HOSTED BY ELSEVIER

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Prior-based bayesian pairwise ranking for one-class collaborative filtering

Qian Zhang [a], Fuji Ren [a,*]

[a]Faculty of Engineering, Tokushima University, 7700814, Japan

ARTICLE INFO

Article history
Received 00 December 00
Received in revised form 00 January 00
Accepted 00 February 00

Keywords:
One-class collaborative filtering
Prior information
Pairwise ranking
Recommendation method

ABSTRACT

In many real-world applications, only user-item interactions (one-class feedback) can be observed. The recommendation methods have been studied for personalized ranking with one-class feedback in recent years. Pairwise ranking methods have been widely used for dealing with the one-class problem with the assumption that users prefer their observed items over unobserved items. However, existing some items that users have not seen yet. It is unsuitable for treating all unobserved items of the user as negative feedback. In this paper, we propose a Prior-based Bayesian Pairwise Ranking (PBPR) model, which relaxes the simple pairwise preference assumption in previous works by further considering the pairwise preference between two unobserved items. Moreover, we calculate users' potential preference scores on unobserved items, i.e., prior information, based on historical interactions. The prior information can be used to measure the fine-grained preference difference between any two unobserved items of each user. Through extensive experiments on real-world datasets, we demonstrate the effectiveness of our proposed recommendation method.

1 Introduction

With the rapid spread of multimedia Web applications, a large number of contents are being generated online in real-time. Users will spend much effort in finding interesting things from the massive data. Therefore, recommender systems are proposed to deal with information overload and meet users' personalized interests. Almost every service (search engines, social media sites, E-commerce, and news portals) that provides the content to users is equipped with a recommender system [1]. In recommender systems, item represents different kinds of contents consumed by users [2], like a movie, a song, or a book. Collaborative filtering (CF) based methods [3], which take account of user-item interactions, have made great satisfactory success, among different recommend strategies [4]. However, the sparsity of user-item interactions usually influences the performance of CF-based recommendation methods.

The recommendation problem for the dataset, like MovieLens [5], in which users give the rating values scale 1-5, can be considered as a multi-class task. For example, the accurate preference scores of users on different movies are predicted based on multi-class methods. However, in many datasets, only one-class feedback is available [6], e.g., bought products, watched movies, and clicked Web pages. Such datasets only have each user's positive feedback, usually called one-class feedback [7] or implicit feedback [8]. The one-class problem is different from that of the five stars rating prediction problem since the former only contain observed items rather than both positive items and negative items in the latter. The one-class collaborative filtering methods aim to rank unobserved items for users, and the multi-class methods aim to predict users' rating scores on items.

The recommendation methods for solving the one-class problem, which is also the implicit feedback problem, can be divided into two branches based on previous researches: pointwise regression methods [9,10] and pairwise ranking methods [11,12]. The pointwise regression methods, which take observed items as absolute preference scores [13] for the one-class problem, learn latent representations of users and items to represent users' preference scores and minimize a pointwise square loss to approximate the absolute scores. The pairwise ranking methods take an observed item and an unobserved item of a user as a triple <user, an observed item, an unobserved item>, assuming that users prefer observed items to unobserved items, and maximize the likelihood of pairwise preferences over observed items and unobserved items. Empirically, the pairwise ranking methods [14,15], which have

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: 501947005@tokushima-u.ac.jp (Q. Zhang), ren@is.tokushima-u.ac.jp (F. Ren)

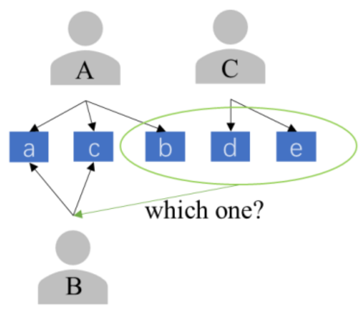

Fig. 1: Illustration of the example of three users' implicit feedback on five movies.

been successfully adopted in many scenarios [16,17], achieve much better performance than pointwise methods [7,10]. Bayesian personalized ranking (BPR) [14] is one of the most popular pairwise ranking methods. The optimization process of the BPR model assumes equal importance of the huge unobserved items.

We find that the historical interactions between users and items provide resources about users' partial preferences. Specifically, we could discover a user's potential preference on his/her unobserved items, according to some explainable connections existing between observed items and unobserved items. For example, a recommender system contains three users (*A*, *B*, and *C*) and five movies (a, b, c, d, and e). The interactions between users and items are shown in Fig. 1. Movie *b* is a nice choice while recommending a new movie for user *B*. Considering that user *B* has the same historical interactions (*a* and *c*) with user *A*, user *A* can be considered as the like-minded user of user *B*. Thus, user *B* may be interested in the movie, which has been watched by user *A*. Additionally, users' potential preferences on items provide resources about potential interactions between users and items, which can alleviate the data sparsity problem and can also be used to evaluate the fine-grained preference difference between any two unobserved items.

In this paper, we propose a novel recommendation model called Prior-based Bayesian Pairwise Ranking (PBPR), which further considering the pairwise preference between two unobserved items, for attempting to relax the assumption adopted by BPR with equal importance of the unobserved items. The preference difference between two unobserved items can be determined by users' potential preference scores, which are calculated based on historical interactions. Our proposed recommendation method does not need any side information except the user-item interactions. The key contributions in this research are summarized as below:

- We propose a novel pairwise ranking model named Prior-based Bayesian Pairwise Ranking (PBPR), which relaxes the assumption in BPR about unobserved items by taking into account the preference difference between unobserved items of a user.
- We calculate potential preference scores between users and items based on user-item interactions to measure the fine-grained preference difference between any two unobserved items of the user.
- We conduct extensive experiments on three real-world datasets, demonstrating the effectiveness of our proposed recommendation method.

The remainder of this paper is organized as follows: Section 2 discusses related works. The problem definition and base models are given in Section 3. Section 4 describes our solution in detail, and Section 5 provides experimental results and analysis based on real-world datasets. Section 6 is the presentation of conclusions and future work.

2 Related works

One-class collaborative filtering is an emerging setup in collaborative filtering. One-class collaborative filtering recommendation methods are proposed for solving problems in real-world scenarios that only positive examples can be observed [18]. Compared with the traditional collaborative filtering setting where the data has ratings, one-class collaborative filtering is more realistic in many scenarios when no explicit feedbacks are available [19].

The existing personalized recommendations with one-class feedback generally focus on how to discover the missing interactions into personalized preference modeling. Zhou et al. [20] proposed the MUPL (Multi-facet user preference learning) framework for the fine-grained item recommendation, which considers different types of auxiliary feedback, to generate a high-quality personalized recommendation. Guo et al. [21] devised a linear regression model to learn the correlations between auxiliary feedback and target feedback and selected a number of nearest neighbors as positive items based on the set of items purchased by the user. They proposed a novel ranking model to help accommodate both the original and generated data and conducted experiments to demonstrate its performance. However, in practice, not all recommendation datasets contain social information. Besides, the aforementioned personalized recommendation methods are designed for applications including some specific information like the auxiliary implicit feedback of view or click in [20] and [21].

