Joint Collaborative Ranking with Social Relationships in Top-N Recommendation

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

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

ABSTRACT

With the advent of learning to rank methods, relevant studies showed that Collaborative Ranking (CR) models can produce accurate ranked lists in the top-N recommendation problem. However, in practice several real-world problems decrease their ranking performance, such as the sparsity and cold-start problems, which often occur in recommendation systems for inactive or new users. In this study, to account for the fact that the selections of social friends can improve the recommendation accuracy, we propose a joint CR model based on the users' social relationships. We propose two different CR strategies based on the notions of Social Reverse Height and Social Height, which consider how well the relevant and irrelevant items of users and their social friends have been ranked at the top of the list, respectively. We focus on the top of the list mainly because users see the top-N recommendations in real-world applications, and not the whole ranked list. Furthermore, we formulate a joint objective function to consider both CR strategies, and propose an alternating minimization algorithm to learn our joint CR model. Our experiments on benchmark datasets show that our proposed joint CR model outperforms other state-of-the-art 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

Collaborative ranking; learning to rank; recommendation systems; 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 [13, 24, 27]. User preferences are expressed explicitly in the form of ratings or implicitly in the form of number of views, clicks, purchases, and so on. Relevant studies examine how to predict the rating of a user on an unseen item, also known as the rating prediction problem [19]. In practice though, only the top-N items are presented to the user as a ranked list; therefore, instead of focusing on the accuracy of rating prediction, ranking-based models try to produce accurate ranked lists for the top-N recommendation problem [1, 21]. In relevant studies [10, 34], 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 [21, 26]. In [3], authors examine different optimization algorithms for 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. This means that the user is more likely to pay attention to the items at the top of the list than the lower-ranked items [1]. In doing so, the CR models of [3], 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 [13]. Several approaches have been proposed for handling them in the rating prediction problem, such as the studies in [7, 11, 18, 19], assuming that users tend to trust the recommendations of their social friends [31]. In addition, various approaches have been introduced to exploit social relationships in the top-N recommendation problem, such as the studies in [2, 10, 14, 16, 36]. Nonetheless, these studies do not focus on the ranking performance at the top of the list when learning the ranking functions, and as a consequence may have limited ranking accuracy in the top-N recommendation problem [3].

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 which learns the personalized ranking function of each individual in a collaborative manner. When learning the ranking functions, the proposed CR model attempts to push up users' relevant items above

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983839

the irrelevant ones at the top of the list. In our model, we present two different CR strategies based the notions of Social Reverse Height and Social Height. The two proposed CR strategies consider how well the relevant and irrelevant items of users and their social friends have been ranked at the top of the list, respectively. As our two CR strategies may have different ranking performances when focusing on the top of the list, we propose a joint CR model, where we define a joint objective function to learn the model. We formulate the joint objective function of our joint CR model as a minimization problem, and we provide an efficient optimization algorithm based on alternating minimization and gradient descent [8]. Our experiments on benchmark datasets from the social media platforms of Epinions[1], Flixter[2] and Ciao[3] demonstrate the superiority of our joint CR model over other state-of-the-art models.

The remainder of the paper is organized as follows, Section 2 reviews the related study and in Section 3 we formally define our problem. In Section 4 we present the proposed CR model, where we define the joint objective function and detail the proposed optimization algorithm to learn our model. In Section 5 we evaluate the performance of the proposed CR model and Section 6 concludes the study.

2. RELATED WORK

2.1 CR in Top-N Recommendation

Learning to rank methods have been widely studied in Information Retrieval and in recommendation systems [1]. The goal is to define a ranking function on each item, and then learn toward some loss functions. Liu et al. [17] 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` [32] and `CLiMF` [26], 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`) [21], 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 [3] 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, thus not performing well in the *cold-start* scenario, where users have poor history record [31].

[1]http://www.epinions.com/
[2]www.flixster.com/
[3]www.ciao.com/

2.2 Social Recommendation

Several studies incorporate social relationships in the rating prediction problem, for example, Trust-based Singular Value Decomposition (`TrustSVD`) [7], Social Matrix Factorization (`SocialMF`) [11], learning to recommend with Social Trust Ensemble (`STE`) [18], and Social Recommendation (`SoRec`) [19]. However, these methods treat the recommendation problem as a rating prediction problem in which different squared loss functions are used to minimize the prediction error; this means that methods that achieve low rating errors do not necessarily have high recommendation accuracy [34]. This mainly happens because optimizing the predicted values of the ratings may not provide the best recommendation lists to the end-user in many cases; for example, if all the low-ranked ratings are predicted very accurately, but significant errors are made on the higher-ranked ratings, the resulting solution will not provide a high-quality recommendation list to the end-user [1].

Therefore, given that users are mainly focused only on the top-N recommendations and the other predictions are ignored in real-world applications, various approaches have been introduced which exploit the social relationships in the top-N recommendation problem [10]. For example, Rendle et al. [14] present the `MR-BPR` model, where they combine Multi-Relational matrix factorization with the `BPR` framework [21] to model both users' feedback on items and on social relationships. Zhao et al. [36] 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. Both `MR-BPR` and `SBRP` follow a "one-class" collaborative strategy, which means that they consider only the positive/relevant user feedback when generating the recommendation lists. In [10], the Random Walk approach `TrustWalker` [9] is combined with Collaborative Filtering, namely `Trust-CF`, to generate top-N recommendations with social relationships. Liu et al. [16] introduce `ST-CoR`, a Social Temporal Collaborative Ranking model which considers both the social relationships and preference evolution over time. Chaney et al. [2] present a Bayesian model, that performs Social Poisson factorization, namely `SPF`, accounting for the fact that a social friend does not necessarily match the user preferences. However, all the aforementioned approaches that exploit social relationships 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). User preferences are stored in a ($|\mathcal{U}| \times |\mathcal{I}|$) matrix R, with $[R]_{ij}$=1 if user i has selected item j as relevant, $[R]_{ij}$=-1 as irrelevant, and $[R]_{ij}$=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 $i \in \mathcal{U}$, the Reverse Height of a relevant item x_i^+ is defined as follows [3]:

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^+.

