Collaborative Ranking with Social Relationships for Top-N Recommendations

Dimitrios Rafailidis
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
draf@csd.auth.gr

Fabio Crestani
Faculty of Informatics
Università della Svizzera italiana (USI)
Lugano, Switzerland
fabio.crestani@usi.ch

ABSTRACT

Recommendation systems have gained a lot of attention because of their importance for handling the unprecedentedly large amount of available content on the Web, such as movies, music, books, etc. Although Collaborative Ranking (CR) models can produce accurate recommendation lists, in practice several real-world problems decrease their ranking performance, such as the sparsity and cold start problems. Here, to account for the fact that the selections of social friends can leverage the recommendation accuracy, we propose `SCR`, a Social CR model. Our model learns personalized ranking functions collaboratively, using the notion of Social Reverse Height, that is, considering how well the relevant items of users and their social friends have been ranked at the top of the list. The reason that we focus on the top of the list is that users mainly see the top-N recommendations, and not the whole ranked list. In our experiments with a benchmark data set from Epinions, we show that our `SCR` model performs better than state-of-the-art CR models that either consider social relationships or focus on the ranking performance at the top of the list.

CCS Concepts

•Information systems → Collaborative and social computing systems and tools;

Keywords

Recommendation systems; collaborative ranking; social relationships

1. INTRODUCTION

The collaborative filtering strategy has been widely followed in recommendation systems, where users with similar preferences tend to get similar recommendations [5]. User preferences are expressed explicitly in the form of ratings or implicitly in the form of number of views, clicks, purchases, etc. Relevant studies examine how to predict the rating of a user on an unseen item, also known as the rating prediction problem [8]. Instead of focusing on the accuracy of rating prediction, ranking-based models try to produce accurate ranked lists for the top-N recommendation problem [9]. In relevant studies [2, 13], it has been shown that ranking-based models achieve higher recommendation accuracy than rating prediction-based approaches in the top-N recommendation problem.

Collaborative Ranking (CR) models learn a personalized scoring/ranking function to rank the recommended items for each individual, while all functions are constructed collaboratively across all the users [9, 10]. In [2], authors introduce CR models that perform push at the top of the recommendation list, by considering that what matters in a recommendation system is the ranking performance at the top of the list, that is, the items the user will actually see. In doing so, the CR models that focus on the ranking performance at the top of the list achieve high ranking accuracy in the top-N recommendation problem.

Although CR models can generate accurate recommendation lists, in practice several real-world problems decrease the recommendation accuracy, such as the sparsity and the cold-start problems [5]. Several approaches have been proposed for handling them in the rating prediction problem, assuming that users tend to trust the recommendations of their social friends [1]. However, a little effort has been made to exploit social relationships in CR models, such as the studies in [6, 14]. Nonetheless, the CR models in [6, 14] do not focus on the ranking performance at the top of the list, when learning the ranking functions.

Therefore, a pressing challenge resides on generating accurate recommendations able to exploit the social relationships that simultaneously focus on the top of the list. In this paper, we propose a Social CR model, namely `SCR`. Our CR model learns the personalized ranking function of each individual in a collaborative manner using the notion of Social Reverse Height. When learning the ranking functions, `SCR` attempts to push up users' relevant items above the irrelevant ones at the top of the list, by considering the ranking positions that the relevant items of their friends have. We formulate the objective function of the proposed `SCR` model as a minimization problem, and we provide an efficient optimization algorithm based on alternating minimization [4] and gradient descent. Our experiments on a data set from Epinions[1] demonstrate the superiority of our `SCR` model over state-of-the-art CR models.

