2015 IEEE 15th International Conference on Data Mining Workshops

CroRank: Cross Domain Personalized Transfer Ranking for Collaborative Filtering

Yunhui Guo
Institute of Artificial Intelligence
Zhejiang University
Hangzhou,China
gyhui@zju.edu.cn

Xin Wang
Institute of Artificial Intelligence
Zhejiang University
Hangzhou,China
cswangxinm@zju.edu.cn

Congfu Xu
Institute of Artificial Intelligence
Zhejiang University
Hangzhou,China
xucongfu@cs.zju.edu.cn

***Abstract*—Collaborative filtering techniques aim at recommending products to users based on their historical feedback. And many algorithms focus on personalized ranking problem with implicit feedback due to the "one-class" nature of many real-world datasets in a variety of services. Most of the existing personalized ranking methods are confined to one domain of data source and the question of how to model users' preferences information across distinct domains is usually be ignored. There are some transfer learning approaches that try to transfer numerical ratings, auxiliary social relations and other information across different domains but they do not address how users' preferences information varies from one domain to another accordingly. And they mainly exploit rating prediction problem rather than personalized ranking problem. In this paper, we propose an algorithm called CroRank to address the question, "How to bridge users' preferences information across different domains to promote better personalized ranking performance?". There are two main steps in CroRank, we first present an algorithm called multiple binomial matrix factorization (MBMF) to bridge the gap between items from distinct sources and then we introduce transfer Bayesian personalized ranking (TBPR) to recommend items for each user in the target domain. In CroRank, users' inclinations can transfer from the auxiliary domain to the target domain to provide better personalized ranking results. We compare CroRank to the state-of-the-art non-transfer models to demonstrate the improvements in flexibility and effectiveness.**

***Keywords*—tranfer learning; personalied ranking; collaborative filtering;**

I. Introduction

As the fast growing of the amount of information on the Web, recommendation becomes more and more crucial since it aims at excluding noisy information and provide users with useful feedback. Many commercial websites, such as Amazon.com and Alibaba.com, have employed recommender systems to promote better sales profit and user experience. Many algorithms have been proposed to handle recommendation task. Among them, collaborative filtering techniques [2], [24] have proven useful and various techniques based on collaborative filtering [6], [11], [18] have been developed recently to improve recommendation accuracy. Most of these methods try to model the observed ratings in the user-item matrix. However, due to the limitation of real-world datasets, the rating matrix can be extremely sparse or even unavailable. Meanwhile, users' implicit feedback such as "like" or "dislike" is easier to collect and many websites accumulate such information by providing users with corresponding links directly. These "one-class" form information [12] only contains positive feedback, and indicates that a user has a higher preference ranking for observed items over unobserved ones. However, the relations between unobserved items are usually unclear. "One-class" form feedback is often utilized to recommend items to users instead of for predicting ratings, which is of more significance since for commercial websites it is more important to recommend items for users directly.

There are several methods for solving the one-class collaborative filtering problem. Roughly, the previous works can be categorized into two main trends, (1) one is to formulate the problem by optimizing an objective function over both observed and missing data [12], and the weights over different edges must be carefully tuned during training process. This is sometimes called pointwise approach. And (2) the other is Bayesian personalized ranking that treats the one-class collaborative filtering problem as a pairwise ranking problem [20]. Empirical studies show that pairwise ranking approach is preferable to pointwise approach. [15].

There are many extensions to Bayesian personalized ranking (BPR) that try to provide more accurate recommendations. For example, [15] extends BPR by combining *group preference* with *individual preference* to induce more flexible interactions. And MR-BPR [7] models users' preferences and social relations simultaneously and achieve the state-of-the-art result. But none of them considers the question, "How to bridge users' preferences information across different domains to promote better personalized ranking performance?" This question arises when a commercial website intends to expand its business to cover more fields, the preference information in the existing domains is of great value and should be employed to develop the users' preference structures in the new domains.

To link data from different sources, transfer learning techniques [13] are proposed recently which aim at leveraging information across different domains to enhance the performance in the target domain. And transfer learning has been extended to classification, regression and clustering problems and has achieved desirable results [13]. Collaborative filtering models are also integrated with transfer learning methods recently [9], [15], [3] trying to deal with data sparsity problem and other issues. But most of them focus on optimizing a rating prediction objective function rather than conducting personalized ranking. As we stated above, ranking problem is sometimes more significant than rating prediction in a sense. So in this paper, we propose an algorithm called *cross domain personalized*

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.46

transfer ranking (CroRank) to enhance personalized ranking performance by transferring users' preference structures in different domains. To the extend of our knowledge, CroRank is the first model that trys to aggregate information across different domains to address personalized ranking problem.

To elaborate CroRank, we first make the following assumptions:

- The pairwise preference information of user u is independent of that of user v.
- If item i has been viewed by user u, user u prefers item i over all the unobserved items.
- In the target domain, user's preference for an item i is affected by the similar items of item i in the auxiliary domain.

The first two assumptions are the same as the ones made in BPR [20]. We call the last assumption *related item effect* and illustrate it in Figure. 1.

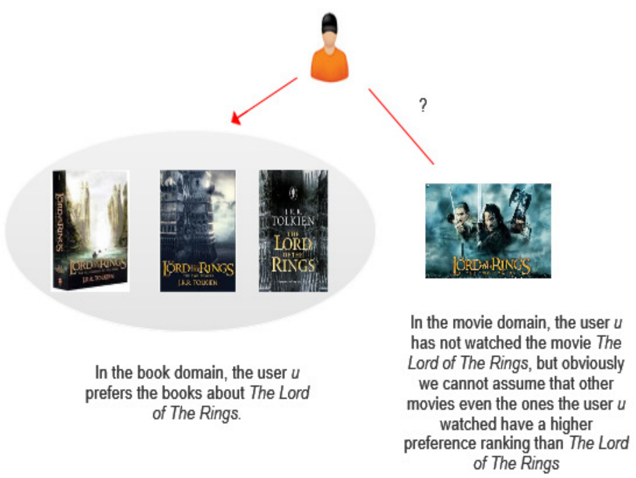


Figure 1. User's preference structure of one domain may be heavily affected by the user's behaviors in another domain.