Bayesian personalized ranking (BPR) is a well-performed pairwise ranking recommendation approach, many previous methods optimized by BPR [22,23]. Existing pairwise ranking methods extend BPR for better recommendation results [24,25] by relaxing the assumptions before or taking into account more features like visual features from product images [26]. Inspired by the limitation of the assumption that individual preference alone in [14], Pan et al. [27] proposed a novel model called GBPR (Group Bayesian personalized ranking), which introduces group preference into the novel model to relax the individual and independence assumptions in BPR, enhance the richer interactions among users. The experimental results demonstrate the effect of group preference assumption in GBPR. Nevertheless, the equal importance for a large number of unobserved items of each user is unreasonable because the user might not notice some items rather than dislikes them. For distinguishing the difference between unobserved items, Yu et al. [28] divided the unobserved items into the possibly negative items and the unknown preference for the items. Zhou et al. [29] categorized the unobserved feedback into potential feedback and negative feedback and assumed user *u* prefers items with positive feedback over items with potential feedback, and user *u* prefers items with potential feedback over items with negative feedback. The preference comparison strategies for unobserved items are carried between unobserved items from different groups. The preference difference between two unobserved items in the same group could not be measured.

We propose a generic recommendation approach, which is also an extension of BPR and does not need any social information, for datasets containing user-item interactions. Moreover, we aim to measure the fine-grained preference difference between any two unobserved items of the user rather than dividing unobserved items into different groups and distinguishing the preference difference between unobserved items from different groups.

3 Preliminaries

We give the problem definition and then review the MF algorithm and the BPR algorithm in this section.

3.1 Problem Definition

We use $U^{tr} = \{u\}_{u=1}^{M}$ and $I^{tr} = \{i\}_{i=1}^{N}$ to denote the sets of users and items and denote a user-item interaction matrix as $R \in \mathbb{R}^{M\times N}$, where M and N represent the number of users and items, respectively. We use R_{ui} to record the interaction between a user u and an item i. $R_{ui} > 0$ indicates user u has interacted with the item i, and $R_{ui} = 0$ indicates user u has no interaction with the item i. Our goal is to recommend items to users that they may like from their unobserved items.

3.2 Matrix Factorization

Collaborative filtering, which can be divided into memory-based collaborative filtering [30] (user-based collaborative filtering and item-based collaborative filtering) and model-based collaborative filtering [31], is an essential technique for solving the problem of recommender systems. Matrix factorization [32] is widely applied in model-based collaborative filtering tasks because of its scalability and flexibility.

Matrix factorization algorithm decomposes the user-item interaction matrix $R_{M\times N}$ into the inner product of two low-rank matrices $U_{M\times k}$ and $V_{N\times k}$.

$$R_{M\times N} \approx U_{M\times k} \cdot V_{N\times k}^{T} \tag{1}$$

where $U_{M\times k}$ and $V_{N\times k}$ represent the user latent factor matrix and the item latent factor matrix. k represents the number of dimensionality and k≪min (M, N). We use U_u to denote the latent factor vector of user u and V_i to denote the latent factor vector of item i.

3.3 Bayesian Personalized Ranking

BPR is a widely used classical matrix factorization algorithm [33]. It makes pairwise learning and models a triplet of a user and two items, where one item is the observed item of the user, and the other one is the user's unobserved item. The individual preference score of user u on item i is denoted as r_{ui}, which is calculated by the inner product of U_u and V_i.

$$r_{ui} = U_u \cdot V_i^{T}. \tag{2}$$

BPR assumes the user prefers the observed item to the unobserved item, and the likelihood of pairwise preferences of users is independent. The overall likelihood of BPR among users could be formulated as

$$\text{BPR} = \prod_{u\in U^{tr}} \prod_{i\in I_u^{+}} \prod_{j\in I^{tr}\setminus I_u^{+}} Pr\left(r_{ui} > r_{uj}\right)\left[1 - Pr\left(r_{uj} > r_{ui}\right)\right] \tag{3}$$

where user u has observed items I_u^{+} and unobserved items $I^{tr}\setminus I_u^{+}$. Item i is an observed item from I_u^{+}, and item j is a random sampled unobserved item. $r_{ui} > r_{uj}$ represents user u prefers item i to item j.

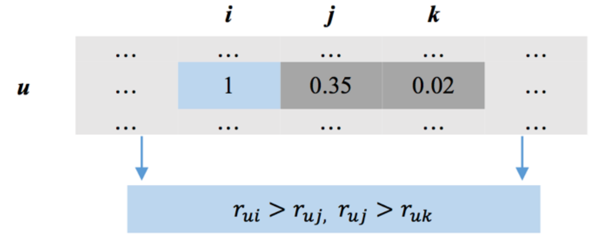

Fig. 2: Illustration of preference assumption based on prior information.

4 Our Solution

4.1 Prior-based Bayesian Pairwise Ranking

Besides observed items I_u^{+}, the remaining items which are unobserved by users could be attributed to two reasons: the user has not seen these items, or the user is not interested in these items [28]. Different from the assumption adopted by the BPR with equal importance of the unobserved items, we consider user's preferences on unobserved items are not equal, and the preference difference between unobserved items could be measured by potential preference scores between users and items. We give an illustration of preference assumption based on prior information in Fig. 1. For user u, we record "1" for his/her observed items and record the user's potential preference scores on unobserved items in the grey area with a value varied from 0 to 1. We consider that item j has a higher possibility of attracting the user u's attention than item k according to their potential preference scores. $r_{uj} > r_{uk}$ represents the preference difference between item j and item k. Inspired by this assumption, we can simply extend BPR to our proposed new model called Prior-based Bayesian Pairwise Ranking (PBPR). The tentative likelihood for users and items could be written as follows:

$$\text{PBPR} = \prod_{u\in U^{tr}} \prod_{i\in I_u^{+}} \prod_{j,k\in I^{tr}\setminus I_u^{+}} Pr\left(r_{ui} > r_{uj}, r_{uj} > r_{uk}\right) \left[1 - Pr\left(r_{ui} < r_{uj}, r_{uj} < r_{uk}\right)\right] \tag{4}$$

where I_u^{+} represents observed items of user u. j and k are two random sampled unobserved items of user u, and item j is more likely to be observed by the user than item k.

The shortcoming of PBPR in Eq. (4) is that its performance largely depends on the accuracy of potential preference scores. We consider that when the individual preference r_{uj} is fused with an individual preference r_{wj} ($w \in U_j^{tr}$, U_j^{tr} represents the set of users who observe item j), the fused preference r_{Nuj} is more likely to be higher than the individual preference r_{uk} comparing with the individual preference r_{uj}. Based on the above consideration, the likelihood of our proposed PBPR model could be given by:

$$\text{PBPR} = \prod_{u\in U^{tr}} \prod_{i\in I_u^{+}} \prod_{j,k\in I^{tr}\setminus I_u^{+}} Pr\left(r_{ui} > r_{uj}, r_{Nuj} > r_{uk}\right) \left[1 - Pr\left(r_{ui} < r_{uj}, r_{Nuj} < r_{uk}\right)\right] \tag{5}$$

where $r_{Nuj} = \lambda_1 r_{wj} + (1-\lambda_1) r_{uj}$ represents the fused preference on item j, λ_1 is a trade-off parameter used to fuse two different users' preferences on item j.