Accordingly, the Height of an irrelevant item x_i^- is defined as follows:

DEFINITION 2. *The **Height** $H_i(x_i^-)$ of an irrelevant item x_i^- in the recommendation list of user i is the number of relevant items ranked below x_i^-.*

Note that Reverse Height is calculated for each relevant marked item, while the Height is computed for each irrelevant item.

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

DEFINITION 3. *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 4. *The **Social Height** $SH_i(x_i^-)$ of an irrelevant item x_i^- is the sum of (1) the number of relevant items x_i^+ ranked below x_i^-; and (2) the number of the relevant items x_i^+ ranked below the irrelevant items x_w^- of all her friends.*

DEFINITION 5. *The **goal of our joint CR model** is to learn the personalized ranking functions collaboratively by jointly trying to minimize $SRH_i(x_i^+)$ and $SH_i(x_i^-)$.*

4. PROPOSED CR MODEL

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 [2, 31]. Hence, when computing the Social Reverse Height of a relevant item, we measure the social influence between user i and her friend w using the *relevant/positive social coefficient* $s_{iw}^+ = |\mathcal{I}_i^+ \cap \mathcal{I}_w^+|$, 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 value s_{iw}^+ with the maximum value of all the $|\mathcal{F}_i|$ friends.

Given a personalized ranking function $r_i(x)$ for each user i, with $i \in \mathcal{U}$ and $x \in \mathcal{I}$, according to Definition 3 we have:

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

where $\mathbb{1}$ is the indicator function and s_{iw}^+ the relevant/positive social coefficient. 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 $rg_i(\cdot)$ as follows:

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

4.2 Social Height

When computing the Social Height of an irrelevant item, we calculate the *irrelevant/negative social coefficient* between user i and her friend w as follows: $s_{iw}^- = |\mathcal{I}_i^- \cap \mathcal{I}_w^-|$, which is the intersection of the sets of irrelevant items of user i and her friend w. According to Definition 4 the Social Height of an irrelevant item x_i^- is defined as follows:

$$SH_i(x_i^-) = \sum_{x_i^+ \in \mathcal{I}_i^+} \Big\{ \mathbb{1}[r_i(x_i^+) \leq r_i(x_i^-)] + \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^-|} \sum_{x_w^- \in \mathcal{I}_w^-} s_{iw}^- \mathbb{1}[r_i(x_i^+) \leq r_i(x_w^-)] \Big\} \quad (3)$$

Similarly, we take the following surrogate $g_i(\cdot)$ for the Social Height:

$$g_i(x_i^+, x_i^-) = r_i(x_i^+) - r_i(x_i^-) + \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^-|} \sum_{x_w^- \in \mathcal{I}_w^-} s_{iw}^- \big(r_i(x_i^+) - r_i(x_w^-)\big) \quad (4)$$

4.3 Objective Function

In the proposed 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 $\hat{R} = U^T V$ 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 (i) the relevant item of user i as $a{=}x_i^+$; (ii) the irrelevant item of user i as $b{=}x_i^-$; (iii) the relevant item of a friend $w \in \mathcal{F}_i$ as $c{=}x_w^+$; and (iv) her irrelevant item as $d{=}x_w^-$. According to the representation of the latent factors $\mathbf{u}_i$ and $\mathbf{v}_j$, the surrogate $rg(\cdot)$ of Social Reverse Height in Eq. (2) can be rewritten as follows:

$$rg_i(a, b) = \mathbf{u}_i^T(\mathbf{v}_a - \mathbf{v}_b) + \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^+|} \sum_{c \in \mathcal{I}_w^+} s_{iw}^+ (\mathbf{u}_w^T(\mathbf{v}_c - \mathbf{v}_b)) \quad (5)$$

As the logistic loss function is a convex upper bound to the indicator function $\mathbb{1}$, based on Eq. (5) we can reformulate Eq. (1) of the Social Reverse Height as follows:

$$SRH_i(a) = \sum_{b \in \mathcal{I}_i^-} \Big(\log\left(1 + \exp\left(-rg_i(a, b)\right)\right) \Big) \quad (6)$$

Accordingly, the surrogate $g(\cdot)$ of the Social Height in Eq. (4) can be calculated based on the representation of the latent factors as follows:

$$g_i(a, b) = \mathbf{u}_i^T(\mathbf{v}_a - \mathbf{v}_b) + \frac{1}{|\mathcal{F}_i|} \sum_{w \in \mathcal{F}_i} \frac{1}{|\mathcal{I}_w^-|} \sum_{d \in \mathcal{I}_w^-} s_{iw}^- (\mathbf{u}_w^T(\mathbf{v}_a - \mathbf{v}_d)) \quad (7)$$

Based on Eq. (7) we can reformulate the Social Height in Eq. (3) as follows:

$$SH_i(b) = \sum_{a \in \mathcal{I}_i^+} \Big(\log\left(1 + \exp\left(-g_i(a, b)\right)\right) \Big) \quad (8)$$

According to Definition 5, for each user i we have to jointly minimize (i) the Social Reverse Height $SRH_i(a)$ of Eq. (6) for all her relevant items $a \in \mathcal{I}_i^+$, and (ii) the Social Height

$SH_i(b)$ of Eq. (8) for all her irrelevant items $b \in \mathcal{I}_i^-$. In our model, we consider the formulation of the p-norm push [23] for the Social Height $SH_i(b)$ and the log-log function [3] for the Social Reverse Height $SRH_i(a)$. Hence, $\forall\, i \in \mathcal{U}$ the goal of the proposed joint CR model is to minimize the following joint objective function with respect to matrices U and V:

$$\mathfrak{L}(U,V) = \sum_{i=1}^{|\mathcal{U}|} \frac{1}{n_i}\Big\{ \sum_{b\in\mathcal{I}_i^-} SH_i(b)^p + \sum_{a\in\mathcal{I}_i^+} \log(1+SRH_i(a))\Big\} + \frac{\lambda}{2}\Big(||U||^2 + ||V||^2\Big) \tag{9}$$

