# Collaborative Recommendation with Multiclass Preference Context

Weike Pan and Zhong Ming*
{panweike,mingz}@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University

## Abstract

*Factorization- and neighborhood-based methods have been recognized as the state-of-the-art methods for collaborative recommendation tasks. Those two methods are known complementary to each other, while very few works have been proposed to combine them together. SVD++ tries to combine the main idea of latent features and neighborhood of those two methods, but ignores the existent* categorical scores *of the rated items. In this paper, we address this limitation and take a user's ratings as categorical* multiclass preferences*. In this regard, we propose a novel method called matrix factorization with multiclass preference context (MF-MPC), which integrates an enhanced neighborhood based on the assumption that users with similar past multiclass preferences (instead of oneclass preferences in SVD++) will have similar taste in the future. Mathematically, SVD++ is a special case of our MF-MPC. The main merit of our MF-MPC is its ability to make use of the multiclass preference context in the factorization framework in a fine-grained manner and thus inherits the advantages of those two methods in a better way. Experimental results on three real-world data sets show that our solution can perform significantly better than factorization-based method, neighborhood-based method and the integrated method with oneclass preference context.*

Keywords: Collaborative Recommendation, Matrix Factorization, Multiclass Preference Context

## 1 Introduction

Intelligent recommendation technology has been an effective solution to address the information overload and personalization challenges in many e-commerce and entertainment systems [1, 10]. The main assumption of most recommendation algorithms is that users' true preferences are closely related to users' explicit or implicit feedback such as ratings, transactions and examinations, which provides a way to learn and mine users' preferences for personalized services. There are mainly two branches of algorithms for learning users' preferences in a recommender system, i.e., model-based methods and neighborhood-based methods.

A model-based method assumes that users' feedback to items are governed by some latent model parameters such as users' latent interests and items' latent attributes, among which matrix factorization (MF) has been a state-of-the-art solution in many recommendation tasks. Basically, MF assumes that two latent feature matrices, one for users' interests and one for items' attributes, can be obtained via factorizing the original rating matrix of the observed user feedback. Typically, some loss functions and regularization terms are used to guide the factorization process. MF is known to be very effective and efficient in modeling users' feedback, which has been verified in various contests [3, 6].

An alternative solution to model-based methods is called neighborhood-based methods, which make recommendations based on the neighbors' aggregated preferences. A neighborhood-based method usually contains three steps, including similarity calculation, neighborhood construction and preference prediction. The main assumption is that users with similar taste in the past will be clustered into the same neighborhood and thus will have similar taste in the future, on which the prediction rule is based. The merit of neighborhood-based methods is its locality, simplicity and good interpretability, which is also known to be complementary to the model-based methods [2, 3].

A natural question we ask in this paper is that whether we can integrate those two complementary methods, i.e., the factorization-based method and the neighborhood-based method, in a principled way. There has been very few attempts in this line of research. One of the most representative work is called SVD++ [3], where the prediction rule contains two parts, one from the factorization-based method for a target (user, item) pair and one from the neighborhood-based method for the other rated items of the corresponding user. The resulted enhanced prediction rule can then inherit the merits of both methods. Specifically, the learned latent feature vectors of two users with similar rating behaviors will also be similar in the latent space, which reflects the effect of the neighborhood-based method.

However, there is still some information not exploited by SVD++. For a typical rating prediction problem in a recommender system, we usually have the categorical scores, e.g., $\{1,2,3,4,5\}$ for {bad, fair, good, excellent, perfect}, which is neglected by the newly introduced term of the expanded prediction rule in SVD++. Generally, whether an item is rated or not is a certain oneclass preference or feedback, and how the item is rated is a multiclass preference. Hence, SVD++ can be considered as a matrix factorization method with oneclass preference context (MF-OPC). In this paper, we propose a novel recommendation algorithm that integrates the factorization-based method and the neighborhood-based method in a fine-grained manner via exploiting the multiclass preference context (MPC). For this reason, we call our approach matrix factorization with multiclass preference context (MF-MPC). Empirical studies on three public data sets show that exploiting multiclass preference context can significantly improve the recommendation performance of MF and MF-OPC.

