

# MFMAP: Learning to Maximize MAP with Matrix Factorization for Implicit Feedback in Recommender System

Jianli Zhao<sup>1\*</sup>, Zhengbin Fu<sup>1</sup>, Qiuxia Sun<sup>2</sup>, Sheng Fang<sup>1</sup>, Wenmin Wu<sup>1,3</sup>,  
Yang Zhang<sup>1</sup>, Wei Wang<sup>1</sup>

<sup>1</sup> College of Computer Science & Engineering, Shandong University of Science and Technology  
Qingdao, China

[E-mail : jlzhao@sdust.edu.cn]

<sup>2</sup> College of Mathematics and Systems Science, Shandong University of Science and Technology  
Qingdao, China

<sup>3</sup> Qihoo 360 Show Ads Algorithm Group, Shanghai, China

\*Corresponding author : Jianli Zhao

*Received June 2, 2018; revised October 12, 2018; accepted November 24, 2018;  
published May 31, 2019*

---

## Abstract

Traditional recommendation algorithms on Collaborative Filtering (CF) mainly focus on the rating prediction with explicit ratings, and cannot be applied to the top-N recommendation with implicit feedbacks. To tackle this problem, we propose a new collaborative filtering approach namely Maximize MAP with Matrix Factorization (MFMAP). In addition, in order to solve the problem of non-smoothing loss function in learning to rank (LTR) algorithm based on pairwise, we also propose a smooth MAP measure which can be easily implemented by standard optimization approaches. We perform experiments on three different datasets, and the experimental results show that the performance of MFMAP is significantly better than other recommendation approaches.

---

**Keywords:** Top-N recommendation, Collaborative Filtering (CF), learning to rank (LTR), Mean Average Precision (MAP), implicit feedback.

---

This paper is supported by the National Key R&D Plan (No. 2018YFC0831002, No.2017YFC0804406), the Key R&D Plan of Shandong Province (No.2018GGX101045), Humanity and Social Science Fund of the Ministry of Education (No.18YJAZH136, N. 17YJCZH262), the Key Project of Industrial Transformation and Upgrading (Made in China 2025 No.TC170A5SW), the National Natural Science Foundation of China (No. 61433012, U1435215).

## 1. Introduction

Recommender Systems collect preference information for a set of users or items [1]. The information can be explicit, such as users' ratings on movies, music and books. However, most of the feedbacks given by its users are implicit, we only know which items user has interacted, e.g., clicked and browsed, etc. In these cases, we do not know the explicit rating information, a specific score that represents the user's preference for the item, thus we need to learn a prediction function from implicit feedback data. This task can be considered as classification problem, which need to classify items into relevant or irrelevant, or a LTR problem which need to produce an optimal item recommendation list.

In order to improve user' satisfaction, Top-N recommendation has become an increasing attractive research in recent years, because it generates a Top-N recommendation list [2]. Conventional recommender systems focus on computing scores. A predicted score reflects a user's preference for an item. However, some works have showed that low-rated prediction error does not mean that the user have a strong preference for the item [3-5]. On the other hand, almost all of recommender systems present a Top-N recommendation list rather than the preference strength of users.

**Table 1.** Mathematical notations

Notations	Description
$P_u$	User $u$ 's latent factor vector
$Q_i$	Item $i$ 's latent factor vector
$d$	The dimension of latent vector
$r_{ui}$	The rank of item $i$ in user $u$ 's recommendation list
$y_{ui}$	The implicit feedback data user $u$ marked on item $i$
$f_{ui}$	User $u$ 's preference to item $i$
$\lambda$	The regularization coefficient
$\gamma$	The learning rate
$t$	Number of iterations
AP	Average precision
MAP	Mean average precision

Some ranking-oriented CF approaches such as EigenRank [6] and CoFiRank [7] have been proposed for explicit feedback domains, those approaches are invalid for implicit feedback data, since they need users' explicit preference information like ratings to various items. As the implicit feedback in Top-N recommender systems is often binary, the quality of recommendation list can be measured using the MAP. And MAP provides a single-figure measure of quality with especially good discrimination and stability, and roughly corresponds to the average area under the prediction-recall curve [8]. So MAP is a common method to measure recommendation performance when the system provides its users with a ranked list of the most relevant items [9]. Rendle Steffen proposed an approach named Bayesian Personalized Ranking (BPR) in [10], and BPR trains models by maximizing the measure of Area under the ROC Curve (AUC), which is based on pairwise comparisons between items. Note that AUC is not a top-biased measure as mistakes at the top of the recommendation list carry the same weight to mistakes at the bottom of the list. To address this drawback, a new approach named CLiMF was proposed in [11] which learns the model's parameters by optimizing the Mean Reciprocal Rank (MRR). However, from the definition of MRR, one can find that MRR just take the first relevant item in the list into account, thus MRR cannot guarantee to get a most relevant top-N recommendation list.

And on the contrary with AUC, MAP is a list-wise metric, for which the mistakes at the top of the recommendation list carry a higher penalty than mistakes at the bottom of the list [12-14]. The value of MAP depends on the sorting of items in the recommendation list. However, the ordering of items related to predicted user preferences is changed in a non-smooth manner, it results in a non-smoothing change in the MAP evaluation metric with respect to the user implicit feature and item implicit feature. This makes it impossible to optimize it using conventional methods.