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

4.4 Model Learning

The goal of our joint CR model is to learn (compute) matrices U and V, to minimize the loss function $\mathfrak{L}(U,V)$ in Eq. (9). In our learning process, we follow the alternating minimization strategy [8] using the gradient descent optimization algorithm[4], 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 updating rules for each iteration $t+1$ are:

$$\mathbf{u}_i^{t+1} \leftarrow \mathbf{u}_i^t - \eta \nabla_{\mathbf{u}_i} \mathfrak{L}(U^t, V^t), \quad i = 1 \dots |\mathcal{U}| \tag{10}$$

$$\mathbf{v}_j^{t+1} \leftarrow \mathbf{v}_j^t - \eta \nabla_{\mathbf{v}_j} \mathfrak{L}(U^{t+1}, V^t), \quad j = 1 \dots |\mathcal{I}| \tag{11}$$

where parameter η controls the learning rate.

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

We apply the chain rule to Eq. (9) to take the gradient of $\mathfrak{L}$ with respect to $\mathbf{u}_i$ for the Social Reverse Height $SRH_i(a)$ and the Social Height $SH_i(b)$. In particular, for the Social Reverse Height $SRH_i(a)$ we have the composition of gradients $\nabla_{\mathbf{u}_i} rg_i(a,b)$ and $\nabla_{rg_i(a,b)} SRH_i(a)$ with respect to $\mathbf{u}_i$ in Eq. (5) and $rg_i(a,b)$ in Eq. (6), respectively; while for the Social Height $SH_i(b)$ we consider the composition of gradients $\nabla_{\mathbf{u}_i} g_i(a,b)$ and $\nabla_{g_i(a,b)} SH_i(a)$ with respect to $\mathbf{u}_i$ in Eq. (7) and $g_i(a,b)$ in Eq. (8), accordingly. Note that the gradient of the regularization term in Eq. (9) with respect to $\mathbf{u}_i$ equals $\lambda\mathbf{u}_i$. Thus, given that $\mathbf{u}_i \neq \mathbf{u}_w$, with $i \neq w$ (Section 3), we have the following gradients:

$$\nabla_{\mathbf{u}_i} rg_i(a,b) = \nabla_{\mathbf{u}_i} g_i(a,b) = \mathbf{v}_a - \mathbf{v}_b \tag{12}$$

$$\nabla_{rg_i(a,b)} SRH_i(a) = \frac{-1}{1+\exp(rg_i(a,b))} \tag{13}$$

$$\nabla_{g_i(a,b)} SH_i(b) = \frac{-1}{1+\exp(g_i(a,b))} \tag{14}$$

Based on the gradients in Eqs. (12), (13) and (14), the gradient of $\mathfrak{L}$ with respect to $\mathbf{u}_i$ in Eq. (9) is equivalent to:

$$\nabla_{\mathbf{u}_i}\mathfrak{L}(U,V) = \frac{p}{n_i}\sum_{b\in\mathcal{I}_i^-}\Big\{SH_i(b)^{p-1}\sum_{a\in\mathcal{I}_i^+}\frac{1}{1+\exp(g_i(a,b))}(\mathbf{v}_b-\mathbf{v}_a)\Big\} + \frac{1}{n_i}\sum_{a\in\mathcal{I}_i^+}\Big\{\frac{1}{1+SRH_i(a)}\sum_{b\in\mathcal{I}_i^-}\frac{1}{1+\exp(rg_i(a,b))}(\mathbf{v}_b-\mathbf{v}_a)\Big\} + \lambda\mathbf{u}_i \tag{15}$$

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

When calculating the gradients of $rg_i(a,b)$ and $g_i(a,b)$ with respect to $\mathbf{v}_j$ in Eqs. (5) and (7), respectively, we have the following cases:

$$\nabla_{\mathbf{v}_j} rg_i(a,b) = \begin{cases} \mathbf{u}_i & \text{, if } j{=}a \\ -\big[\mathbf{u}_i + \frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^+|}\sum_{c\in\mathcal{I}_w^+} s_{iw}^+\mathbf{u}_w\big] & \text{, if } j{=}b \\ \frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^+|}\sum_{c\in\mathcal{I}_w^+} s_{iw}^+\mathbf{u}_w & \text{, if } j{=}c \\ 0 & \text{, if } j{=}d \end{cases} \tag{16}$$

$$\nabla_{\mathbf{v}_j} g_i(a,b) = \begin{cases} \mathbf{u}_i + \frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^-|}\sum_{d\in\mathcal{I}_w^-} s_{iw}^-\mathbf{u}_w & \text{, if } j{=}a \\ -\mathbf{u}_i & \text{, if } j{=}b \\ 0 & \text{, if } j{=}c \\ -\frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^-|}\sum_{d\in\mathcal{I}_w^-} s_{iw}^-\mathbf{u}_w & \text{, if } j{=}d \end{cases} \tag{17}$$

The gradient of $\mathfrak{L}$ with respect to $\mathbf{v}_j$ in Eq.(9) consists of the gradient $\nabla_{\mathbf{v}_j} rg_i(a,b)$ and $\nabla_{\mathbf{v}_j} g_i(a,b)$ based on the cases in Eqs. (16) and (17), respectively, as well as the gradients $\nabla_{rg_i(a,b)} SRH_i(a)$ and $\nabla_{g_i(a,b)} SH_i(b)$, in Eqs. (13) and (14). 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 $\mathcal{U}_j^+$. This means that the set $\mathcal{U}_j^-$ corresponds to the irrelevant marked items, that is, the cases of $j{=}b$ and $j{=}d$; while the set $\mathcal{U}_j^+$ reflects on the relevant items, that is, the cases of $j{=}a$ and $j{=}c$ in Eqs. (16) and (17). Thus by setting the following terms:

$$\Lambda_{rg} = \sum_{b\in\mathcal{I}_i^-}\frac{1}{1+\exp(rg_i(a,b))}\Big(\mathbf{u}_i + \frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^+|}\sum_{c\in\mathcal{I}_w^+} s_{iw}^+\mathbf{u}_w\Big) \tag{18}$$

$$\Lambda_{g} = \sum_{a\in\mathcal{I}_i^+}\frac{1}{1+\exp(g_i(a,b))}\Big(\mathbf{u}_i + \frac{1}{|\mathcal{F}_i|}\sum_{w\in\mathcal{F}_i}\frac{1}{|\mathcal{I}_w^-|}\sum_{d\in\mathcal{I}_w^-} s_{iw}^-\mathbf{u}_w\Big) \tag{19}$$

the gradient of $\mathfrak{L}$ with respect to $\mathbf{v}_j$ is equivalent to:

$$\nabla_{\mathbf{v}_j}\mathfrak{L}(U,V) = \sum_{i\in\mathcal{U}_j^-}\Big\{\frac{p}{n_i}\sum_{b\in\mathcal{I}_i^-} SH_i(b)^{p-1}\Lambda_g + \frac{1}{n_i}\sum_{a\in\mathcal{I}_i^+}\frac{1}{1+SRH_i(a)}\Lambda_{rg}\Big\} - \sum_{i\in\mathcal{U}_j^+}\Big\{\frac{p}{n_i}\sum_{b\in\mathcal{I}_i^-} SH_i(b)^{p-1}\Lambda_g + \frac{1}{n_i}\sum_{a\in\mathcal{I}_i^+}\frac{1}{1+SRH_i(a)}\Lambda_{rg}\Big\} + \lambda\mathbf{v}_j \tag{20}$$

[4]In this study, we implement the gradient descent algorithm in a single node architecture. As a future work we plan to implement the proposed approach in a distributed framework to ensure scalability [6, 15, 20, 37].

Using the respective gradients in (15) and (20) in the updating rules (10) and (11), we compute matrices U and V. Finally, we calculate the low-rank d approximation $\hat{R} = U^T V$ of the initial $(|\mathcal{U}| \times |\mathcal{I}|)$ matrix R, and for each user i we generate a final ranked (recommendation) list of items based on the respective i-th row of $\hat{R}$.

5. EXPERIMENTAL EVALUATION

5.1 Evaluation Setup

Datasets. In our experiments we use the following datasets, **Epinions**[5] [22], **Flixster**[6] [11] and **Ciao**[7] [30]. The main statistics of the evaluation datasets are summarized in Table 1. In Epinions and Ciao the ratings are at a 5-star scale by a step of 1, where we considered 4-5 star ratings as relevant to a user and the remaining ratings as irrelevant; in Flixster the ratings are also at a 5 star scale by a step of 0.5, where 3.5-5 star ratings are considered as relevant and 0.5-3 as irrelevant [10, 34].

Table 1: Dataset statistics.

	Epinions	Flixster	Ciao
#users	71,002	147,612	7,375
#items	104,356	48,794	105,114
#ratings	571,235	8,196,077	284,086
#social rel.	508,960	7,058,819	111,781
Av. #ratings per user	8.04	55.52	38.52
Av. #social rel. per user	7.16	47.82	15.16

Training/test set split. Each dataset is divided into two subsets: the training set and the test set. Following the evaluation protocol of similar studies [34, 36], 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) cross-validation test set over the runs.

Evaluation metrics. Popular commercial systems make top-N recommendations to users, and relevant studies showed that rating error metrics, such as RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) do not necessarily reflect on the top-N recommendation performance [1, 34]. Therefore, in our experiments we used the ranking-based metrics *Recall* ($R@N$) and *Normalized Discounted Cumulative Gain* ($NDCG@N$) to evaluate the top-N recommendation performance of the examined models directly [3, 14, 36]. *Recall* ($R@N$) is defined as the ratio of the relevant items in the top-N ranked list over all the relevant items for each user. The Normalized Discounted Cumulative Gain ($NDCG@N$) metric considers the ranking of the relevant items in the top-N list. For each user the Discounted Cumulative Gain is defined as:

$$DCG@N = \sum_{j=1}^{N} \frac{2^{rel_j} - 1}{\log_2 j + 1} \quad (21)$$

where rel_j represents the relevance score of the item j, binary relevance in our case, that is, relevant or irrelevant to the user. The *Normalized Discounted Cumulative Gain* $NDCG@N$ 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.

5.2 Compared Methods

We evaluate the performance of the following *baseline CR models*:

- `CofiRank`[8] [32]: is a baseline CR model that does not focus on the top of the list nor exploits the users' social relationships, when generating the top-N recommendations.

- `P-Push` / `Inf-Push` / `RH-Push`[9] [3]: are the three different push CR models based on p-norm, infinite and reverse height that focus on the ranking performance at the top of the list. In `P-Push`, we set p to 2 as in [3]. All the three different push CR models do not use social relationships, as well.

In addition, we compare the following *social CR models* that exploit users' social relationships in the top-N recommendation problem:

- `SBRP` [36]: is a ranking model that considers social relationships in the learning process, assuming that users tend to assign higher ranks to items that their friends prefer. This model follows an "one-class" collaborative filtering strategy, thus trained only on the positive/relevant marked items.

- `SPF`[10] [2]: is a probabilistic model that performs Social Poisson factorization. `SPF` incorporates user latent preferences for items with the latent influences of her friends, thus matching user preferences with her social friends when generating the top-N recommendations.

- `SH`: is a variant of the proposed joint model which tries to minimize only the Social Height in Eq. (8), by ignoring the Social Reverse Height when learning the model.

- `SRH`: is a variant of our model, which considers only the Social Reverse Height in Eq. (6).

- `JSCR`: is the proposed joint CR model, which jointly minimizes the Social Height and the Social Reverse Height.

In all examined models, we tune the parameters based on the same cross validation strategy (Section 5.1), and in our experiments we report the best results.