# 2 Our Solution: Matrix Factorization with Multiclass Preference Context

## 2.1 Problem Definition

In this paper, we study a typical matrix factorization problem that exists in various learning and recommendation applications. Specifically, we have $n$ users (or rows) and $m$ items (or columns), and some observed multiclass preferences such as ratings that are recorded in $\mathcal{R} = \{(u, i, r_{ui})\}$ with $r_{ui} \in \mathbb{M}$. The multiclass preference set $\mathbb{M}$ can be $\{1, 2, 3, 4, 5\}$, $\{0.5, 1, 1.5, \ldots, 5\}$ or other ranges. Our goal is then to build a model so that the missing entries of the original matrix can be predicted. The studied problem is usually called rating prediction in collaborative filtering or matrix completion in machine learning. We put some notations in Table 1.

Table 1: Some notations.

| | |
|---|---|
| $n$ | user number |
| $m$ | item number |
| $u, u'$ | user ID |
| $i, i'$ | item ID |
| $\mathbb{M}$ | multiclass preference set |
| $r_{ui} \in \mathbb{M}$ | rating of user $u$ on item $i$ |
| $\mathcal{R} = \{(u, i, r_{ui})\}$ | rating records of training data |
| $y_{ui} \in \{0, 1\}$ | indicator, $y_{ui} = 1$ if $(u, i, r_{ui}) \in \mathcal{R}$ |
| $\mathcal{I}_u^r$, $r \in \mathbb{M}$ | items rated by user $u$ with rating $r$ |
| $\mathcal{I}_u$ | items rated by user $u$ |
| $\mu \in \mathbb{R}$ | global average rating value |
| $b_u \in \mathbb{R}$ | user bias |
| $b_i \in \mathbb{R}$ | item bias |
| $d \in \mathbb{R}$ | number of latent dimensions |
| $U_{u\cdot} \in \mathbb{R}^{1\times d}$ | user-specific latent feature vector |
| $V_{i\cdot}, O_{i\cdot}, M_{i\cdot}^r \in \mathbb{R}^{1\times d}$ | item-specific latent feature vector |
| $\mathcal{R}^{te} = \{(u, i, r_{ui})\}$ | rating records of test data |
| $\hat{r}_{ui}$ | predicted rating of user $u$ on item $i$ |
| $T$ | iteration number in the algorithm |

## 2.2 Multiclass Preference Contenxt (MPC)

For a traditional matrix factorization (MF) model [6], the rating of user $u$ on item $i$, $r_{ui}$, is assumed to be dependent on latent features of user $u$ and item $i$ only. We can represent it in a probabilistic way as follows,

$$P(r_{ui}|(u, i)), \tag{1}$$

which means that the probability of generating the rating $r_{ui}$ is conditioned on the (user, item) pair $(u, i)$ or their latent features only. Empirically, it is a very effective solution in exploiting the collective information among users and items, even when the observed ratings in the (user, item) matrix are very few.

Some advanced models [3] assume that the rating $r_{ui}$ is related to not only the user $u$ and item $i$ but also the other rated items by user $u$ as a certain *context*, denoted as $\mathcal{I}_u\backslash\{i\}$. Similarly, the preference generalization probability can be represented as follows,

$$P(r_{ui}|(u, i); (u, i'), i' \in \mathcal{I}_u\backslash\{i\}), \tag{2}$$

where both $(u, i)$ and $(u, i'), i' \in \mathcal{I}_u\backslash\{i\}$ denote the factors that govern the generalization of the rating $r_{ui}$. The advantage of the conditional probability in Eq.(2) is its ability to allow users with similar rated item sets to have similar latent features in the learned model. However, the exact values of the ratings assigned by the user $u$ have not been exploited yet. Hence, we call the condition $(u, i'), i' \in \mathcal{I}_u\backslash\{i\}$ in Eq.(2) *oneclass* preference context (OPC) as inspired by the well-known one-class feedback in collaborative filtering.