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2914711

[1] http://www.epinions.com/

2. RELATED WORK

Learning to rank methods have been widely studied in Information Retrieval and in recommendation systems. The goal is to define a ranking function on each item, and then learn toward some loss functions. Liu et al. [7] categorize learning to rank methods into point-wise, list-wise and pair-wise. In short, point-wise approaches predict ranking scores for individual items. List-wise approaches consider an individual training example as an entire list of items and use loss functions to express the distance between the reference list and the output list from the ranking model. Representative list-wise approaches in recommendation systems are `CofiRank` [12] and `CLiMF` [10], which use loss functions based on Normalized Discounted Cumulative Gain and Reciprocal Rank, respectively. Pair-wise approaches make a prediction for every pair of items concerning their relative ordering in the final list. A pair-wise recommendation approach is the Bayesian Personalized Ranking framework (`BPR`) [9], a CR model that learns the personalized ranking functions collaboratively. To account for the fact that users focus on recommendations at the top of the list, in [2] authors introduce optimization algorithms for three CR models `P-Push`, `Inf-Push` and `RH-Push`, which follow different strategies when learning the personalized ranking functions. `P-Push` tries to push irrelevant items in the top-N list below those that have been selected as relevant; `Inf-Push` attempts to move the most irrelevant item in the list below all the relevant ones; while `RH-Push` tries to move the relevant items above the irrelevant ones. However, all the aforementioned models do not exploit the social relationships in the learning process.

Although several approaches incorporate social relationships in the rating prediction problem, e.g. [8], a few attempts have been made in the top-N recommendation problem. Rendle et al. [6] present the `MR-BPR` model, where they combine Multi-Relational matrix factorization with the `BPR` framework [9] to model both users' feedback on items and on social relationships. Zhao et al. [14] propose `SBRP`, a Social Bayesian Personalized Ranking model that incorporates social relationships into a pair-wise ranking model, assuming that users tend to assign higher ranks to items that their friends prefer. However, `MR-BPR` and `SBRP` do not focus on the ranking performance at the top of the list.

3. PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{I}$ be the sets of users and items, respectively. We assume that users express their preferences over the items by marking them as relevant or irrelevant (either via explicit or implicit feedback), stored in a $(|\mathcal{U}| \times |\mathcal{I}|)$ matrix R, with $[R]_{ij}$=1 if user i has selected item j as relevant, -1 as irrelevant, and 0 if user i has not expressed her preference over item j. For each user i the sets of relevant and irrelevant items are denoted as $\mathcal{I}_i^+$ and $\mathcal{I}_i^-$, with $x_i^+ \in \mathcal{I}_i^+$, $x_i^- \in \mathcal{I}_i^-$ and $n_i = |\mathcal{I}_i^+ \cup \mathcal{I}_i^-|$ being the total number of marked items by user i. For each user $i \in \mathcal{U}$, we define the Reverse Height of a relevant item x_i^+ as follows [2]:

DEFINITION 1. *The **Reverse Height** $RH_i(x_i^+)$ of a relevant item x_i^+ in the recommendation list of user i is the number of irrelevant items ranked above x_i^+.*

In our setting, we assume that each user i has a set $\mathcal{F}_i$ of friends with trusted social relationships. Given $x_i^+ \in \mathcal{I}_i^+$, $x_i^- \in \mathcal{I}_i^-$ and $x_w^+ \in \mathcal{I}_w^+$, with $w \in \mathcal{F}_i$, we define the Social Reverse Height of a relevant item x_i^+ as follows:

DEFINITION 2. *The **Social Reverse Height** $SRH_i(x_i^+)$ of a relevant item x_i^+ in the recommendation list of user i is the sum of (1) the number of irrelevant items x_i^- ranked above x_i^+; and (2) the number of the irrelevant items x_i^- ranked above the relevant items x_w^+ of all her $|\mathcal{F}_i|$ friends.*

DEFINITION 3. *The **goal of the proposed** `SCR` **model** is to learn the personalized ranking functions collaboratively by trying to minimize $SRH_i(x_i^+)$, that is, pushing up the relevant items x_i^+ of user i above the irrelevant items x_i^-, by considering the ranking positions of the relevant items x_w^+ of her $|\mathcal{F}_i|$ friends.*

4. SOCIAL COLLABORATIVE RANKING

4.1 Social Reverse Height

A trusted friend $w \in \mathcal{F}_i$ might point to an interesting item that does not match the preferences of user i [1]. Hence, to measure the social influence between user i and her friend w we define the *social regularization term* $s_{iw} = |\mathcal{I}_i^+ \cap \mathcal{I}_w^+|$, which is the intersection of the sets of relevant items of user i and her friend w. For each user i we normalize s_{iw} in $[0, 1]$, by dividing each s_{iw} value with the maximum value of all the $|\mathcal{F}_i|$ friends.