In many real-world scenarios, a user may has few or even no feedback (cold-start user) in the target domain, meanwhile this user may has abundant behaviors in a related domain. Obviously, it is worthwhile to exploit the ranking information in the related domain so as to uplift the ranking performance in the target domain. This is especially important for the commercial website that wants to broaden its coverage to more domains. However, the explicit item correlations in different domains are usually unclear and manual annotations are obviously impossible. To alleviate this problem, we first adopt a multiple binomial matrix factorization technique to link items in different domains, by doing so, the implicit ranking structures in all domains are connected via the latent item vectors, and then we develop a procedure called *transfer Bayesian personalized ranking* (TBRP) to leverage users' preference information for items in the auxiliary domain to promote personalized ranking accuracy in the target domain. Apart from enhancing ranking performance for cold-start users, CroRank can also provide better personalized ranking for ordinary users by exploiting their inclinations for items across different domains.

The main contributions of CroRank can be summarized as follows,

- CroRank extends Bayesian personalized ranking by transferring user's preference information across different domains, to the best of our knowledge, CroRank is the first algorithm that utilizes such information for personalized ranking.
- CroRank consists of two main steps, MBMF and TBPR, both of them are effective in the dedicated tasks.
- Experimental results on several datasets demonstrate the effectiveness of CroRank over the state-of-the-art models for personalized ranking problem.
- For cold-start one-class recommendation problem, CroRank outperforms the state-of-the-art ranking approaches that address this problem.

II. Preliminaries

In this section, we first introduce binomial matrix factorization [27] and Bayesian personalized ranking [20], and then give the problem definition and notations.

A. Binomial Matrix Factorization

Traditional matrix factorization approaches for collaborative filtering usually assume that the latent user vectors and item vectors are sampled from normal distributions [22], [23] and the rating values are also drawn from normal distributions. However, in real recommender systems, users' feedback is always collected in discrete rating values, thus the normal distribution assumptions are not reasonable in this scenario. In [27], the author proposes a matrix factorization model based on binomial assumptions which is more reasonable.

In binomial matrix factorization (BMF), latent user vectors and item vectors are drawn from normal distributions as PMF [22], however the rating value that a user u for an item i is given by a binomial distribution. Given a latent user vector U_u and a latent item vector I_i, the probability that the corresponding rating value is R_{ui} is given by,

$$p(R_{ui}|U_u, I_i) = \mathcal{B}\ (R_{ui}|M, \hat{x}_{ui})$$

Where $\mathcal{B}(t|n,p)$ is a binomial distribution with parameters n and p. The maximal rating value that users can give is denoted by M, in many real-world recommender systems, this value is usually set to 5. And $\hat{x}_{ui}$ is the preference level that user u over item i. A sigmoid function is used to model $\hat{x}_{ui}$,

$$\hat{x}_{ui} = \frac{1}{1+e^{-U_u^T I_i}}$$

Given above assumptions, the probability that a user u gives a rating value r (r = 0,1,...,M) for an item i is given by,

$$p(R_{ui} = r|U_u = w_u, I_i = v_i) = \binom{M}{r}\ \hat{x_{ui}}^r\ (1 - \hat{x_{ui}})^{M-r}$$

Binomial assumptions can confine the rating values to be discrete ones and furthermore binomial distribution is unimodal, which is more reasonable. Meanwhile, $\hat{x}_{ui}$ can endow the user u with a preference ranking structure, since we can compare user u's preference level for item i and item j by computing $\hat{x}_{uij}$ = $\hat{x}_{ui}$ - $\hat{x}_{uj}$. A positive value of $\hat{x}_{uij}$ indicates that user u prefers item i over item j.

B. Bayesian Personalized Ranking

In Bayesian personalized ranking (BPR) [20], the item set that the user u has viewed is represented as I_+^u, and meanwhile the unobserved item set is denoted as I_-^u. We use I to denote the whole item set and U to denote the whole user set.

A user u's relative preference over item i and item j can be denoted by,

$$i \succ_u j \qquad i \in I_+^u,\, j \in I_-^u$$

Which indicates that the user u has a higher preference ranking for observed item i over unobserved item j.

The likelihood of BPR for a user u can be written as,

$$p(\succ_u) = \prod_{i,j \in I} p(\hat{x}_{ui} > \hat{x}_{uj})^{\sigma(i \succ_u j)} \times [1 - p(\hat{x}_{ui} > \hat{x}_{uj})]^{[1-\sigma(i \succ_u j)]}$$

Where $\hat{x}_{ui} > \hat{x}_{uj}$ indicates the latent preference configuration of user u over item i and item j and $\sigma(*)$ is a indicator function which equals to 1 when $*$ is true, otherwise equals to 0.

The basic assumptions of BPR are as follows:

- If item i has been observed by user u, user u prefers item i over all the unobserved items.
- The pairwise preference information of user u is independent of that of user v.

Based on the above assumptions, we can reach the following overall likelihood function of BPR,

$$\prod_{u \in U} p(\succ_u) = \prod_{u \in U} \prod_{i \in I_+^u} \prod_{j \in I_-^u} p(\hat{x}_{ui} > \hat{x}_{uj})[1 - p(\hat{x}_{uj} > \hat{x}_{ui}]$$

To optimize the objective function of BPR, the authors derive a stochastic gradient descent method based on bootstrap sampling.

