**0.7189**	1.7%
Delicious	R@5	9.3×10^{-4}	0.0012	0.0029	0.0091	0.0058	0.0128	0.0083	0.0045	**0.0369**	188%
	R@10	0.0017	0.0022	0.0059	0.0162	0.0103	0.0221	0.0174	0.0126	**0.0770**	250%
	NDCG	0.1422	0.1425	0.1474	0.1708	0.1807	0.1884	0.1935	0.1822	**0.2147**	10.9%
	AUC	0.5051	0.5158	0.5555	0.6613	0.7303	0.7577	0.7372	0.7576	**0.7618**	0.5%
Lthing	R@5	6.27×10^{-5}	0.0118	0.0089	0.0133	0.0083	0.0062	0.0107	0.0096	**0.0164**	23.3%
	R@10	3.193×10^{-4}	0.0186	0.0125	0.0203	0.0119	0.0107	0.0191	0.0148	**0.0245**	20.6%
	NDCG	0.1043	0.1328	0.1255	0.1365	0.1220	0.1297	0.1322	0.1306	**0.1380**	1.1%
	AUC	0.5026	0.6922	0.6035	0.7255	0.5816	0.7496	0.6904	0.7377	**0.7622**	5.1%
Epinions	R@5	0.0081	0.0054	0.0045	0.0053	0.0045	0.0054	0.0053	0.0062	**0.0082**	1.2%
	R@10	0.0106	0.0103	0.0121	0.0078	0.0079	0.0099	0.0112	0.0125	**0.0143**	18%
	NDCG	0.1321	0.1312	0.1316	0.1320	0.1313	0.1315	0.1326	0.1342	**0.1365**	2.9%
	AUC	0.4898	0.4969	0.5042	0.4991	0.5054	0.5023	0.5212	0.5257	**0.5314**	1.9%

- **MR-BPR:** This method [8] combines multi-relational matrix factorization models and *BPR-MF* models in order to model users' feedback on items and on social relations simultaneously. *MR-BPR* is a state-of-the-art method for one-class recommendation with social information.

- **GBPR:** This work [13] relaxes *BPR*'s assumption to a group pairwise preference assumption. The authors try to smooth the individual positive feedback model and increase the confidence for pairwise classification by aggregating a group of users' preferences. Here we fix the number of grouped users to 5.

- **Social BPR-1 (SBPR-1):** This method follows the assumption of Eq. 3. The model formulation and learning method are similar to Algorithm 1.

- **Social BPR-2 (SBPR-2):** This is our proposed method that strictly follows the assumption of Eq. 2 and the learning algorithm is shown in Algorithm 1.

Many of the above baseline methods can be found in [3].

Evaluation Metrics. We use three popular metrics, *Recall@K* (R@K), *NDCG* (Normalized Discounted Cumulative Gain) and *Area under the curve* (AUC), to measure the recommendation quality of our proposed approach in comparison to baseline methods.

The average AUC statistic is defined as

$$AUC = \frac{1}{M} \sum_{u \in M} \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(x_{ui} > x_{uj}), \quad (10)$$

where $E(u) = \{(i,j)|(u,i) \in T_u \wedge (u,j) \notin (P_u \cup T_u)\}$.

DCG@K considers the ranking of the recommended items by discounting the importance and is defined as

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \quad (11)$$

where rel_i represents the relevance score of the item i (we use a binary value for this quantity). *NDCG* is the ratio of the *DCG* value to the ideal *DCG* value for that user. The ideal value of *DCG* comes from the best ranking function for the user.
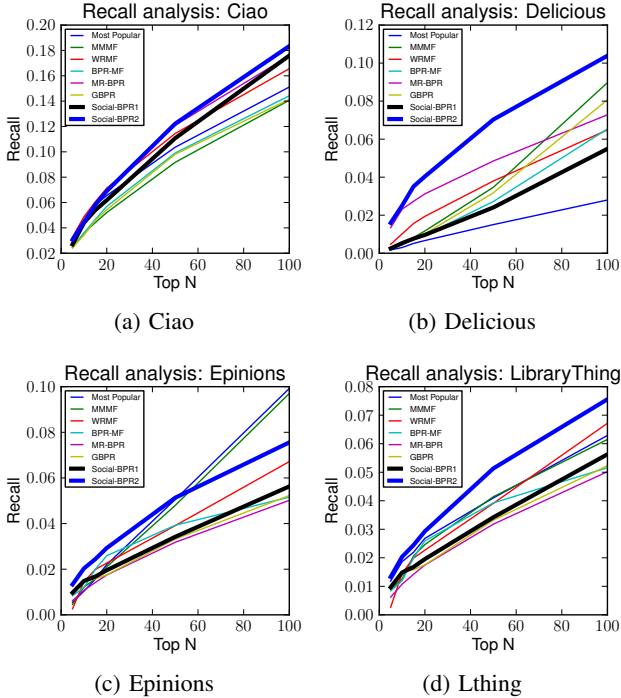
5.2 Recommendation Performance

