A Relaxed Ranking-based Factor Model for Recommender System from Implicit Feedback

Huayu Li^{+}, Richang Hong*, Defu Lian^{-}, Zhiang Wu$^{\times}$, Meng Wang* and Yong Ge^{+}

$^{+}$ UNC Charlotte, {hli38, yong.ge }@uncc.edu

* Hefei University of Technology, {hongrc, wangmeng}@hfut.edu.cn

$^{-}$ University of Electronic Science and Technology of China, dove.ustc@gmail.com

$^{\times}$ Nanjing University of Finance and Economics, zawu@seu.edu.cn

Abstract

Implicit feedback based recommendation has recently been an important task with the accumulated user-item interaction data. However, it is very challenging to produce recommendations from implicit feedback due to the sparseness of data and the lack of negative feedback/rating. Although various factor models have been proposed to tackle this problem, they either focus on rating prediction that may lead to inaccurate top-k recommendations or are dependent on the sampling of negative feedback that often results in bias. To this end, we propose a Relaxed Ranking-based Factor Model, RRFM, to relax pairwise ranking into a SVM-like task, where positive and negative feedbacks are separated by the soft boundaries, and their non-separate property is employed to capture the characteristic of unobserved data. A smooth and scalable algorithm is developed to solve group- and instance- level's optimization and parameter estimation. Extensive experiments based on real-world datasets demonstrate the effectiveness and advantage of our approach.

1 Introduction

Recommender systems have been an important feature to recommend relevant items to relevant users in many online communities, e.g. Amazon, Netflix, and Foursquare. Some online systems allow users to provide an explicit rating for an item to express how much they like it. A higher (or lower) rating indicates that the user likes (or dislikes) the item more. Nevertheless, many recommender systems only have user's implicit feedback, such as browsing activity, purchasing history, watching history, click behavior and check-in information. As implicit feedback becomes more and more prevalent, this type of recommender system has attracted many researchers' attention [Joachims *et al.*, 2005; Lim *et al.*, 2015]. However, implicit feedback based recommender systems suffer from many challenges. For example, the sparseness of observed data (i.e., only a small percentage of user-item pairs have implicit feedback) increases the difficulty to learn user's exact taste on items. Also, different from explicit feedback, only positive preference is observed in implicit feedback. In other words, we have no prior knowledge about which items users dislike. This has been a thorny issue for learning task.

In the literature, some related work has been proposed to take advantage of implicit feedback for item recommendations. For example, [Hu *et al.*, 2008] regards user's preference for an item as a binary value, where a user's preference for observed item and unobserved one are viewed as one and zero, respectively. Then it fits these predefined ratings with vastly varying confidential levels based on matrix factorization framework. Although it assumes that a user prefers the observed items to unobserved ones, the quadratic in the formulation weakens their instinct ranking order. There is no guarantee that the higher accuracy in rating prediction will result in the better ranking effectiveness [N. and Qiang, 2008]. For instance, the true ratings for two items are $\{1, 0.5\}$. The predicted ratings $\{0.4, 0.6\}$ and $\{1.6, 0.6\}$ have the same prediction accuracy. But in fact, they lead to totally different ranking orders of items. To this end, Rendle formulates user's consuming behavior into the pairwise ranking problem, i.e., users are much more interested in their consumed items than unconsumed ones [Rendle *et al.*, 2009]. Due to a large number of such pairs, it only samples some negative items for the learning procedure. However, there are two limitations. First, the pairwise ranking increases the number of comparisons. Second, although Rendle [Rendle and Freudenthaler, 2014] improves the sampling skill by oversampling the top ranked items, sampling technique itself easily leads to bias. It is likely that the sampled negative item is already ranked below the positive one, which as a result has no contribution to the optimization.

To address these issues, we propose to relax the ranking model in [Rendle *et al.*, 2009] to eliminate the pairwise ranking. Specifically, the positive and negative feedbacks are separated by the positive and negative boundaries. In fact, the unobserved implicit feedback is often a mixture of negative and missed positive data, so a slack variable is introduced to capture such characteristic, which allows some negative feedbacks and positive feedbacks to be non-separate. Furthermore, instead of sampling, a smooth and scalable algorithm is designed to learn model's parameters based on group and instance level's optimization. The proposed algorithm allows to take all the unobserved items into account for optimization, and as a result addresses the bias caused by the sampling technique in [Rendle *et al.*, 2009;

Rendle and Freudenthaler, 2014]. Finally, the proposed model is evaluated with many state-of-the-art baseline models and different validation metrics on three real-world data sets. The experimental results demonstrate the superiority of our model for tackling implicit feedback based recommendations.