In order to tackle these shortcomings of previous works, we propose a new collaborative filtering approach, learning to MFMAP which models the data by maximizing MAP directly. Besides, unlike MRR, MFMAP take all the relevant items in the ranked list into account, thus can guarantee to get an optimized top-N recommendation list.

The contributions of this paper can be summarized as following:

- (1) We propose a novel CF recommendation approach, MFMAP, which can be easily applied to top-N recommendation with implicit feedbacks. MFMAP is a list-level evaluation metric that considers the position of all items in the list, and it is top-biased. The penalty for an item at the top of the list is heavier than the penalty for an item at the bottom of the list. We perform experiments on three different datasets with various conditions to demonstrate our approach outperforms other recommendation approaches.
- (2) We propose a smoothed version of MAP on the basis of Matrix Factorization Model which makes the parameters can be learned by standard optimization methods. By using the smoothing function approximation to replace the non-smooth part of the MAP,

the objective function can be optimized using an optimization method, and the parameters of the model are trained. And our experiments can also prove that the performance of our proposed smoothed MAP method is superior to the traditional LTR recommendation algorithms.

The rest of this paper is organized as follows. In Section 2 we discuss the related works. Section 3 presents the proposed MFMAP model in detail. In Section 4, we discuss our experimental evaluation. Finally, we summarize our work and highlight a few of future work in Section 5.

## 2. Related Work

In this paper, our work is related to collaborative filtering with implicit feedback and learning to rank. Most relevant related work will be presented in the following respectively.

### 2.1 Collaborative Filtering with Implicit Feedback

Among the existing works of collaborative filtering, most of them focus on rating prediction [15-18]. Past works have shown that matrix factorization, also known as latent factor model, works well on this problem. The logic behind such models is to fit the observed rating by two latent factor vectors' inner product  $f_{ui} = \langle P_u, Q_i \rangle$ .  $P_u$  and  $Q_i$  are user  $u$ 's and item  $i$ 's latent factor vectors respectively. It has been shown that these models perform well in recommendation accuracy and stability [19-20].

BPR is one of the matrix factorization models for implicit feedback. It uses relevance strength measured by the count of user's interaction such as click or purchase to the item rather than score as the model's input. Some extensions of this approach are introduced in [21-23].

### 2.2 Learning to Rank

In recent years, LTR has become a research hotspot in the field of information retrieval and recommender systems. [24]. The work of this paper is a branch of LTR which learn the model parameters by optimize the IR metrics [25-26], for which the processing of the non-smoothing problem becomes the research difficulty [27-28]. Previous works used two methods to deal with this problem, [29] and [30] minimized convex upper bounds of loss functions based on the evaluation measures, while [31] and [32] optimized a smoothed version of an evaluation measure directly. The most similar work to our paper is CLiMF. In CLiMF, the author proposes a smoothed MRR which is easy to optimize.

Traditional LTR algorithms based on pairwise establish recommendation model by maximizing the number of correct item pairs. However, these methods are not optimized for a

top-biased ranking evaluation metric, that is, when the items in the front position and the back position in the list have errors in ranking, the weight of the penalty for the error is the same. But in practice, when the order of items in the front position in the recommendation list is wrong, it is obviously much more important than the order of the items in the back position is wrong.

In order to tackle these problems, we propose a new algorithm named MFMAP, which is optimized by the matrix factorization algorithm combined with the smoothed MAP evaluation metric as the objective function. Besides, MFMAP considers all the items in the list, which guarantees a better list of recommendations.

### 3. MFMAP Model

In this section, we first formalize the problem then we present our main work of this paper: (1) we propose a smooth approximation method of MAP; and (2) the optimization mechanism of MFMAP.

#### 3.1 Problem and Terminology

The main research content of this paper is as follows: given implicit feedbacks, provide a list of optimal (from a MAP perspective) recommendation items to each user.

In a recommender system, we denote the implicit feedback data from  $M$  users to  $N$  items as a two-dimensional binary matrix  $Y$ , with  $M \times N$  entries. Each entry in  $Y$  is denoted with  $y_{ui}=1$  means that user  $u$  has interacted with item  $i$  which indicates that user  $u$  has a preference for item  $i$ , while  $y_{ui}=0$  represents the missing value, thus user  $u$ 's rating on item  $i$  is unknown.

As discussed in Section 2, matrix factorization models are to find two low rank matrices  $P$  and  $Q$  to approximate the original user-item interaction matrix. Thus, user  $u$ 's preference to item  $i$  can be predicted as the inner product of two latent factors as below:

$$f_{ui} = \langle P_u, Q_i \rangle = \sum_{d=1}^D P_{ud} \cdot Q_{id} \quad (1)$$

where  $P_u$  and  $Q_i$  are user  $u$  and item  $i$ 's latent factor vector respectively, and  $d$  is the dimension of latent factor.  $\langle \cdot \rangle$  denotes the inner product of two vectors.

Using the user's ratings rated on items, we can generate a recommendation list by sorting all items in descending order of the calculated ratings. Then, the Average Precision (AP) can be defined as:

$$AP_u = \frac{1}{\sum_{i=1}^N y_{ui}} \sum_{i=1}^N \frac{y_{ui}}{r_{ui}} \sum_{j=1}^N y_{uj} I(r_{uj} \leq r_{ui}) \quad (2)$$

$$I(r_{uj} \leq r_{ui}) = \begin{cases} 1, & r_{uj} \leq r_{ui} \\ 0, & \text{else} \end{cases} \quad (3)$$

where  $r_{ui}$  represents the rank of item  $i$  in user  $u$ 's recommendation list.