In this paper, we go one step beyond and propose a fine-grained preference generalization probability,

$$P(r_{ui}|(u, i); (u, i', r_{ui'}), i' \in \cup_{r\in\mathbb{M}}\mathcal{I}_u^r\backslash\{i\}), \tag{3}$$

which includes the rating $r_{ui'}$ of each rated item by user $u$. This new probability is based on three parts, including (i) the (user, item) pair $(u, i)$ in Eq.(1), (ii) the examined items $\cup_{r\in\mathbb{M}}\mathcal{I}_u^r\backslash\{i\}$ in Eq.(2), and (iii) the categorical score $r_{ui'}$ of each rated item. We can see that the new probability is more sophisticated and captures more information when modeling the observed rating records.

The difference between the oneclass preference context $(u, i'), i' \in \mathcal{I}_u \backslash \{i\}$ in Eq.(2) and the condition $(u, i', r_{ui'}), i' \in \cup_{r \in \mathbb{M}} \mathcal{I}_u^r \backslash \{i\}$ in Eq.(3) is the categorical multiclass scores (or ratings), $r_{ui'}$, and thus we call it *multiclass* preference context (MPC). We can see that the MPC in Eq.(3) will be reduced to the OPC in Eq.(2) when we treat all ratings as a constant.

We illustrate those three types of preference generalization probability in Figure 1. In Figure 1, we can see that the probability of generating the rating $r_{23} = 3$ is dependent on different conditions in different models, i.e., (i) (user, item) pair $(2, 3)$ in MF, (ii) (user, item) pairs $(2, 3)$, $(2, 4)$ and $(2, 5)$ in MF-OPC, and (iii) (user, item) pair $(2, 3)$, and (user, item, rating) triples $(2, 4, 1)$ and $(2, 5, 4)$ in MF-MPC.

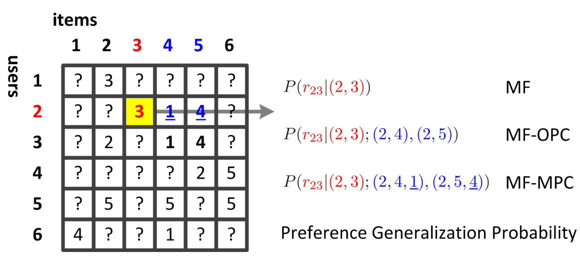



Figure 1: Illustration of preference generalization probabilities in different models, including matrix factorization (MF), MF with oneclass preference context (MF-OPC) and MF with multiclass preference context (MF-MPC). Note that MF-OPC is equivalent to SVD++ [3].

## 2.3 Matrix Factorization with MPC

For a basic matrix factorization model, the prediction rule of the rating assigned by user $u$ to item $i$ is defined as follows [6],

$$\hat{r}_{ui} = U_{u \cdot} V_{i \cdot}^T + b_u + b_i + \mu, \tag{4}$$

where $U_{u \cdot} \in \mathbb{R}^{1 \times d}$ and $V_{i \cdot} \in \mathbb{R}^{1 \times d}$ are the user-specific and item-specific latent feature vectors, respectively, and $b_u$, $b_i$ and $\mu$ are the user bias, the item bias and the global average, respectively.

For matrix factorization with oneclass preference context, we can define the prediction rule of a rating as follows [3],

$$\hat{r}_{ui} = U_{u \cdot} V_{i \cdot}^T + \bar{U}_{u \cdot}^{\text{OPC}} V_{i \cdot}^T + b_u + b_i + \mu, \tag{5}$$

where $\bar{U}_{u \cdot}^{\text{OPC}}$ is based on the corresponding oneclass preference context $\mathcal{I}_u \backslash \{i\}$ [3],

$$\bar{U}_{u \cdot}^{\text{OPC}} = \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} O_{i' \cdot}. \tag{6}$$

From the definition of $\bar{U}_{u \cdot}^{\text{OPC}}$ in Eq.(6), we can see that two users, $u$ and $u'$, with similar examined item sets, $\mathcal{I}_u$ and $\mathcal{I}_{u'}$, will have similar latent representations $\bar{U}_{u \cdot}^{\text{OPC}}$ and $\bar{U}_{u' \cdot}^{\text{OPC}}$. Hence, the prediction rule in Eq.(5) can be used to integrate certain neighborhood information.

