Personalized Ranking Recommendation via Integrating Multiple Feedbacks

Jian Liu[1], Chuan Shi[4,1](✉), Binbin Hu[1] and Shenghua Liu[2], Philip S. Yu[3]

[1]Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications
[2] Institute of Computing Technology, Chinese Academy of Sciences
[3] University of Illinois at Chicago
[4] Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University
fullback@yeah.net,shichuan@bupt.edu.cn,hubinbin@bupt.edu.cn
liushenghua@ict.ac.cn,psyu@uic.edu

Abstract. Recently, recommender system has attracted a lot of attentions, which helps users to find items of interest through utilizing the user-item interaction information and/or content information associated with users and items. The interaction information (i.e., feedback) between users and items are widely exploited to build recommendation models. The feedback data in recommender systems usually comes in the form of both explicit feedback (e.g., rating) and implicit feedback (e.g., browsing histories, click logs). Although existing works have begun to utilize either explicit or implicit feedback for better recommendation, they did not make best use of these feedback information together. In this paper, we first study the personalized ranking recommendation problem by integrating multiple feedbacks, i.e., one type of explicit feedback and multiple types of implicit feedbacks. Then we propose a unified and flexible personalized ranking framework MFPR to integrate multiple feedbacks. Moreover, as there are no readily available training data, an explicit feedback based training data generation algorithm is designed to generate item pairs with more accurate partial order consistent with the multiple feedbacks for the proposed ranking model. Extensive experiments on two real-world datasets validate the effectiveness of the MFPR model, and the integration of multiple feedbacks making up better complementary information significantly improves recommendation performance.

Keywords: Recommender system, multiple feedbacks, explicit feedback, implicit feedback, Bayesian Personalized Ranking

1 Introduction

In recent years, recommender systems have attracted much attention from multiple disciplines. The interaction information (i.e., feedback) between users and items are widely exploited to build recommendation models. The feedback data in recommender systems usually comes in the form of explicit or implicit feedback [4]. Explicit feedback is the interaction information that directly expresses user preferences to items, such as the rating information of users to items. While implicit feedback indirectly reflects user opinions and can imply user probable preferences [9], such as the “collect” and “share” of users to items. Fig. 1 shows a toy example of multiple feedbacks in Douban Book. The rating (1-5 scales) is the explicit feedback and there are two types of implicit feedbacks. Thereinto, the “wish” means the user wishes to read the book but has not begun yet; the “reading” means the user is currently in reading process. It is obvious that the explicit feedback (i.e., rating) is critical for recommendation, while the implicit feedbacks also provide important supplementary information.

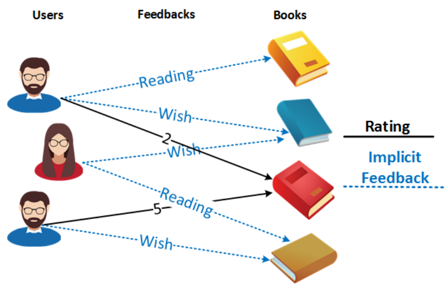


Fig. 1. A toy example of multiple feedback between users and books in Douban Book

Many methods exploit the feedbacks to build recommender systems. Fig. 2 shows how those methods utilize these information. Traditional collaborative filtering usually utilizes explicit feedback information (i.e., ratings) [5, 7, 14](see Fig. 2 (a)). Since implicit feedback information is widely and cheaply available, researchers began to exploit the implicit feedback. Some works considered to use one single type of implicit feedback [6, 10, 13] (see Fig. 2(b)), and Costa Fortes et al. [2] combined several types of implicit feedbacks using a simple ensemble approach not long ago (see Fig. 2(c)). In addition, SVD++ [7] was designed to combine rating information with a single type of implicit feedback for more accurate rating prediction, as shown in Fig. 2(d). Unfortunately, all these works have not utilized comprehensive feedback information in recommender systems. In

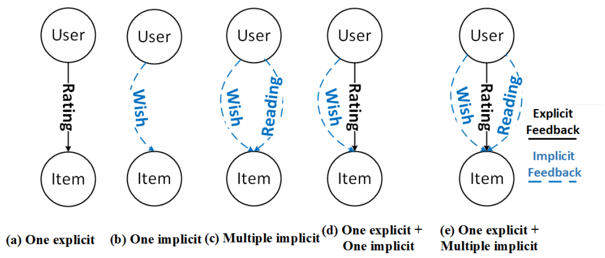


Fig. 2. The schemas of utilizing feedback information

this paper, we propose to solve the personalized ranking problem by integrating multiple feedbacks, as shown in Fig. 2(e). For convenience, multiple feedbacks mean one type of explicit feedback and multiple types of implicit feedbacks in the following sections. In many review web sites, such as Yelp and Dianping, users are required to give a rating score (i.e., explicit feedback) to a business, and they can also have other interactions with businesses, such as "checking in" and "viewing". Obviously, our problem setting is a general framework to utilize feedback information, and existing problems are special cases of our problem setting. In addition, from recommendation perspective, the predicted ranking over an item is much more meaningful than the predicted rating. Thus in this work, we focus on developing a personalized ranking model that integrates multiple feedbacks. Although many methods have been proposed to utilize the feedbacks, these models are usually designed for special problem settings, and they cannot be directly applied in multiple-feedback setting.