MAP represents the average of AP for all users, as shown below:

$$MAP = \frac{1}{M} \sum_{u=1}^M \frac{1}{\sum_{i=1}^N y_{ui}} \sum_{i=1}^N \frac{y_{ui}}{r_{ui}} \sum_{j=1}^N y_{uj} I(r_{uj} \leq r_{ui}) \quad (4)$$

### 3.2 Smoothed Mean Average Precision

Based on the ranking of items, the AP changes in a non-smooth way according to the predicted user preference ratings, and thus AP measure is a non-smoothing function about the users' and items' latent features. Therefore, we cannot use the optimization method of smooth loss function to train the model.

As we mentioned in Section 2, explicit optimization of evaluation metrics has made significant progress in the field of LTR, including MAP. The main problem is to construct a smoothing function for the model parameters to approximate  $r_{ui}$  and  $I(r_{uj} < r_{ui})$ .

Following the method in [30], we approximate  $r_{uj} < r_{ui}$  by the following logistic function:

$$I(r_{uj} \leq r_{ui}) \approx g(f_{uj} - f_{ui}) = g(\langle P_u, Q_j - Q_i \rangle) \quad (5)$$

$$g(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

If item  $j$  has a relatively higher correlation rating than item  $i$ , the condition that item  $j$  is ranked higher than item  $i$  is more likely to be satisfied. A sophisticated approximation of  $r_{ui}$  is proposed in [32], however, it has not been implemented in practice. Note that only  $1/r_{ui}$  used in the definition of MAP, thus there is no need to approximate  $r_{ui}$ . We use another logistic function to approximate  $1/r_{ui}$  directly:

$$\frac{1}{r_{ui}} \approx g(\langle P_u, Q_i \rangle) \quad (7)$$

Note that the higher the predicted relevance rating  $f_{ui}$ , the closer  $g(f_{ui})$  gets to 1, resulting in a low value of  $r_{ui}$ . Reversely, the lower  $f_{ui}$ , the larger is  $r_{ui}$ . Substituting (5) and (7) into (4) we can get a smooth MAP approximation:

$$MAP = \frac{1}{M} \sum_{u=1}^M \frac{1}{\sum_{i=1}^N y_{ui}} \sum_{i=1}^N y_{ui} g(f_{ui}) \sum_{j=1}^N y_{uj} g(f_{u(j-i)}) \quad (8)$$

### 3.3 Optimization

Since (8) is a smooth function with respect to  $P_u$  and  $Q_i$ , it can be optimized with standard optimization methods such as gradient ascent. We add L2 norms of  $P$  and  $Q$  into the loss function to avoid overfitting. The final MFMAP objective function is as following:

$$L = \sum_{u=1}^M \frac{1}{\sum_{i=1}^N y_{ui}} \sum_{i=1}^N y_{ui} g(f_{ui}) \sum_{j=1}^N y_{uj} g(f_{u(j-i)}) - \frac{1}{2} (\|P\|^2 + \|Q\|^2) \quad (9)$$

Note that the regularization term is negative, because our objective is maximized  $L$  and control the model complexity simultaneously.

We use SGD to maximize the objective function. The gradient of the objective function can be computed as below:

$$\frac{\partial L(P_u, Q)}{\partial P_u} = \frac{1}{\sum_{i=1}^N y_{ui}} \sum_{i=1}^N y_{ui} [g'(f_{ui}) \cdot Q_j \sum_{j=1}^N y_{uj} g(f_{u(j-i)}) + g(f_{ui}) \sum_{j=1}^N y_{uj} g'(f_{u(j-i)}) \cdot (Q_j - Q_i)] - \lambda P_u \quad (10)$$

$$\frac{\partial L(P_u, Q)}{\partial Q_i} = \frac{y_{ui} P_u}{\sum_{i=1}^N y_{ui}} \sum_{j=1}^N \{g'(f_{ui}) \cdot g(f_{u(j-i)}) + [g(f_{uj}) - g(f_{ui})] \cdot g'(f_{u(j-i)})\} - \lambda Q_i \quad (11)$$

where,

$$\begin{cases} f_{ui} = \langle P_u, Q_i \rangle \\ f_{u(j-i)} = \langle P_u, Q_j - Q_i \rangle \\ g'(x) = 1 / (2 + e^x + e^{-x}) \end{cases} \quad (12)$$

Then, parameters in MFMAP can be updated as:

$$\begin{cases} P_u + = \gamma \cdot \frac{\partial L(P_u, Q)}{\partial P_u} \\ Q_i + = \gamma \cdot \frac{\partial L(P_u, Q)}{\partial Q_i} \end{cases} \quad (13)$$

where  $\gamma$  is the learning rate.

The constructing process of MFMAP shown in **Table 2**.