C. Notations And Problem Definition

We denote the target domain as D_t and the auxiliary domain as D_a. As for the auxiliary domain D_a, the user set is denoted as U and the item set is denoted as I_a. In CroRank, users are shared across different domains. In the target domain, the item set is represented as I_t. We use X to denote the observed implicit feedback matrix of D_a and Y to denote that of D_t. Then we derive the following concepts of CroRank for ease of illustration,

Positive item set and negative item set: positive item set for the user u is denoted as I_+^u, thus I_+^u are those items that have been viewed by the user u. Similarly, negative item set for the user u is denoted as I_-^u. We assume that items in I_+^u has a higher preference ranking than items in I_-^u.

Related Item Group: For an item i in the target domain, we assume that there are a group of items in the auxiliary domain that the user u shares the same inclination, and we denote this group as I_i^g.

Inner Domain Preference: In one domain, the preference structure of user u over two items can be formulated as,

$$i \succ_u j \qquad i \in I_+^u,\, j \in I_-^u$$

Notice that *inner domain preference* only tells us the preference relations between observed items and unobserved ones and is inherently noisy

Related Item Preference: For item i and item j in the target domain, the preference structure of user u for I_i^g and I_j^g can be represented as,

$$\hat{x}_{uI_i^g} > \hat{x}_{uI_j^g} \quad i,j \in I_t \text{ and } i \neq j \quad I_i^g, I_j^g \subseteq I_a$$

To illustrate this formulation further, for example, if a user u has not watched the movie *The Lord of The Rings* and meanwhile the user has watched the movie *The Shawshank Redemption*. In the book domain, the user u read several books about *The Lord of The Rings* but few books related to *The Shawshank Redemption*. Due to this extra information, we cannot simply reach the following assumption in the movie domain,

$$\textit{The Shawshank Redemption} \succ_u \textit{The Lord of The Rings}$$

The information hidden in the book domain may reverse this preference order. It is easy to recognize that the preference structure in the auxiliary domain can have a significant impact in the target domain in personalized ranking.

In this paper, we propose an algorithm called *cross domain personalized transfer ranking* (CroRank) to consider the two types of impacts. Our aim is to promote personalized ranking performance in the target domain by transferring user's preference structure from the auxiliary domain.

III. Our Solution

In this section, we introduce the CroRank algorithm and derive learning procedures to efficiently solve it.

A. CroRank

CroRank consists of two main steps. In MBMF, we try to rank items in the auxiliary domain and link the ranking structure with that of the target domain by multiple binomial matrix factorization. Then in TBPR, we consider connecting the preference structure of the auxiliary domain with the preference information hidden in the target domain to uplift ranking accuracy.

1) Multiple Binomial Matrix Factorization: Multiple binomial matrix factorization (MBMF) is inspired by [26], [27] and [5]. We give the graphical model of multiple binomial matrix factorization (MBMF) in Figure 2.

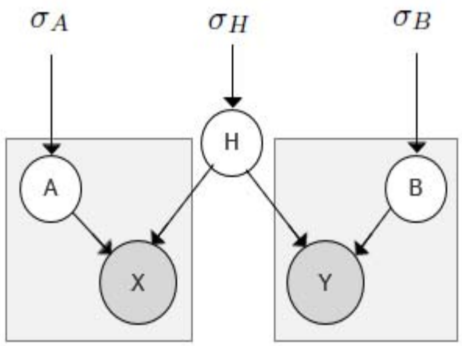


Figure 2. Graphical model of the multiple binomial matrix factorization

In Figure 2, we use X to denote the implicit feedback matrix of the auxiliary domain and use Y to denote the implicit feedback matrix of the target domain. A and B is the latent item matrix of the auxiliary domain and the target domain separately. The latent user matrix is represented as H. We place zero-mean spherical Gaussian priors on the user and item latent vectors, given all the prior parameters thus we get,

$$P(A|\sigma_A) = \prod_{l=1}^{L} \mathcal{N}(A_l|0, \sigma_A^2 I) \tag{1}$$

$$P(H|\sigma_H) = \prod_{u=1}^{U} \mathcal{N}(H_u|0, \sigma_H^2 I) \tag{2}$$

$$P(B|\sigma_B) = \prod_{m=1}^{M} \mathcal{N}(B_m|0, \sigma_B^2 I) \tag{3}$$

For modeling implicit feedback matrices, we adopt binomial assumptions,

$$p(X_{ul}|H_u, A_l) = \mathcal{B}(X_{ul}|S, \hat{x}_{ul}) \tag{4}$$

$$p(Y_{um}|H_u, B_m) = \mathcal{B}(Y_{um}|S, \hat{x}_{um}) \tag{5}$$

Notice that S equals to 1 in MBMF. Similar to BMF [27], the user u's inclination for the item i is expressed as a sigmoid function,

$$\hat{x}_{ul} = \frac{1}{1 + e^{-H_u A_l^T}} \tag{6}$$

And the experssion of $\hat{x}_{um}$ takes a similar manner.

In regard to link the preference structures of the auxiliary and target domain, we correlate them by the shared latent user matrix H in Eq. 4 and Eq. 5. Intuitively, if a user u prefers both item l in the auxiliary domain and item m in the target domain, the latent vectors of the item l and the item m tend to be pushed alike by factorizing the corresponding values of the implicit feedback matrices simultaneously.

2) Transfer Bayesian Personalized Ranking: In *transfer Bayesian personalized ranking* (TBPR), we unite *inner domain preference* with *related item preference* to model user's personalized ranking structure in the target domain. This is the *related item effect* assumption of CroRank.

