MF-DMPC: Matrix Factorization with Dual Multiclass Preference Context for Rating Prediction

Jing Lin[1,2], Weike Pan[1,2](✉), and Zhong Ming[1,2](✉)

[1] National Engineering Laboratory for Big Data System Computing Technology,
Shenzhen University, Shenzhen, China
linjing4@email.szu.edu.cn, {panweike,mingz}@szu.edu.cn
[2] College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen, China

Abstract. Thanks to the development of big data, more and more context are available in web services. In this paper, we focus on recommendation with internal context (antithetical to external context such as social connections), which refers to ratings assigned by users to items only. Inspired by a very recent work that embeds the item neighborhood information represented by multiclass preference context (or simply "MPC") into a model-based method, we further extend MPC to both user-based MPC for user neighborhood and item-based MPC for item neighborhood. In our new model termed matrix factorization with dual multiclass preference context (MF-DMPC), both user-based and item-based MPC are encoded in a matrix factorization framework. By studying the effectiveness of user-based MPC, item-based MPC, and our dual MPC through experiments on three public data sets, we find that our proposed model with dual MPC performs the best in accuracy. As a matter of fact, our model successfully strikes a balance between user-based and item-based neighborhood information, i.e., it exploits the complementarity well.

Keywords: Dual multiclass preference context · Matrix factorization
Collaborative filtering

1 Introduction

Recommendation technology [1], which is designed to help users pick out the items they are interested in from a wide range of items, is an important tool for solving the information overload problem and therefore attracts a great deal of attention from both the academia and business communities. Actually, it is of great practical application value to many web services scenarios such as e-commerce, advertisement, multimedia (including movie, music, etc.), and even social networking.

It is natural to imagine that the type of information (also called "context" in a broad sense) introduced in the recommendation system has an important impact

H. Jin et al. (Eds.): ICWS 2018, LNCS 10966, pp. 337–349, 2018.
https://doi.org/10.1007/978-3-319-94289-6_22

on the quality of recommendation service, with different type of context having its own advantages. In this paper, we divide context in a recommendation system into two categories, including external context and internal context, which is illustrated in Fig. 1. External context includes physical context (e.g., time, location) [2], social context (e.g., trust, companion, festival) [14] and so on, while internal context mainly refers to user's direct preference in the form of complex latent pattern hidden in the known rating matrix [7,9]. In recent years, more and more researchers are paying attention to context-aware recommendation systems [4] that make predictions through the introduction of various external context, believing that introducing appropriate external context can effectively improve the recommendation performance. However, it may cause the problem of model inflexibility and computational burden. Furthermore, external context may not be available for many systems due to design flaws in early versions, so algorithms that only make use of internal context still occupy an important place in the research community.

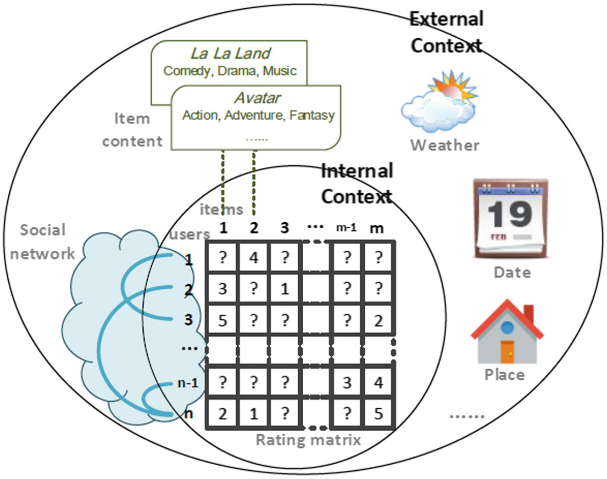


Fig. 1. Illustration of two categories of context in recommendation systems.