In our matrix factorization with multiclass preference context, we propose a novel and generic prediction rule for the rating of user $u$ to item $i$,

$$\hat{r}_{ui} = U_{u \cdot} V_{i \cdot}^T + \bar{U}_{u \cdot}^{\text{MPC}} V_{i \cdot}^T + b_u + b_i + \mu, \tag{7}$$

where $\bar{U}_{u \cdot}^{\text{MPC}}$ is from the multiclass preference context,,

$$\bar{U}_{u \cdot}^{\text{MPC}} = \sum_{r \in \mathbb{M}} \frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u^r \backslash \{i\}} M_{i' \cdot}^r. \tag{8}$$

Notice that $\frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}}$ plays as a normalization term for the preference of class $r$. We can see that $\bar{U}_{u \cdot}^{\text{MPC}}$ in Eq.(8) is different from $\bar{U}_{u \cdot}^{\text{OPC}}$ in Eq.(6), because it contains more information, i.e., the fine-grained categorical preference of each rated item. And the virtual user profile $\bar{U}_{u \cdot}^{\text{MPC}}$ in Eq.(8) is more closely related to the similarity measurement in neighborhood-based methods such as PCC (Pearson correlation coefficient) [7], because both of them are defined on the fine-grained categorical scores. The neighborhood information in Eq.(8) is thus more accurate than that of Eq.(6), which is expected to generate better recommendation performance.

With the prediction rule in Eq.(7), we can learn the model parameters in the following minimization problem,

$$\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} [\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + reg(u, i)] \tag{9}$$

where $y_{ui} \in \{0, 1\}$ is an indicator variable denoting whether $(u, i, r_{ui})$ is in the set of rating records $\mathcal{R}$, $reg(u, i) = \frac{\lambda}{2}\|U_{u \cdot}\|^2 + \frac{\lambda}{2}\|V_{i \cdot}\|^2 + \frac{\lambda}{2}\|b_u\|^2 + \frac{\lambda}{2}\|b_i\|^2 + \frac{\lambda}{2}\sum_{r \in \mathbb{M}} \sum_{i' \in \mathcal{I}_u^r \backslash \{i\}} ||M_{i' \cdot}^r||_F^2$ is the regularization term used to avoid overfitting, and $\Theta = \{U_{u \cdot}, V_{i \cdot}, b_u, b_i, \mu, M_{i \cdot}^r\}$, $u = 1, 2 \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$ are model parameters to be learned. Note that the form of the objective function in Eq.(9) is exactly the same with that of the basic matrix factorization [6], because our improvement is reflected in the prediction rule for $\hat{r}_{ui}$.

Initialize model parameters $\Theta$
**for** $t = 1, \ldots, T$ **do**
  **for** $t_2 = 1, \ldots, |\mathcal{R}|$ **do**
    Randomly pick up a rating from $\mathcal{R}$
    Calculate the gradients via Eq.(10-15)
    Update the parameters via Eq.(16)
  **end for**
  Decrease the learning rate $\gamma \leftarrow \gamma \times 0.9$
**end for**

Figure 2: The algorithm of MF-MPC.