2 Preliminaries

The recommendation task addressed in this paper is defined as: given the consumption behaviors of N users over M items, we aim at recommending each user with top-K new items that he might be interested in but has never consumed before. Matrix factorization based models assume that $U \in \mathbb{R}^{K\times N}$ and $V \in \mathbb{R}^{K\times M}$ are the user and item latent feature matrices, with column vectors U_i and V_j representing the K-dimension user-specific and item-specific feature vectors of user i and item j, respectively. The predicted preference (rating) of user i for item j, denoted as $\hat{R}_{ij}$, is approximated by:

$$\hat{R}_{ij} = U_i^T V_j. \tag{1}$$

In implicit feedback datasets, only positive feedback is observed. As we lack substantial evidence on which items users dislike, their preference for an unobserved item is regarded as a mixture of negative and missing values. Along this line, [Rendle *et al.*, 2009] assumes that each user prefers the observed items over unobserved ones. Let us denote $\mathcal{M}_i^o$ as a set of items that user i has consumed and $\mathcal{M}_i^u$ as the remaining items that he never consumed. Then for user i, the ranking based on user's preference for an observed item j over an unobserved item k is given by:

$$U_i^T V_j > U_i^T V_k, \qquad \forall j \in \mathcal{M}_i^o \wedge \forall k \in \mathcal{M}_i^u \tag{2}$$

For convenience we will henceforth refer to i as user, j as observed item, and k as unobserved item unless stated otherwise. Eq.(2) models the correlation of user's preference for each pair of observed item and unobserved one. It is actually maximizing the Area Under the ROC Curve (AUC) for matrix factorization. The optimization problem for pairwise ranking is formulated as follows:

$$\max_{U,V} \sum_{i=1}^{N} \sum_{j\in\mathcal{M}_i^o} \sum_{k\in\mathcal{M}_i^u} \log\sigma(U_i^T V_j - U_i^T V_k) - \frac{\lambda_U}{2}||U||_F^2$$
$$- \sum_{i=1}^{N} \left(\frac{\lambda_{V_1}}{2} \sum_{j\in\mathcal{M}_i^o} ||V_j||_F^2 + \frac{\lambda_{V_2}}{2} \sum_{k\in\mathcal{M}_i^u} ||V_k||_F^2 \right),$$

where $\sigma(x)$ is the sigmoid function, i.e., $\sigma(x) = 1/(1+e^{-x})$; $||\cdot||_F$ is the Frobenius norm; and λ_U, λ_{V_1} and λ_{V_2} are regularization constraints. In optimization process, sampling negative items is adopted to avoid comparing with all unobserved items for each individual user. The optimal solution, as a result, can be obtained by Stochastic Gradient Descent (SGD).

3 Methodologies

3.1 The Relaxed Model

In implicit feedback systems, user's preference for the observed item is usually regarded as positive rating and there is no negative rating. Most research work argues that one user's preference for the observed item is supposed to be larger than that for any unobserved one [Rendle *et al.*, 2009; Hu *et al.*, 2008], which indicates the presence of ranking between positive and negative ratings. However, the pairwise ranking of a user's preference for the observed items over the unobserved ones is quite inefficient, especially when the user's historical data increases. To address this issue, we propose to relax the pairwise ranking. We treat one user's preference for an item as an point, where his positive (or negative) rating is viewed as the positive (or negative) point. Our goal is to make all positive points reside above all negative points. Inspired by the soft margin idea of SVM, we separate these two types of points by two different boundaries. In other words, user's preference for the observed items and unobserved ones are separated by the boundaries. Specifically, user's positive rating is located on or above a boundary represented as a numeric value r_+. On the other hand, his negative rating resides on or below another boundary. It not only improves the efficiency of comparisons, but also preserves the ranking information. Along this line, we relax Eq.(2) for $\forall j \in \mathcal{M}_i^o \wedge \forall k \in \mathcal{M}_i^u$ in the following:

$$\begin{cases} U_i^T V_j \geq r_+, \\ U_i^T V_k \leq r_- + \xi_{ik}, \end{cases} \tag{3}$$

where ξ_{ik} is the slack variable for user i on the unobserved item k. Similar to [Hu *et al.*, 2008], r_+ and r_- are set as one and zero, respectively. Even though there may be many unobserved items for a user, it does not necessarily indicate that he dislikes them. Probably he may be just unaware of them. In other words, some of the unobserved items might be those users are interested in, while others are actually those they dislike. To capture this characteristic, the slack variable ξ_{ik} in Eq.(3) is introduced to allow the mixture of negative feedback and positive feedback in unobserved data. Hence, we apply the following constraint for ξ_{ik} as:

$$r_- + \xi_{ik} \geq r_+ \quad \Rightarrow \quad \xi_{ik} \geq r_+ - r_-. \tag{4}$$

It is worth to note that for each user i, pairwise ranking in Eq.(2) needs $|\mathcal{M}_i^o||\mathcal{M}_i^u|$ comparisons; but our relaxed ranking in Eq.(3) only requires $M{=}|\mathcal{M}_i^o|{+}|\mathcal{M}_i^u|$ comparisons. Specifically, when $|\mathcal{M}_i^o|{=}M/2$, pairwise ranking needs $M^2/4$ comparisons while relaxed ranking only requires M comparisons.