For two items i and j in the target domain, the preference structure of the user u for the item i and the item j can be written as,

$$\hat{x}_{ui} > \hat{x}_{uj} \qquad i,j \in I_t \text{ and } i \neq j$$

For *related item group* of the item i and the item j in the auxiliary domain, the user u's preference structure is,

$$\hat{x}_{uI_i^g} > \hat{x}_{uI_j^g} \quad i,j \in I_t \text{ and } i \neq j \quad I_i^g, I_j^g \subseteq I_a$$

To join the two kinds of effects, we linearly combine the two preference information,

$$\hat{x}_{ui} + \hat{x}_{uI_i^g} > \hat{x}_{uj} + \hat{x}_{uI_j^g} \;\; i,j \in I_t \text{ and } i \neq j \;\; I_i^g, I_j^g \subseteq I_a$$

For clarity, we substitute $\hat{x}^u_{i+I_i^g}$ for $\hat{x}_{ui} + \hat{x}_{uI_i^g}$, thus the we reach the following formulation,

$$\hat{x}^u_{i+I_i^g} > \hat{x}^u_{j+I_j^g} \quad i,j \in I_t \text{ and } i \neq j \quad I_i^g, I_j^g \subseteq I_a$$

For a user u, the likelihood function of TBPR can be written as,

$$TBPR(\succ_u) = \prod_{i \in I_+^u} \prod_{i \in I_-^u} p(\hat{x}^u_{i+I_i^g} > \hat{x}^u_{j+I_j^g})[1 - p(\hat{x}^u_{j+I_j^g} > \hat{x}^u_{i+I_i^g})] \tag{7}$$

Which is different from Bayesian personalized ranking [20] since we cover both *inner domain preference* and *related item preference*.

To identify *related item group*, for an item i in the target domain, we compute the similarity between B_i with the latent item vectors in the auxiliary domain. Where B_i is the latent item vector of item i. We choose cosine similarity to measure this distance, and there are also other choices such as Pearson similarity and Euclidean-based similarity. We compute I_i^g based on cosine similarity and fix the number of grouped items to 3. In the experiment setup, we also consider varying this parameter and examine the model responses.

Based on above analyses, we can write $\hat{x}^u_{i+I_i^g}$ as follows,

$$\hat{x}^u_{i+I_i^g} = \rho H_u \cdot b_i^T + (1-\rho) \frac{1}{|\mathcal{G}|} \sum_{g=1}^{|\mathcal{G}|} H_u \cdot a_g^{i^T} \tag{8}$$

In Eq. 8, the first part is the contribution of *inner domain preference* and the second part measures the influence of *related item preference*. And $\rho \in [0,1]$ is a tradeoff parameter that balances the two effects. When ρ equals to 1, CroRank is reduced to Bayesian personalized ranking. Cross validation can be conducted to empirically determine the value of ρ.

3) Learning the CroRank: To learn model parameters, we adopt the standard stochastic gradient descent (SGD) method based on different sampling strategies.

In MBMF, based on above assumptions, we can write the log-posterior likelihood function as,

$$\begin{aligned}
&\log p(A,H,B|X,Y,\Phi) \\
&= \log[p(A|0,\sigma_A^2)p(H|0,\sigma_H^2)p(B|0,\sigma_B^2)p(X|A,H)p(Y|B,H)] + C \\
&= \sum_{u=1}^{U}\sum_{l=1}^{L}[\log \tbinom{S}{x_{ul}} - Slog(1+e^{-H_u A_l^T}) - (S-x_{ul})H_u A_l^T] \\
&+ \sum_{u=1}^{U}\sum_{m=1}^{M}[\log \tbinom{S}{y_{um}} - S\log(1+e^{-H_u B_m^T}) - (S-y_{um})H_u B_m^T)] \\
&- \frac{1}{2\sigma_A^2}\sum_{l=1}^{L} A_l A_l^T - \frac{1}{2\sigma_B^2}\sum_{m=1}^{M} B_m B_m^T - \frac{1}{2\sigma_H^2}\sum_{u=1}^{H} H_u H_U^T \\
&- \frac{1}{2}[KL\log\sigma_A^2 + KM\log\sigma_M^2 + KU\log\sigma_H^2] + C
\end{aligned} \tag{9}$$

Where Φ is the prior parameter set and C is used to denote all the constant values. Maximizing Eq. 9 is equivalent to minimize the following equation,

$$\begin{aligned}
\mathcal{L}(A,H,B) &= \frac{1}{2}\sum_{u=1}^{U}\sum_{l=1}^{L}[Slog(1+e^{-H_u A_l^T}) + (S-x_{ul})H_u A_l^T] \\
&+ \frac{1}{2}\sum_{u=1}^{U}\sum_{m=1}^{M}[Slog(1+e^{-H_u B_m^T}) + (S-y_{um})H_u B_m^T)] \\
&+ \frac{\lambda}{2}\sum_{l=1}^{L} A_l A_l^T + \frac{\lambda}{2}\sum_{m=1}^{M} B_m B_m^T + \frac{\lambda}{2}\sum_{u=1}^{H} H_u H_u^T
\end{aligned} \tag{10}$$

Since $\mathcal{L}(A,H,B)$ is differentiable, we use the standard stochastic gradient descent (SGD) to optimize $\mathcal{L}(A,H,B)$, thus we can find a local optimum of the parameters. The gradient in regard to the user-specific latent vector is given by,

$$\frac{\partial\mathcal{L}}{\partial H_u} = \frac{1}{2}(\frac{S}{1+e^{-H_u A_l^T}} - x_{ul})A_l + \frac{1}{2}(\frac{S}{1+e^{-H_u B_m^T}} - y_{um})B_m + \lambda H_u \tag{11}$$

Similarly, the gradients of $\mathcal{L}$ with respect to latent item vectors are given by,

$$\frac{\partial\mathcal{L}}{\partial A_l} = \frac{1}{2}(\frac{S}{1+e^{-H_u A_l^T}} - x_{ul})H_u + \lambda A_l \tag{12}$$

$$\frac{\partial\mathcal{L}}{\partial B_m} = \frac{1}{2}(\frac{S}{1+e^{-H_u A_l^T}} - y_{um})H_u + \lambda B_m \tag{13}$$

The detailed steps to solve MBMF are given in Algorithm 1.