### 2.4 Learning MF-MPC

For a tentative objective function $\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + reg(u,i)$, we have the gradients of the model parameters,

$$\nabla U_{u\cdot} = -e_{ui}V_{i\cdot} + \lambda U_{u\cdot} \tag{10}$$
$$\nabla V_{i\cdot} = -e_{ui}(U_{u\cdot} + \bar{U}_{u\cdot}^{\text{MPC}}) + \lambda V_{i\cdot} \tag{11}$$
$$\nabla b_u = -e_{ui} + \lambda b_u \tag{12}$$
$$\nabla b_i = -e_{ui} + \lambda b_i \tag{13}$$
$$\nabla \mu = -e_{ui} \tag{14}$$
$$\nabla M_{i'\cdot}^r = \frac{-e_{ui}V_{i\cdot}}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}} + \lambda M_{i'\cdot}^r, i' \in \mathcal{I}_u^r \backslash \{i\}, r \in \mathbb{M} \tag{15}$$

where $e_{ui} = (r_{ui} - \hat{r}_{ui})$ is the difference between the true rating and the predicted rating.

Finally, we have the update rules,

$$\theta = \theta - \gamma \nabla \theta, \tag{16}$$

where $\gamma$ is the learning rate, and $\theta \in \Theta$ is a model parameter to be learned.

We describe the algorithm in Figure 2. The algorithm basically consists of the following steps. Firstly, it randomly samples a rating record from the training data. Secondly, it calculates the gradients via Eq.(10-15). Thirdly, it updates each model parameter via Eq.(16). The stochastic gradient descent algorithm framework is similar to that of traditional matrix factorization, while the major difference is from the prediction rule as shown in Eq.(7) and the corresponding gradients.

## 3 Experiments

In this section, we conduct empirical studies in order to verify whether the multiclass preference context is helpful in modeling user feedback.

### 3.1 Data Sets and Evaluation Metrics

We use three commonly used public data sets from the grouplens research lab[1], including MovieLens100K (i.e., ML100K), MovieLens1M (i.e., ML1M) and MovieLens10M (i.e., ML10M), which contain 100000 ratings by 943 users and 1682 items, 1000209 ratings by 6040 users and 3952 items, and 10000054 ratings by 71567 users and 10681 items, respectively. Note that the multiclass preference sets are $\mathbb{M} = \{1, 2, 3, 4, 5\}$ for ML100K and ML1M, and $\mathbb{M} = \{0.5, 1, 1.5, \ldots, 5\}$ for ML10M. In the experiments, we use five-fold cross validation. Specifically, for each data set, we first divide it into five parts with equal size. Then, we take one part as test data and the remaining four parts as training data, which is repeated for five times so that we have five copies of training data and test data for each of the three data sets. The averaged rating prediction performance on those five copies of test data will be reported.

We adopt two commonly used evaluation metrics for collaborative recommendation tasks, including mean absolute error (MAE) and root mean square error (RMSE).

### 3.2 Baselines and Parameter Settings

For empirical studies, we compare our MF-MPC with the baselines from neighborhood-based approaches to latent factor methods,

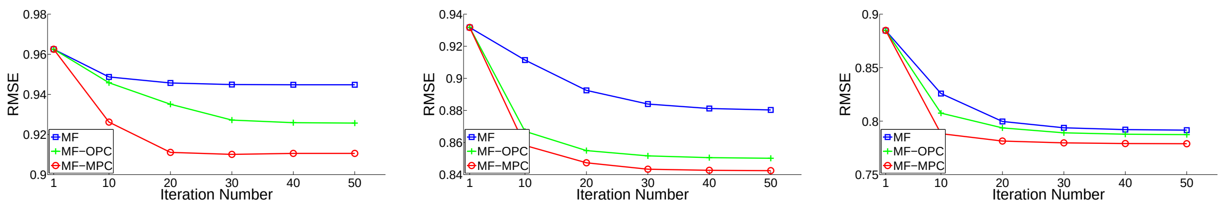
- AF (average filling): we use the average rating of each user as calculated from the training data $\mathcal{R}$ to predict each rating in the test data;
- CF (collaborative filtering): we implement a user-oriented neighborhood-based collaborative filtering method using PCC (Pearson correlation coefficient) [7] as the similarity measurement;
- MF (matrix factorization): we use the basic latent factor model, i.e., matrix factorization without preference context [6] as shown in Eq.(4), as a major baseline; and
- MF-OPC (matrix factorization with oneclass preference context): for direct comparative studies between MPC and OPC, we also use MF-OPC as shown in Eq.(5). Note that MF-OPC is the same with SVD++ [3].