However, Eq.(3) is a "hard" version of optimization problem due to the strict constraints. Thus, we make the constraints "soft" by introducing a plus function to penalize the violated constraints. The optimization problem with the softened constraints is formulated as follows:

$$\min_{U,V,\xi} \sum_{i=1}^{N} \left(\sum_{j\in\mathcal{M}_i^o} \left(-U_i^T V_j + r_+\right)_+ + \right.$$
$$\left. \sum_{k\in\mathcal{M}_i^u} \left(U_i^T V_k - \xi_{ik} - r_-\right)_+ \right) + ||\Theta||, \tag{5}$$

where $(\cdot)_+$ is the plus function [Zhu *et al.*, 2003], i.e., $(x)_+ = max(x, 0)$, and $||\Theta||$ is the regularization term given by:

$$||\Theta|| = \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_V}{2}||V||_F^2 + \lambda_\xi \sum_i^N ||\xi_i||_1,$$

where λ_U, λ_V and λ_ξ are the regularization constants, and $||\xi_i||_1$ is ℓ_1 norm of vector ξ_i. More important, one reason for imposing such a vector norm on ξ is that users are usually interested in a small percentage of unobserved items among all the remaining unobserved ones. It is worth to note that the formulation in Eq.(5) is different from a pure SVM scheme: (1) We relax the pairwise ranking problem which is adaptable for any pairwise ranking based application; (2) A novel and scalable optimization is designed for the objective function (in the next section). (3) The slack variable with ℓ_1 norm is introduced to capture the characteristics of unobserved data (i.e., the mixture of positive and negative data).

3.2 Optimization

The bottleneck for the optimization problem shown in Eq.(5) is the calculation of each user's ratings on all the unobserved items, which results in at least $\mathcal{O}(MNK)$ time complexity. Particularly, with the increase of the item number, it becomes more and more inefficient. As sampling negative items possibly leads to bias, in this paper we propose a smooth and scalable optimization algorithm to take all the unobserved items into consideration.

First, we solve the problem in a group-based optimization. The entire item set is randomly divided into G groups. It is worth to note that we adopt the random group technique in the experiments for the sake of efficient computation. Instead of requiring the rating on each unobserved item to reside on or below the negative boundary, we softly make the average rating of the unobserved items in each group locate on or below this boundary. Suppose $\mathcal{G}_h$ is the set of all items which belong to group h, and $\mathcal{G}_h^i$ is the set of unobserved items for user i with group as h, then we have the following inequality:

$$\frac{1}{|\mathcal{G}_h^i|}\sum\nolimits_{k\in\mathcal{G}_h^i} U_i^T V_k \le r_- + \xi_{ih},$$

where ξ_{ih} is the slack variable for user i on group h. Let us define the following variables:

$$V_h^s \equiv \sum_{p\in\mathcal{G}_h} V_p,\quad \tilde{V}_{ih}^s \equiv \sum_{j\in\mathcal{M}_i^o\wedge h(j)=h} V_j,\quad \bar{V}_h^i \equiv \frac{1}{|\mathcal{G}_h^i|}(V_h^s - \tilde{V}_{ih}^s),$$

where $h(j)$ is the group index of item j. In addition, the plus function is not twice differentiable and can be smoothly approximated by the integral to a smooth approximation of the sigmoid function [Lee and Mangasarian, 1999; Chen and Mangasarian, 1993] as follows:

$$(x)_+ \approx p(x,\alpha) = x + \frac{1}{\alpha}\log(1+\exp(-\alpha x)).$$

In the experiments, we find the above proximation form is slightly better than the logistic function. Hence, the objective function in Eq.(5) can be modified as the following:

$$\mathcal{L} = \underset{U,V,\xi}{\operatorname{argmin}} \sum_{i=1}^{N}\left\{\sum_{j\in\mathcal{M}_i^o} p\left(-U_i^T V_j + r_+, \alpha\right) + \sum_{h}^{G} p\left(U_i^T \bar{V}_h^i - \xi_{ih} - r_-, \alpha\right)\right\} + ||\Theta||, \tag{6}$$

where the slack variable has the constraint as $\xi_{ih} \ge r_+ - r_-$. We exploit the gradient descent based optimization procedure to obtain the optimal solutions for U, V and ξ in above problem. Their gradients are given by:

$$\frac{\partial\mathcal{L}}{\partial U_i} = -\sum_{j\in\mathcal{M}_i^o} p'_{ij}(U,V,\xi)V_j + \sum_{h=1}^{G}\bar{p}'_{ih}(U,V,\xi)\bar{V}_h^i + \lambda_U U_i, \tag{7}$$

$$\frac{\partial\mathcal{L}}{\partial V_j} = -\sum_{i\in\mathcal{N}_j^o} p'_{ij}(U,V,\xi)U_i + \sum_{i\in\mathcal{N}_j^u}\frac{\bar{p}'_{ih(j)}(U,V,\xi)U_i}{|\mathcal{G}_{h(j)}^i|} + \lambda_V V_j, \tag{8}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_{ih}} = -\bar{p}'_{ih}(U,V,\xi) + \lambda_\xi, \tag{9}$$

where $\mathcal{N}_j^o$ is the set of users who have consumed item j and $\mathcal{N}_j^u$ is the remaining users who have never consumed it. One caveat is that as the constraint placed on ξ guarantees it to be larger than 0 (i.e., its sign holds as positive), we can simply remove the absolute value sign to obtain its gradient. Also, $p'_{ij}(U,V,\xi)$ and $\bar{p}'_{ih}(U,V,\xi)$ are defined as the following:

$$p'_{ij}(U,V,\xi) = 1 - 1/(1+\exp(\alpha(-U_i^T V_j + r_+)),$$
$$\bar{p}'_{ih}(U,V,\xi) = 1 - 1/(1+\exp(\alpha(U_i^T\bar{V}_h^i - \xi_{ih} - r_-))).$$