**Table 2.** The constructing process of MFMAP

---

Algorithm 1: Train MFMAP model with SGA

---

**Input :** Implicit data  $R$ , iterations  $t$ ,  $d$ ,  $\gamma$   
**Output :** The parameter of MFMAP model

1. Initialize  $P, Q$  with random values
2. **while**  $t > 0$  **do**
3.   **foreach**  $(u, i) \in R$  **do**
4.     Calculate  $f_{u(j-i)}$  by (10),
5.      $loss = L(P, Q)$
6.     Calculate  $\frac{\partial L}{\partial P_u}, \frac{\partial L}{\partial Q_i}$  by formula (8), (9)
7.      $P_u = P_u + \gamma \cdot \frac{\partial L}{\partial P_u}$
8.      $Q_i = Q_i + \gamma \cdot \frac{\partial L}{\partial Q_i}$
9.   **end**
10. **end**
11.    $t = t - 1$
12. **end**

---

## 4. Experimental Evaluation

In this section, we do experiments to verify the performance of the MFMAP. Firstly, we introduce the experimental datasets and experimental setup. Secondly, we compare the recommendation performance of MFMAP in terms of top-N recommendation with other three approaches including BPR and CLiMF.

### 4.1 Datasets

We conduct experiments using a social dataset from Epinions<sup>1</sup>, a music system dataset from Last.fm<sup>2</sup> and a film dataset from MovieLens(ML-100k)<sup>3</sup>. These three datasets used in this paper are all commonly used datasets in the recommendation systems, which are real data

---

1. <http://www.epinions.com>

2. <http://www.last.fm/>

3. <http://www.grouplens.org>



extracted from real websites. The statistics of these three datasets is shown in **Table 3**. All of these datasets are publicly available. The Epinions dataset contains trust relationship between 49288 users. The Epinions dataset represents a directed social network, i.e., if user  $u$  is a trustee of user  $j$ , user  $j$  is not necessary a trustee of user  $i$ . And in our experiment, we excluded users with fewer than 25 trustees from the dataset. The Last.fm is an implicit feedback dataset consisting of 1892 users and 17632 singers, each record indicating the number of times a user listen to a song. We use each user's average listening counts as his threshold. For each user in the dataset, when an artist's listening counts larger than user's average listening counts we treated this artist as relevant. The MovieLens(ML-100k) is an explicit feedback dataset which consists of 100000 ratings by 943 users on 1682 films, the rating ranges from 1 to 5. In order to convert this dataset into an implicit feedback dataset, we refer to a method commonly used in most of papers, we removed the rating scores from the dataset and treat a film as the user's related film if he has rated this film. By this way, we convert the ratings into implicit feedbacks.

**Table 3.** Statistics of three datasets

Datasets	Epinions	Last.fm	ML-100k
non-zeros	346035	92834	100000
users	4718	1892	943
friends/related artists /related films	49288	17632	1682
sparseness	99.85%	99.72%	93.70%
Avg. friends/related artists per user/ related films per user	73.34	73.34	106.04

## 4.2 Experimental Setup and Evaluation Metrics

We divide each dataset into training-set and test-set according to different conditions. For instance, the condition of "Given 5" denotes that for each user we randomly selected 5 out of her trustees/related artists/related films to form the training-set, and use the remaining trustees/related artists/related films to form the test-set. The task is to generate a recommendation list for each user in the training-set and measure performance using holdout data in the test-set. We don't design experiments on the Last.fm and MovieLens datasets under the condition "Given 15", due to we found it would result in too few samples left in the test-set sometimes. We consider the optimal balance between performance and cost, and

choose the most appropriate parameter in MFMAP: the regularization parameter  $\gamma = 0.1$ , the feature dimension  $d=10$  and the learning rate  $\lambda = 0.1$ .

The main evaluation metric used in our experiment is MAP which is the optimization object of our model. In addition, we also use precision and recall to measure the performance. Precision at top-k (P@k) reflects the ratio of relevant items in the recommendation items. And recall at top-k (R@k) reflects the ratio of relevant items selected out of all relevant items in the test-set. In order to further prove the effectiveness of MFMAP model, we also use AUC and MRR as evaluation metrics to measure the recommendation quality.

### 4.3 Baseline Models

We compare our proposed model MFMAP with a baseline model and two state-of-the-art models which namely BPR and CLiMF.

The models used for comparative test are listed as below:

- 1) PopRec. A naive baseline model, the recommendation results of all users in the test-set are always the popular trustees, artists or films.
- 2) BPR-MF. A CF model based implicit feedback data. BPR-MF uses matrix factorization (MF) to learn the model's parameters with BPR optimization criterion.
- 3) CLiMF. A CF model that optimizes the evaluation measure. CLiMF learns the model's parameters by directly maximizing the MRR measure.

### 4.4 Results

The performance comparison of MFMAP and other three algorithms on different datasets in terms of MAP, Precision, Recall, AUC and MRR are respectively shown in [Table 4](#), [Table 5](#), [Table 6](#) and [Table 7](#). We highlight the results of MFMAP in the table.

Two main observations can be drawn from the [Table 4](#), [Table 5](#) and [Table 6](#): Firstly, BPR-MF, CLiMF and MFMAP all get some improvements compared with the PopRec on all metrics. Secondly, the proposed MFMAP model significantly outperforms the other three algorithms on all datasets and evaluation metrics. The improvement of MAP, Precision rate and Recall rate shows that by optimizing MAP, the MFMAP model improve the recommendation quality of top-ranked items.

As can be also seen from the results, BPR-MF performs worse than CLiMF and MFMAP in all conditions on Last.fm and MovieLens dataset and in most conditions of the Epinions dataset. The reason might be that the BPR-MF is not designed to improve the recommendation quality of top-ranked items.