Given a personalized ranking function $r_i(x)$, with $i \in \mathcal{U}$ and $x \in \mathcal{I}$, according to Definition 2 we have:

$$\begin{aligned} SRH_i(x_i^+) = &\sum_{x_i^- \in \mathcal{I}_i^-} \Big\{ \mathbb{1}[r_i(x_i^+) \leq r_i(x_i^-)] \\ &+ \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum_{x_w \in \mathcal{I}_w^+} s_{iw} \mathbb{1}[r_i(x_w^+) \leq r_i(x_i^-)] \Big\} \end{aligned} \quad (1)$$

where $\mathbb{1}$ is the indicator function and s_{iw} the social regularization term. Note that $r_i(x_w^+)$ expresses the ranking of the relevant item x_w^+ of friend w using the ranking function of user i. The above optimization problem in Eq. (1) is intractable due to the non-convex indicator function $\mathbb{1}$. Thus, we take a surrogate $g_i(\cdot)$ as follows:

$$\begin{aligned} g_i(x_i^+, x_i^-) &= r_i(x_i^+) - r_i(x_i^-) \\ &+ \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum_{x_w \in \mathcal{I}_w^+} s_{iw} \big(r_i(x_w^+) - r_i(x_i^-)\big) \end{aligned} \quad (2)$$

4.2 Objective Function

In the proposed `SCR` model we follow a collaborative strategy, instead of learning a ranking function $r_i(\cdot)$ for each individual i. Given the low-rank d decomposition $\hat{R} \approx U^T V$ of the initial matrix R, with U and V being $(d \times |\mathcal{U}|)$ and $(d \times |\mathcal{I}|)$ matrices, we consider $\mathbf{u}_i$ and $\mathbf{v}_j$ as the $(d \times 1)$ latent vectors of user i and item j respectively, that is, the i-th and j-th columns of U and V, respectively. For readability, we denote the relevant item of user i as $a = x_i^+$, the irrelevant item of user i as $b = x_i^-$, and the relevant item of a friend $w \in \mathcal{F}_i$ as $c = x_w^+$. Using the representation of the latent factors $\mathbf{u}_i$ and $\mathbf{v}_j$, Eq. (2) can be rewritten as:

$$\begin{aligned} g_i(a, b) &= \mathbf{u}_i^T(\mathbf{v}_a - \mathbf{v}_b) \\ &+ \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum_{c \in \mathcal{I}_w^+} s_{iw}(\mathbf{u}_w^T(\mathbf{v}_c - \mathbf{v}_b)) \end{aligned} \quad (3)$$

where $a \in \mathcal{I}_i^+$, $b \in \mathcal{I}_i^-$ and $c \in \mathcal{I}_w^+$. As the logistic loss function is a convex upper bound to the indicator function $\mathbb{1}$, based on Eq. (3) we can reformulate Eq. (1) as:

$$SRH_i(a) = \sum_{b \in \mathcal{I}_i^-} \Big(\log\left(1 + \exp\left(-g_i(a,b)\right)\right) \Big) \tag{4}$$

According to Definition 3, for all the relevant items $a \in \mathcal{I}_i^+$ we have to minimize the $SRH_i(a)$ value of Eq. (4). Hence, $\forall$ $i \in \mathcal{U}$ the goal is to minimize the following objective function (log-log loss function [2]) with respect to matrices U and V:

$$\mathfrak{L}(U,V) = \\ \sum_{i=1}^{|\mathcal{U}|} \left\{ \frac{1}{n_i} \sum_{a \in \mathcal{I}_i^+} \log(1 + SRH_i(a)) \right\} + \frac{\lambda}{2} \Big(||U||^2 + ||V||^2 \Big) \tag{5}$$

where the last term, called regularization term, is used to avoid model overfitting, and λ is the regularization parameter.