However, integration of multiple feedbacks faces two challenges. (1) Design a unified ranking model integrating multiple feedbacks. In order to make the best use of these feedback information, we need to design an effective mechanism to handle relations between explicit and implicit feedbacks as well as relations among implicit feedbacks. (2) Generate training samples. As a ranking method, we need to generate preference pairs or lists for training. However, there are multiple types of feedbacks. What kind of feedbacks could we utilize for better preference pair or sequence?

The major contributions of our paper are summarized as follows: (1) We first try to solve the personalized ranking recommendation problem by integrating multiple feedbacks. The problem widely exists in real recommender system, and it is a general problem setting to encompass existing works. (2) We propose a Bayesian Personalized Ranking (BPR) based model MFPR to integrate multiple feedbacks. Moreover, as there are no readily available training data for this problem, an effective algorithm is designed to generate the training data that is more consistent with multiple feedbacks for the MFPR model. (3) We crawl comprehensive Douban Book and Dianping datasets [1] including ratings and multiple types of implicit feedbacks.

2 Preliminary

2.1 Explicit & Implicit Feedback and Problem Formulation

Formally, when the data is in the form of explicit feedback with single implicit feedback, each user u is associated with two types of item sets: implicit item set $N(u)$ and explicit feedback set $E(u)$. *Explicit feedback* is intentionally provided by users to directly express user preferences (e.g., likes or dislikes) to items. For an item $i \in E(u)$, the rating given by user u to item i is denoted as R_{ui}. *Implicit feedback* reflects user opinions indirectly and can imply user probable preferences [9]. For an item $i \in N(u)$, the implicit feedback does not necessarily mean that user u likes the item i.

When data consists of explicit feedback with multiple types of implicit feedbacks, each user is associated with single explicit feedback and τ types of implicit feedbacks ($\tau \geq 2$). For user u, the explicit item set is still denoted as $E(u)$ which contains items user u has rated (i.e., rating) on, and the implicit item sets are denoted as $N^1(u), N^2(u), \cdots, N^\tau(u)$ where $N^t(u)$ contains items user u has expressed the t-type implicit feedback on ($t = 1, \cdots, \tau$).

[1] The datasets are available at https://github.com/7thsword/MFPR-Datasets.

Let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items respectively. We define a ranking recommendation problem on multiple feedback data $R_d = \{\mathcal{U}, \mathcal{I}, E_f, I_f\}$. E_f, defined as $E_f = \{E(u)|u \in \mathcal{U}\}$, denotes the explicit feedback data consisting of all users' explicit item sets. I_f, defined as $I_f = \{N^t(u)|u \in \mathcal{U}, t = 1, \cdots, \tau\}$, denotes the implicit feedback data consisting of all users' implicit item sets. Hence, as shown in Fig. 2(e), our task is to design a model for better personalized ranking recommendation through making full use of the explicit feedback data E_f and the implicit feedback data I_f.

2.2 Base Learner Integrating Explicit and Implicit Feedback

The explicit feedback (i.e. rating) is very important for recommendation but rare, and the implicit feedback is popular in real systems. Some researchers began to consider the integration of explicit and implicit feedback for more accurate rating prediction. Assume that there are m users and n items (i.e., $|\mathcal{U}| = m$, $|\mathcal{I}| = n$). Given a rating matrix $R = (R_{ui})^{m \times n}$, where R_{ui} denotes the score user u has rated on item i. The predicted rating $\hat{R}_{ui}$ user u may give to item i in SVD++ [7] can be modeled as:

$$\hat{R}_{ui} = (p_u + |N(u)|^{-\frac{1}{2}} \sum_{k \in N(u)} \gamma_k) q_i^{\mathrm{T}}, \tag{1}$$

where $p_u \in \mathbb{R}^d$ is the explicit latent vector of user u, $q_i \in \mathbb{R}^d$ is the explicit latent vector of item i and $d \ll min(m, n)$. $\gamma_k \in \mathbb{R}^d$ is the implicit latent vector of item k and $N(u)$ is the implicit item set as mentioned above. Here a user u is modeled as $p_u + |N(u)|^{-\frac{1}{2}} \sum_{k \in N(u)} \gamma_k$, and the complemented sum term $|N(u)|^{-\frac{1}{2}} \sum_{k \in N(u)} \gamma_k$ represents the perspective of implicit feedback. SVD++ treats the explicit and implicit feedback differently. It makes best use of explicit feedback and adds implicit feedback as supplements.