In TBPR, for a user u, we try to optimize,

$$\arg\min_{\Theta_t}\mathcal{L}_t = \arg\min_{\Theta_t} -\frac{1}{2}\ln TBPR(\succ_{u_{i+I_i^g}}) + \frac{1}{2}\mathcal{R}(\Theta_t) \tag{14}$$

Based on previous assumptions, we can reach the following formulation,

$$\mathcal{L}_t = \prod_{i\in I_+^u}\prod_{j\in J_-^u} -\ln\frac{1}{1+e^{-(x_{i+I_i^u}^u - x_{j+I_j^u}^u)}} + \frac{1}{2}\mathcal{R}(\Theta_t) \tag{15}$$

Where $\Theta_t = \{B,H\}$, and $\mathcal{R}(\Theta_t)$ can be regarded as the regularization term to avoid overfitting. Similar to MBPR, We compute the gradients of $\mathcal{L}_t$ for each model parameter. The

Algorithm 1 The algorithm of Multiple Binomial Matrix Factorization (MBMF)

Input: Training data set $T_r = \{X,Y\}$, where X is the implicit feedback matrix of the auxiliary domain and Y is the implicit feedback matrix of the target domain. The max epoch parameter T. The learning rate parameter μ.
Output: The learned model parameters set $\Theta = \{A,H,B\}$.
For t_1 = 1,...,T.
  For t_2 = 1,...,$|U|$.
    Step.1 Randomly sample item $l \in I_a$.
    Step.2 Randomly sample item $m \in I_t$.
    Step.3 Update H_u by $H_u = H_u - \mu\frac{\partial\mathcal{L}}{\partial H_u}$ and $\frac{\partial\mathcal{L}}{\partial H_u}$ is given by Eq. 11.
    Step.4 Update A_l by $A_l = A_l - \mu\frac{\partial\mathcal{L}}{\partial A_l}$ and $\frac{\partial\mathcal{L}}{\partial A_l}$ is given by Eq. 12.
    Step.5 Update B_m by $B_m = B_m - \mu\frac{\partial\mathcal{L}}{\partial B_m}$ and $\frac{\partial\mathcal{L}}{\partial B_m}$ is given by Eq. 13.
  End
End

Algorithm 2 The algorithm of Transfer Bayesian Personalized Ranking (TBPR)

Input: Training data set $T_r = \{A,B,H,\{u,m\}\}$, where A is the learned latent item matrix of the auxiliary domain and B is the learned latent item matrix of the target domain. H is the learned latent user matrix. $\{u,m\}$ denotes the observed user-item pair of the target domain. The size of the $related\ item\ group$ $|\mathcal{G}|$. The max epoch parameter T. The learning rate parameter μ.
Output: The learned model parameters set $\Theta = \{B,H\}$.
For t_1 = 1,...,T.
  For t_2 = 1,...,$|U|$.
    Step.1 Randomly sample user $u \in U$
    Step.2 Randomly sample item $i \in I_+^t$.
    Step.3 Randomly sample item $j \in I_-^t$.
    Step.4 Derive the $related\ item\ group$ of size $|\mathcal{G}|$ for item i based on the cosine similarity measure.
    Step.5 Derive the $related\ item\ group$ of size $|\mathcal{G}|$ for item j based on the cosine similarity measure.
    Step.6 Update H_u by $H_u = H_u - \mu\frac{\partial\mathcal{L}_m}{\partial H_u}$ and $\frac{\partial\mathcal{L}_m}{\partial H_u}$ is given by Eq. 16.
    Step.7 Update B_i by $B_i = B_i - \mu\frac{\partial\mathcal{L}}{\partial B_i}$ and $\frac{\partial\mathcal{L}}{\partial B_i}$ is given by Eq. 17.
    Step.8 Update B_m by $B_j = B_j - \mu\frac{\partial\mathcal{L}}{\partial B_j}$ and $\frac{\partial\mathcal{L}}{\partial B_j}$ is given by Eq. 18.
  End
End

gradient of the user-specific parameter is given by,

$$\frac{\partial \mathcal{L}_m}{\partial H_u} = \frac{1}{1+e^{\Phi}}(\rho B_j^{\mathrm{T}} - \rho B_i^{\mathrm{T}} + (1-\rho)\frac{1}{|\mathcal{G}|}\sum_{g=1}^{|\mathcal{G}|} A_g^j - (1-\rho)\frac{1}{|\mathcal{G}|}\sum_{g=1}^{|\mathcal{G}|} A_g^i) + \lambda_\phi H_u \tag{16}$$

Similarly, the gradients for the item-specific parameters are,

$$\frac{\partial \mathcal{L}_m}{\partial B_i} = \frac{-H_u}{1+e^{\Phi}} + \lambda_\phi B_i \tag{17}$$

$$\frac{\partial \mathcal{L}_m}{\partial B_j} = \frac{H_u}{1+e^{\Phi}} + \lambda_\phi B_j \tag{18}$$

Where $\Phi = [\rho H_u \cdot B_i - \rho H_u \cdot B_j + (1-\rho)\frac{1}{|\mathcal{G}|}\sum_{g=1}^{|\mathcal{G}|} H_u \cdot A_i^g - (1-\rho)\frac{1}{|\mathcal{G}|}\sum_{g=1}^{|\mathcal{G}|} H_u \cdot A_j^g]$. The detailed optimization steps of TBPR is listed in Algorithm 2.