Different from context-aware recommendation systems, traditional recommendation systems (typically collaborative filtering recommendation systems [4, 7,9,10]) only consider internal context, that is, ratings assigned by users to items. A newly proposed model called matrix factorization with multiclass preference context (MF-MPC) [9] is a unified method which combines the two major categories of collaborative filtering – neighborhood-based [4] and model-based [7,10]. Briefly, MF-MPC is an improved method of SVD (a kind of matrix factorization method) [10] by adding a matrix transformed from the multiclass preference context of a certain user, which represents the user similarities in a neighborhood-based method. In this paper, we further introduce a matrix factorization model that combines not only user similarities but also item similarities, which is thus called MF with dual MPC (MF-DMPC for short). We also study the effect of user similarities and item similarities separately. Experimental results show that our MF-DMPC performs better than MF-MPC [9], and the degree of performance improvement is probably effected by the ratio of the user group size to the item group size, as well as the density of the rating matrix. MF-DMPC inherits not only high accuracy of model-based recommendation algorithms, but also good explainability of neighborhood-based algorithms.

2 Preliminaries

2.1 Problem Definition

In this paper, we study the problem of making good use of internal context in recommendation systems, which means that we will only need an incomplete rating matrix represented by $\mathcal{R} = \{(u, i, r_{ui})\}$ for our prediction task, where u

Table 1. Some notations and explanations.

Symbol	Meaning
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 to item i
$\mathcal{R} = \{(u, i, r_{ui})\}$	Rating records of training data
$y_{ui} \in \{0, 1\}$	Indicator, $y_{ui} = 1$ if $(u, i, r_{ui}) \in \mathcal{R}$ and $y_{ui} = 0$ otherwise
$\mathcal{I}_u^r, r \in \mathbb{M}$	Items rated by user u with rating r
$\mathcal{I}_u$	Items rated by user u
$\mathcal{U}_i^r, r \in \mathbb{M}$	Users who rate item i with rating r
$\mathcal{U}_i$	Users who rate item i
$\mu \in \mathbb{R}$	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
$U_{u\cdot}, N_{u\cdot}^r \in \mathbb{R}^{1\times d}$	User-specific latent feature vector
$V_{i\cdot}, M_{i\cdot}^r \in \mathbb{R}^{1\times d}$	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 to item i
T	Iteration number in the algorithm

represents one of the ID numbers of n users (or rows), i represents one of the ID numbers of m items (or columns), and r_{ui} is the recorded rating of user u to item i ($r_{ui} \in \mathbb{M}$, $\mathbb{M}$ can be $\{1, 2, 3, 4, 5\}$, $\{0.5, 1, 1.5, \ldots, 5\}$ or other ranges). As a result, we build an improved model based on MF-MPC [9] to estimate the missing entries of the rating matrix. In other words, we are studying a problem called rating prediction. Some notations used to establish our model are shown in Table 1.

2.2 Multiclass Preference Context

In the state-of-the-art matrix factorization based model – SVD model [10], the prediction rule for the rating of user u to item i is as follows,

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + b_u + b_i + \mu, \tag{1}$$

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 μ are the user bias, the item bias and the global average, respectively.

In the MF-MPC model [9], the rating of user u to item i, i.e., r_{ui}, can be represented in a probabilistic way as follows:

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

which means that r_{ui} is dependent on not only the (user, item) pair (u, i), but also the examined items $i' \in \mathcal{I}_u \backslash \{i\}$ and the categorical score $r_{ui'} \in \mathbb{M}$ of each item. Notice that multiclass preference context (MPC) refers to the condition $(u, i', r_{ui'}), i' \in \cup_{r\in\mathbb{M}} \mathcal{I}_u^r \backslash \{i\}$.

In order to introduce multiclass preference context into matrix factorization based model, we need a user-specific latent preference vector $\bar{U}_{u\cdot}^{MPC}$ for user u from the multiclass preference context [9],

$$\bar{U}_{u\cdot}^{MPC} = \sum_{r\in\mathbb{M}} \frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}} \sum_{i'\in\mathcal{I}_u^r \backslash \{i\}} M_{i'\cdot}^r. \tag{3}$$

Notice that $M_{i\cdot}^r \in \mathbb{R}^{1\times d}$ is a classified item-specific latent feature vector and $\frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}}$ plays as a normalization term for the preference of class r.