To efficiently calculate the gradient of latent factor V, we rewrite Eq.(8) as the following (see the next section):

$$\frac{\partial\mathcal{L}(U,V,\xi)}{\partial V_j} = -\sum_{i\in\mathcal{N}_j^o} p'_{ij}(U,V,\xi)U_i - \sum_{i\in\mathcal{N}_j^o}\widetilde{V}_{h(j)i}(U,V,\xi) + \widetilde{V}_{h(j)}^s(U,V,\xi) + \lambda_V V_j, \tag{10}$$

where $\widetilde{V}_{hi}(U,V,\xi)$ and $\widetilde{V}_h^s(U,V,\xi)$ are defined as follows:

$$\widetilde{V}_{hi}(U,V,\xi) = \frac{1}{|\mathcal{G}_h^i|}\bar{p}'_{ih}(U,V,\xi)U_i,\quad \widetilde{V}_h^s(U,V,\xi) = \sum_{i=1}^{N}\widetilde{V}_{hi}(U,V,\xi).$$

The algorithm detail is shown in Figure 1, where adaptive learning rate is used to improve convergence rate [Orr, 1999]. Specifically, we sample a set of unobserved items for each user to reinforce the optimization granularity. It allows us to optimize the problem in both group and instance levels. Suppose for user i, we sample a set of unconsumed items, denoted as $\mathcal{A}$, and then select those items whose predicted ratings are larger than its group's mean rating to learn parameters. The selected item l for user i satisfies the following rule:

$$U_i^T V_l > U_i^T \bar{V}_{h(l)}^i. \tag{11}$$

Hence the objective function in Eq.(6) is refined by:

$$\mathcal{L}^{(new)} = \mathcal{L} + \sum_{i=1}^{N}\sum_{q\in\mathcal{A}_i} p(U_i^T V_q - \xi_{ih(q)} - r_-, \alpha)),$$

where $\mathcal{A}_i$ is the set of selected items that satisfy Eq.(11) and are unconsumed by user i. Therefore, the gradients of U, V, and ξ can be obtained similarly as above inference, given by:

$$\frac{\partial\mathcal{L}^{(new)}(U,V,\xi)}{\partial U_i} = \frac{\partial\mathcal{L}(U,V,\xi)}{\partial U_i} + \sum_{q\in\mathcal{A}_i}\bar{p}'_{ih(q)}(U,V,\xi)V_q,$$

$$\frac{\partial\mathcal{L}^{(new)}(U,V,\xi)}{\partial V_j} = \frac{\partial\mathcal{L}(U,V,\xi)}{\partial V_j} + \sum_{i\in\mathcal{A}_j}\bar{p}'_{ih(j)}(U,V,\xi)U_i,$$

$$\frac{\partial\mathcal{L}^{(new)}(U,V,\xi)}{\partial\xi_{ih}} = \frac{\partial\mathcal{L}(U,V,\xi)}{\partial\xi_{ih}} - \sum_{q\in\mathcal{A}_{ih}}\bar{p}'_{ih}(U,V,\xi),$$

where $\mathcal{A}_j$ is the set of users who have not consumed item j and $\mathcal{A}_{ih}$ is the set of items that belong to group h and are unconsumed by user i.

3.3 Complexity Analysis

In each iteration, updating U, V and ξ dominates the major time complexity for solving the optimal problem shown in Eq.(6). To update U_i, the first term of Eq.(7) requires time complexity as $\mathcal{O}(n_i K)$, where n_i is the number of items that user i has consumed. V_h^s is independent of i and can be pre-computed. Hence, computing $\tilde{V}_{ih}^s$ costs $\mathcal{O}(n_{ih}K)$, where n_{ih} is the number of items that user i has consumed and belong to group h, and $n_i = \sum_h n_{ih}$. Then, the second term of updating U_i needs time complexity as $\mathcal{O}(n_i K + GK)$. Consequently, the computation of U_i is performed in $\mathcal{O}(n_i K + GK)$. This procedure is performed over N users, so the total time is $\mathcal{O}(nK + NGK)$, where $n \equiv \sum_i n_i$.

In the procedure of optimizing V_j, the searching direction is dependent on $V^{(old)}$ obtained in the last iteration. The first term in Eq.(10) costs $\mathcal{O}(n_j K)$, where n_j is the number of users who have consumed item j. Similar to above, $\widetilde{V}_{ih}^{(old)\ s}$ in Eq.(10) can be pre-calculated. On the other hand, $\widetilde{V}_{hi}(U, V^{(old)}, \xi)$ is independent of j and could be also pre-computed. Then $\widetilde{V}_h^s(U, V^{(old)}, \xi)$ is pre-stored in the memory due to the sum of $\widetilde{V}_{hi}(U, V^{(old)}, \xi)$ over all users. Hence, the second term in Eq.(10) costs $\mathcal{O}(n_j K)$ time complexity. Thus, updating V_j needs running time as $\mathcal{O}(n_j K)$, and the total cost time over M items is $\mathcal{O}(nK)$, where $n = \sum_j n_j$.