Unfortunately, these existing models cannot be directly applied to our problem setting. Although SVD++ also considers explicit and implicit feedbacks, it just integrates one type of implicit feedback. In addition, SVD++ is originally designed for the rating prediction problem. Since predicting exact ratings is not necessary for recommendation, we propose to use ranking framework.

3 Personalized Ranking with Multiple Feedbacks

The explicit and implicit feedbacks have different characteristics, we need to treat them differently. Through adapting the Bayesian Personalized Ranking framework [13], we first design a Personalized Ranking model which integrates explicit and one Single implicit Feedbacks (called SFPR). Then we extend the SFPR model to integrate more implicit feedbacks and propose a unified Multiple Feedbacks based Personalized Ranking model (called MFPR).

3.1 The SFPR Model

Firstly, we design a ranking model to combine explicit feedback and one type of implicit feedback. Assume that a training set $\mathcal{T}_r$ consists of triples of the form (u, i, j) with $i \succ j$ denoting that user u prefers item i to item j.Note that the generation of training set $\mathcal{T}_r$ is an important issue and it will be discussed in

Sec. 4. The Bayesian formulation of finding the correct personalized ranking is to maximize the following posterior probability:

$$p(\theta|\mathcal{T}_r) \propto p(\mathcal{T}_r|\theta)p(\theta), \tag{2}$$

where θ is the parameter of a certain base learner and $p(\theta)$ is the prior probability of base leaner parameter. We use $p(i \succ j; u|\theta)$ to denote the probability that user u prefers item i over item j under the model expressed by θ. With the assumption that each triple $(u, i, j) \in \mathcal{T}_r$ is independent, the likelihood function can be expanded as follows:

$$p(\mathcal{T}_r|\theta) = \prod_{(u,i,j)\in\mathcal{T}_r} p(i \succ j; u|\theta). \tag{3}$$

Since the SVD++ can effectively differentiates explicit and implicit feedback and fully utilize the explicit feedback, we utilize the SVD++ as our base learner. Then the individual probability $p(i \succ j; u|\theta)$ can be modeled as:

$$p(i \succ j; u|\theta) = \sigma(\hat{R}_{ui} - \hat{R}_{uj}), \tag{4}$$

where σ is the logistic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

For convenience, we simplify $\hat{R}_{ui} - \hat{R}_{uj}$ in Eq. 4 as $\hat{x}_{uij}$. Note that $\hat{x}_{uij}$ is a real-valued function of θ which captures ranking relation between item i and item j with the given user u. Assume that $p(\theta)$ is a Gaussian distribution with zero mean and variance-covariance matrix $\sum_\theta = \lambda_\theta I$. Now we can estimate parameter θ of the base learner through maximizing the posterior probability in Eq. 2 as follows:

$$\begin{aligned}
\max_\theta \mathcal{L} &= ln\ p(\theta|\mathcal{T}_r) \\
&= ln\ p(\mathcal{T}_r|\theta)p(\theta) \\
&= \sum_{(u,i,j)\in\mathcal{T}_r} ln\ p(i \succ j; u|\theta) - \lambda_\theta \|\theta\|^2 \\
&= \sum_{(u,i,j)\in\mathcal{T}_r} ln\ \sigma(\hat{x}_{uij}) - \lambda_\theta \|\theta\|^2,
\end{aligned} \tag{5}$$

where $\lambda_\theta \|\theta\|^2$ is a L2 regularization term which can be derived from the Gaussian distribution $p(\theta)$ mentioned above.

3.2 Learning SFPR Model

The objective function Eq. 5 is differentiable, gradient ascent based algorithms can be employed as optimizer. The gradient of Eq. 5 with respect to the parameter θ is:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{(u,i,j)\in\mathcal{T}_r} \frac{\partial}{\partial \theta} ln\sigma(\hat{x}_{uij}) - \lambda_\theta \frac{\partial}{\partial \theta}\|\theta\|^2 \\
&\propto \sum_{(u,i,j)\in\mathcal{T}_r} \frac{1}{1+e^{\hat{x}_{uij}}} \frac{\partial}{\partial \theta}\hat{x}_{uij} - \lambda_\theta \theta.
\end{aligned} \tag{6}$$