Recommendation Evaluation Table 3 details the average recommendation performance of different methods. Here we fix the number of latent factors to 10. From the results, we can see that our approach shows significant improvement compared with other algorithms on all four datasets. Since most of the social recommendation methods are concerned with rating estimation and cannot handle one-class recommendation problems, here we include comparisons with two state-of-the-art one-class recommendation methods, *MR-BPR* and *GBPR*. In particular, when comparing the *SBPR* models with the two closely related methods, we find that our proposed approach wins in most cases. One possible reason may be that although *MR-BPR* models users' social preference and item preference simultaneously making it better than many baseline methods, it fails to model how social relations directly influence users' preferences on items. As for *GBPR*, it replaces *individual preferences* with *group preferences*. However, the group of users in *GBPR* is sampled by *random selection* based only on a simple rule that users in the group have the same positive preference towards an item. We also find that all models show poor performance on the Epinions dataset, the reason we consider is the sparsity of users' positive feedback (As Table 1 shows, the average number of positive feedback in Epinions is only 4).

In Section 4, we proposed two preference order assumptions based on social feedback. In this section, we compare the performance of those two assumptions. From the experimental results, we find that the performance of the first assumption always outperforms the other one, which shows that users are more likely to prefer the items selected by their friends to others. These results are also consistent with the observational analysis in Section 1.

Detailed results for *Recall@N* can be found in Figure 3. Although the curves of different methods on Ciao seem to be very close to each other, both of the proposed methods outperform baselines on different *Recall@N* measure points. On Epinions, it is much clearer that *SBPR*-2 improves the recommendation performance when *N* is small. On the other two datasets, the proposed methods consistently outperform all baseline methods.

Table 4: Sampling Strategy Comparisons for SBPR-2. AUC is used for evaluation.

	Ciao	Delicious	Lthing	Epinions
Uniform Sampling	0.7133±0.0012	0.7536±0.0011	0.7554±0.0014	0.5274±0.0022
Static Sampling	**0.7189±0.0014**	**0.7618±0.0019**	**0.7622±0.0004**	**0.5314±0.0012**
Adaptive Sampling	0.6855±0.0162	0.7528±0.0056	0.7522±0.0012	0.5269±0.0017
Dynamic Negative Sampling	0.6823±0.0130	0.7463±0.0101	0.7510±0.0034	0.5222±0.0032

(a) Ciao   (b) Delicious

(c) Epinions   (d) Lthing

Figure 3: Recommendation performance comparisons (Recall@N) on four datasets.

Recommendation for Cold-Start Users. In Section 3, we have shown that although cold-start users' have limited positive feedback (or none), the introduction of social feedback provides information that can be used to model their preferences. In this section, we perform experiments to investigate whether the proposed assumption and algorithms can improve the recommendation accuracy for such cold-start users. We randomly select 10% of feedback as a test-set and use the remainder for training. Among training feedback, we identify users with fewer than 5 positive actions as cold-start users. The experimental results are shown in Table 5. From these results, we observe that our proposed methods significantly outperform other baseline methods on all four datasets for cold-start recommendation. When training instances are scarce, social information provides evidence to represent or interpret cold-start users' preferences. Moreover, when comparing the performance of *SBPR* with *MR-BPR*, the improvement of *SBPR* indicates that modeling the influence of the social network to cold-start users' preference ranking is more suitable than explicitly modeling their social relations, since the latter might bring too much bias when users' available training instances are limited.