[5] https://alchemy.cs.washington.edu/data/epinions/
[6] http://www.cs.ubc.ca/~jamalim/datasets/
[7] http://www.jiliang.xyz/trust.html
[8] https://github.com/markusweimer/cofirank
[9] http://www-users.cs.umn.edu/~christa/
[10] https://github.com/ajbc/spf

Table 2: Effect on $NDCG@10$. Bold values denote the best scores for $^*p< 0.05$ in paired t-test. SBPR, SPF, SH, RSH and the proposed JSCR model exploit social relationships, when generating the recommendations.

		Epinions	Flixster	Ciao
Baseline CR models	CofiRank	0.1423 ± 0.0043	0.1879 ± 0.0255	0.1247 ± 0.0029
	P-Push	0.1671 ± 0.0058	0.2497 ± 0.0217	0.1540 ± 0.0052
	Inf-Push	0.1563 ± 0.0091	0.2349 ± 0.0618	0.1456 ± 0.0037
	RH-Push	0.1574 ± 0.0079	0.2456 ± 0.0391	0.1679 ± 0.0044
Social CR models	SBPR	0.1886 ± 0.0076	0.2736 ± 0.0421	0.2167 ± 0.0032
	SPF	0.1798 ± 0.0084	0.3290 ± 0.0484	0.1978 ± 0.0026
	SH	0.2145 ± 0.0092	0.3162 ± 0.0416	0.2067 ± 0.0024
	RSH	0.2065 ± 0.0064	0.2984 ± 0.0315	0.2134 ± 0.0030
	JSCR	$\mathbf{0.2451 \pm 0.0053}^*$	$\mathbf{0.3546 \pm 0.0224}^*$	$\mathbf{0.2446 \pm 0.0042}^*$

5.3 Performance Evaluation

In the first set of experiments we evaluate the performance of the examined CR models. Table 2 shows the effect on Normalized Discounted Cumulative Gain $NDCG@10$ and Figures 1(a)-(c) show the effect on recall $R@$, in Epinions, Flixter and Ciao, respectively. We can make the following observations:

- **Performance of the baseline CR models:** The baseline CR models P-Push, Inf-Push and RH-Push outperform CofiRank in all datasets, as these models focus on the recommendation accuracy at the top of the list.
- **Comparison of baseline with social CR:** All the social CR models SBPR, SPF, SH RSH and JSCR significantly leverage the recommendation accuracy by exploiting the social relationships when generating the top-N ranked lists. When comparing the social CR models with the baselines, Table 2 shows that the social models improve $NDCG@$ on average, by a 31.4, 36.9 and 45.8% in Epinions, Flixter and Ciao, respectively. In terms of $R@N$, Figures 1(a)-(c) show that the social CR models achieve an average improvement of 35.8, 33.6 and 45.9% in Epinions, Flixter and Ciao, respectively.
- **Performance of the variants SH and SHR:** Our two variants follow different social CR strategies, which explains their different ranking performances in Table 2 and Figures 1(a)-(c). The SH variant considers only the Social Height when generating the top-N recommendations, while SHR tries to minimize the Social Reverse Height. SH and SHR achieve higher accuracy in Epinions than the competitive social CR models SBPR and SPF. However, in Flixter and Ciao SH and SHR have similar ranking performances with at least one of the social CR models SBPR and SPF. This happens because both Flixter and Ciao have larger numbers of average ratings and relationships per user than Epinions, as shown in Table 1; consequently, relatively high ranking performances are observed for all the social CR models.
- **Performance of JSCR:** Table 2 and Figures 1(a)-(c) show that the proposed JSCR model beats all the competitive models by exploiting social relationships and focusing on the top of the ranking list in a jointly manner. As expected, the joint consideration of the Social Height and Social Reverse Height in the objective function in Eq. (9) makes JSCR superior over all the competitors, including its two variants, that is, the social CR models SH and SHR. With respect to the second best method, JSCR achieves a relative improvement of 8.9 and 10.8% on average, in terms of $NDCG@$ and $R@$, respectively. Using the paired t-test, we found that the differences between the reported results for JSCR against the competitive approaches were statistically significant for $p<0.05$, in all settings.

5.4 Cold-Start Analysis

In addition, we study the performance of the ranking models on the *cold-start* scenario. We define users with less than 5 ratings as *cold-start* users, corresponding to the 49% and 53% of the total users in Epinions and Flixter, respectively [10, 11]. As in the evaluation dataset of Ciao only the 0.03% has less than 5 ratings, we define *cold-start* users with less than 10 ratings, that is, the 22% of the total users in Ciao.

Table 3 and Figures 1(d)-(f) show the effect on $NDCG@$ and $R@$ for the *cold-start* users, respectively. In relation to the experimental results for all users, presented in Table 2 and Figures 1(a)-(c), we observe that there is a significant drop on both evaluation metrics for *cold-start* users; the ranking performances of the baseline CR models significantly degrade for *cold-start* users, with a 32.5, 30.7 and 19.2% average drop in Epinions, Flixter and Ciao, accordingly; while the social CR models preserve their ranking performances relatively high, with a 17.1, 22.5 and 12.8% drop, on average. This occurs because the social CR models incorparate the users' relationships when generating the top-N recommendations, thus handling the *cold-start* problem by exploiting the selections of their friends.

In this set of experiments, the proposed JSCR model improves the second best model at 10.8 and 12.24% on average, in terms of $NDCG@$ and $R@$, accordingly, confirming its superiority in the *cold-start* scenario as well. This occurs because JSCR exploits both the social CR strategies of Social Height and Social Reverse Height, when trying to minimize the joint objective function in Eq. (9), thus being less affected than the other social CR models when generating recommendations for *cold-start* users. This is very important in recommendation systems, as in real-world applications there are often many inactive or new users with poor history record, corresponding to *cold-start* users [1, 4, 13, 31].