If we multiply the MPC expressions of two users u and u' ($\bar{U}_{u\cdot}^{MPC}$ and $\bar{U}_{u'\cdot}^{MPC}$) together, and set $\mathbb{M} = \{1\}$ specially, we will have

$$\langle \bar{U}_{u\cdot}^{MPC}, \bar{U}_{u'\cdot}^{MPC} \rangle \approx \frac{\sum_{i'\in\mathcal{I}_u} M_{i'\cdot}^r \cdot \sum_{i'\in\mathcal{I}_{u'}} M_{i'\cdot}^r}{\sqrt{|\mathcal{I}_u||\mathcal{I}_{u'}|}}, \tag{4}$$

which is quite similar to the cosine-based similarity [11] – a measure of the similarity between two users shown as follows:

$$sim(u, u') = cos(\boldsymbol{R}_{u,*}, \boldsymbol{R}_{u',*}) = \frac{\boldsymbol{R}_{u,*} \cdot \boldsymbol{R}_{u',*}}{\|\boldsymbol{R}_{u,*}\|_2 \, \|\boldsymbol{R}_{u',*}\|_2}, \tag{5}$$

where $\boldsymbol{R}$ is the $n \times m$ user-item matrix. Therefore we believe that multiclass preference context can represent user similarities.

By adding the neighborhood information $\bar{U}_{u\cdot}^{MPC}$ to SVD model, we can get the MF-MPC prediction rule for the rating of user u to item i [9]:

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + \bar{U}_{u\cdot}^{MPC} V_{i\cdot}^T + b_u + b_i + \mu, \tag{6}$$

where $U_{u\cdot}$ and $V_{i\cdot}$, b_u, b_i and μ are exactly the same with that of the SVD model. MF-MPC is proved to generate better recommendation performance than SVD [10] and SVD++ [7], and also contains them as particular cases.

3 Matrix Factorization with Dual Multiclass Preference Context

Inspired by the differences between user-based and item-based collaborative filtering [4], we can infer that item similarities (item-based multiclass preference context) can also be introduced to improve the performance of matrix factorization model. Furthermore, thanks to the extendibility of MF model, we can hopefully introduce both user-based and item-based MPC into the prediction rule so as to obtain an improved model – matrix factorization with dual multiclass preference context (MF-DMPC). The derivation process can be found in Fig. 2.

3.1 Dual Multiclass Preference Context

Now we call $\bar{U}_{u\cdot}^{MPC}$ (mentioned in previous section) user-based multiclass preference context (user-based MPC). Before defining dual multiclass preference context (DMPC), we should first define item-based multiclass preference context (item-based MPC) $\bar{V}_{i\cdot}^{MPC}$ to represent item similarities. Symmetrically, we have