IV. EXPERIMENTS

In this section, we evaluate CroRank in several datasets to demonstrate the efficiency and effectiveness of CroRank over the state-of-the-art models for personalized ranking.

A. Datasets

we evaluate CroRank using a series of real-life datasets collected by [10]. These datasets contain a variety of categories ranging from movies to electronic goods, which is suitable for our study of cross domain effect. For demonstration, We choose $\{Books, Movies\}$ pair, $\{Music, Movies\}$ pair, $\{Toys, Sports\}$ pair and $\{Cloth, Shoes\}$ pair, since intuitively they are closely related categories. Due to the limitation of machines, we randomly sample a part of the datasets to conduct experiments. The statistics of the datasets is summarized in Table I.

Pairs	Dataset	users	items	user-item pairs
Books & Movies	Books	1411	2864	25066
	Movies	1411	4658	49735
Music & Movies	Music	1530	2541	18406
	Movies	1530	4308	60115
Toys & Sports	Toys	1203	2571	7053
	Sports	1203	4229	12410
Clothes & Shoes	Clothes	1341	2512	7877
	Shoes	1341	2123	11206

Table I. Statistics of the Datasets

Meanwhile, we pre-process the rating matrices that only the rating values that higher than 3 are collected as positive feedback to form the implicit feedback matrices.

B. Evaluation Methods and Metrics

For all the datasets, we randomly extract half of the observed $\{user, item\}$ pair as training data, and the rest half is served as test data. We repeat this process 3 times to generate 3 copies of the training data and test data, and report the final results after averaging the performance across those 3 copies of data.

After we have learned the model parameters $\Theta_t = \{B, H\}$, the preference score of user u for item i can be computed as $S_{ui} = \rho H_u \cdot b_i^T + (1-\rho)\frac{1}{|\mathcal{G}|}\sum_{g=1}^{|\mathcal{G}|} H_u \cdot {a_g^i}^T$. Thus we can obtain a personalized ranking list for the user u based on the value of S_{ui} for each item i in the test dataset. A large value of S_{ui} suggests that the user u prefers item i over other items.

There are a variety of metrics for assessing ranking-oriented algorithms for collaborative filtering. The most widely used ones are $Pre@k$ [25], $NDGG@k$ [14] and $Rec@k$ [15]. And we can derive $F1$ score [15] based on $Pre@k$ and $Rec@k$. For each of the evaluation metric, we obtain the corresponding performance for each user u in the test data and then average the performance over all users to attain the ultimate results.

C. Compared Algorithms

We implement several baseline algorithms for comparison , and we give simple summaries below.

MP: Most Popular (MP) is a basic algorithm for solving one-class collaborative filtering problem, which ranks the items based on their popularity among users.

BPR: This the seminal work that solves the one-class collaborative filtering problem by means of a pairwise oriented approach.

GBPR: Group preference based Bayesian personalized ranking (GBPR) is propose in [15]. It relaxes the constraint of the assumptions of BPR by introducing $group\ preference$.

D. Parameter Settings

For all the experiments, we choose the optimization iteration number $T \in \{10^3, 10^4, 10^5\}$. We first fix the tradeoff parameter as $\rho = 0.4$, and then search $\rho \in \{0.2, 0.4, 0.6, 0.8\}$. The learning rate μ of BRP, GBPR and CroRank is fixed as 0.01. The regularization terms are searched from $\{0.001, 0.01, 0.1\}$. We first fix the size of the *related item group* to be 3, and then vary $|\mathcal{G}| \in \{1, 2, 3, 4, 5\}$. The latent dimension K is fixed as 20 as in [15].

E. Summary of Experimental Results

The experimental results are shown in Table II. We take $Movies$, $Shoes$ and $Sports$ as target domains while take $Books$, $Music$, $Clothes$ and $Toys$ as auxiliary domains. The best result among all the method is in boldface. We summarize the results in two points,

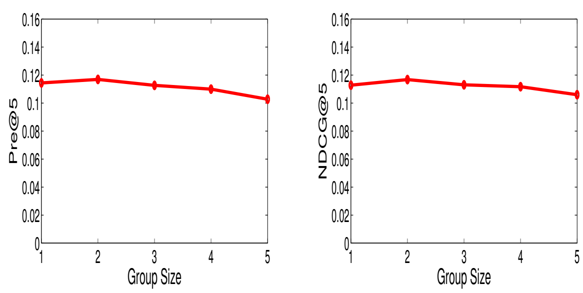
- Compared with MP, the *pairwise approaches* are much better, which shows that the *pairwise assumptions* indeed capture the user's preference structure.
- CroRank achieves the best results on all the four datasets, which demonstrates that incorporating cross domain information can promote better personalized ranking performance than only modeling *inner domain preference*. Meanwhile, allowing to unify multiple data source can induce more flexibility. And flexibility usually plays an important role since the diversity of real applications.