Similar to update user latent matrix U, for each slack variable ξ_{ih}, we only need $\mathcal{O}(K)$ time complexity due to the pre-computation of V_{ih}^s. Totally, updating ξ over N users and G groups costs time complexity as $\mathcal{O}(NGK)$.

In instance-level optimization, similar to above analysis, in the worst situation, each iteration needs $\mathcal{O}(NAK)$ running time. In a summary, each iteration of our algorithm costs $\mathcal{O}(nK + N(A+G)K)$ time complexity, where n is the number of the observed entries in user-item matrix, and $\frac{N(A+G)}{n}$ is small. Therefore, the time complexity can be approximated by $\mathcal{O}(nK)$. In other words, the time complexity of each iteration for the optimization is a linear proportion to the number of observed user-item pairs.

```
Input: Observed user-item pairs, regularization constants $\lambda_U$, $\lambda_V$ and
       $\lambda_\xi$, dimension $K$ of latent space, group number $G$, stop criteria $\tau$
       and $maxIter$, and learning rate $\eta$
Output: $U^{(k)}$, $V^{(k)}$, $\xi^{(k)}$
Randomly initialize $U^{(0)}$ and $V^{(0)}$, $k \leftarrow 1$, $\varepsilon \leftarrow \infty$;
Randomly initialize $\xi^{(0)}$ and ensure $\xi_{ih}^{(0)} \geq r_+ - r_-$;
Randomly divide items into $G$ equal groups;
while $k \leqslant maxIter$ && $\varepsilon \geqslant \tau$ do
    Uniformly sample $\mathcal{A}$ unobserved items for each user $i$ and select those
      items which satisfies Eq.(11) to join to learn latent factors;
    for $j = 1$ to $M$ do
        $V_j^{(k+1)} \leftarrow V_j^{(k)} - \eta \frac{\partial \mathcal{L}^{(new)}(U^{(k)}, V^{(k)}, \xi^{(k)})}{\partial V_j}$;
    for $i = 1$ to $N$ do
        $U_i^{(k+1)} \leftarrow U_i^{(k)} - \eta \frac{\partial \mathcal{L}^{(new)}(U^{(k)}, V^{(k+1)}, \xi^{(k)})}{\partial U_i}$;
    for $(i, h) = (1, 1)$ to $(N, G)$ do
        $\xi_{ih}^{(k+1)} \leftarrow \max(r_+ - r_-, \xi_{ih}^{(k)} - \eta \frac{\partial \mathcal{L}^{(new)}(U^{(k+1)}, V^{(k+1)}, \xi^{(k)})}{\partial \xi_{ih}})$
    $\Delta \leftarrow \mathcal{L}^{(new)}(U^{(k)}, V^{(k)}, \xi^{(k)}) - \mathcal{L}^{(new)}(U^{(k+1)}, V^{(k+1)}, \xi^{(k+1)})$;
    Update $\eta$:   $\eta \leftarrow \begin{cases} 1.05\eta & \text{if } \Delta > 0, \\ 0.5\eta & \text{otherwise;} \end{cases}$
    $\varepsilon \leftarrow \frac{|\Delta|}{|\mathcal{L}^{(new)}(U^{(k)}, V^{(k)}, \xi^{(k)})|}$
    $k \leftarrow k + 1$
return $U^{(k)}$, $V^{(k)}$, $\xi^{(k)}$
```

Figure 1: Algorithm of Relaxed Ranking-based Factor Model

4 Experimental Results

4.1 Datasets

We use three datasets to evaluate the performance of our proposed model. The first two datasets are check-in data collected from Gowalla and Foursquare. Each check-in record in the dataset includes a user ID, a location ID, a check-in frequency and a timestamp that he first time checked-in this location. As we only know user's check-in action, these two data sets are the implicit feedback based data sets. The third data set is the movie data of MovieLens, where each user provides explicit ratings, 1 to 5 stars, for some movies. As our task focuses on the implicit feedback, similar to [Rendle *et al.*, 2009], we remove the rating score from the dataset. We utilize *consuming* to represent user's *checking-in* or *watching* behavior in these three datasets. We remove those users who have consumed less than 10 items. The detailed statistics of datasets are reported in Table 1.

As recommender system targets at recommending new items for users, we split the training and testing as the following. In check-in data, for each user, we first sort the records according to the check-in timestamp; and then select the earliest 80% to train the model and use the next 20% as testing. In MovieLens dataset, we randomly select 80% data as training and use the rest as testing due to the absent timestamp.

4.2 Parameter Settings

In the experiments, the regularization constants λ_U, λ_V and λ_ξ are set as 0.01. The number K of latent space is set as 10, the initial learning rate η is 0.001, and parameter α is set as 5. We divide 100 groups for three datasets. A in check-in data and MovieLens are 150 and 400, respectively.