4.2 Generation prior information

As mentioned before, our proposed PBPR model is designed, further considering the pairwise preference between two unobserved items of each user. Therefore, we need prior information

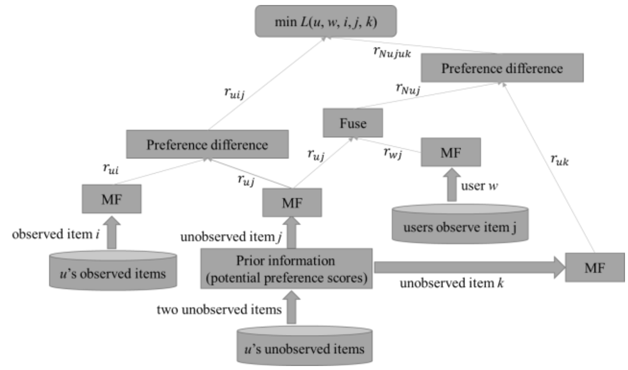

Fig. 3: Illustration of each randomly sampled record in PBPR. For each user u, an observed item i and two unobserved items are selected. The unobserved item with a higher potential preference score is item j, another item k. User w has positive feedback on item j.

about each user's potential preference scores on items to measure the fine-grained preference difference between unobserved items. In this section, we focus on describing the calculation method of potential preference scores between users and items based on their historical interactions.

We consider that the user's potential preferences on items could be measured by his/her like-minded users' preferences on items and the similarity between observed items and unobserved items. Therefore, we firstly employ cosine similarity to measure the item-based similarity (IUS, IUS $\in \mathbb{R}^{M\times M}$) between user u and user w based on the user-item interaction matrix R:

$$\mathrm{IUS}(u,w) = \frac{\sum_k R_{u_k} R_{w_k}}{\sqrt{\sum_k R_{u_k}^2}\sqrt{\sum_k R_{w_k}^2}} \tag{6}$$

where k is the dimensionality of user latent factor vector, $u \epsilon U^{tr}$, and $w \epsilon U^{tr}$. R_u represents the latent factor vector of user u. For each user, we rank other users according to user similarities calculated by Eq. (6). Only the top-5 users with the highest similarities are chosen as like-minded users. The user-based item similarity (UIS, UIS $\in \mathbb{R}^{N\times N}$) between item i and item j can also be calculated as follows:

$$\mathrm{UIS}(i,j) = \frac{\sum_k R_{:,i_k} R_{:,j_k}}{\sqrt{\sum_k R_{:,i_k}^2}\sqrt{\sum_k R_{:,j_k}^2}} \tag{7}$$

where k is the dimensionality of the item latent factor vector, $i \epsilon I^{tr}$, and $j \epsilon I^{tr}$. $R_{:,i}$ represents the latent factor vector of item i.

We denote the potential preference score matrix as S ($S \in \mathbb{R}^{M\times N}$), and S_{ui} ($S_{ui} \in [0,1]$) represents the potential preference score of user u on item i. S_{ui} is fixed as 1 when the item i is an observed item of user u. It will not be changed during the calculation of potential preference scores. Other values in S_{ui} are calculated in the sense of the user-based and item-based collaborative filtering, which could be formulated as follows.

$$S'_u = \frac{R_u UIS}{\sum_i R_{ui}} \tag{8}$$

$$S''_u = \frac{1}{|G_u|}\sum_{w\in G_u}(\mathrm{IUS}(u,w)S'_w) \tag{9}$$

$$S_u = \frac{1}{2}S'_u + \frac{1}{2}S''_u \tag{10}$$

In Eq. (8), S'_u represents potential preference score vector of user u from user u's observed items, which are calculated based on the multiplication of R_u, UIS and the number of observed items of user u. S''_u represents the potential preference score vector of user u from user u's like-minded users. G_u represents the group of like-minded users of user u, and $\mathrm{IUS}(u,w)$ denotes the user similarity between user u and user w. In S'_u and S''_u, the observed items of user u are all fixed as 1. Finally, the potential preference score vector of user u (S_u) is calculated by Eq. (10) based on S'_u and S''_u. Considering the ranking list of user's unobserved items can be determined by his/her potential preference scores, we denote this calculation method as U & I recommendation method and evaluate its performance in Section 5.

4.3 Learning the model

In this paper, we represent $r_{ui} > r_{uj}$, $r_{N_{uj}} > r_{uk}$ as $\lambda(r_{ui} - r_{uj}) + (1-\lambda)(r_{N_{uj}} - r_{uk})$, where λ is a control coefficient used to fuse their relations. To maximize the posterior probability for PBPR, we employ $\sigma(x) = 1/(1+\exp(-x))$ to approximate the probability $Pr(\cdot)$, following the BPR algorithm, where σ represents the logistic sigmoid function. The log-likelihood is also employed to reduce the calculation complexity of PBPR. For each randomly sampled record, it includes a user u, an observed item i of user u, two unobserved items j, k of user u, and a user w, where $w \in U_j^{tr}$. The illustration of this process is shown in Fig. 3. The objective function can be written as:

$$\begin{aligned} L(u,w,i,j,k) = &-\ln\sigma\Big(\lambda\big(r_{ui} - r_{uj}\big) \\ &+ (1-\lambda)\big(r_{N_{uj}} - r_{uk}\big)\Big) + \frac{\beta}{2}(\|U_u\|^2 \\ &+ \|U_w\|^2 + \|V_i\|^2 + \|V_j\|^2 + \|V_k\|^2) \end{aligned} \tag{11}$$

where $\|U_u\|^2$, $\|U_w\|^2$, $\|V_i\|^2$, $\|V_j\|^2$ and $\|V_k\|^2$ are regularization terms to prevent overfitting in the learning process, and β is the hyper-parameter to tune the regularization terms. The individual preference score is modeled by matrix factorization. Different implementations of prediction functions and their experimental results will be discussed in Section 5.2.1 and Section 5.3.

Following the well-known stochastic gradient descent (SGD) algorithm, each parameter could be updated as follows,

$$\theta \leftarrow \theta - \eta\frac{\partial L(u,w,i,j,k)}{\partial\theta} \tag{12}$$

where θ can be U_u, U_w, V_i, V_j, or V_k, and $\eta > 0$ is the learning rate. We represent the fuse of the preference difference between r_{ui} and r_{uj}, and the preference difference between r_{Nuj} and r_{uk} as $r_{uiuj;Nujuk}$. So, we can derive the following gradients for users:

$$\begin{aligned} \frac{\partial L(u,w,i,j,k)}{\partial U_u} = &\frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} \\ &\times\big(\lambda V_i + \big((1-\lambda)(1-\lambda_1) - \lambda\big)V_j \\ &- (1-\lambda)V_k\big) + \beta U_u \end{aligned} \tag{13}$$

Algorithm 1: Learning parameters for the PBPR model

Input:

User-item interaction matrix R;

Potential preference score matrix S;

Parameters η, λ, λ_1, and β;

Output:

The learned model parameters $\{U_u, V_i, u \in U^{tr},\ i \in I^{tr}\}$.