We adopt stochastic gradient ascent (SGA) to optimize the model SFPR. Then with a training sample (u, i, j), the model parameter θ can be updated as:

$$\theta \leftarrow \theta + \eta(\frac{1}{1+e^{\hat{x}_{uij}}}\frac{\partial}{\partial\theta}\hat{x}_{uij} - \lambda_\theta\theta), \tag{7}$$

where η is the given learning rate and generally tuned via cross validation. The gradient of $\hat{x}_{uij}$ with respect to each model parameter has to be known before gradient ascent process. $\hat{x}_{uij} = \hat{R}_{ui} - \hat{R}_{uj}$ is defined above and we can get the derivatives:

$$\frac{\partial\hat{x}_{uij}}{\partial\theta} = \begin{cases} q_i - q_j & \text{if } \theta = p_u, \\ p_u + |N(u)|^{-\frac{1}{2}}\sum_{k\in N(u)}\gamma_k & \text{if } \theta = q_i, \\ -(p_u + |N(u)|^{-\frac{1}{2}}\sum_{k\in N(u)}\gamma_k) & \text{if } \theta = q_j, \\ |N(u)|^{-\frac{1}{2}}(q_i - q_j) & \text{if } \theta = \gamma_k. \end{cases}$$

The predicted $\hat{R}_{ui}$ in SFPR model cannot be regarded as the usual predicted rating (i.e. 1 to 5 scales). Here, we call $\hat{R}_{ui}$ the predicted ranking score, which implies that degree of user u prefers item i. The larger the ranking score is, the higher preference it implies.

3.3 The MFPR model

The proposed SFPR is designed to integrate single explicit feedback and single implicit feedback. Here we extend the SFPR model to integrate more implicit feedbacks. When considering multiple feedbacks, as mentioned in Sec. 2.1, each user u is associated with an explicit item set $E(u)$ and τ types of implicit item sets $N^1(u), N^2(u), \cdots, N^\tau(u)$. For integrating multiple implicit feedbacks, our extended preference predictor can be designed as

$$\hat{R}_{ui} = (p_u + \frac{1}{\tau}\sum_{t=1}^{\tau}|N^t(u)|^{-\frac{1}{2}}\sum_{k\in N^t(u)}\gamma_k^t)q_i^{\mathrm{T}}, \tag{8}$$

where $\gamma_k^t \in \mathbb{R}^d$ represents the implicit latent vector of item k under the t-th implicit feedback. The model in Eq. 8 can be seen as a more general version of the SFPR model. Now we have the $\hat{x}_{uij} = \hat{R}_{ui} - \hat{R}_{uj}$ as:

$$\hat{x}_{uij} = (p_u + \frac{1}{\tau}\sum_{t=1}^{\tau}|N^t(u)|^{-\frac{1}{2}}\sum_{k\in N^t(u)}\gamma_k^t)(q_i - q_j)^{\mathrm{T}}. \tag{9}$$

Similarly, we apply SGA to solve the optimization problem.

4 Training Set Generation Algorithm

The MFPR model is fed with training data in the form of (u, i, j) with $i \succ j$ denoting that user u prefers item i over item j. Since the preference partial pairs significantly affect performances [1], it is an important issue that how we can effectively generate (u, i, j) from multiple feedbacks. For those traditional personalized ranking models utilizing only one or more types of implicit feedbacks, such as BPR-MF in [13] and the approach in [2], their training set generation algorithms just take implicit feedbacks into account. Specifically, they draw partially ordered item pairs from the cartesian product of user's interacted items (items belong to user's implicit item set) and user's non-interacted items (items do not belong to user's implicit item set). However, in terms of multiple feedbacks, such training set generation algorithm is inapplicable for MFPR. Besides implicit feedbacks, there are quality rating information in our problem setting, which can better reflect user preference. Hence, we need to design a new training data generation algorithm.

Burgess and Shaked et al. [1] have proved that if the ranking probabilities of every adjacent document pair in a permutation of all documents to be ranked are known, then the ranking probabilities of any document pair can be derived. Inspired by this conclusion, we design the training set generation algorithm which utilizes the most significant preference information in the multiple feedbacks: rating information. For each user u, we randomly split his or her explicit item set $E(u)$ into two subsets $E_{tr}(u)$ and $E_{te}(u)$ with the given split ratio, where $E_{tr}(u)$ is designed for constructing training set $\mathcal{T}_r$ and $E_{te}(u)$ is for test set $\mathcal{T}_e$. When constructing $\mathcal{T}_r$, we first get a random permutation of $E_{tr}(u)$. Then, for every adjacent item pair (i, j) in the permutation: (1) if $R_{ui} > R_{uj}$, put the triple (u, i, j) into $\mathcal{T}_r$; (2) if $R_{ui} < R_{uj}$, put the triple (u, j, i) into $\mathcal{T}_r$; (3) if $R_{ui} = R_{uj}$, skip and continue to check next adjacent pair. Through the process for every user, we can get the training set $\mathcal{T}_r$ eventually. And the similar process is done for the test set $\mathcal{T}_e$.