4.3 Evaluation Metrics

We quantitatively evaluate model performance in terms of top-K recommendation performance, i.e., Precion@K and Recall@K (where K is different from the latent space dimension), and ranking performance, i.e., MAP and AUC [Li *et al.*, 2015]. Formally, their definitions are shown as:

$$P@K = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_i(K) \cap T_i|}{K}, MAP = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{\widetilde{M}_i} p(j) \times rel(j)}{|T_i|},$$

$$R@K = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_i(K) \cap T_i|}{|T_i|}, AUC = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{(j,k)\in E(i)} I(\hat{R}_{ij} > \hat{R}_{ik})}{|E(i)|},$$

where $S_i(K)$ is a set of top-K new items recommended to user i excluding those items in training, and T_i is a set of items that have been consumed in the testing. $\widetilde{M_i}$ is the number of the returned items in the list of user i, $p(j)$ is the precision of a cut-off rank list from 1 to j, and $rel(j)$ is an indicator function that equals to 1 if the item is in the testing, otherwise is 0. $E(i)$ is defined as $E(i) := \{(j,k) | j \in T_i \wedge k \notin (T_i \cup T_i^t)\}$, where T_i^t is a set of items consumed by user i in the training. $I(\cdot)$ is a indicator function, which equals to 1 if its argument is true and 0 otherwise.

Table 1: Statistics of Data Sets.

Dataset	#User	#Item	#Records	Sparsity
Gowalla	52,216	98,351	2,577,336	0.0399%
Foursquare	74,343	198,161	3,501,608	0.0238%
MovieLens	6,040	3,681	1,000,179	4.4986%

4.4 Baseline Methods

To comprehensively demonstrate the effectiveness of our proposed model, named as RRFM, we compare it with the following popular recommendation models.

- **UBPR** [Rendle *et al.*, 2009] models the pairwise ranking for each pair of the observed and unobserved items, and employs SGD with uniformly sampling for optimization.
- **ABPR** [Rendle and Freudenthaler, 2014] is similar to UBPR, but it over-samples the top ranked negative items based on a context-dependent sampling schema.
- **WRMF** [Hu *et al.*, 2008], treats user's rating as a binary value and fits the ratings on the observed and unobserved items with different confidential values.
- **PMF** [Salakhutdinov and Mnih, 2007] regards the rating as the dot product of user-specific and item-specific feature factors. Logistic function is used as rating conversion method to avoid the data's bias.

4.5 Performance Comparison

Ranking Performance Comparison.

The ranking performance in terms of MAP and AUC for our proposed model with baseline models is reported in Table 2. We summarize the following observations.

First, the method only modeling the observed feedback (i.e., PMF), performs much worse than those taking both observed and unobserved data into consideration (i.e., other four models). Specifically, in Gowalla data, RRFM achieves result as 95.673% in terms of AUC while PMF only obtains 62.005%. Also, RRFM is 3.793% in terms of MAP while PMF is 1.357%. Different from explicit feedback, it is difficult to know user's negative preference from implicit feedback. As a result, explicit feedback based model is difficult to achieve better performance for implicit feedback. The result indicates that the unobserved data also provides meaningful information and assist to improve ranking performance.

Second, ranking-based methods (i.e., UBPR, ABPR and RRFM) are nearly better than rating prediction-based models (i.e., WRMF and PMF), particularly in terms of AUC metric. Ranking-based methods are modeling the ranking order of user's positive rating over negative rating, which actually maximizes the AUC metric; while rating prediction-based models focus on the task how to correctly predict ratings. Although WRMF regards user's positive and negative ratings as one and zero respectively, there is no guarantee for the correctness of their ranking order due to rating prediction based loss. This explains why it has poor performance in AUC.

Third, our proposed model is superior to others, such as in Foursquare, RRFM outperforms WRMF 100.7% and 12.9% in terms of MAP and AUC, respectively; RRFM achieves 15.2% improvement over ABPR in MAP. It happens due to

Table 2: Performance comparison in terms of MAP and AUC metrics. The result is reported in percentage (%).

Gowalla Dataset					
Metric	RRFM	WRMF	ABPR	UBPR	PMF
MAP	**3.793**	2.634	3.169	3.080	1.357
AUC	**95.673**	88.748	94.732	94.642	62.005
Foursquare Dataset					
MAP	**2.302**	1.147	1.999	1.913	0.024
AUC	**95.814**	84.866	94.726	94.554	61.211
MovieLens Dataset					
MAP	**21.108**	20.528	19.659	19.512	10.159
AUC	**93.795**	91.743	92.417	92.372	85.685

two reasons. First, we separate the positive and negative ratings by the soft boundaries, where some of them are non-separate. It captures user's actual consuming behavior where he does not consume an item probably due to his dislike or unawareness. Second, we optimize the model in group and instance granularity, where group-based optimization makes parameters reach their optimal solutions in a direction towards the minimum of cost function and instance-based optimization reinforces the approximation to the exact optimization problem in Eq.(5). In addition, WRMF has different performance in MAP due to the much less items and much denser user-item entries in MovieLens than check-in data.

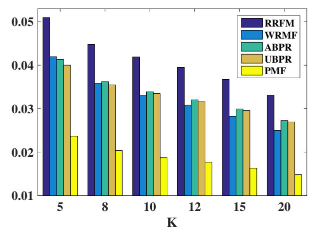
Last, ABPR performs slightly better than UBPR. Instead of uniformly sampling negative items, ABPR samples with an adaptive distribution. It exploits item tailed characteristics (i.e., over-sampling the top ranked negative items) to speed up the convergence of SGD. But with the increase of iteration, UBPR approximates to the true optimal solution as ABPR. It is the reason why it performs similarly as ABPR.