Randomly initialize U and V;
For $t_1 = 1, \dots, T$
  For $t_2 = 1, \dots, t$
    Randomly pick a user $u \in U^{tr}$;
    Randomly pick an item $i \in I_u^+$;
    Randomly pick two items $j, k \in I^{tr} \backslash I_u^+, S_{uj} > S_{uk}$;
    Randomly pick a user $w \in U_j^{tr}$;
    Calculate r_{Nuj} and $\frac{\partial L}{\partial r_{uiuj;Nujuk}}$;
    Update U_u via Eq. (13);
    Update U_w via Eq. (14);
    Update V_i via Eq. (15);
    Update V_j via Eq. (16);
    Update V_k via Eq. (17);
  End
End

$$\frac{\partial L(u,w,i,j,k)}{\partial U_w} = \frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} \times \left(\lambda_1(1-\lambda)V_j\right) + \beta U_w \tag{14}$$

and we can get the gradients for items as follows:

$$\frac{\partial L(u,w,i,j,k)}{\partial V_i} = \frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} \times (\lambda U_u) + \beta V_i \tag{15}$$

$$\frac{\partial L(u,w,i,j,k)}{\partial V_j} = \frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} \times \Big(\big((1-\lambda)(1-\lambda_1)-\lambda\big)U_u + \lambda_1(1-\lambda)U_w\Big) + \beta V_j \tag{16}$$

$$\frac{\partial L(u,w,i,j,k)}{\partial V_k} = \frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} \times \left(-(1-\lambda)U_u\right) + \beta V_k \tag{17}$$

where $\frac{\partial L(u,w,i,j,k)}{\partial r_{uiuj;Nujuk}} = -\frac{\exp(-r_{uiuj;Nujuk})}{1+\exp(-r_{uiuj;Nujuk})} = -\frac{1}{1+\exp\,(r_{uiuj;Nujuk})}$. The steps of PBPR is summarized in Algorithm 1 in detail.

5 Experiments

In this section, we demonstrate the effectiveness of our proposed recommendation method by a series of experiments on three real-world datasets. We compare the performance of the PBPR model with several benchmark methods in terms of precision (Prec), recall (Rec), and normalized discounted cumulative gain (NDCG). Then we evaluate the impacts of different parameter settings on the performance of our proposed recommendation method.

5.1 Datasets

Three real-world datasets are employed as experimental data, including Last-FM[1], MovieLens 100K[2], and MovieLens 1M[2]. Last-FM provides a dataset that users' behaviors of listening to music, which has 92,834 interactions from 1,892 users and 17,632 items. MovieLens is a collection of movie ratings. MovieLens 100K includes 100,000 ratings given by 943 users and 1,682 movies, and MovieLens 1M contains 951,612 interactions assigned by 6,040 users and 3,952 movies. For studying the one-class feedback problem on datasets MovieLens 100K and MovieLens 1M, we do not preprocess datasets according to their rating values scale 1-5, such as keeping the ratings larger than 3 as the observed feedback. We consider all observed user-item pairs as positive feedbacks in all experiments [29]. The description of the experimental datasets is presented in Table 1.

For all three datasets, we randomly sample 20%, 50%, and 80% of user-item interactions on each dataset as training data, respectively, and the rest as test data.

Table 1 Description of the experimental datasets

Dataset	#Users	#Items	#Interactions	Sparsity
Last-FM	1,892	17,632	92,834	99.72%
MovieLens 100K	943	1,682	100,000	93.70%
MovieLens 1M	6,040	3,952	951,612	96.01%

5.2 Experimental design

To evaluate the performance of the PBPR model, we compare our proposed method with benchmark methods by convincing evaluation metrics. In addition, we empirically investigate the effects of different parameter settings in PBPR on the recommendation results.

5.2.1 Benchmark methods

We compare our proposed method with several recommendation methods, including

- PopRank [20]: PopRank recommends each user a ranking list of items according to the item's popularity in the training data. It is not a personalized recommendation approach usually used for solving the user cold-start problem.
- U & I: U & I represents the user-based and item-based recommendation method, which we propose in Section 4.2. It can also be used to predict users' preferences via aggregation of the item-item similarity and the like-minded users' preferences.
- BPR [2]: As introduced in Section 3.3, the Bayesian Personalized Ranking method is a state-of-the-art [20] pairwise learning method based on matrix factorization. It recommends a personalized ranking list of items for each user only based on the user's historical interactions.
- GBPR [3]: Group preference-based Bayesian Personalized Ranking method is an extension of the BPR model, which relaxes individual and independence assumptions in the BPR model. This work is able to accommodate richer interactions among users.

We define the following two prediction functions for PBPR:

$$r_{ui} = U_u \cdot V_i^T \tag{18}$$

$$r_{ui} = U_u \cdot V_i^T + \alpha \cdot S_{ui} \tag{19}$$

where α is a trade-off parameter, and S represents the potential preference score matrix. We denote the recommendation methods, in which the prediction functions are Eq. (18) and Eq. (19), as PBPR and PBPR$^+$, respectively.

[1] https://grouplens.org/datasets/hetrec-2011/.

[2] https://grouplens.org/datasets/movielens/.

5.2.2 Experimental setting

We set the iteration number T=10,000. To make the comparison fair, we initialize all pairwise ranking models (BPR, GBPR and PBPR) with the same random distribution. We set the same dimensionality, the same learning rate, and the same hyper-parameter of the regularization terms. For all recommendation models, the dimensionality is empirically set to $k = 20$, the learning rate is set to $\eta = 0.05$, and the hyper-parameter is set to $\beta = 0.01$.

For the GBPR model, we fix the user group size as $|G| = 3$, the parameter ρ=1 and ρ=0.6 are set for the datasets MovieLens 100K and MovieLens 1M, respectively, considering the settings in [27]. The parameter ρ=0.8 is set for the dataset Last-FM. The control coefficient $\lambda = 0.7$ by default in PBPR and PBPR$^+$, the control coefficient for the dataset Last-FM is set as $\lambda = 0.8$. The trade-off parameter $\lambda_1 = 0.9$ is set for the datasets MovieLens 100K and MovieLens 1M, and $\lambda_1 = 0.3$ is set for the dataset Last-FM. The trade-off parameter α in the prediction function is set as $\alpha = 0.7$ and $\alpha = 0.1$ for the Last-FM dataset and other datasets. We could observe that parameter settings in PBPR are the same for the datasets MovieLens 100K and MovieLens 1M. The reason may be that two datasets are all collected from the MovieLens web site.

Different dimensionalities have influences on the performance of the models. We will change the dimensionality to study the effects of different dimensionalities on recommendation results. In the PBPR$^+$ model, α is a trade-off parameter in the prediction function, and we will discuss its influence on the recommendation results in Section 5.4.

5.2.3 Evaluation metrics

We report the average performance of users in the test data. Users pay attention to a few top-ranked items, so we use top-N evaluation metrics [34], which are Prec@N, Rec@N, and NDCG@N, to evaluate the performance of our method.