Fig. 3 gives a toy example for user u. We have explicit item set $E_{tr}(u) = \{6, 8, 9, 11, 17\}$ and the corresponding ratings are $R_{u,6} = 4$, $R_{u,8} = 3$, $R_{u,9} = 2$, $R_{u,11} = 5$ and $R_{u,17} = 4$. Assume that a random permutation of E_{tr} is $P_{tr} = \{11, 8, 17, 6, 9\}$, then we in turn check every adjacent item pairs $(11, 8)$, $(8, 17)$, $(17, 6)$, $(6, 9)$ of the permutation. Finally, the triples $(u, 11, 8)$, $(u, 17, 8)$ and $(u, 6, 9)$ are selected and put into the training set $\mathcal{T}_r$.

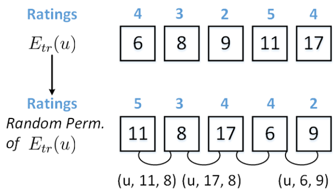


Fig. 3. The toy example of generating training data for user u

We name this algorithm as *IPPE* which means that *I*tem *P*airs with partial order are obtained from checking adjacent items in a *P*ermutation of *E*xplicit item set. The IPPE method considers every adjacent item pair, rather than any item pair. This strategy significantly reduces the size of training samples without much sacrifice in recommendation performance.

5 Experiment

5.1 Datasets

In this paper, we focus on exploiting multiple feedbacks. As far as we know, it is difficult to obtain such public datasets. Hence, we crawled two real-world datasets for the experiments.

The Douban Book dataset [2] contains 190,590 ratings (1-5 scales) involving 12,850 users and 22,040 books. The ratings to books are considered as explicit

[2] http://book.douban.com

Table 1. Statistics of Datasets

Dataset	Type	Feedbacks(A-B)	No. of A	No. of B	No. of (A-B)
Douban Book	explicit	user-rating	12850	22040	190590
	implicit	user-wish	11107	16406	162565
		user-reading	9776	12787	71662
		user-read	12029	20014	174726
		user-tag	8487	19942	162070
		user-comment	8776	18888	151758
		user-rated	12850	22040	190590
Dianping	explicit	user-rating	10549	17707	188813
	implicit	user-good taste	10473	14043	122060
		user-good env.	10293	12135	90350
		user-good service	10354	13271	105846
		user-good overall	10425	14283	125173

feedback. There are 6 types of implicit feedbacks: "wish", "reading", "read", "tag", "comment" and "rated". All these implicit feedbacks are recorded using a binary matrix ("1" for done and "0" for not). Note that the "rated" implicit feedback comes from rating information through degrading the rating matrix into a binary matrix ("1" means "rated" and "0" for "not rated").

The Dianping dataset [3] contains 188,813 ratings (1-5 scales) involving 10,549 users and 17,707 restaurants. There are four types of ratings in Dianping, including overall rating (1-5 scales) and ratings (1-5 scales) on taste, environment and service. We use the overall ratings as explicit feedback and degrade overall, taste, environment and service ratings into "1" if rating ≥ 3 otherwise "0". Then four types of implicit feedbacks are obtained: "good taste", "good environment", "good service" and "good overall". The details can be seen in Table 1.

5.2 Comparison Methods and Evaluation Metrics

We compare the performance of the proposed SFPR and MFPR with five representative methods:

• Most Popular(MP). This baseline ranks items according to their popularity and is non-personalized.

• SVD[7]. This method is a typical matrix factorization based model. It is a rating prediction model and the input data needs only the rating information. We rank items using the predicted ratings in our experiments.

• BPR-MF[13]. This pairwise ranking method was introduced by Rendle et al. and is a state-of-the-art personalized ranking model using only one type of implicit feedback.

•Ensemble of BPRMF (EN-BPRMF)[2]. This method is an ensemble approach to unify different types of implicit feedbacks based on BPR-MF. In experiments, we ensemble all types of implicit feedbacks using this approach.

• SVD++[7]. This method is also a matrix factorization based rating prediction model and the first to integrate rating information with one type of implicit feedback. We rank items using the predicted ratings.

• Factorization Machine (FM)[11]. This method is a general predictor which works with any real valued feature vector and combines the advantages of support vector machines with factorization models. We integrate rating information and all types of implicit feedbacks into the feature vector. It is a rating prediction model and we rank items using the predicted rating.