For the parameter settings of factorization-based methods, we follow [5]. Specifically, (i) we fix the learning rate $\gamma = 0.01$, the number of latent dimensions $d = 20$, and the iteration number $T = 50$; (ii) we search the best value of the tradeoff parameter $\lambda$ from $\{0.001, 0.01, 0.1\}$ using the first

[1] http://grouplens.org/datasets/movielens/

Table 2: Recommendation performance of AF, CF, MF, MF-OPC and MF-MPC on MovieLens100K, MovieLens1M and MovieLens10M. Note that MF-OPC is equivalent to SVD++ [3]. The significantly best results are marked in bold ($p < 0.01$). We also include the searched best value of the tradeoff parameter $\lambda$ for each method and data set for easy reproducibility of the experimental results.

| Data | Metric | AF | CF | MF | MF-OPC/SVD++ | MF-MPC |
|---|---|---|---|---|---|---|
| ML100K | MAE | 0.8348$\pm_{0.0025}$ | 0.7576$\pm_{0.0028}$ | 0.7478$\pm_{0.0032}$ | 0.7266$\pm_{0.0032}$ | **0.7092**$\pm_{0.0032}$ |
| | RMSE | 1.0417$\pm_{0.0019}$ | 0.9637$\pm_{0.0039}$ | 0.9448$\pm_{0.0030}$ | 0.9253$\pm_{0.0032}$ | **0.9091**$\pm_{0.0026}$ |
| | | | | ($\lambda = 0.1$) | ($\lambda = 0.001$) | ($\lambda = 0.001$) |
| ML1M | MAE | 0.8289$\pm_{0.0020}$ | 0.7560$\pm_{0.0020}$ | 0.6956$\pm_{0.0021}$ | 0.6655$\pm_{0.0014}$ | **0.6596**$\pm_{0.0017}$ |
| | RMSE | 1.0355$\pm_{0.0024}$ | 0.9531$\pm_{0.0024}$ | 0.8832$\pm_{0.0023}$ | 0.8511$\pm_{0.0017}$ | **0.8439**$\pm_{0.0018}$ |
| | | | | ($\lambda = 0.001$) | ($\lambda = 0.001$) | ($\lambda = 0.01$) |
| ML10M | MAE | 0.7688$\pm_{0.0007}$ | 0.7138$\pm_{0.0003}$ | 0.6068$\pm_{0.0006}$ | 0.6028$\pm_{0.0003}$ | **0.5947**$\pm_{0.0003}$ |
| | RMSE | 0.9784$\pm_{0.0008}$ | 0.9148$\pm_{0.0005}$ | 0.7911$\pm_{0.0008}$ | 0.7870$\pm_{0.0006}$ | **0.7783**$\pm_{0.0005}$ |
| | | | | ($\lambda = 0.01$) | ($\lambda = 0.01$) | ($\lambda = 0.01$) |




MovieLens100K MovieLens1M MovieLens10M

Figure 3: Recommendation performance of MF, MF-OPC (i.e., SVD++ [3]) and MF-MPC on the first copy of MovieLens100K, MovieLens1M and MovieLens10M with different iteration numbers.

copy of each data and the RMSE metric; (iii) we initialize the user bias $b_u$, the item bias $b_i$ and the global average $\mu$ using the statistics of the training data $\mathcal{R}$; (iv) we initialize the value of each latent feature $U_{uk}$, $V_{ik}$ and $M^r_{i'k}$ with a small random value. For the number of neighbors in CF, we set it to be the same with the value of $d$, i.e., 20. We also use different dimensions for MF and MF-OPC in order to study the effectiveness of our MF-MPC from the perspective of the number of model parameters.