The metrics Prec@N and Rec@N for user u are defined as follows:

$$Prec@N = \frac{1}{N}\sum_{i=1}^{N}\delta(I_i \in I_u^+) \tag{20}$$

$$Rec@N = \frac{1}{|U_u^{test}|}\sum_{i=1}^{N}\delta(I_i \in I_u^+) \tag{21}$$

where I_i represents the i-th item in the ranking list. $\delta(\cdot)$ represents the indicator function. Its value is equal to 1 when $I_i \in I_u^+$, otherwise 0. $|U_u^{test}|$ represents the number of all observed items of user u in the test data.

The normalized discounted cumulative gain (NDCG) [35], which takes into account the position of correctly recommended items, is a standard measure of ranking quality. It is defined as

$$NDCG@N = \frac{1}{Z_N}\sum_{j}^{N}\frac{2^{r(j)}-1}{\log(j+1)} \tag{22}$$

where j represents the j-th position in the ranking list, and r(j) presents the relevance of the item in position j. Z_N represents the ideal value for the discounted cumulative gain (DCG, $\mathrm{DCG} = \sum_j^N \frac{2^{r(j)}-1}{\log(j+1)}$). Here, we set N as 5.

5.3 Performance comparison results and analysis

Our experiments are performed on a Windows 10 with 3.6GHz Intel Core i3, 8 GB. The results of all recommendation methods on the datasets Last-FM, MovieLens 100K and MovieLens 1M are presented in Table 2, Table 3, Table 4, respectively. The best values for each dataset are in bold. Several observations stand out:

- PBPR$^+$ performs the best on all the three datasets according to evaluation metrics in Section 5.2.3. The results of PBPR$^+$ outperform PBPR on all datasets, which shows the effect of the integration strategy in Eq. (19). PBPR$^+$ and PBPR further improve BPR on all top-5 metrics on all datasets, which demonstrates the effectiveness of injected pairwise preference between unobserved items into the traditional pairwise ranking model.
- Note that the sparsity of the user-item interaction matrix has an influence on the recommendation results. The results of the top-5 metrics are much lower for the dataset Last-FM than other datasets because the sparsity of its user-item interaction matrix is significantly larger than MovieLens 100K and MovieLens 1M. The overall results of the dataset MovieLens 100K are slightly better than MovieLens 1M.
- Compare with the training data with 20% user-item pairs, the results of the training data with 50% user-item pairs increase in terms of Prec@5 and NDCG@5 because more user-item pairs could provide more information about users' preferences. However, the results of Prec@5 and NDCG@5 decrease when the training data includes 80% user-item pairs. We consider reasons may be that the observed items of some users might be smaller than 5 because we do not preprocess the original datasets. For example, the best precision is 0.2 when only one item of user u in the test data. In addition, it is difficult to find this item from a large number of unobserved items for user u.
- In Eq. (21), $|U_u^{test}|$ represents the number of all observed items of the user u in the test data. The results are very poor in Table 2, Table 3 and Table 4 in terms of Rec@5. The reason may be that many users' observed items in the test data are larger than 5. For example, user u has 24 user-item pairs in the test data; the best recall is 0.208 when the top-5 items are all user u's observed items. Therefore, the results of Rec@5 improve with the increase of the training user-item pairs on all three datasets, i.e., the decrease of the number of user-item pairs in the test data.
- The gaps between U ＆ I method and recommendation models (BPR, GBPR and PBPR) are reduced with the increase of the training data, especially on the dataset Last-FM (80%), the U ＆ I method performs better than almost all recommendation models (BPR, GBPR and PBPR) on all evaluation metrics. It is obvious that the U ＆ I method could provide well results when the training user-item pairs are enough. However, data with high density is difficult to obtain in real-life applications. The results of the U ＆ I method could be integrated with the recommendation model to enhance its performance and provide more accurate recommendations.
- PopRank performs better than the U ＆ I method and BPR on MovieLens 100K (20%). It demonstrates the effectiveness of recommending popular items to users to solve the user cold-

start problem. In this study, we take this idea into the sampling strategy of our proposed model, which will be discussed in Section 5.5.1.

Table 2: Performance comparison for PopRank, U & I, BPR, GBPR, PBPR and PBPR⁺ in terms of Prec@5, Rec@5 and NDCG@5 on Last-FM.

Method/Dataset	Last-FM (20%)			Last-FM (50%)			Last-FM (80%)		
	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5
PopRank	0.1588	0.0406	0.1314	0.1434	0.0583	0.1538	0.0722	0.0735	0.0987
U & I	0.1751	0.0220	0.1794	0.2740	0.0558	0.2870	0.2163	0.1112	0.2379
BPR	0.2874	0.0365	0.2947	0.3346	0.0680	0.3500	0.1932	0.0986	0.2108
GBPR	0.3131	0.0398	0.3200	0.3425	0.0703	0.3543	0.1997	0.1019	0.2221
PBPR	0.3226	0.0411	0.3327	0.3653	0.0745	0.3852	0.2083	0.1065	0.2267
PBPR⁺	**0.3317**	**0.0423**	**0.3416**	**0.3720**	**0.0758**	**0.3957**	**0.2185**	**0.1116**	**0.2406**

Table 3: Performance comparison for PopRank, U & I, BPR, GBPR, PBPR and PBPR⁺ in terms of Prec@5, Rec@5 and NDCG@5 on MovieLens 100K.

Method/Dataset	MovieLens 100K (20%)			MovieLens 100K (50%)			MovieLens 100K (80%)		
	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5
PopRank	0.4838	0.0416	0.4960	0.3862	0.0573	0.4009	0.2180	0.0768	0.2303
U & I	0.4592	0.0403	0.4729	0.5007	0.0781	0.5201	0.3423	0.1251	0.3638
BPR	0.4802	0.0465	0.4868	0.5262	0.0804	0.5388	0.3665	0.1274	0.3840
GBPR	0.5393	0.0538	0.5462	0.5790	0.0925	0.5967	0.3871	0.1361	0.4105
PBPR	0.5642	0.0569	0.5758	0.5922	0.0931	0.6098	0.3981	0.1389	0.4239
PBPR⁺	**0.5684**	**0.0574**	**0.5788**	**0.5930**	**0.0932**	**0.6117**	**0.3996**	**0.1394**	**0.4249**

Table 4: Performance comparison for PopRank, U & I, BPR, GBPR, PBPR and PBPR⁺ in terms of Prec@5, Rec@5 and NDCG@5 on MovieLens 1M.