[3] http://www.dianping.com

Since BPR-MF, SVD++ and SFPR need one type of implicit feedback, we choose the "read" feedback in Douban Book and the "good overall" feedback in Dianping for them. The reason is that the best performance is achieved in these conditions. In addition, some baselines are obtained from open resources. FM is from libFM [12], while MP and BPR-MF are from MyMediaLite [3].

We use two evaluation metrics, which are widely used to evaluate ranking performance. *Zero-One Error* [8] is the average ratio of correctly ordered item pairs of triples (u, i, j) in test set $\mathcal{T}_e$:

$$\varepsilon_{0/1} = \frac{1}{|\mathcal{T}_e|} \sum_{(u,i,j)\in\mathcal{T}_e} [\hat{x}_{uij}(R_{ui} - R_{uj}) > 0], \tag{10}$$

where $\hat{x}_{uij}$ is the difference between predicted ranking score $\hat{R}_{ui}$ and $\hat{R}_{uj}$ as defined above. And $[c]$ denotes a condition indicator that return 1 iff c is true otherwise 0.

$NDCG@k$ [8] is designed to take into count the order of items in the recommendation list. To define $NDCG_u@k$ for a user u, $DCG_u@k = \sum_{i=1}^{k} \frac{2^{R_{ui}}-1}{log_2(i+1)}$ should be given formally first, thereinto i ranges over positions in the recommended list of user u, and we use the observed rating R_{ui} to weigh the degree user u prefers item i. $NDCG_u@k$ is the ratio of $DCG_u@k$ to ideal DCG for that user:

$$NDCG_u@k = \frac{DCG_u@k}{IDCG_u@k}, \tag{11}$$

where $IDCG_u@k$ is the maximum possible DCG when the recommended items are just in descending order by user u preference. $NDCG@k$ is the mean value of $NDCG_u@k$ over all users, reflecting model performance of recommended list at the top k ranking.

5.3 Effectiveness

This section will validate the effectiveness of the proposed SFPR and MFPR compared to those baselines. For Douban Book and Dianping datasets, we generate training set $\mathcal{T}_r$ and test set $\mathcal{T}_e$ using different split ratios $30\%, 50\%, 70\%$, respectively. The random split was carried out 5 times independently in all experiments and we report the mean values of $\varepsilon_{0/1}$ and $NDCG$.

Parameters of all methods are tuned to the optimal values through cross validation on the datasets. For fair comparison, we set the same number of latent dimension $d = 10$ for all matrix factorization based methods. We select $\varepsilon_{0/1}$, $NDCG@5$ and $NDCG@10$ as evaluation metrics. We also record the improvement ratio on these evaluation metrics of all methods compared to the SVD. Moreover, we also conduct the t-test experiments with 95% confidence, which shows that the $\varepsilon_{0/1}$ and the $NDCG$ improvements is statistically stable and non-contingent.

The experimental results are shown in Table 2, the main findings from the experimental comparisons are summarized as follows: (1) MFPR achieves the best performance in all conditions, which validates the significant benefits of integrating both explicit feedback and multiple implicit feedbacks. The experiments also confirm that better performance will be achieved through integrating

Table 2. Performance Comparisons on Douban Book and Dianping (d=10, the baseline of improvement ratio is SVD)