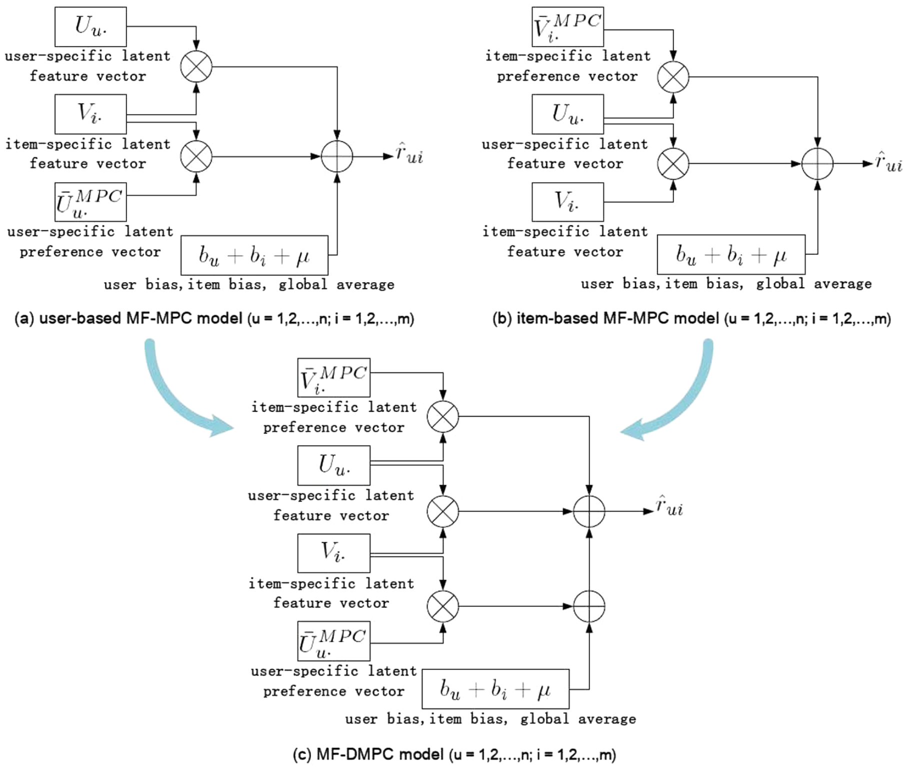
$$\bar{V}_{i\cdot}^{MPC} = \sum_{r \in \mathbb{M}} \frac{1}{\sqrt{|\mathcal{U}_i^r \backslash \{u\}|}} \sum_{u' \in \mathcal{U}_i^r \backslash \{u\}} N_{u'\cdot}^r, \tag{7}$$

where $N_{u\cdot}^r \in \mathbb{R}^{1 \times d}$ is a classified user-specific latent feature vector. So now we have the item-based MF-MPC prediction rule,

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + \bar{V}_{i\cdot}^{MPC} U_{u\cdot}^T + b_u + b_i + \mu. \tag{8}$$

When choosing between the implementation of a user-based and an item-based neighborhood recommender system, we often consider criteria such as accuracy, efficiency, and so on. However these properties always depend on the ratio between the number of users and items in the system.

The good news is that through the advanced matrix factorization method, now we can introduce both user-based and item-base neighborhood information into our model by keeping both $\bar{U}_{u\cdot}^{MPC}$ and $\bar{V}_{i\cdot}^{MPC}$ in the model, collectively called dual multiclass preference context (DMPC).


$U_{u\cdot}$
user-specific latent feature vector
$V_{i\cdot}$
item-specific latent feature vector
$\bar{U}_{u\cdot}^{MPC}$
user-specific latent preference vector
$b_u + b_i + \mu$
user bias, item bias, global average
$\hat{r}_{ui}$
$\bar{V}_{i\cdot}^{MPC}$
item-specific latent preference vector
$U_{u\cdot}$
user-specific latent feature vector
$V_{i\cdot}$
item-specific latent feature vector
$b_u + b_i + \mu$
user bias, item bias, global average
$\hat{r}_{ui}$
$\bar{V}_{i\cdot}^{MPC}$
item-specific latent preference vector
$U_{u\cdot}$
user-specific latent feature vector
$V_{i\cdot}$
item-specific latent feature vector
$\bar{U}_{u\cdot}^{MPC}$
user-specific latent preference vector
$b_u + b_i + \mu$
user bias, item bias, global average
$\hat{r}_{ui}$

(a) user-based MF-MPC model (u = 1,2,...,n; i = 1,2,...,m)

(b) item-based MF-MPC model (u = 1,2,...,n; i = 1,2,...,m)

(c) MF-DMPC model (u = 1,2,...,n; i = 1,2,...,m)

Fig. 2. Illustration of user-based MF-MPC, item-based MF-MPC and our MF-DMPC.