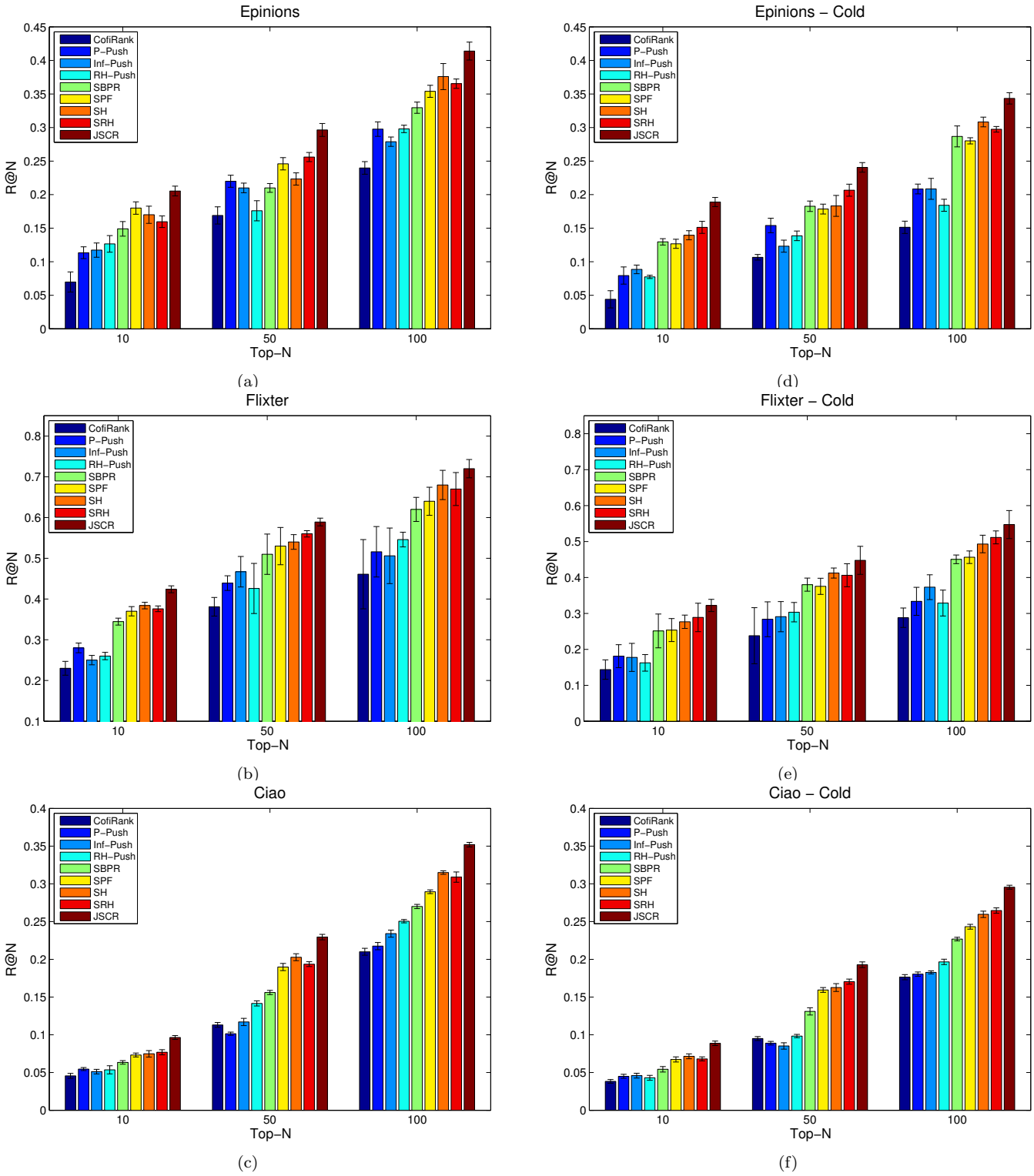

Figure 1: (a)-(c) effect on $R@N$ by varying the top-N recommendations in Epinons, Flixter and Ciao, for all users. (d)-(e) effect on $R@N$ for *cold-start* users.

Table 3: Methods comparison in terms of $NDCG@10$ for *cold-start* users, with bold values denoting the best scores for $^*p<0.05$ in paired t-test.

		Epinions	Flixster	Ciao
Baseline CR models	CofiRank	0.0898 ± 0.0022	0.1467 ± 0.0335	0.1047 ± 0.0045
	P-Push	0.1170 ± 0.0046	0.2018 ± 0.0112	0.1294 ± 0.0026
	Inf-Push	0.1102 ± 0.0068	0.2097 ± 0.0219	0.1119 ± 0.0053
	RH-Push	0.1032 ± 0.0094	0.1901 ± 0.0394	0.1223 ± 0.0024
Social CR models	SBPR	0.1603 ± 0.0058	0.2512 ± 0.0174	0.1994 ± 0.0027
	SPF	0.1641 ± 0.0039	0.2846 ± 0.0322	0.1678 ± 0.0033
	SH	0.1764 ± 0.0066	0.2767 ± 0.0275	0.1881 ± 0.0032
	RSH	0.1675 ± 0.0084	0.2872 ± 0.0395	0.1985 ± 0.0042
	JSCR	$\mathbf{0.1941 \pm 0.0063}^*$	$\mathbf{0.3269 \pm 0.0245}^*$	$\mathbf{0.2250 \pm 0.0028}^*$

5.5 Discussion

Our experimental results showed that it is essential to consider both the ranking performance at the top of the list and the social relationships, when generating top-N recommendations. For example, despite the fact that P-Push, Inf-Push and RH-Push focus on the ranking performance at the top of the list, these baseline CR models ignore users' social relationships, and as a consequence fail to generate accurate recommendations, due to the data sparsity and *cold-start* problems. More precisely, in the *cold-start* scenario P-Push, Inf-Push and RH-Push have low ranking performances with an average relative drop of 27.6% in the three evaluation datasets.