**Table 4.** Performance comparison of algorithms in terms of MAP

Dataset/Method		Given 5	Given 10	Given 15
		MAP	MAP	MAP
Epinions	<i>PopRec</i>	0.010	0.010	0.010
	<i>BPR-MF</i>	0.011	0.017	0.016
	<i>CLiMF</i>	0.012	0.016	0.016
	<i>MFMAP</i>	<b>0.013</b>	<b>0.018</b>	<b>0.017</b>
Last.fm	<i>PopRec</i>	0.020	0.024	--
	<i>BPR-MF</i>	0.020	0.029	--
	<i>CLiMF</i>	0.023	0.029	--
	<i>MFMAP</i>	<b>0.027</b>	<b>0.032</b>	--
MovieLens	<i>PopRec</i>	0.032	0.033	--
	<i>BPR-MF</i>	0.039	0.044	--
	<i>CLiMF</i>	0.044	0.045	--
	<i>MFMAP</i>	<b>0.052</b>	<b>0.054</b>	--

**Table 5.** Performance comparison of algorithms in terms of Precision

Dataset/Method		Given 5		Given 10		Given 15	
		P@5	P@10	P@5	P@10	P@5	P@10
Epinions	<i>PopRec</i>	0.139	0.069	0.135	0.067	0.123	0.062
	<i>BPR-MF</i>	0.147	0.145	0.191	0.170	0.180	0.162
	<i>CLiMF</i>	0.159	0.154	0.189	0.169	0.182	0.163
	<i>MFMAP</i>	<b>0.183</b>	<b>0.175</b>	<b>0.192</b>	<b>0.173</b>	<b>0.185</b>	<b>0.167</b>
Last.fm	<i>PopRec</i>	0.063	0.031	0.040	0.020	--	--
	<i>BPR-MF</i>	0.067	0.031	0.055	0.027	--	--
	<i>CLiMF</i>	0.071	0.038	0.056	0.029	--	--
	<i>MFMAP</i>	<b>0.083</b>	<b>0.041</b>	<b>0.058</b>	<b>0.036</b>	--	--

Movielens	<i>PopRec</i>	0.121	0.083	0.152	0.122	--	--
	<i>BPR-MF</i>	0.353	0.312	0.371	0.334	--	--
	<i>CLiMF</i>	0.382	0.324	0.401	0.351	--	--
	<i>MFMAP</i>	<b>0.462</b>	<b>0.421</b>	<b>0.470</b>	<b>0.414</b>	--	--

**Table 6.** Performance comparison of algorithms in terms of Recall

Dataset/Method		Given 5		Given 10		Given 15	
		R@5	R@10	R@5	R@10	R@5	R@10
Epinions	<i>PopRec</i>	0.009	0.014	0.010	0.016	0.013	0.017
	<i>BPR-MF</i>	0.015	0.030	0.016	0.031	0.018	0.032
	<i>CLiMF</i>	0.018	0.031	0.017	0.030	0.018	0.032
	<i>MFMAP</i>	<b>0.019</b>	<b>0.032</b>	<b>0.018</b>	<b>0.032</b>	<b>0.019</b>	<b>0.033</b>
Last.fm	<i>PopRec</i>	0.012	0.021	0.017	0.029	--	--
	<i>BPR-MF</i>	0.032	0.054	0.045	0.071	--	--
	<i>CLiMF</i>	0.039	0.059	0.043	0.070	--	--
	<i>MFMAP</i>	<b>0.043</b>	<b>0.060</b>	<b>0.058</b>	<b>0.076</b>	--	--
Movielens	<i>PopRec</i>	0.013	0.027	0.015	0.032	--	--
	<i>BPR-MF</i>	0.035	0.064	0.046	0.085	--	--
	<i>CLiMF</i>	0.041	0.070	0.047	0.083	--	--
	<i>MFMAP</i>	<b>0.045</b>	<b>0.082</b>	<b>0.054</b>	<b>0.090</b>	--	--

**Table 7.** Performance comparison of algorithms in terms of AUC & MRR

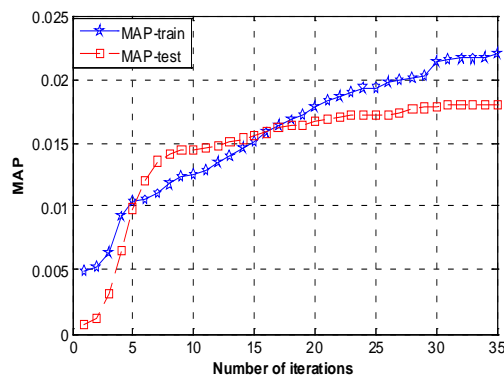
Dataset/Method		Given 5		Given 10		Given 15	
		AUC	MRR	AUC	MRR	AUC	MRR
Epinions	<i>PopRec</i>	0.576	0.132	0.535	0.107	0.523	0.105
	<i>BPR-MF</i>	0.810	0.301	0.809	0.317	0.810	0.306
	<i>CLiMF</i>	0.831	0.354	0.832	0.369	0.822	0.350
	<i>MFMAP</i>	<b>0.835</b>	<b>0.375</b>	<b>0.842</b>	<b>0.380</b>	<b>0.827</b>	<b>0.367</b>