Method/Dataset	MovieLens 1M (20%)			MovieLens 1M (50%)			MovieLens 1M (80%)		
	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5	Prec@5	Rec@5	NDCG@5
PopRank	0.3611	0.0217	0.3362	0.3301	0.0306	0.3454	0.2110	0.0405	0.2209
U & I	0.3917	0.0255	0.3861	0.4294	0.0495	0.4379	0.3274	0.0865	0.3471
BPR	0.5145	0.0357	0.5213	0.5092	0.0556	0.5177	0.3510	0.0813	0.3652
GBPR	0.5497	0.0391	0.5552	0.5307	0.0612	0.5424	0.3477	0.0847	0.3638
PBPR	0.5550	0.0391	0.5575	0.5527	0.0629	0.5653	0.3664	0.0874	0.3833
PBPR⁺	**0.5554**	**0.0391**	**0.5580**	**0.5541**	**0.0631**	**0.5667**	**0.3672**	**0.0874**	**0.3842**

5.4 *Investigation of parameters*

The dimensionality is an important parameter for pairwise ranking models. Small dimensionalities will not represent users and items well, and large dimensionalities will spend more time training the model. We investigate the performance of different models with different dimensionalities ($k = 10, 20, 30, 40, 50$) on Last-FM (50%), MovieLens 100K (50%) and MovieLens 1M (50%). Other parameter settings of models are the same as in Section 5.3. The results are shown in Fig. 4. We can observe that all evaluation metrics are improved with the increase of the dimensionality. The experimental results of all models are best on all datasets when the dimensionality k=50. From Fig. 4, we can also see that PBPR-based models achieve high performance in contrast to the BPR model and the GBPR model in terms of Prec@5, Rec@5 and NDCG@5.

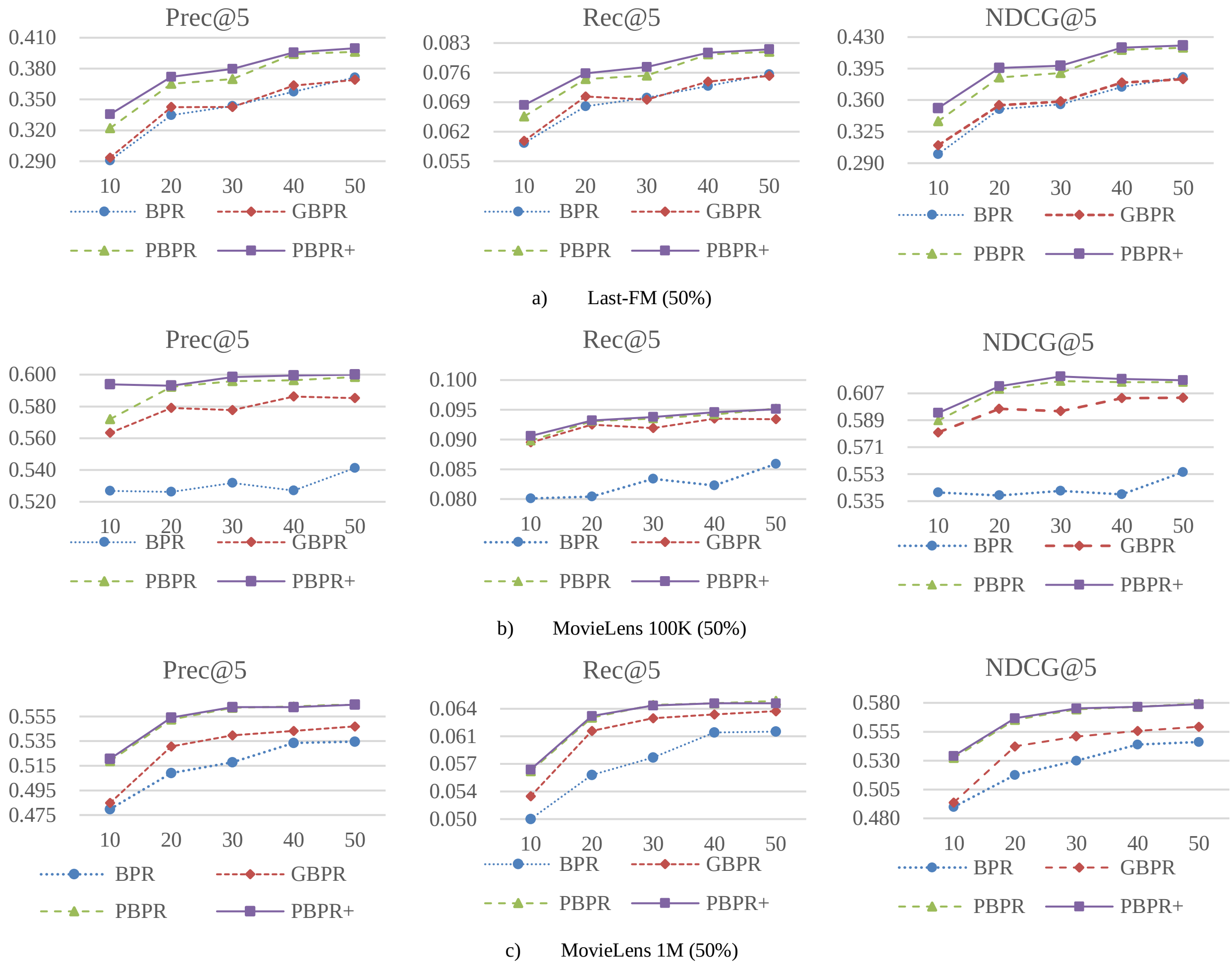

Fig. 4: Performance investigation different dimensionalities on Last-FM (50%), MovieLens 100K (50%) and MovieLens 1M (50%).

The User-based and Item-based (U & I) recommendation method could achieve great results when 80% of the user-item interactions are chosen as training data. Therefore, we discuss the impact of the trade-off parameter α on three datasets Last-FM (80%), MovieLens 100K (80%) and MovieLens 1M (80%)), on which the performance of potential preference scores is relatively good.

We change the trade-off parameter α with different values of {0.1, 0.2, …, 0.9, 1}. The experimental results of the PBPR model and the $PBPR^+$ model in terms of Prec@5, Rec@5 and NDCG@5 are shown in Fig. 5. Fig. 5 shows that the results of the $PBPR^+$ model increase on all evaluation metrics when α varies from 0.1 to 1 on the dataset Last-FM (80%). In Fig. 5 b), the $PBPR^+$ model achieves high performance in term of Rec@5 when the parameter α set on some specific values. The results of $PBPR^+$ decrease with the increase of α in term of Rec@5 on the dataset MovieLen 1M (80%). The performance of the $PBPR^+$ model improves when the parameter α varies from 0.1 to 0.3 and then decreases with the increase of α, in terms of Prec@5 and NDCG@5 on the dataset MovieLens 100K (80%). $PBPR^+$ performs better than PBPR when the parameter α is smaller than 0.8, in terms of Prec@5 and NDCG@5 on the dataset MovieLens 1M (80%).

We observe that prior information calculated based on user-item interactions has a significant impact on the results of the PBPR-based models. We consider that more accurate potential preference scores and the suitable trade-off parameter could make the $PBPR^+$ model provide more accurate recommendations. In this paper, our proposed PBPR a generic recommendation approach. It could be improved by considering more auxiliary information for specific recommended scenarios like calculating the potential preference scores between users and microblogs considering the topic information [36] and emotions [37] of microblogs while recommending microblogs for users.