Although we have shown that *SBPR* yields good performance when training instances are scarce, an important case is when there are no training instances, i.e., for brand-new users of the system. Such users may have social relationships (e.g. if they signed

on to a service using a social media account), but have performed no actions and provided no feedback. This is a genuine cold- (rather than cool-)start setting. To evaluate the performance in this setting we test the recommendation accuracy for users as a function of the number of available training instances (from 0 to 5) and compare the performance of *SBPR* with *MR-BPR* on these users. A user with 0 training instances represents the case of a brand-new user of the system. Figure 4 shows results. We can see that on all four datasets, *SBPR* outperforms the baseline method. The reason is that *MR-BPR* suffers from modeling biases brought from fitting users' feedback on social relations when users' feedback on items is insufficient. This improvement verifies that exploiting feedback from users' social relations to model the ranked list of users' preferred items is a promising way to solve cold-start recommendation problems.

5.3 Analysis and Discussion

Social Feedback Analysis. In the *SBPR* model, we define *social* feedback based on users' social relations. In order to demonstrate the contribution of this definition, we conduct experiments by comparing several different definitions of social feedback. The definitions compared include *random user selection*, *random item selection* and *all items*. The former two definitions are illustrated in the Introduction, and *all items* uses all items as social feedback. The results are shown in Figure 6 and we use the AUC value for evaluation. From the results, we can see that only by using users' explicit social relations to identify social feedback does the proposed method achieve optimal performance on all four datasets. This analysis verifies the empirical observation in Section 1.

Social Coefficient Analysis. The social coefficient controls the contribution of social feedback. In this section, we study the impact of the social coefficient on recommendation accuracy by measuring it in different ways. This coefficient for social feedback can be viewed as the summarization of social strength from related users, indicating the preference distance between positive feedback and social feedback. Social feedback with a large social coefficient implies that items have a higher probability of being adopted or preferred by users. Here we choose three definitions to capture the social strength between each linked-user pair:

- constant: we assign a constant, 1, to each social relation; in this way, the user's social preference towards an item is determined simply by counting the number of friends who prefer the item.

- common neighbor: we define the value of each social relation by the number of their common neighbor users; in this way, the coefficient, as a summation of social strength, represents how much users' friends prefer social feedback by considering the social network structure.

- common preference: different from the definition above, we here define the value of each social relation by the number of
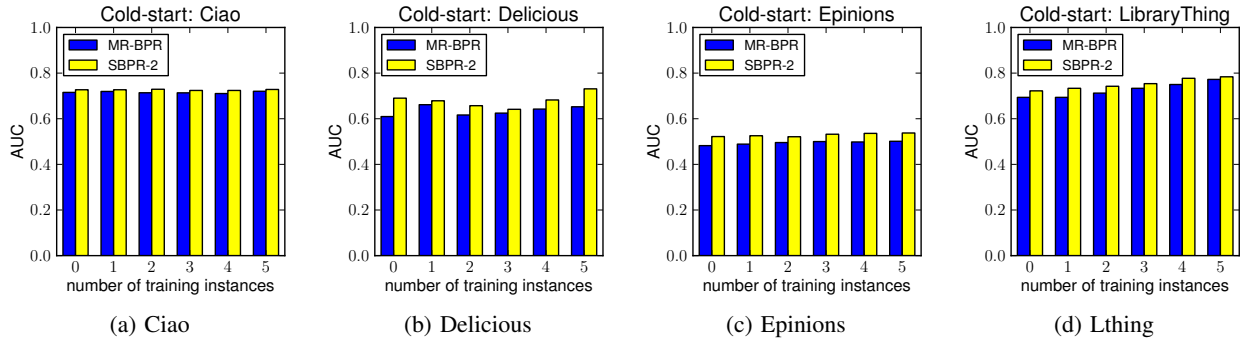
(a) Ciao   (b) Delicious   (c) Epinions   (d) Lthing

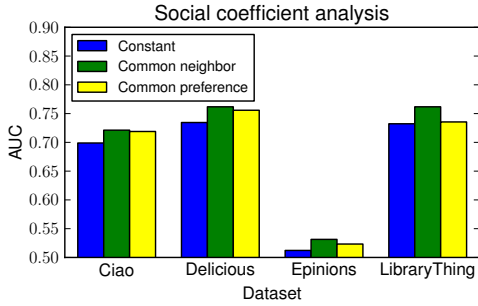**Figure 4: Detailed cold-start recommendation analysis.**

Figure 5: Social coefficient Analysis.

common items preferred by the linked users; thus, the coefficient measures the preference of social feedback by considering users' common preferences with their friends.