Last.fm	<i>PopRec</i>	0.502	0.091	0.506	0.076	--	--
	<i>BPR-MF</i>	0.653	0.135	0.643	0.123	--	--
	<i>CLiMF</i>	0.667	0.166	0.647	0.129	--	--
	<i>MFMAP</i>	<b>0.674</b>	<b>0.174</b>	<b>0.653</b>	<b>0.132</b>	--	--
MovieLens	<i>PopRec</i>	0.621	0.383	0.652	0.412	--	--
	<i>BPR-MF</i>	0.856	0.532	0.891	0.624	--	--
	<i>CLiMF</i>	0.877	0.635	0.897	0.631	--	--
	<i>MFMAP</i>	<b>0.885</b>	<b>0.667</b>	<b>0.902</b>	<b>0.651</b>	--	--

**Table 7** shows that MFMAP model performs better than the other two algorithms on AUC and MRR. Compared with BPR-MF and CLiMF, which optimize AUC and MRR respectively, the MFMAP improves the recommendation quality of top-ranked items by optimizing MAP. For MAP is a list-wise metric that take all the relevant items into account in the top-ranked items, while AUC is not a top-biased metric and MRR only takes the first relevant item into account.

Meanwhile, we investigate the effectiveness of our proposed learning algorithm of MFMAP. **Fig. 1** shows the change of MAP evaluation with the number of iterations in the Epinions dataset under the condition of "given 10". **Fig. 2** and **Fig. 3** show the change of MAP evaluation with the number of iterations under the "Given 5" condition for the Last.fm and the MovieLens dataset respectively.

As shown in **Fig.1, Fig.2** and **Fig.3**, the MAP increase as the number of iterations increase until convergence. It means that MFMAP is effective for increasing MAP to a local maximum. This observation also indicates that MFMAP learns users' and items' latent factors effectively from the training set by optimizing MAP which consequently help to improve MAP value in the test set.



**Fig. 1.** MFMAP performance on Epinions.

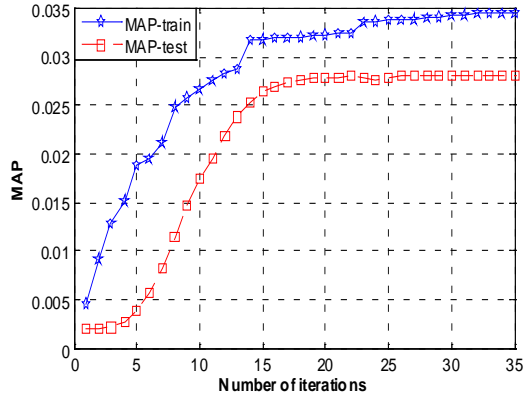


Fig. 2. MFMAP performance on Last.fm

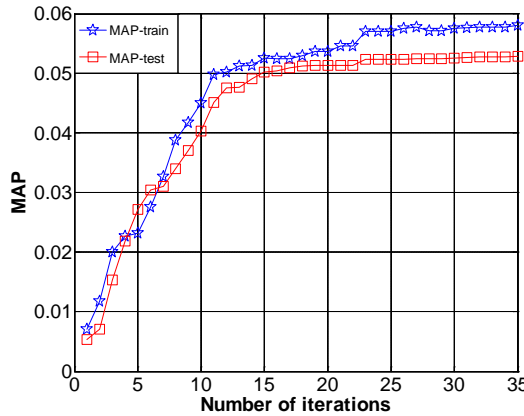


Fig. 3. MFMAP performance on MovieLens

Fig. 4 shows the performance comparison of MFMAP and other algorithms on Epinions dataset under the “Given 10” condition. Accordingly, Fig. 5 and Fig. 6 shows the performance comparisons of MFMAP and other algorithms on the Last.fm and MovieLens dataset under the “Given 5” condition. All these figures show that MFMAP model coverage faster and get a higher MAP value than other three algorithms.

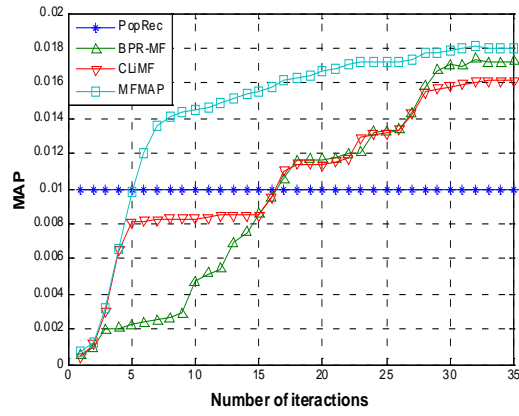


Fig. 4. Performances on Epinion

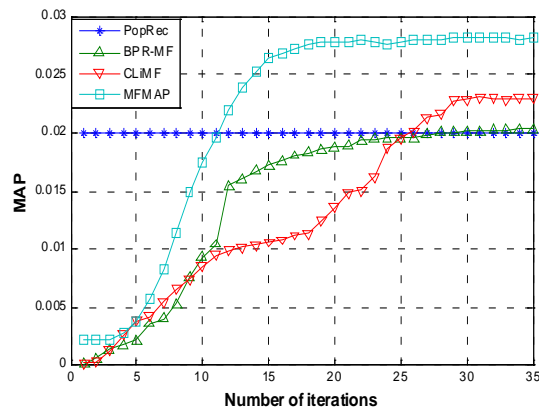


Fig. 5. Performances on Last.fm

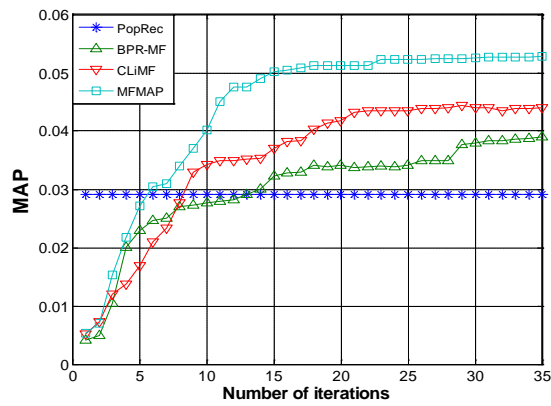


Fig. 6. Performances on MovieLens

From all these figures and tables, we can see that our MFMAP model is more effective than other three algorithms.