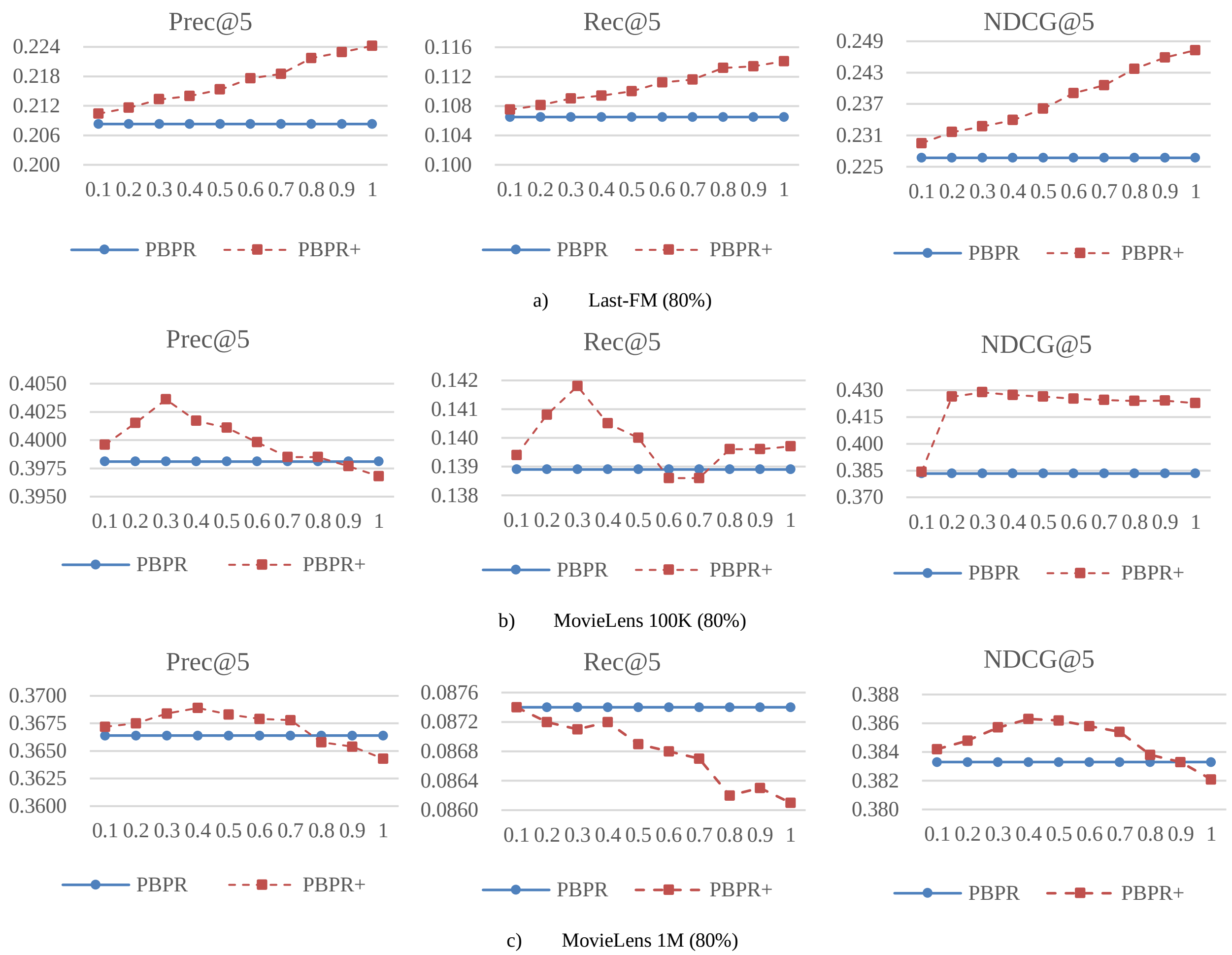

Fig. 5: Performance investigation of trade-off parameters on Last-FM (80%), MovieLens 100K (80%) and MovieLens 1M (80%).

5.5 Discussion

5.5.1 More details in the sampling process

In Algorithm 1, we randomly sample two unobserved items j and k, with the condition that S_{uj} is larger than S_{uk} in step 6, where the matrix S ($S \in \mathbb{R}^{M\times N}$) records potential preference scores between users and items. In this study, the sparsity of the dataset Last-FM is 99.72%. The sparsity of the dataset Last-FM (20%) reaches 99.94%, which may cause like-minded users can not be found, or values of user-based item similarity are zero. Existing users that their potential preference scores on unobserved items are all zero.

Recommending popular items for users is a solution for solving the user cold-start problem. Inspired by this, we assume that users may like items that are popular among other users. During the random sampling process on the dataset Last-FM, we randomly sampled a popular item from the user's unobserved items as the unobserved item j of user u when the potential preference scores between the user u and unobserved items are all zero.

5.5.2 Explainability of prior information

Explainability becomes critically important for recommender systems to provide convincing results [33]. Explainable recommender systems [38] aim to reveal why a user might like the item, and it helps improve users' satisfaction or acceptance of recommendation results [39]. The explanations for recommender systems are usually based on phrase sentiment [40], aspect [41], and social networks [42].

Since we adopt the user-based and item-based collaborative filtering method to calculate the potential preference scores between users and items, it provides an explainable reason for why the user might have an interest in the items with higher potential preference scores. For instance, recommending item 1 to user 2 might be explained by “$item\ 1 \xrightarrow{\text{watched}} user\ 1 \xrightarrow{\text{like}-\text{minded}} user\ 2$”. Therefore, in PBPR, the prior information provides an explainable reason for why the user u might prefer the unobserved item j over the unobserved item k. Additionally, the preference score matrix S, which considering the user's potential preference score on items, provides more potential interaction information between users and items to relax the sparsity issues from the user-item interaction matrix.

6 Conclusions

In this paper, we propose a novel model called PBPR for solving the one-class problem. PBPR relaxes the assumption in BPR with equal importance of the unobserved items, further considering the pairwise preference between two unobserved items. Our proposed recommendation method merely needs positive feedbacks without any additional information. It is a generic recommendation approach and could be improved by considering the unique social information about the specific application. Moreover, we measure the preference difference between any two unobserved items based on prior information rather than dividing unobserved items into different groups and studying the preference difference between unobserved items from different groups. We calculate potential preference scores as prior information based on users' historical interactions, which can be used to (a) measure fine-grained preference difference between any two unobserved items; (b) relax the sparsity of user-item interactions by providing more potential interactions be-tween users and items; (c) integrate with the individual preference for the more accurate recommendations. Experimental results demonstrate the effectiveness of our method on real-world datasets.

For future work, we are interested in (a) considering side information from other domains (affective computing, data mining). Emotion expressions [43], which are opinion targets with users' emotions, can be extracted from microblogging sites; and provide more information about user-item interactions for the recommender systems. Moreover, we can use sentiment analysis techniques [44] to analyze why users like items (user *A* prefers action movies) or study users' preferences by considering the auxiliary feedback via the hybrid emotion recognition system [45]; this helps to construct explainable recommender systems; (b) exploring the multi-task learning tasks, which conduct the calculation of potential preference scores and recommendation model learning together.

Acknowledgments

This work was partially supported by the Research Clusters program of Tokushima University (No. 2003002).