### 3.3 Results

We report the rating prediction performance of our MF-MPC and the compared baselines on those three data sets in Table 2. We can have the following observations:

- MF-MPC performs significantly better than all baselines on all three data sets, which clearly shows the effectiveness of our proposed multiclass preference context and the way we integrate it into the factorization framework;
- MF-OPC or SVD++ is much better than MF, which shows the usefulness of the exploited preference context, and the complementarity of factorization-based methods and neighborhood-based methods; and
- factorization-based methods are better than the neighborhood-based method and the basic average filling method, which is consistent with previous reported results that latent factor models are usually very competitive.

We also study the convergence property of each factorization-based method, which is shown in Figure 3. Note that the results on MAE is very similar and is not included due to length limitation. From Figure 3, we can see:

- all three factorization-based methods converge smoothly before or around 50 iterations, which shows that they can converge within in a reasonable number of iterations; and
- the overall performance ordering, i.e., MF<MF-OPC<MF-MPC, is the same with that of Table 2, which again shows the helpfulness of exploiting the

Table 3: Recommendation performance on RMSE of M-F, MF-OPC (i.e., SVD++) and MF-MPC using comparable model size.

| Data | MF | MF-OPC | MF-MPC |
|---|---|---|---|
| ML100K | $0.9439{\pm}_{0.0036}$ ($\lambda = 0.001$) | $0.9209{\pm}_{0.0034}$ ($\lambda = 0.001$) | $\mathbf{0.9091}{\pm}_{0.0026}$ ($\lambda = 0.001$) |
| ML1M | $0.8719{\pm}_{0.0023}$ ($\lambda = 0.001$) | $0.8477{\pm}_{0.0017}$ ($\lambda = 0.001$) | $\mathbf{0.8439}{\pm}_{0.0018}$ ($\lambda = 0.01$) |
| ML10M | $0.7821{\pm}_{0.0005}$ ($\lambda = 0.01$) | $0.7810{\pm}_{0.0005}$ ($\lambda = 0.01$) | $\mathbf{0.7783}{\pm}_{0.0005}$ ($\lambda = 0.01$) |

oneclass preference context and multiclass preference context in MF-OPC and MF-MPC, respectively, and multiclass preference context indeed contain more fine-grained information.

We further study whether the improvement of our MF-MPC is from more model parameters. Specifically, we fix $d = 20$ for MF-MPC, and set $d = 120$ for MF and $d = 80$ for MF-OPC (i.e., SVD++), where MF-MPC does not have any advantage of using more model parameters for all three data sets. We report the performance on RMSE in Table 3 (the results on MAE is similar), and find that MF-MPC is again significantly better ($p < 0.01$) than MF and MF-OPC, which clearly shows the advantage of the proposed factorization model.

## 4 Related Work

Matrix factorization has been well recognized as an effective and efficient solution to various recommendation problems. However, most factorization-based methods focus on each observed explicit or implicit feedback and do not exploit the available preference context such as other examined or rated items of a target user. In this section, we discuss some closely related works that combine matrix factorization and preference context. Specifically, we categorize the related works into two branches from the perspective of preference context, including oneclass preference context and multiclass preference context.

### 4.1 Oneclass Preference Context

There are different types of oneclass preferences such as transactions and examinations, which are usually called explicit oneclass feedback and implicit oneclass feedback, respectively. Hence, for oneclass preference context, we have two subtypes of preference context, including explicit oneclass preference context and implicit oneclass preference context.

Table 4: Summary of some related works. Note that SVD++ [3] is equivalent to MF-OPC (matrix factorization with oneclass preference context).

| Recommendation task | Preference context: Oneclass | Preference context: Multiclass |
|---|---|---|
| Item recommendation | FISM [2], etc. | |
| Rating prediction | SVD++ [3], etc. | **MF-MPC (proposed)** |