As Figure 5 shows, the common-neighbor setting achieves better performance than the other two settings. In fact, when training data is scarce, the common-neighbor setting captures more social information for social feedback than the common-preference and constant settings.

Sampling Strategy Analysis. Table 4 demonstrates the AUC values of *SBPR* with different sampling strategies for training. One can find that static sampling always performs better than other strategies. The reason is that the sampling distribution based on the social coefficient provides valuable information for social feedback selection. We also conduct experiments to investigate whether negative feedback should be sampled from a static distribution according to their global popularity. Unfortunately, the performance is poor; our analysis of the datasets suggests that negative feedback with high global popularity does not indicate that a user dislikes an item. Moreover, both adaptive sampling and dynamic negative sampling strategies show poor performance. One possible reason is that these two strategies select training instances based on intermediate estimated parameters. However, since model learning has not yet converged, the intermediate parameters are unreliable for selecting training instances. The uniform sampling strategy shows better results than adaptive and dynamic negative sampling strategies but still cannot achieve performance as good as static sampling. The only difference between uniform sampling and static sampling in this paper is whether the social coefficient is used to select social feedback for training. By using the social coefficient, the static sampling strategy can leverage users' social connections to select the most informative feedback pairs for model training.

Convergence Analysis. We further investigate the convergence of the *SBPR* model. Figure 7 shows the convergence analysis of the *SBPR* model on different datasets. For each iteration, we select $\frac{\sqrt{|F^{train}|}}{50}$ instances for training. We see that our method converges within 30 iterations on Ciao and always in fewer than 80 iterations on other datasets. In particular, we also find that when using different strategies for training instance sampling, our proposed method demonstrates similar convergence performance. This fact indicates that the proposed method can achieve promising convergence regardless of how we select training instances. Combined with the above sampling strategy analysis, we conclude that different strategies of training instance selection show a great impact on the recommendation accuracy but not on the convergence performance of *SBPR*.

6. RELATED WORK

In this section, we will briefly review related work along two highly related aspects: one-class collaborative filtering and social recommendation.

One-class Collaborative Filtering Several works have studied one-class collaborative filtering and can be mainly divided into two branches: pointwise methods [5, 12] and pairwise methods [14, 15, 16, 18, 22].

Pointwise methods aim to fit a numeric value associated with each evaluated item. These methods view positive feedback as high preference scores and use several strategies to sample negative feedback as low preference scores. Then existing matrix factorization methods can be used to fit the preference scores. Pan et al. [12] solve the one-class recommendation problem in two ways: negative example weighting and negative example sampling. Hu et al. [5] introduce a novel concept, called a "confidence level," associated with positive and negative feedback, and propose an efficient optimization method for confidence-based matrix factorization.

Different from pointwise methods, pairwise methods focus on modeling the order, or ranking of the feedback. Pairwise methods always consider implicit feedback as relative relationships indicating that users show higher preference on positive feedback than on negative feedback. In [15], Rendle et al. propose a bayesian personalized ranking (*BPR*) framework. Following this, various ideas have been proposed that incorporate different types of contextual information into the *BPR* framework. [8] extends the *BPR* framework to model both users' feedback on items and on their social relations. In [18], Rendle et al. extend the *BPR* framework from matrix factorization to tensor factorization for tag recommendation. Pan et al. [13] aggregate the features of a group of related users to reduce the uncertainty of the selected training instances.
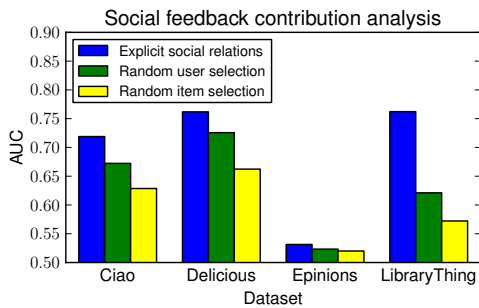
Figure 6: Social Feedback Contribution Analysis.

Du et al. [2] improve one-class recommendation performance by incorporating a social regularization term into the *BPR* framework.

Since the learning method used in most related works is stochastic gradient descent, how to select the training instances has great impact on model performance. Zhang et al. [24] use current estimation results to dynamically select negative feedback. Lee et al. [9] propose a pairwise method by combining rating information and a kernel function to improve recommendation performance. Zhang et al. [24] use the current parameter estimates to dynamically select negative feedback for training. In [17], the authors propose two strategies to sample negative feedback and also design an efficient implementation to improve the convergence of the learning algorithm.