References

[1] X. Wang, X. He, Y. Cao, M. Liu, T, Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950-958.

[2] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T. Chua, Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, 2017, pp. 335-344.

[3] Zhang Y, Shi Z, Zuo W, et al. Joint Personalized Markov Chains with social network embedding for cold-start recommendation[J]. Neurocomputing, 2020, 386: 208-220.

[4] F. Zhang, N. J. Yuan, D. Lian, X. Xie, W. Ma, Collaborative knowledge base embedding for recommender system, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016, pp. 353-362.

[5] Y. Hu, F. Xiong, D. Lu, H. Chen, Movie collaborative filtering with multiplex implicit feedbacks, Neurocomputing 398 (2020) 485-494.

[6] Q. Zhang, F. Ren, Potential Preference Based Bayesian Personalized Ranking for One-Class Collaborative Filtering, in: Proceedings of the fifth International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'20), 2020, pp. 11.

[7] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 502–511.

[8] D. Nichols, Implicit rating and filtering, in: Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering, 1998, pp. 31-36.

[9] Z. Cheng, X. Chang, L. Zhu, R. Kanjirathinkal, M. Kankanhalli, MMALFM: Explainable recommendation by leveraging reviews and images, ACM Transactions on Information Systems (TOIS), 2019, 37(2): 16.

[10] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 263–272.

[11] Y. Guo, Z. Cheng, L. Nie, Y. Wang, J. Ma, M. Kankanhalli, Attentive long short-term preference modeling for personalized product search, ACM Transactions on Information Systems (TOIS) 37(2) (2019) 19.

[12] W. Cai, J. Zheng, W. Pan, J. Lin, L. Li, L. Chen, X. Peng, Z. Ming, Neighborhood-enhanced transfer learning for one-class collaborative filtering, Neurocomputing 341 (2019) 80-87.

[13] W. Pan, M. Liu, Z. Ming, Transfer learning for heterogeneous one-class collaborative filtering, IEEE Intelligent Systems 31(4) (2016) 43-49.

[14] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: *UAI*, 2009, pp. 452-461.

[15] S. Rendle, C. Freudenthaler, Improving pairwise learning for item recommendation from implicit feedback, in: Proceedings of the 7th ACM international conference on Web search and data mining, 2014, pp. 273-282.

[16] S. Yang, B. Long, A. Smola, H. Zha, Z. Zheng, Collaborative competitive filtering: learning recommender using context of user choice, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, pp. 295–304.

[17] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, 2010, pp. 81–90.

[18] Y. Li, J. Hu, C. Zhai, Y. Chen, Improving one-class collaborative filtering by incorporating rich user information, in: Proceedings of the 19th ACM international conference on Information and knowledge management. 2010, pp. 959-968.

[19] J. Chen, D. Lian, K. Zheng, Improving One-Class Collaborative Filtering via Ranking-Based Implicit Regularizer, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 37-44.

[20] X. Zhou, G. Guo, Z. Sun, Y. Liu, Multi-facet user preference learning for fine-grained item recommendation, Neurocomputing 385 (2020) 258-268.

[21] G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma, X. Wang, Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems, Knowledge-Based Systems 138 (2017) 202-207.

[22] L. Yu, C. Zhang, S. Liang, X. Zhang, Multi-order attentive ranking model for sequential recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 5709-5716.

[23] X. Xin, X. He, Y. Zhang, Y. Zhang, J. Jose, Relational collaborative filtering: Modeling multiple item relations for recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 125-134.

[24] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, L. Schmidt-Thieme, Multi-relational matrix factorization using bayesian personalized ranking for social network data, in: Proceedings of the fifth ACM international conference on Web search and data mining, 2012, pp. 173-182.

[25] L. Guo, H. Jiang, X. Wang, F. Liu, Learning to recommend point-of-interest with the weighted Bayesian personalized ranking method in LBSNs, Information 8(1) (2017) 20.

[26] R He, J McAuley, VBPR: visual bayesian personalized ranking from implicit feedback, in: Proceedings of the thirtieth AAAI conference on artificial intelligence, 2016, pp. 144-150.

[27] W. Pan, L. Chen, GBPR: group preference based Bayesian personalized ranking for one-class collaborative filtering, in: Proceedings of the twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 2691-2697.

[28] R. Yu, Y. Zhang, Y. Ye, L. Wu, C. Wu, Q. Liu, E. Chen, Multiple pairwise ranking with implicit feedback, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1727-1730.

[29] W. Zhou, J. Li, Y. Zhou, M H. Memon, Bayesian pairwise learning to rank via one-class collaborative filtering, Neurocomputing 367 (2019) 176-187.

[30] X. Ning, C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, Recommender systems handbook, Springer, Boston MA (2015) 37-76.

[31] Y. Koren, R, Bell, Advances in collaborative filtering, Recommender systems handbook, Springer, Boston MA (2015) 77-118.

[32] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Proceedings of the twenty-second Advances in neural information processing systems. 2008, pp. 1257-1264.

[33] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Jointly learning explainable rules for recommendation with knowledge graph, in: Proceedings of the nineteenth World Wide Web Conference. 2019, pp. 1210-1221.

[34] G. Li, Z. Zhang, L. Wang, Q. Chen, J. Pan, One-class collaborative filtering based on rating prediction and ranking prediction, Knowledge-Based Systems 124 (2017) 46-54.

[35] S. Balakrishnan, S. Chopra, Collaborative ranking, in: Proceedings of the fifth ACM international conference on Web search and data mining, 2012, pp. 143-152.

[36] Q. Zhang, S. Zhang, F. Ren, X. Zhang, Aspect-Based Opinion and Affective Meaning in Microblogging Comments Via Topic Model, JOURNAL OF CHINESE INFORMATION PROCESSING 33(6) (2019) 116-123,140.

[37] J. Deng, F. Ren, Multi-label Emotion Detection via Emotion-Specified Feature Extraction and Emotion Correlation Learning, IEEE Transactions on Affective Computing, DOI 10.1109/TAFFC.2020.3034215 (2020).

[38] N. Tintarev, J. Masthoff, A survey of explanations in recommender systems in: Proceedings of the IEEE 23rd international conference on data engineering workshop, 2007, pp. 801-810.

[39] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 417-426.

[40] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 2014, pp. 83-92.

[41] K. Bauman, B. Liu, A. Tuzhilin, Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 717-725.

[42] B. Wang, M. Ester, J. Bu, D. Cai, Who also likes it? generating the most persuasive social explanations in recommender systems, in: AAAI, 2014, pp 173-179.

[43] F. Ren, Q. Zhang, An emotion expression extraction method for Chinese microblog sentences, IEEE Access 8(1) (2020) 69244-69255.

[44] F. Ren, Affective Information Processing and Recognizing Human Emotion, Electronic Notes in Theoretical Computer Science 225(2009) (2009) 39-50.

[45] Y. Gu, X. Zhang, Z. Liu, F. Ren, WiFE: WiFi and Vision based Intelligent Facial-Gesture Emotion Recognition, arXiv preprint arXiv:2004.09889 (2020).