For explicit oneclass preference context, the most representative work is probably FISM (factored item similarity model) [2]. The prediction rule of FISM is as follows,

$$\hat{r}_{ui} = \bar{U}_{u\cdot}^{\text{OPC}} V_{i\cdot}^T + b_u + b_i, \tag{17}$$

which is very similar to the prediction rule of MP-OPC as shown in Eq.(5), except the terms $U_{u\cdot} V_{i\cdot}^T$ and $\mu$. With the prediction rule in Eq.(17), the model parameters can then be learned in a pointwise or pairwise way. For implicit oneclass preference context, the most famous work is SVD++ [3], where the main idea is to constrain two users with similar examination behaviors (e.g., similar rated item sets) to have similar latent features. From the perspective of recommendation tasks, FISM [2] is for item recommendation while SVD++ [3] is for rating prediction. Hence, SVD++ (i.e., MF-OPC) is more close to our MF-MPC.

### 4.2 Multiclass Preference Context

As far as we know, there are no previous works that exploit the multiclass preference context in a factorization framework as ours. Our MF-MPC is the first work that generalizes the oneclass preference context by digesting the rating behaviors or categorical feedback in a fine-grained manner. For a recommendation scenario with categorical preferences such as graded ratings or signed links [8], we can always make use of the proposed multiclass preference context via revising the prediction rule in a similar way to that of Eq.(7). From the perspective of recommendation task, our MF-MPC is for rating prediction. Note that our multiclass preference context is not limited to missing value prediction, because the enhanced prediction rule can also be embedded in a ranking-oriented loss function for item recommendation.

We put our MF-MPC and the above discussed representative works in Table 4, from which we can see that our MF-MPC is a novel solution for rating prediction via exploiting multiclass preference context.

## 5 Conclusions and Future Work

In this paper, we propose a novel factorization-based method for rating prediction and matrix completion. Specifically, we integrate multiclass preference context (MPC) into the matrix factorization framework and achieve significantly better recommendation performance than the state-of-the-art methods.

For future works, we are interested in generalizing the idea of multiclass preference context to recommendation with categorical preference information in cross-domain scenarios [4, 9]. We are also interested in designing some advanced sampling strategy instead of the random sampling approach in the learning algorithm.

## Acknowledgments

We thank the handling editor and reviewers for their expert comments and constructive suggestions, and Yuchao Duan for some preliminary empirical studies and helpful discussions. We thank the support of National Natural Science Foundation of China No. 61502307 and No. 61672358, and Natural Science Foundation of Guangdong Province No. 2014A030310268 and No. 2016A030313038. Zhong Ming is the corresponding author for this work.

## References

[1] Zan Huang, Daniel Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5):68–78, September 2007.

[2] Santosh Kabbur, Xia Ning, and George Karypis. Fism: Factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 659–667, 2013.

[3] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, 2008.

[4] Weike Pan. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, 177:447–453, 2016.

[5] Weike Pan and Zhong Ming. Interaction-rich transfer learning for collaborative filtering with heterogeneous user feedback. *IEEE Intelligent Systems*, 29(6):48–54, 2014.

[6] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology*, 3(3):57:1–57:22, May 2012.

[7] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, 1994.

[8] Jiliang Tang, Charu Aggarwal, and Huan Liu. Recommendation with signed social networks. In *Proceedings of the 25th International World Wide Web Conference*, WWW '16, 2016.

[9] Yu Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data*, 1(1):16–33, 2015.

[10] Xiangmin Zhou, Lei Chen, Yanchun Zhang, Longbing Cao, Guangyan Huang, and Chen Wang. Online video recommendation in sharing community. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1645–1656, 2015.

## The Authors

**Weike Pan** is a lecturer in the College of Computer Science and Software Engineering at Shenzhen University, China. His research interests include transfer learning, intelligent recommendation, and machine learning. He received a PhD in computer science and engineering from the Hong Kong University of Science and Technology. Contact him at panweike@szu.edu.cn.

**Zhong Ming**, corresponding author, is a professor in the College of Computer Science and Software Engineering at Shenzhen University, China. His research interests include software engineering and Web intelligence. Ming received a PhD in computer science and technology from Sun Yat-Sen University, China. Contact him at mingz@szu.edu.cn.