Datasets	Model	Pre@5	Rec@5	F1@5	NDCG@5
Books & Movies	MP	0.0623	0.0467	0.0534	0.0661
	BPR	0.0969	0.0690	0.0806	0.0979
	GBPR	0.1021	0.0698	0.0829	0.1098
	CroRank	**0.1258**	**0.0928**	**0.1068**	**0.1274**
	Improv.	23.21%	32.95%	28.83%	16.03%
Music & Movies	MP	0.0422	0.0318	0.0320	0.0456
	BPR	0.0624	0.0513	0.0630	0.0602
	GBPR	0.0718	0.0654	0.0685	0.0696
	CroRank	**0.0984**	**0.0733**	**0.0840**	**0.0997**
	Improv.	37.05%	12.08%	22.63%	43.25%
Clothes & Shoes	MP	0.1009	0.1322	0.1144	0.1179
	BPR	0.1490	0.2828	0.1952	0.1788
	GBPR	0.1524	0.3188	0.2028	0.1893
	CroRank	**0.2041**	**0.4281**	**0.2706**	**0.2310**
	Improv.	33.92%	34.28%	33.43%	22.03%
Toys & Sports	MP	0.0311	0.0421	0.0362	0.0298
	BPR	0.0574	0.0783	0.0662	0.0528
	GBPR	0.0563	0.0745	0.0641	0.0510
	CroRank	**0.0850**	**0.0879**	**0.0864**	**0.0870**
	Improv.	48.08%	12.26%	34.79%	64.77%

Table II. Ranking Performance of MP, BPR, GBPR and CroRank on Different Datasets.

F. Analysis and Discussion

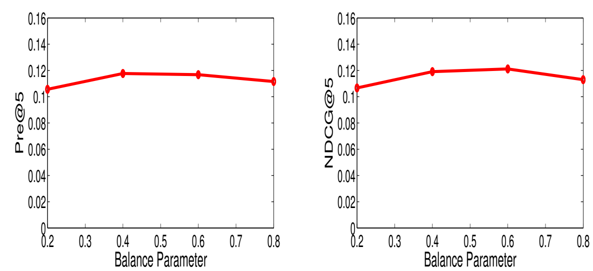
1) Group Size Analysis: To study the affect of the size of the *related item group*, we adjust the group size $|\mathcal{G}|$ $\in$ $\{1, 2, 3, 4, 5\}$, and we show the results of $Pre@5$ and $NDCG@5$ of the datasets $\{Books, Movies\}$ in Figure 3. The balance parameter is fixed as 0.4. We can recognize that a relatively larger $|\mathcal{G}|$ corresponds to a better performance. Intuitively, to incorporate more related items can prevent the user u's preference structure of the target domain to be biased, so the ranking performance tends to be better. Meanwhile, a large group size $|\mathcal{G}|$ may introduce noises, so the ranking performance does not grow continually as more related items are added, the best performance is achieved when the group size is around 2 or 3, which we can find in Figure 3.



(a) (b)

Figure 3. Ranking performance of CroRank with different sizes of *related item group*.

2) Balance Parameter Analysis: The balance parameter ρ controls the influence of the *inner domain preference* and the *related item preference*. When ρ equals to 1, CroRank reduces to Bayesian personalized ranking (BPR). We adjust $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ to analyze the effect of the balance parameter. We present the results of $Pre@5$ and $NDCG@5$ of datasets $\{Books, Movies\}$ in Figure 4. We can find that the ranking performance is relatively better when ρ is around 0.4 or 0.6. It is can be explained that the ranking structure of a particular user cannot be determined by the *inner domain preference* or *related domain preference* only, which is in accordance with our intuition. Combining these two impacts reasonably can exclude noises and lead to better ranking performance.



(a) (b)

Figure 4. Ranking performance of CroRank with different balance parameters.

3) Sampling Strategy Analysis: The sampling strategy for selecting training instances may have significant impact on the recommendation performance [19], [29]. For each user u, we investigate three sampling methods for choosing training instances, and adopt $Pre@5$ for evaluation.

i. Uniform Sampling: all the positive and negative feedback are drawn uniformly from I_+^u and I_-^u.
ii. Adaptive Sampling: For each user u, the positive instances are sampled uniformly from I_+^u, and the negative instances are sampled from the following distribution,

$$p(i \in I_-^u | u) \propto exp(\frac{\hat{x}_{ui}}{\lambda}), \lambda \in \mathbb{R}^+. \quad (19)$$

	Uniform Sampling	Adaptive Sampling	Dynamic Negative Sampling
Books & Movies	0.1249	0.1121	0.1102
Toys & Sports	0.0852	0.0834	0.0810

Table III. Sampling Strategy Analysis for CroRank.

iii. Dynamic Negative Sampling: In [28], Zhang et al. propose a rejection sampling strategy for sampling negative instances. Here we adopt a linear weight function. We uniformly sample $i,j \in I^u_-$ and compute $\hat{x}_{ui}$ and $\hat{x}_{uj}$. If $\hat{x}_{ui} > \hat{x}_{uj}$, return i as the selected one with probability $\frac{1}{1+\beta}$ or return j otherwise, if $\hat{x}_{uj} > \hat{x}_{ui}$, return j as the selected one with probability $\frac{1}{1+\beta}$ or return i otherwise. Where β is set to 1 as in [28]. The positive instances are sampled uniformly from I^u_+.

Table III shows the $Pre@5$ values of CroRank based on different sampling strategies on the datasets $\{Books, Movies\}$ and $\{Toys, Sports\}$. We notice that the uniform sampling method achieves best results among all the sampling strategies. As for adaptive and dynamic negative sampling, the training instances are selected based on intermediate results, and the model parameters have not converged yet, this may account for the effectiveness of uniform sampling over adaptive and dynamic negative sampling.

V. Related Work

We tersely review related works in two main tracks : one-class collaborative filtering and transfer learning for collaborative filtering.

One-class collaborative filtering problem one-class collaborative filtering problem emerges recently due to the explicit feedback are sometimes hard to accumulate in real-world applications. There are two main trends for solving one-class collaborative filtering problem and can be roughly classified as (1) pointwise approach, and (2) pairwise approach.