However, the aforementioned works mainly focus on modeling the feedback order by using users' positive and negative feedback, but do not investigate how the feedback from users' friends can be used to model users' preference order on items. Moreover, no existing works aim to leverage social connections to distinguish the different contribution of the sampled training pairs.

Social Recommendation. Social recommendation methods focus on using the social network to improve rating estimation and item recommendation. SoRec [10] is proposed as a probabilistic matrix factorization framework which incorporates trust network information into user taste analysis. Ma et al. [11] also propose a matrix factorization framework with social regularization based on the assumption that users' interests should be similar to those of their friends. Based on these works, Yang et al. [21] devise a factor-based random walk model to explain friendship connections, and simultaneously use a coupled latent factor model to uncover interest interactions.

Similarly, Jamali et al. [6] propose a matrix factorization method to handle the transitivity of trust relations and trust propagation. Jiang et al. [7] design a matrix factorization framework which exhibits the contribution of two important factors: individual preference and interpersonal influence. Furthermore, In [23], Ye et al. propose a generative model that captures social influence between friends quantitatively and employs social influence to mine the personal preference of users. Shen et al. [19] also propose a joint personal and social latent factor model for social recommendation.

Besides the use of feedback from users and items, recently, other types of contextual information have been incorporated into social recommendation methods for various specific applications. Cho et al. [1] develop a model of human mobility that combines periodic short range movements with travel due to the social network structure for user movement prediction. Zhao et al. [25] use tag information to incorporate topic mining into social recommender systems. However, most of the existing social recommendation methods fo-

cus on rating estimation problems rather than the one-class recommendation setting that we consider.

7. CONCLUSION AND FUTURE WORK

In this paper, we exploit users' social connections in order to improve the recommendation accuracy on one-class recommendation problems. We study four real-world datasets and observe that the probability that a user selects an item increases monotonically as a function of the number of friends who have selected the item. Based on this observation, we design a pairwise algorithm, called *SBPR* for recommendation. Experiments on four real-world datasets show that *SBPR* effectively improves the recommendation accuracy in one-class recommendation problems. Besides providing improved ranking accuracy in the general case, the detailed experimental analysis also verifies the contribution of SBPR for solving cold-start recommendation problems.

For future work, we are interested in extending *Social-BPR* in three ways, (1) Investigating how to incorporate rating information into the *SBPR* model; (2) Employing an active learning framework to select training pairs more effectively in the *Social-BPR* model; 3) Exploiting context information to model users' preference order on items.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. *In KDD*, 2011.

[2] L. Du, X. Li, and Y.-D. Shen. User graph regularized pairwise matrix factorization for item recommendation. *In ADMA*, 2011.

[3] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: a free recommender system library. *In RecSys*, 2011.

[4] L. Hong, R. Bekkerman, J. Adler, and B. D. Davison. Learning to rank social update streams. *In SIGIR*, 2012.

[5] Y. Hu, Y. Koren, and C. Volin-sky. Collaborative filtering for implicit feedback datasets. *In ICDM*, 2008.

[6] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. *In RecSys*, 2010.

[7] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. Social contextual recommendation. *In CIKM*, 2012.

[8] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization us-ing bayesian personalized ranking for social network data. *In WSDM*, 2012.

[9] J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer. Local collaborative ranking. *In WWW*, 2014.

[10] H. Ma, H. Yang, M. R.Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. *In CIKM*, 2008.