Top-K Recommendation Performance Comparison.

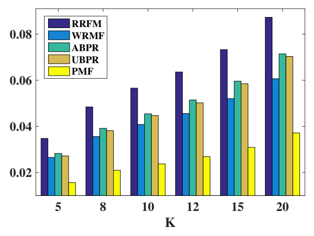
The top-K performance of our proposed model versus baseline methods over three datasets is plotted in Figure 2. Based on the observations, we summarize as follows:

First, RRFM outperforms all baseline methods. In particular, it has significant improvement over PMF, which contributes to the involvement of unobserved data for modeling. Its superior performance than ABPR and UBPR is beneficial from the following reasons: (1) The relaxed model allows user's preference for some unobserved items to mix with positive preference. This is based on the assumption that the unobserved data might be actually negative or missed positive value. (2) It optimizes the ranking problem based on the entire unobserved items with a smooth approximation, which addresses the bias caused by ABPR and UBPR. Furthermore, RRFM obtains much better performance than WRMF. Although WRMF also considers the ranking of positive rating over negative one, it models the recommendation problem by fitting the binary ratings, which cannot guarantee their actual ranking order. Hence, it does not perform as well as RRMF. In MovieLens, the improvement of RRFM over WRMF is small due to that the denser observed data helps WRMF approximate to the true optimal point.

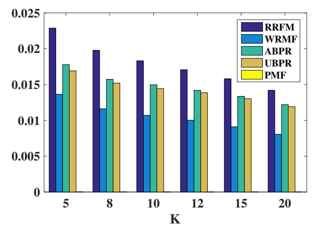
Second, ABPR and UBPR are superior to WRMF in check-



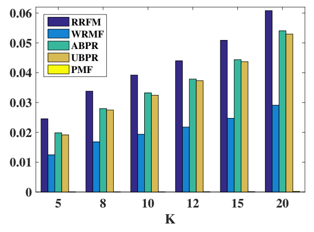
(a) Precision@K on Gowalla



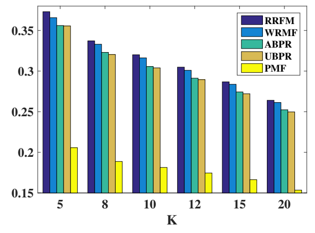
(b) Recall@K on Gowalla



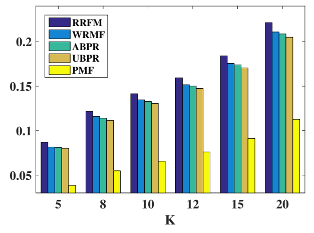
(c) Precision@K on Foursquare



(d) Recall@K on Foursquare



(e) Precision@K on MovieLens



(f) Recall@K on MovieLens

Figure 2: Performance comparison in terms of precision and recall on Gowalla, Foursquare and MovieLens datasets.

in data, while they are a little worse than WRMF in MovieLens data. ABPR and UBPR actually optimize the AUC metric which evaluates the ranking of one user's preference for the testing items over the remaining unobserved items. To make user's positive ratings larger than most of the negative ones, they might sacrifice some top-K performance. It is a tradeoff between the performance in AUC and top-K. Hence although they obtain extremely superior performance in terms of AUC, it cannot guarantee that they can also perform well in top-K performance. That's why they have poor performance in terms of precision@K and recall@K in MovieLens dataset.

Last, ABPR, UBPR and PMF perform consistently with the ranking performance, where ABPR is better than UBPR, and PMF performs the worst among all models. Although the way to sample negative items has been improved, ABPR is difficult to make significant improvements due to the limitation of sampling technique. PMF's poor result further demonstrates the importance of modeling all data.

5 Related Work

In this section we briefly review some existing work on recommender system. From the point of view of optimization method, we group them into two categories.

The first category is ranking-based model [Park *et al.*, 2015; Pan and Chen, 2013]. The ranking-based models optimize different ranking metrics [Weston *et al.*, 2013; Yang *et al.*, 2011; Liu *et al.*, 2015; 2014]. The first common type is based on AUC [Dhanjal *et al.*, 2015]. Specifically, [Rendle *et al.*, 2009] proposed to model pairwise ranking of user's preference for observed item over unobserved item, where the ranking is approximated by a logistic function. Parameter estimation was obtained by SGD with bootstrap sampling. Later, [Rendle and Freudenthaler, 2014] improved the convergence rate of such SGD by oversampling the top ranked negative items. However, [Balakrishnan and Chopra, 2012; Taylor *et al.*, 2008; Weimer *et al.*, 2008] proposed to optimize the Normalized Discounted Cumulative Gain (NDCG) and [Shi *et al.*, 2012a] optimized the Mean Average Precision (MAP). On the other hand, [Shi *et al.*, 2013; 2012b] proposed to optimize the Mean Reciprocal Rank (MRR). Most of them either only model the observed data or highly rely on sampling negative items, which are different from our work.