Pointwise approach try to learn latent representations of users and items to fit the absolute numeric scores. The preference scores are usually fitted by matrix factorization techniques. [4] classifies the feedback as positive and negative ones based on a notion of $confidence\ level$ and then adopt a latent factor matrix factorization associated with the $confidence\ level$ to solve the one-class collaborative filtering problem. In [12], the authors propose two frameworks to alleviate this problem, one is to randomly sample the negative examples, and the other is to introduce several new weighting schemes to incorporate more user-item interactions.

Pairwise approaches take a different view of one-class collaborative filtering problem and aim at modeling the pairwise interactions rather than fitting rating scores directly. [20] introduce the basic Bayesian personalized ranking framework. Many other works that relate to pairwise models are mainly originated from [20]. In [29], Zhao et al. extend the BPR by leveraging social connections of users to improve personalized ranking performance. Pan et al. [15] consider the $group\ preference$ information to alter basic assumptions of BPR. In [14], the authors try to aggregate related items to form item-sets, and model user's preference level over the item-sets which is maybe more reasonable the original assumptions of BPR. [1] adds a social regularization term to BPR to promote better performance. Tag recommendation task is considered in [21] by extending BPR from matrix factorization techniques to tensor factorization.

Transfer learning for collaborative filtering Transfer learning is a emerging research field of machine learning aiming at transferring the feature space of one domain into another domain to contribute to classification, regression and clustering performance. And recently researchers in collaborative filtering field have tried to apply transfer learning to recommendation task so that the auxiliary data can help improve recommendation accuracy.

In [9], Li et al. propose a rating-matrix generative method showing that the relatedness of multiple rating matrices can be dig out by means of revealing a common implicit cluster-level rating matrix to promote better rating prediction performance. [16] tentatively deliver the rating knowledge from some auxiliary data source in binary form to a target numerical rating matrix, thus alleviate the data sparseness problem and achieve higher rating prediction accuracy. Due to the unavailability of precise rating values, [17] tries to utilize the uncertain ratings of the auxiliary data to bring a benefit for rating prediction of the target domain. In [8], the authors try to solve the data sparsity problem by transferring user-item rating patterns from a denser auxiliary rating matrix.

Nevertheless, all the previous works that leverage transfer learning technique for collaborative filtering problem only concentrate on numerical rating prediction but ignore personalized ranking problem. Furthermore, the cross domain preference structure information is disregarded by most of the previous works. CroRank is a novel algorithm that try to transfer the ranking structure of the auxiliary domain into the target domain, and empirical study suggests that our proposed solution outperforms the methods without using auxiliary data.

VI. Conclusion

In this paper, we extend Bayesian personalized ranking by incorporating the auxiliary domain ranking structure, thus the ranking information hidden in the auxiliary domain and target domain can be fused together to complete the user's preference profile. We reformulate the original assumptions of BPR to provide a sounder modeling of item pairwise interactions, and this leads to better experimental results than the state-of-the-art models. The optimization procedure of CroRank is efficient by means of stochastic gradient descent (SGD).

In the future work, we plan to extend CroRank in three ways. (1) Exploiting more compact methods to find *related item group* and transfer preference structure. (2) Incorporating social information and group preference to promote better ranking performance. (3) Developing more efficient sampling strategy for selecting training instances.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (NSFC) No. 61272303 and Natural Science Foundation of Guangdong Province No. 2014A030310268.

REFERENCES

[1] Liang Du, Xuan Li, and Yi-Dong Shen. User graph regularized pairwise matrix factorization for item recommendation. In Jie Tang, Irwin King, Ling Chen, and Jianyong Wang, editors, *Advanced Data Mining and Applications*, volume 7121 of *Lecture Notes in Computer Science*, pages 372–385. Springer Berlin Heidelberg, 2011.

[2] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992.

[3] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 595–606, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

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

[5] Yong-Deok Kim and Seungjin Choi. Bayesian binomial mixture model for collaborative prediction with non-random missing data. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 201–208, New York, NY, USA, 2014. ACM.

[6] 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*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[7] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 173–182, New York, NY, USA, 2012. ACM.

[8] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 2052–2057, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[9] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 617–624. ACM, 2009.

[10] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172. ACM, 2013.

[11] Nathan Srebro Nati and Tommi Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

[12] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 502–511, Washington, DC, USA, 2008. IEEE Computer Society.

[13] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[14] Weike Pan and Li Chen. *CoFiSet: Collaborative Filtering via Learning Pairwise Preferences over Item-sets*, chapter 19, pages 180–188.

[15] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2691–2697. AAAI Press, 2013.

[16] Weike Pan, Nathan N. Liu, Evan W. Xiang, and Qiang Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2318–2323. AAAI Press, 2011.

[17] Weike Pan, Evan Wei Xiang, and Qiang Yang. Transfer learning in collaborative filtering with uncertain ratings. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* AAAI Press, 2012.

[18] Steffen Rendle. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[19] 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*, WSDM '14, pages 273–282, New York, NY, USA, 2014. ACM.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[21] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 81–90, New York, NY, USA, 2010. ACM.

[22] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *In ICML 08: Proceedings of the 25th International Conference on Machine Learning*, 2008.

[23] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

[25] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, New York, NY, USA, 2012. ACM.

[26] Koh Takeuchi, Katsuhiko Ishiguro, Akisato Kimura, and Hiroshi Sawada. Non-negative multiple matrix factorization. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013.

[27] Jinlong Wu. Binomial matrix factorization for discrete collaborative filtering. In *ICDM*, pages 1046–1051, 2009.

[28] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 785–788, New York, NY, USA, 2013. ACM.

[29] 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*, CIKM '14, pages 261–270, New York, NY, USA, 2014. ACM.