Our two variants SH and RSH exploit social relationships, as well as they focus on the ranking performance at the top of the list, by considering how well the relevant and irrelevant items of users and their social friends have been ranked. SH follows the CR strategy of minimizing the Social Height, that is, given an irrelevant item, SH tries to minimize the number of relevant items ranked below the irrelevant item, as well as the number of relevant items of all her friends that have been ranked below the same irrelevant item. Meanwhile, RSH follows the CR strategy of minimizing the Social Reverse Height, that is, given a relevant item, RSH attempts to minimize the number of irrelevant items ranked above the relevant item and the number of irrelevant items of all her friends that have been ranked above the same relevant item. This means that SH tries to push irrelevant items below those that have been selected as relevant from a user and her friends; while RSH tries to push relevant items above those that been marked as irrelevant from a user and her friends. Our experiments showed that both the SH and RSH variants achieve higher or comparable performances with other social CR models in all settings.

In addition, we evaluated the recommendation performance of the proposed JSCR model, which combines its two variants by learning the joint objective function in Eq. (9). This means that JSCR tries to minimize both the Social Height and Social Reverse Height at the same time. We observed that the proposed JSCR model beats all the competitors in all settings, based on the pairwise t-test ($p<0.05$). JSCR has a very stable and relatively high top-N recommendation performance; more precisely, compared to the second best method for each set of experiments, JSCR achieves an average relative improvement of 10.1 and 11.5% in terms of Recall $R@$ and Normalized Discounted Gain $NDCG@$, respectively. This indicates that JSCR is capable of (i) retrieving more relevant items and (ii) performing better ranking than the other CR models, both important functionalities in recommendation systems [4].

Although the proposed JSCR model achieves high ranking performance in the top-N recommendation problem, the following two key factors have to be considered in a real-world application: (i) the fact that *user preferences and their social relationships evolve over time* [31]; and (ii) *the implementation in a distributed framework to ensure scalability.* Regarding the first key factor, users in recommendation systems shift their preferences over time mainly for the following reasons [12]:

- users in recommendation systems tend to explore new items over time, instead of interacting with the same items multiple times;
- users may interact with popular items to a great extent, irrespective of their history record, thus being biased by popular items e.g., movies and songs;
- user preferences may be updated based on the preferences of their friends over time; meanwhile, their social relationships may change over time as well [25].

Therefore, a challenge resides on how to consider the time dimension, so as to capture both users' preference dynamics and the social relationships' evolution over time in JSCR.

Regarding the second key factor, our implementation to learn our JSCR model is based on a gradient descent algorithm in a single node architecture, thus making the model learning computational intensive. Real-world applications have to generate recommendations for millions of users based on million or billion items; consequently, to account for the fact that real-world recommendation systems have to scale, several distributed implementations of gradient descent have been proposed in the literature, such as the studies reported in [15, 20]. Provided that in-memory processing platforms is a mainstay for handling Big Data [33, 35], it is worth to examine the implementation of JSCR in these platforms to ensure scalability. For example, popular in-memory Big Data processing platforms are Apache Spark[11] and Apache Flink[12], both supporting machine learning algorithms in their libraries, such as MLlib[13] and Flink-ML[14], respectively.

[11]http://spark.apache.org/
[12]https://flink.apache.org/
[13]http://spark.apache.org/mllib/
[14]https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/ml/index.html

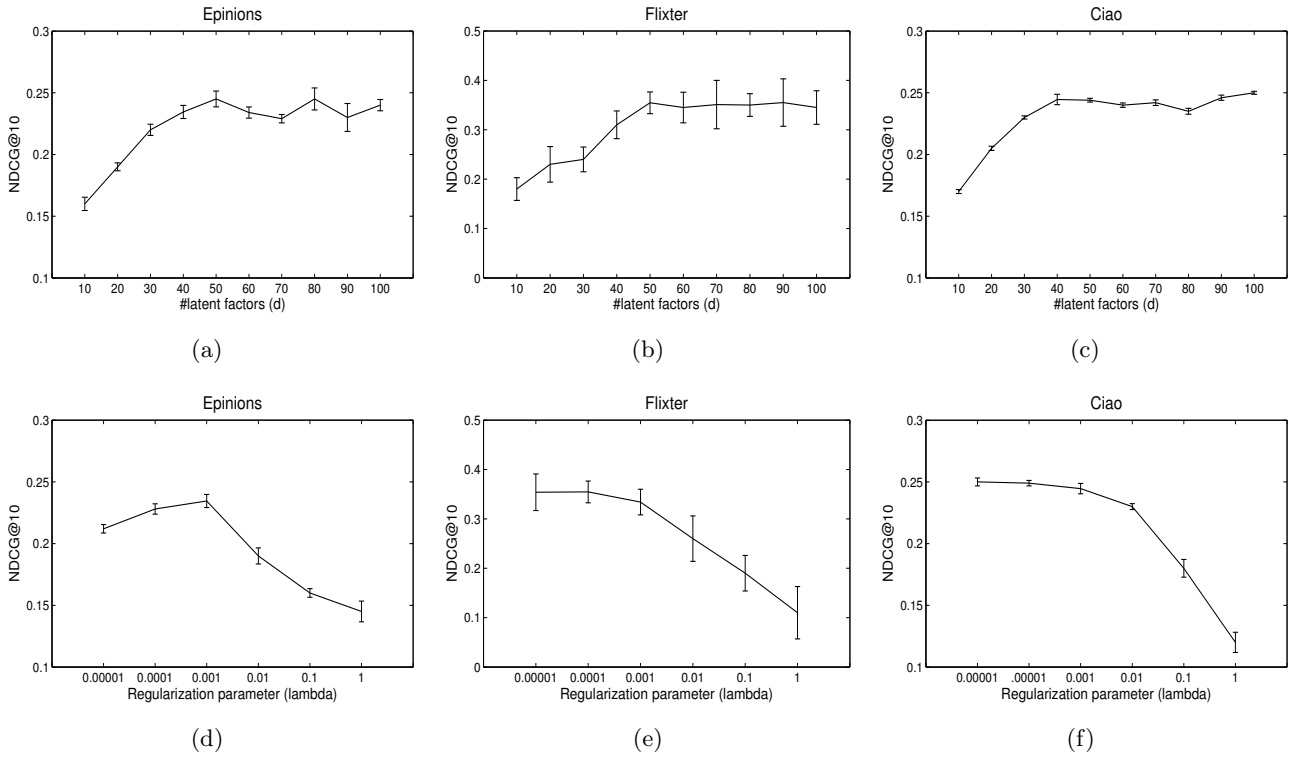

Figure 2: Effect on on $NDCG@10$ by varying (a)-(c) #latent factors d in the factorized matrix $\hat{R}$, and (d)-(f) the regularization parameter λ in the joint objective function of Eq. (9).