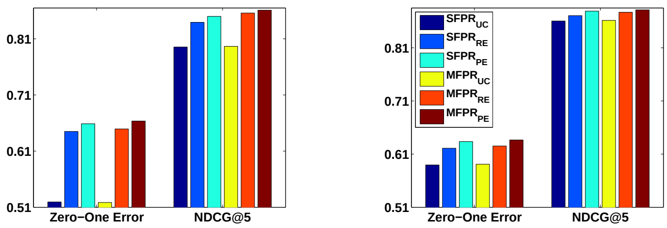
Datasets	Training	Metric	MP	SVD	BPR-MF	EN-BPRMF	SVD++	FM	SFPR	MFPR
Douban Book	30%	$\varepsilon_{0/1}$	0.5210	0.5251	0.5314	0.5372	0.6089	0.6145	0.6270	**0.6307**
		Improve	-0.66%		1.20%	2.30%	15.96%	17.03%	19.41%	20.11%
		NDCG@5	0.7831	0.7879	0.7845	0.7861	0.8291	0.8288	0.8371	**0.8399**
		Improve	-0.78%		-0.43%	-0.23%	5.23%	5.19%	6.24%	6.60%
		NDCG@10	0.8301	0.8332	0.8318	0.8323	0.8656	0.8691	0.8706	**0.8726**
		Improve	-0.37%		-0.17%	-0.11%	3.89%	4.31%	4.49%	4.73%
	50%	$\varepsilon_{0/1}$	0.5225	0.5909	0.5299	0.5374	0.6396	0.6399	0.6605	**0.6636**
		Improve	-11.58%		-10.32%	-9.05%	8.24%	8.29%	11.78%	12.30%
		NDCG@5	0.7969	0.8347	0.7989	0.7994	0.8516	0.8500	0.8564	**0.8611**
		Improve	-4.53%		-4.29%	-4.23%	2.02%	1.83%	2.60%	3.16%
		NDCG@10	0.8478	0.8747	0.8493	0.8494	0.8887	0.8864	0.8927	**0.8959**
		Improve	-3.08%		-2.90%	-2.89%	1.60%	1.34%	2.06%	2.42%
	70%	$\varepsilon_{0/1}$	0.5239	0.6242	0.5312	0.5397	0.6558	0.6582	0.6676	**0.6756**
		Improve	-16.07%		-14.90%	-13.54%	5.06%	5.45%	6.95%	8.23%
		NDCG@5	0.8338	0.8791	0.8403	0.8409	0.8874	0.8875	0.8895	**0.8932**
		Improve	-5.15%		-4.41%	-4.35%	0.94%	0.96%	1.18%	1.60%
		NDCG@10	0.8814	0.9110	0.8821	0.8824	0.9172	0.9164	0.9196	**0.9220**
		Improve	-3.25%		-3.17%	-3.14%	0.68%	0.59%	0.94%	1.21%
Dianping	30%	$\varepsilon_{0/1}$	0.5967	0.5922	0.5999	0.6072	0.6118	0.6220	0.6248	**0.6253**
		Improve	0.59%		1.30%	2.53%	3.31%	5.03%	5.50%	5.59%
		NDCG@5	0.8214	0.8178	0.8225	0.8261	0.8293	0.8365	0.8377	**0.8387**
		Improve	0.44%		0.57%	1.01%	1.41%	2.29%	2.43%	2.56%
		NDCG@10	0.8619	0.8594	0.8630	0.8658	0.8692	0.8689	0.8721	**0.8752**
		Improve	0.29%		0.42%	0.74%	1.14%	1.11%	1.48%	1.84%
	50%	$\varepsilon_{0/1}$	0.5965	0.6191	0.6009	0.6062	0.6304	0.6307	0.6345	**0.6367**
		Improve	-3.65%		-2.94%	-2.08%	1.83%	1.87%	2.49%	2.84%
		NDCG@5	0.8628	0.8727	0.8643	0.8674	0.8774	0.8778	0.8801	**0.8815**
		Improve	-1.13%		-0.96%	-0.61%	0.54%	0.58%	0.85%	1.01%
		NDCG@10	0.8924	0.8999	0.8940	0.8961	0.9044	0.9040	0.9056	**0.9076**
		Improve	-0.83%		-0.66%	-0.42%	0.50%	0.46%	0.63%	0.86%
	70%	$\varepsilon_{0/1}$	0.5987	0.6348	0.6006	0.6103	0.6411	0.6437	0.6468	**0.6498**
		Improve	-5.69%		-5.39%	-3.86%	0.99%	1.40%	1.89%	2.36%
		NDCG@5	0.8858	0.8982	0.8875	0.8891	0.9012	0.8996	0.9015	**0.9029**
		Improve	-1.38%		-1.19%	-1.01%	0.33%	0.16%	0.37%	0.50%
		NDCG@10	0.9099	0.9196	0.9110	0.9126	0.9217	0.9209	0.9219	**0.9234**
		Improve	-1.05%		-0.94%	-0.76%	0.23%	0.14%	0.25%	0.41%

more feedback information. For example, SFPR outperforms BPR-MF due to integration of ratings, SVD++ outperforms SVD because of implicit feedback, and the superiority of MFPR to SFPR comes from more implicit feedbacks. Note that MFPR and FM both utilize all feedback information, while MFPR always has better performance. The reason lies in that MFPR designs an effective mechanism treating explicit and implicit feedbacks differently, while FM handles all feedbacks equally. In all, exploiting and integrating multiple feedbacks is really helpful to improve the performance in the personalized ranking recommendation task. (2) When considering different training data ratios, we can find that the improvements of those models integrating explicit feedback with implicit feedbacks (i.e., SVD++, FM, SFPR and MFPR) are more significant for less training data. This indicates that integrating implicit feedbacks into models can effectively alleviate data sparsity of rating information. Specifically, FM outperforms SVD++ and MFPR outperforms SFPR because of integrating more implicit feedbacks. More combined implicit feedbacks mean more supplementary information for ratings. Thus, it is desirable to achieve much better recommendation performance through integrating comprehensive multiple feedbacks, particularly when rating information is insufficient. (3) From the results, we can also note that pairwise methods are more suitable for personalized ranking recommendation. Specifically, SVD, SVD++ and FM are rating prediction models, also known as pointwise methods, while SFPR and MFPR are pairwise ranking models. It is obvious that SFPR and MFPR outperform those three pointwise models. Specially, SFPR uses the same base learner as SVD++. Note that the other two pairwise ranking models (i.e. BPR-MF and EN-BPRMF) fail to defeat those pointwise models. We think the reason lies in that BPR-MF and EN-BPRMF utilize only implicit feedback, so they fail to generate accurate partial order item pairs as training set. In contrast, the proposed SFPR and MF-PR generate item pairs with more accurate ranking order as training set from explicit feedback.