The second category is rating prediction-based model [Salakhutdinov and Mnih, 2007; Koren *et al.*, 2009; Pan *et al.*, 2008; Takcs and Tikk, 2012]. The rating prediction-based model is to minimize the square error loss between the observed rating and the estimated rating. [Salakhutdinov and Mnih, 2007] proposed to approximate the observed rating by the dot product of user and item latent factors, where the user and item latent factors were drawn from Guassian distribution. [Hu *et al.*, 2008] treated rating as a binary value, where the user's preferences for observed and unobserved items were regarded as one and zero, respectively. Then it exploited a large weight to fit those observed ratings and a small weight to fit unobserved ones. Similar to this idea, [Pan *et al.*, 2008] formulated the implicit feedback based item recommendation as one-class problem, and proposed to sample some negative items to improve the efficiency. Although these two methods model both positive and negative ratings, they spend too much effort on rating prediction.

6 Conclusion

In this paper, we propose a relaxed ranking-based algorithm for item recommendation with implicit feedback, and design a smooth and scalable optimization method for model's parameter estimation. The relaxed model not only avoids the pairwise ranking by separating the positive feedback and negative feedback with the soft boundaries, but also exploits the non-separate property of negative and positive feedback to capture the characteristic of unobserved data. To evaluate our proposed model, we conduct extensive experiments with many baseline methods and evaluation metrics on the real-world data sets. The experimental results have shown our model's effectiveness.

Acknowledgments

This work is partially supported by the National 973 Program of China under grant 2014CB347600, NIH (1R21AA023975-01), NSFC (71571093, 71372188, 61572032), National Center for International Joint Research on E-Business Information Processing (2013B01035), and National Key Technologies R&D Program of China (2013BAH16F03).

References

[Balakrishnan and Chopra, 2012] Suhrid Balakrishnan and Sumit Chopra. Collaborative ranking. In *the fifth ACM international conference on Web search and data mining (WSDM)*, pages 143–152, 2012.

[Chen and Mangasarian, 1993] Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71:51–69, 1993.

[Dhanjal *et al.*, 2015] Charanpal Dhanjal, Romaric Gaudel, and Stphan Clmenon. Collaborative filtering with localised ranking. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2554–2560, 2015.

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

[Joachims *et al.*, 2005] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005.

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

[Lee and Mangasarian, 1999] YUH-JYE Lee and O. L. Mangasarian. Ssvm: A smooth support vector machine for classification. Technical report, Computational Optimization and Applications, 1999.

[Li *et al.*, 2015] Huayu Li, Richang Hong, Shiai Zhu, and Yong Ge. Point-of-interest recommender systems: A separate-space perspective. In *IEEE International Conference on Data Mining (ICDM)*, pages 231–240, 2015.

[Lim *et al.*, 2015] Daryl Lim, Julian McAuley, and Gert Lanckriet. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 309–312, 2015.

[Liu *et al.*, 2014] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD*, pages 51–62, 2014.

[Liu *et al.*, 2015] Siyuan Liu, Shuhui Wang, and Feida Zhu. Structured learning from heterogeneous behavior for social identity linkage. *IEEE Trans. Knowl. Data Eng.*, 27(7):2005–2019, 2015.

[N. and Qiang, 2008] Liu Nathan N. and Yang Qiang. Eigenrank: A ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–90, 2008.

[Orr, 1999] Genevieve Orr. Momentum and learning rate adaptation, 1999.

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

[Pan *et al.*, 2008] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *ICDM*, pages 502–511, 2008.

[Park *et al.*, 2015] Dohyung Park, Joe Neeman, Jin Zhang, Sujay Sanghavi, and Inderjit S. Dhillon. Preference completion: Large-scale collaborative ranking from pairwise comparisons. In *ICML*, page 19071916, 2015.

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

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 452–461, 2009.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[Shi *et al.*, 2012a] Yue Shi, Karatzoglou Alexandros, Baltrunas Linas, Martha A. Larson, Alan Hanjalic, and Oliver Nuria. Tfmap: Optimizing map for top-n context-aware recommendation. In *SIGIR*, pages 155–164, 2012.

[Shi *et al.*, 2012b] 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 *RecSys*, pages 139–146, 2012.

[Shi *et al.*, 2013] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: Collaborative less-is-more filtering. In *IJCAI*, 2013.

[Takcs and Tikk, 2012] Gbor Takcs and Domonkos Tikk. Alternating least squares for personalized ranking. In *RecSys*, pages 83–90, 2012.

[Taylor *et al.*, 2008] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimizing non-smooth rank metrics, 2008.

[Weimer *et al.*, 2008] Markus Weimer, Alexandros Karatzoglou, Quoc V. Le, and Alex J. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems 20*, pages 1593–1600. Curran Associates, Inc., 2008.

[Weston *et al.*, 2013] Jason Weston, Hector Yee, and Ron J. Weiss. Learning to rank recommendations with the k-order statistic loss. In *RecSys*, pages 245–248, 2013.

[Yang *et al.*, 2011] Shuang-Hong Yang, Bo Long, Alexander J. Smola, Hongyuan Zha, and Zhaohui Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *SIGIR*, pages 295–304, 2011.

[Zhu *et al.*, 2003] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *NIPS*, page 16. MIT Press, 2003.