5.6 Parameter Analysis

To evaluate the parameter sensitivity in the proposed JSCR model, we tune (i) the *learning rate* η which controls the influence of the gradients, computed in Eqs. (15) and (20), when learning the model via the updating rules in Eqs. (10) and (11), respectively; (ii) the *regularization parameter* λ in the joint objective function of Eq. (9) which controls the regularization term to avoid model overfitting; and (iii) the *number of latent factors* d in the factorized matrix $\hat{R}$. We varied the learning rate η in the range of $[10^{-5}\ 10^{-1}]$ and we concluded in η=10^{-4} for Epinions and Ciao, and η=10^{-5} for Flixter. This indicates that a more conservative learning strategy is required in all evaluation datasets, as higher values of η decreased the performance of the proposed CR model, by falling in local optima when minimizing the joint objective function in Eq. (9). In Figure 2, we vary the number of latent factors d and the regularization parameter λ, while keeping η fixed. Figures 2(a)-(c) show the effect on $NDCG@$, when varying the number of latent factors d in 10-100 by a step of 10. We observe that there is a slight increase on $NDCG@$ after $d \geq 40-50$ in all datasets, while in some cases $NDCG@$ is reduced. Therefore, we select $d = 50$ for Epinions and Flixter, and $d = 40$ for Ciao. In Figures 2(d)-(f), we study the effect of the regularization parameter λ; we observe that there is a drop on $NDCG@$ when λ is in the range of 0.01-1, as high values of λ result in model overfitting, thus reducing the ranking performance of our CR model; therefore, we fix λ=10^{-3} in Epinions and Ciao, and λ=10^{-4} in Flixter.

6. CONCLUSION

In this paper, we introduced the notions of Social Reverse Height and Social Height, two different CR strategies that exploit social relationships in the top-N recommendation problem. Both CR strategies focus on the ranking performance at the top of the list, considering how well the relevant and irrelevant items of users and their social friends have been ranked at the top of the list, respectively. To leverage the recommendation accuracy, we formulated a joint objective function in our Joint Social CR model, namely JSCR, to consider both CR strategies when generating the top-N recommendations. Our learning strategy uses an optimization algorithm based on alternating minimization and gradient descent. We evaluated the performance of the proposed JSCR model for different numbers of top-N recommendations, as well as we examined the ranking performance in the case of *cold-start* users. Our experiments on benchmark datasets showed that JSCR is constantly superior over all the competitive approaches in all settings, including its variants as well, achieving a relative improvement of 10.7% on average when comparing with the second best method for each run.

An interesting future direction is to extend the proposed model in order to exploit both the trust and distrust social relationships, accounting for the fact that users with distrust relationships usually do not have similar preferences, [5, 29]. In addition, we plan to evaluate the performance of JSCR in terms of diversity, that is, how well we can generate diversified top-N recommendations in JSCR to capture the interest range of the target user, while keeping the recommendation accuracy relatively high at the same time, an important problem for recommendation systems as well [28].

7. REFERENCES

[1] C. C. Aggarwal. *Recommender Systems - The Textbook*. Springer, 2016.

[2] A. J.-B. 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.

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

[4] M. Deshpande and G. Karypis. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[5] 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.

[6] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *KDD*, pages 69–77, 2011.

[7] G. Guo, J. Zhang, and N. Yorke-Smith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*, 2015.

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

[9] M. Jamali and M. Ester. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *KDD*, pages 397–406, 2009.

[10] M. Jamali and M. Ester. Using a trust network to improve top-n recommendation. In *RecSys*, pages 181–188, 2009.

[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[12] Y. Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4):89–97, 2010.

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

[14] 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.

[15] F. Li, B. Wu, L. Xu, C. Shi, and J. Shi. A fast distributed stochastic gradient descent algorithm for matrix factorization. In *BigMine*, pages 77–87, 2014.

[16] N. N. Liu, L. He, and M. Zhao. Social temporal collaborative ranking for context aware movie recommendation. *ACM Trans. Intell. Syst. Technol.*, 4(1):15:1–15:26, 2013.

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

[18] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

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

[20] J. Oh, W. Han, H. Yu, and X. Jiang. Fast and robust parallel SGD matrix factorization. In *KDD*, pages 865–874, 2015.

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

[22] M. Richardson and P. M. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[23] C. Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, 2009.

[24] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[25] J. Scott. Social network analysis: developments, advances, and prospects. *Social Netw. Analys. Mining*, 1(1):21–26, 2011.

[26] 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.

[27] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, 2014.

[28] 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.

[29] J. Tang, C. C. Aggarwal, and H. Liu. Recommendations in signed social networks. In *WWW*, pages 31–40, 2016.

[30] J. Tang, H. Gao, and H. Liu. mtrust: discerning multi-faceted trust in a connected world. In *WSDM*, pages 93–102, 2012.

[31] J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Netw. Analys. Mining*, 3(4):1113–1133, 2013.

[32] 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.

[33] X. Wu, X. Zhu, G. Wu, and W. Ding. Data mining with big data. *IEEE Trans. Knowl. Data Eng.*, 26(1):97–107, 2014.

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

[35] H. Zhang, G. Chen, B. C. Ooi, K. Tan, and M. Zhang. In-memory big data management and processing: A survey. *IEEE Trans. Knowl. Data Eng.*, 27(7):1920–1948, 2015.

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

[37] M. Zinkevich, M. Weimer, A. J. Smola, and L. Li. Parallelized stochastic gradient descent. In *NIPS*, pages 2595–2603, 2010.