4.3 Model Learning

The goal is to learn (compute) matrices U and V to minimize the loss function $\mathfrak{L}(U,V)$ in Eq. (5). We follow the alternating minimization strategy [4] using the gradient descent optimization algorithm, where we update U keeping V fixed, and then, update V keeping the updated U fixed. Based on the (sub)gradients of objective function $\mathfrak{L}(U,V)$ with respect to $\mathbf{u}_i$ and $\mathbf{v}_j$, the update rules for each iteration $t+1$ are:

$$\mathbf{u}_i^{t+1} \leftarrow \mathbf{u}_i^t - \eta \nabla_{\mathbf{u}_i} \mathfrak{L}(U^t, V^t), \quad i = 1 \dots |\mathcal{U}| \tag{6}$$

$$\mathbf{v}_j^{t+1} \leftarrow \mathbf{v}_j^t - \eta \nabla_{\mathbf{v}_j} \mathfrak{L}(U^{t+1}, V^t), \quad j = 1 \dots |\mathcal{I}| \tag{7}$$

where η controls the learning rate.

4.3.1 Computing the Gradient $\nabla_{\mathbf{u}_i} \mathfrak{L}(U,V)$

We apply the chain rule to Eq. (5) to take the gradient of $\mathfrak{L}$ with respect to $\mathbf{u}_i$, that is, the composition of gradients $\nabla_{\mathbf{u}_i} g_i(a,b)$ and $\nabla_{g_i(a,b)} SRH_i(a)$ with respect to $\mathbf{u}_i$ in Eq. (3) and $g_i(a,b)$ in Eq. (4), respectively. Note that the gradient of the regularization term in Eq. (5) with respect to $\mathbf{u}_i$ equals $\lambda \mathbf{u}_i$. Thus, provided that $i \neq w$, we have:

$$\nabla_{\mathbf{u}_i} g_i(a,b) = \mathbf{v}_a - \mathbf{v}_b \tag{8}$$

$$\nabla_{g_i(a,b)} SRH_i(a) = \frac{-1}{1 + \exp(g_i(a,b))} \tag{9}$$

the gradient of $\mathfrak{L}$ with respect to $\mathbf{u}_i$ in Eq. (5) equals:

$$\nabla_{\mathbf{u}_i} \mathfrak{L}(U,V) = \\ \frac{1}{n_i} \sum_{a \in \mathcal{I}_i^+} \left\{ \frac{1}{1 + SRH_i(a)} \sum_{b \in \mathcal{I}_i^-} \frac{1}{1 + \exp(g_i(a,b))} (\mathbf{v}_b - \mathbf{v}_a) \right\} + \lambda \mathbf{u}_i \tag{10}$$

4.3.2 Computing the Gradient $\nabla_{\mathbf{v}_j} \mathfrak{L}(U,V)$

When calculating the gradient of $g_i(a,b)$ with respect to $\mathbf{v}_j$ in Eq. (3), there are the following two cases:

$$\nabla_{\mathbf{v}_j} g_i(a,b) = \begin{cases} \mathbf{u}_i, & \text{if } j{=}a. \\ -\big[\mathbf{u}_i + \frac{1}{|\mathcal{F}_i|} \sum\limits_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum\limits_{c \in \mathcal{I}_w^+} s_{iw} \mathbf{u}_w \big], & \text{if } j{=}b. \end{cases} \tag{11}$$

Similarly, we apply the chain rule to Eq. (5), to compute the gradient of $\mathfrak{L}$ with respect to $\mathbf{v}_j$, that is, the composition of the gradient $\nabla_{\mathbf{v}_j} g_i(a,b)$ based on the two cases in Eq. (11), and the gradient $\nabla_{g_i(a,b)} SRH_i(a)$, as presented in Eq. (9). The gradient of the regularization term in Eq. (5) with respect to $\mathbf{v}_j$ equals $\lambda \mathbf{v}_j$. Let $\mathcal{U}_j^+$ and $\mathcal{U}_j^-$ be the sets of users who selected item j as relevant and irrelevant, respectively. We can divide the sum over all users into sums over $\mathcal{U}_j^-$ and $\mathcal{U}_j^+$ [2], in order to calculate the gradient of $\mathfrak{L}$ with respect to $\mathbf{v}_j$ as follows:

$$\nabla_{\mathbf{v}_j} \mathfrak{L}(U,V) = \\ \sum_{i \in \mathcal{U}_j^-} \frac{1}{n_i} \sum_{j \in \mathcal{I}_i^-} \left\{ \frac{1}{1 + SRH_i(a)} \right. \\ \left. \sum_{a \in \mathcal{I}_i^+} \frac{1}{1 + \exp(g_i(a,b))} \big[\mathbf{u}_i + \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum_{c \in \mathcal{I}_w^+} s_{iw} \mathbf{u}_w \big] \right\} \\ - \sum_{i \in \mathcal{U}_j^+} \frac{1}{n_i} \sum_{j \in \mathcal{I}_i^-} \left\{ \frac{1}{1 + SRH_i(a)} \sum_{a \in \mathcal{I}_i^+} \frac{1}{1 + \exp(g_i(a,b))} \mathbf{u}_i \right\} + \lambda \mathbf{v}_j \tag{12}$$

Having computed U and V based on Eqs. (6) and (7) with the respective gradients in Eqs. (10) and (12), we calculate the low-rank d approximation $\hat{R} \approx U^T V$ of the initial $(|\mathcal{U}| \times |\mathcal{I}|)$ matrix R, and we generate a recommendation list by sorting the items in the respective i-th row of $\hat{R}$ in descending order, for each user i.

5. EXPERIMENTAL EVALUATION

5.1 Data Set & Evaluation Protocol

In our experiments we used a publicly available data set[2] from Epinions, which consists of 71,002 users and 104,356 items. The total number of ratings is 571,235 at a 5-star scale, where we considered 4-5 star ratings as relevant to a user and 1-3 as irrelevant. The total number of trusted social relationships is 508,960. The data set is divided into two sub-sets: the training set and the test set. Following the evaluation protocol of similar studies [13, 14], for users with less than five ratings, one randomly selected rating is inserted into the test set. For users with five ratings or more, 10% of the randomly selected ratings are moved to the test set. The training set is further split into two subsets: the cross-validation training set and the cross-validation test set, which are used to determine the tuning parameters of each examined ranking model. We repeated our experiment ten times, and we report mean values and standard deviations on the (actual) test set over the runs.

We evaluated the performance of the examined ranking models in terms of Recall ($R@N$), which is defined as the ratio of the relevant items in the top-N ranked list over all the relevant items for each user. In addition, we used the Normalized Discounted Cumulative Gain ($NDCG@N$) metric, which considers the ranking of the relevant items in the top-N list. For each user the Discounted Cumulative Gain is defined as:

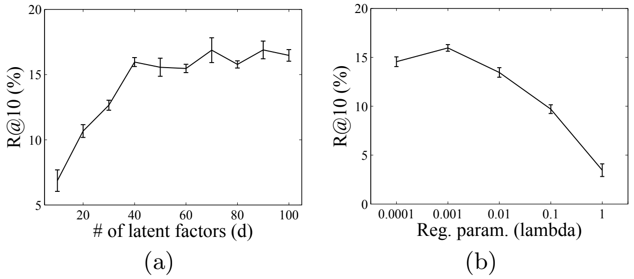
$$DCG@N = \sum_{j=1}^{N} \frac{2^{rel_j} - 1}{\log_2 j + 1} \tag{13}$$

where rel_j represents the relevance score of the item j, binary relevance in our case, that is, relevant or irrelevant to the user. $NDCG@N$ is the ratio of $DCG@N$ over the ideal $iDCG@N$ value for each user, that is, the $DCG@N$ value given the ratings in the test set. In our experiments we averaged $R@N$ and $NDCG@N$ over all users.

[2] http://alchemy.cs.washington.edu/data/epinions/

5.2 Results

To evaluate the parameter sensitivity in SCR, we varied the learning rate η in the range of $[10^{-5}\ \ 10^{-2}]$ and we concluded in η=10^{-4}. This means that a more conservative learning strategy is required, as higher values of η decreased the performance of SCR, by falling in local optima when minimizing the loss function in Eq. (5). In Figure 1, we vary the number of latent factors d and the regularization parameter λ, while keeping η=10^{-4} fixed. For presentation purposes, we plot $R@$ as a percentage (%). Figures 1(a)-(b) show that there is a slight increase on $R@$ after $d \geq 40$, while there is a drop on $R@$ when λ is in the range of 0.01-1, and we select d=40 and λ=0.001.