## 5. Conclusions

### 5.1 Conclusions and future work

In this paper, we focus on the problem that pairwise method didn't consider the item's rank position in recommendation list. Combining with the listwise ideology we present a new collaborative filtering approach that train the recommendation model by directly optimizing the ranking metric MAP, which is named MFMAP. And then we optimized the metric by using smoothed approximation functions instead of the non-continuous parts of original MAP, the model parameters can be obtained by using the stochastic gradient rise algorithm. In addition, experiments on three datasets show that MFMAP algorithm significantly outperforms other LTR approaches.

Many extensions can be done with this work. Firstly, the combination of user's explicit and implicit information can better reflect the user's interest preferences and may further improve the performance of the algorithm. Secondly, we can use some additional information of items such as genres of movies [33] to improve the recommendation diversity. Thirdly, it's fun and useful to apply MFMAP to some special scenarios, such as social network [34-35] or context-aware recommendations [36].

## References

- [1] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, Vol. 46 No. 1, pp. 109-132, 2013. [Article \(CrossRef Link\)](#).
- [2] P. Cremonesi, Y. Koren, R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. of the fourth ACM conference on Recommender systems*, ACM, pp.39-46, 2010. [Article \(CrossRef Link\)](#).
- [3] McNee, Sean M., John Riedl, and Joseph A. Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," *CHI'06 extended abstracts on Human factors in computing systems*, ACM, 2006. [Article \(CrossRef Link\)](#).
- [4] A. Gunawardana, G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *The Journal of Machine Learning Research*, Vol. 10 No. 10, 2009, pp.2935-2962, 2009. [Article \(CrossRef Link\)](#).
- [5] A. Saha, J. Rajendran, S. Shekhar, B. Ravindran., "How popular are your tweets?," in *Proc. of the 2014 Recommender Systems Challenge*, ACM, 2014. [Article \(CrossRef Link\)](#).
- [6] N. N. Liu, Q. Yang, "Eigenrank: a ranking-oriented approach to collaborative filtering," in *Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2008. [Article \(CrossRef Link\)](#).
- [7] M. Weimer, A. Karatzoglou, "Maximum margin matrix factorization for collaborative ranking," *Advances in neural information processing systems*, 2007. [Article \(CrossRef Link\)](#).
- [8] C. D. Manning, P. Raghavan, H. Schütze, "An Introduction to information retrieval," Vol. 1. *Cambridge: Cambridge university press*, 2008. [Article \(CrossRef Link\)](#).



- [9] G. Shani, A. Gunawardana, "Evaluating recommendation systems," *Recommender systems handbook*. Springer US, pp. 257-297, 2011. [Article \(CrossRef Link\)](#).
- [10] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009. [Article \(CrossRef Link\)](#).
- [11] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, A. Hanjalic, "CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering," in *Proc. of the sixth ACM conference on Recommender systems*, ACM, 2012. [Article \(CrossRef Link\)](#).
- [12] J. Davis, M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. of the 23rd international conference on Machine learning*, ACM, 2006. [Article \(CrossRef Link\)](#).
- [13] Y. Yue, T. Finley, F. Radlinski, T. Joachims, "A support vector method for optimizing average precision," in *Proc. of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2007. [Article \(CrossRef Link\)](#).
- [14] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, "TFMAP: optimizing MAP for top-n context-aware recommendation," in *Proc. of the 35th international ACM SIGIR conference on Research and development in information retrieval*, ACM, 155-164, 2012. [Article \(CrossRef Link\)](#).
- [15] Koren Y, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 426-434, 2008. [Article \(CrossRef Link\)](#).
- [16] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, 56(3):156-166, 2014. [Article \(CrossRef Link\)](#).
- [17] Y. Koren, R. Bell, C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, No. 8, pp.30-37, 2009. [Article \(CrossRef Link\)](#).
- [18] B. R. Salakhutdinov, A. Mnih, "Probabilistic matrix factorization," *Advances in neural information processing systems*. 2007. [Article \(CrossRef Link\)](#).
- [19] Manouselis N G, Costopoulou C I, "Designing a Web-based Testing Tool for Multi-Criteria Recommender Systems," *Engineering Letters*, 13(3):417-427, 2006. [Article \(CrossRef Link\)](#).
- [20] Zhong X, Yang G, Li L, et al., "Clustering and correlation based collaborative filtering algorithm for cloud platform," *IAENG International Journal of Computer Science*, 2016. [Article \(CrossRef Link\)](#).
- [21] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, W. Nejdl, "Real-time top-n recommendation in social streams," in *Proc. of ACM Conference on Recommender Systems*, ACM, 59-66, 2012. [Article \(CrossRef Link\)](#).
- [22] Guo W, Wu S, Wang L, et al., "Adaptive Pairwise Learning for Personalized Ranking with Content and Implicit Feedback," in *Proc. of IEEE/ Wic/ ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, 369-376, 2015. [Article \(CrossRef Link\)](#).
- [23] Z.H. Huang, J.W.Zhang, C.Q.Tian, S.L. Sun, Y.Xiang, "Survey on learning-to-rank based recommendation algorithms," *Ruan Jian Xue Bao/ Journal of Software*, 27(3):691-713, 2016 (in Chinese). [Article \(CrossRef Link\)](#).
- [24] Liu T Y, "Learning to Rank for Information Retrieval," *Acm Sigir Forum*, 41(2):58-62, 2011.
- [25] Karatzoglou A, Baltrunas L, Shi Y, "Learning to rank for recommender systems," in *Proc. of ACM Conference on Recommender Systems*, ACM, 493-494, 2013. [Article \(CrossRef Link\)](#).
- [26] Chen W, Liu T, Lan Y, et al., "Ranking measures and loss functions in learning to rank," in *Proc. of International conference on neural information processing*, 315-323, 2009. [Article \(CrossRef Link\)](#).
- [27] S. Chakrabarti, R. Khanna, U. Sawant, C. Bhattacharyya, "Structured learning for non-smooth ranking losses," in *Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008. [Article \(CrossRef Link\)](#).
- [28] Quoc, C., and Viet Le, "Learning to rank with nonsmooth cost functions," in *Proc. of the Advances in Neural Information Processing Systems 19*, 193-200, 2007. [Article \(CrossRef Link\)](#).