Dataset	Metrics	Rand	MP	MMMF	WRMF	BPR	MR-BPR	GBPR	SBPR-1	SBPR-2
Ciao	R@5	9.46×10^{-4}	**0.0359**	0.0268	0.0356	0.0293	7.575×10^{-4}	0.0284	0.0309	0.0261
	R@10	0.0082	0.0476	0.0422	0.0448	0.0366	0.0020	0.0467	**0.0486**	0.0359
	NDCG	0.1210	0.1650	0.1628	0.1643	0.1624	0.1247	0.1635	**0.1681**	0.1630
	AUC	0.5067	0.6515	0.6665	0.6740	0.6645	0.5436	0.6733	0.7212	**0.7272**
Delicious	R@5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0444	**0.1518**
	R@10	0.0111	0.0	0.0	0.0	0.0	0.0	0.0185	0.0444	**0.1852**
	NDCG	0.1019	0.1088	0.1141	0.1104	0.1155	0.1045	0.1278	0.1418	**0.2401**
	AUC	0.4785	0.6228	0.6035	0.6109	0.5969	0.5918	0.7007	0.6747	**0.7310**
Lthing	R@5	0.0	0.0128	0.0111	0.0128	0.0108	0.0072	0.0085	0.0091	**0.0133**
	R@10	0.0	0.0209	0.0149	**0.0226**	0.0143	0.0138	0.0145	0.0164	0.0164
	NDCG	0.0761	0.1006	0.0945	0.1025	0.0917	0.0987	0.0973	0.0969	**0.0992**
	AUC	0.5062	0.7272	0.6239	0.7393	0.5969	0.7724	0.7623	0.7599	**0.7838**
Epinions	R@5	8.2×10^{-4}	0.0038	0.0054	5.4×10^{-4}	0.0070	0.0071	0.0077	0.0073	**0.0090**
	R@10	0.0075	0.0084	0.0092	0.0079	0.0112	0.0117	0.0122	0.0136	**0.0153**
	NDCG	0.1298	0.1309	0.1316	0.1312	0.1319	0.1321	0.1344	**0.1360**	**0.1360**
	AUC	0.4960	0.4947	0.4841	0.5028	0.4847	0.4924	0.5203	0.5219	**0.5375**

(a) Ciao (b) Delicious (c) Lthing (d) Epinions
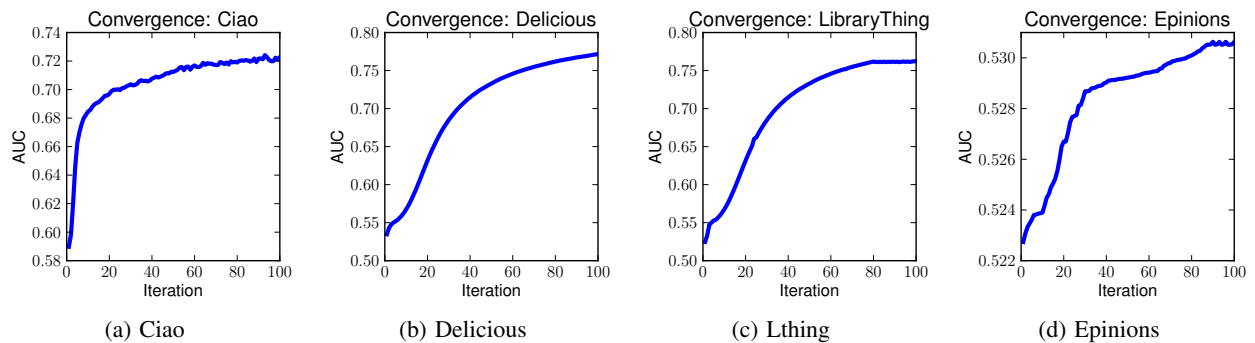
Figure 7: Convergence Analysis on Ciao and Delicious datasets.

[11] H. Ma, D. Zhou, C. Liu, M. R.Lyu, and I. King. Recommender systems with social regularization. *In WSDM*, 2011.

[12] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. *In ICDM*, 2008.

[13] W. Pan and L. Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. *In IJCAI*, 2013.

[14] U. Paquet and N. Koenigstein. One-class collaborative filtering with random graphs. *In WWW*, 2013.

[15] S. Rendle, C. Freuden-thaler, Z. Gantner, and L. Schmidt-Thieme. Bpr:bayesian personalized ranking from implicit feedback. *In UAI*, 2009.

[16] S. Rendle, C. Freuden-thaler, and L. S. Thieme. Factorizing personalized markov chains for next-basket recommendation. *In WWW*, 2010.

[17] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. *In WSDM*, 2014.

[18] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. *In WSDM*, 2010.

[19] Y. Shen and R. Jin. Learning personal + social latent factor model for social recommendation. *In KDD*, 2012.

[20] M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *In Machine Learning Journal*, 2008.

[21] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. *In WWW*, 2011.

[22] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. *In SIGIR*, 2011.

[23] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation: a generative model approach. *In SIGIR*, 2012.

[24] W. Zhang, T. Chen, J. Wang, and Y. Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. *In SIGIR*, 2013.

[25] T. Zhao, C. Li, M. Li, Q. Ding, and L. Li. Social recommendation incorporating topic mining and social trust analysis. *In CIKM*, 2013.