(a) (b)

Figure 1: Effect of (a) # of latent factors d and (b) regularization parameter λ on $R@10$ (%).

We use CofiRank[3] [12] as baseline, and we compare the proposed SCR method with P-Push, Inf-Push and RH-Push using the optimization algorithms[4] of [2]. In addition, we compare SCR with MR-BPR[5] [6], a state-of-the-art method for ranking with social relationships in the recommendation problem. We also evaluate the performance of SCR against SBRP [14], a ranking model that also considers social relationships in the learning process, assuming that users tend to assign higher ranks to items that their friends prefer. In all ranking models, we followed the same cross-validation strategy to select the optimal parameters as in SCR. In Table 1 we compare the examined ranking models, where we observe that P-Push, Inf-Push and RH-Push outperform CofiRank, as these models focus on the ranking performance at the top of the list; while MR-BPR, SBPR and SCR achieve better performance than the models that do not exploit social relationships. The proposed SCR model beats the competitors by exploiting social relationships and focusing on the top of the ranking list. Using the paired t-test, we found that the differences between the reported results for SCR against the competitive approaches were statistically significant for p<0.05.

Table 1: Methods comparison in terms of $NDCG@$ and $R@$, with bold values denoting the best scores ($^*p<$ 0.05 in paired t-test). MR-BPR, SBPR and SCR exploit social relationships.

	$NDCG@10$	$R@10$	$R@100$
CofiRank	.1423 ± .0043	.0698 ± .0075	.2398 ± .0047
P-Push	.1671 ± .0058	.1134 ± .0044	.2977 ± .0054
Inf-Push	.1563 ± .0091	.1073 ± .0053	.2789 ± .0035
RH-Push	.1574 ± .0079	.1266 ± .0062	.2980 ± .0028
MR-BPR	.1798 ± .0084	.1401 ± .0089	.3081 ± .0051
SBPR	.1886 ± .0076	.1490 ± .0055	.3296 ± .0042
SCR	**.2065 ± .0064***	**.1596 ± .0043***	**.3654 ± .0035***

6. CONCLUSION AND FUTURE WORK

In this study, we presented SCR, a CR model that focuses on the ranking accuracy at the top of the list by also considering users' trusted social relationships. Our learning strategy uses the notion of Social Reverse Height to consider the ranking positions of friends' relevant items, while our SCR method outperforms other state-of-the-art ranking models.

An interesting future direction is to exploit both the trust and distrust social relationships in the proposed SCR model, accounting for the fact that users with distrust relationships usually do not have similar preferences [3]. In addition, we plan to evaluate the performance of SCR in terms of diversity, that is, how well we can generate diversified recommendations in SCR to capture the interest range of the target user [11].

[3] https://github.com/markusweimer/cofirank

[4] http://www-users.cs.umn.edu/~christa/

[5] http://www.ismll.uni-hildesheim.de/software

7. REFERENCES

[1] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad. A probabilistic model for using social networks in personalized item recommendation. In *RecSys*, pages 43–50, 2015.

[2] K. Christakopoulou and A. Banerjee. Collaborative ranking with a push at the top. In *WWW*, pages 205–215, 2015.

[3] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *TOIS*, 32(4):17:1–17:38, 2014.

[4] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *STOC*, pages 665–674, 2013.

[5] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[6] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, pages 173–182, 2012.

[7] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331.

[8] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[10] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys*, pages 139–146, 2012.

[11] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. In *SIGIR*, pages 175–184, 2012.

[12] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. COFI RANK - maximum margin matrix factorization for collaborative ranking. In *NIPS*, pages 1593–1600, 2007.

[13] X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k recommendation using social networks. In *RecSys*, pages 67–74, 2012.

[14] T. Zhao, J. McAuley, and I. King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pages 261–270, 2014.