- [29] J. Xu, T. Y. Liu, M. Lu, H. Li, W. Y. Ma, “Directly optimizing evaluation measures in learning to rank,” in *Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2008. [Article \(CrossRef Link\)](#).
- [30] M. Taylor, J. Guiver, S. Robertson, T. Minka, “Softrank: optimizing non-smooth rank metrics,” in *Proc. of the 2008 International Conference on Web Search and Data Mining*, ACM, 2008. [Article \(CrossRef Link\)](#).
- [31] Shi Y, Karatzoglou A, Baltrunas L, et al., “xCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance,” in *Proc. of ACM Recommender Systems Conference*, ACM, 431-434, 2013. [Article \(CrossRef Link\)](#).
- [32] Chapelle, M. Wu, “Gradient descent optimization of smoothed information retrieval metrics,” *Information retrieval*, Vol. 13 No. 3, pp. 216-235, 2010. [Article \(CrossRef Link\)](#).
- [33] J. L. Zhao, W. M. Wu, C. S. Zheng, F. Meng, “A Novel Method for Matrix Factorization in Recommender System using Item's Information,” *Journal of Computational Information Systems*, Vol.11 No. 10, pp. 1–8, 2015. [Article \(CrossRef Link\)](#).
- [34] Zhongying Zhao\*, Chao Li, Xuejian Zhang, Francisco Chiclana, Enrique Herrera Viedma, “An Incremental Method to Detect Communities in Dynamic Evolving Social Networks,” *Knowledge-Based Systems*, 163: 404–415, 2019. [Article \(CrossRef Link\)](#).
- [35] Zhongying Zhao\*, Wenqiang Liu, Yuhua Qian, Liqiang Nie, Yilong Yin, Yong Zhang, “Identif-ying Advisor-advisee Relationships from Co-author Networks via a Novel Deep Model,” *Information Sciences*, 466: 258-269, 2018. [Article \(CrossRef Link\)](#).
- [36] W. M. Wu, J. L. Zhao, C. S. Zhang, F. Meng, Z. L. Zhang, et al., “Improving Performance of Tensor-based Context-aware Recommenders Using Bias Tensor Factorization with Context Feature Auto-encoding,” *Knowledge-Based Systems*, 2017, 2017.04.011. [Article \(CrossRef Link\)](#).



**Jianli Zhao** received his PhD degree in 2006 from the Department of Computer Application Technology, Northeastern University, China. In 2011, He served as an associate professor in the College of Computer Science and Engineering, Shandong University of Science and Technology, Major Research Direction in pervasive computing, personalized recommendation, indoor location.



**Zhengbin Fu** received his BBM degree in College of Computer Science and Engineering from Shandong University of Science and Technology, Qingdao, China, in 2016.



**Qiuxia Sun** received her PhD degree of System Theory in 2011 from Qingdao University, China. Now, she served an associate professor in the College of Mathematics and Systems Science, Shan-dong University of Science and Technology, Major Research Direction in Big Data Analysis.



**Sheng Fang** received his M.S. and Ph.D. degrees from Tsinghua University and Shandong University of Science and Technology in Computer Science in 2002 and 2008 respectively. He is currently a professor in College of Computer Science and Engineering, Shandong University of Science and Technology. He is a senior member of China Computer Federation, a member of CCF Multimedia Technical Committee, and ACM SIGMM China Chapter Member. His research interests are in the fields of multimedia processing.



**Wenmin Wu** received his M.E. degree in College of Computer Science and Engineering from Shandong University of Science and Technology, Qingdao, China, in 2016.



**Yang Zhang** received his B.E. degree in College of Information Science and Engineering from University of Jinan, Jinan, China, in 2014.



**Wei Wang** received his B.E degree in College of Computer Science and Engineering from Shandong University of Science and Technology, Qingdao, China, in 2018.