5.4 Impact of Different Training Set Generation Algorithms

In this section, we verify the effectiveness of the designed training set generation algorithm IPPE. In order to validate the superiority of the IPPE, we compare it with the following two baseline methods. Following the idea of BPR-MF in [13], for user u, we make cartesian product of $E_{tr}(u)$ with user's unknown items to construct training set. We name this approach as *IPUC* which means *I*tem *P*airs of partial order are obtained from *U*nknown item related *C*artesian product. We also consider a variation of the IPPE method. From $E_{tr}(u)$ of each user u, we randomly sample two items each time and generate the item pair with partial order according to their observed ratings. In order to produce the similar training data size as the IPPE, the random process for each user u was conducted $|E_{tr}(u)|$ times. We name this approach as *IPRE* which means *I*tem *P*airs of partial order are obtained from checking *R*andom pairs in *E*xplicit item set. And we retain the same generation strategy for test set as the IPPE for these two approaches.

We apply these three different training set generation algorithms in SFPR and MFPR. As shown in Fig. 4, SFPR based on the methods IPUC, IPRE and IPPE are named as SFPR_{UC}, SFPR_{RE}, SFPR_{PE} respecitvely. It is similar for MFPR. We conduct experiments on both Douban Book and Dianping dataset-s, where the "read" feedback and the "good overall" feedback are still chosen for the SFPR. We can observe that models with IPPE have much better performance than those with IPUC. Specifically, SFPR_{UC} and MFPR_{UC} have very bad



(a) Douban Book (b) Dianping

Fig. 4. Performance under different training set generation algorithms on Douban Book and Dianping

performance, which degrades as BPR-MF in Table 2. Since the method IPPE makes full use of the rating information and thus the corresponding training set $\mathcal{T}_r$ consists of item pairs with more accurate partial order. On the contrary, the approach IPUC just discards the item orders implied by rating information and deals with the rating as ordinary implicit feedback. Moreover, we observe that SFPR_{PE} and MFPR_{PE} outperform SFPR_{RE} and MFPR_{RE} slightly but stably. This shows that sampling adjacent items pairs from random permutations outperforms that sampling item pairs randomly. In summary, for such multiple-feedback data, the proposed IPPE method is more effective to generate training set for the personalized ranking models.

6 Conclusion

In this paper, we study the personalized ranking recommendation by integrating multiple feedbacks, and propose a unified multiple feedbacks personalized ranking framework MFPR. Extensive experiments on two real-world datasets conform the superiority of MFPR. Moreover, we also have designed a delicate algorithm IPPE to generate training data with more accurate partial order for the proposed ranking model. The empirical evaluation results suggest that IPPE through checking adjacent items in a permutation is superior to IPUC and IPRE.

7 Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 61375058, 61572467), National Key Basic Research and Department (973) Program of China (No. 2013CB329606), and the Co-construction Project of Beijing Municipal Commission of Education.

References

1. C. Burges, T. Shaked, and a. et. Learning to rank using gradient descent. In *ICML*, pages 89–96. ACM, 2005.
2. A. da Costa Fortes and Manzato. Ensemble learning in recommender systems: Combining multiple user interactions for ranking personalization. In *PBSMW*, pages 47–54. ACM, 2014.
3. Z. Gantner, S. Rendle, et al. Mymedialite: A free recommender system library. In *RecSys*, pages 305–308. ACM, 2011.
4. R. He and J. McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1510.01784*, 2015.
5. P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 5:1457–1469, 2004.
6. Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE, 2008.
7. Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, pages 426–434. ACM, 2008.
8. J. Lee, S. Bengio, et al. Local collaborative ranking. In *WWW*, pages 85–96. ACM, 2014.
9. D. W. Oard, J. Kim, et al. Implicit feedback for recommender systems. In *AAAI*, pages 81–83, 1998.
10. R. Pan, Y. Zhou, et al. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE, 2008.
11. S. Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.
12. S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, 2012.
13. S. Rendle, C. Freudenthaler, et al. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.
14. R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. Citeseer, 2011.