3.2 Prediction Rule and Optimization Problem

For matrix factorization with dual multiclass preference context, the prediction rule for the rating of user u to item i is defined as follows,

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + \bar{U}_{u\cdot}^{MPC} V_{i\cdot}^T + \bar{V}_{i\cdot}^{MPC} U_{u\cdot}^T + b_u + b_i + \mu, \tag{9}$$

with all notations described above. Finally, we call our new model MF-DMPC in short.

With the prediction rule, we can learn the model parameters in the following minimization problem,

$$\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} [\frac{1}{2}(r_{ui} - \hat{r}_{ri})^2 + reg(u,i)], \tag{10}$$

where $reg(u,i) = \frac{\alpha_m}{2} \sum_{r \in \mathbb{M}} \sum_{i' \in \mathcal{I}_u^r \setminus \{i\}} ||M_{i'}^r||_F^2 + \frac{\alpha_n}{2} \sum_{r \in \mathbb{M}} \sum_{u' \in \mathcal{U}_i^r \setminus \{u\}} ||N_{u'}^r||_F^2$ $+ \frac{\alpha_u}{2} ||U_{u\cdot}||^2 + \frac{\alpha_v}{2} ||V_{i\cdot}||^2 + \frac{\beta_u}{2} ||b_{u\cdot}||^2 + \frac{\beta_v}{2} ||b_{i\cdot}||^2$ is the regularization term used to avoid overfitting, and $\Theta = \{U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu, M_{i\cdot}^r \mu, N_{u\cdot}^r\}$, $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$. Notice that the objective function of MF-DMPC is quite similar to that of MF-MPC. The difference lies in the "dual" MPC, i.e., $\bar{V}_{i\cdot}^{MPC} U_{u\cdot}^T$ in the prediction rule, and $\frac{\alpha_n}{2} \sum_{r \in \mathbb{M}} \sum_{u' \in \mathcal{U}_i^r \backslash \{u\}} ||N_{u'}^r||_F^2$ in the regularization term.

3.3 Algorithm

Using the stochastic gradient descent (SGD) algorithm, we have the gradients of the model parameters for a randomly sampled rating record (u, i, r_{ui}),

$$\nabla U_{u\cdot} = -e_{ui}(V_{i\cdot} + \bar{V}_{i\cdot}^{MPC}) + \alpha_u U_{u\cdot} \quad (11)$$

$$\nabla V_{i\cdot} = -e_{ui}(U_{u\cdot} + \bar{U}_{u\cdot}^{MPC}) + \alpha_v V_{i\cdot} \quad (12)$$

$$\nabla b_u = -e_{ui} + \beta_u b_u \quad (13)$$

$$\nabla b_i = -e_{ui} + \beta_v b_i \quad (14)$$

$$\nabla \mu = -e_{ui} \quad (15)$$

$$\nabla M_{i'\cdot}^r = -\frac{-e_{ui} V_{i\cdot}}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}} + \alpha_m M_{i'\cdot}^r, i' \in \mathcal{I}_u^r \backslash \{i\}, r \in \mathbb{M}. \quad (16)$$

$$\nabla N_{u'\cdot}^r = -\frac{-e_{ui} U_{u\cdot}}{\sqrt{|\mathcal{U}_i^r \backslash \{u\}|}} + \alpha_n N_{u'\cdot}^r, u' \in \mathcal{U}_i^r \backslash \{u\}, r \in \mathbb{M}. \quad (17)$$

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

And we have the update rules,

$$\theta = \theta - \gamma \nabla \theta \quad (18)$$

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

The algorithm of MF-DMPC (see Fig. 3) consists of three major steps. Firstly, we randomly pick out a rating record sample from the training data. Secondly, we calculate the gradients via Eqs. (11–17). Thirdly, we update each model parameter via Eq. (18). The major difference between the algorithm of MF-DMPC and that of MF-MPC [9] lies in the prediction rule as shown in Eq. (9) and the corresponding gradients.

Judging from algorithmic efficiency, the time complexity of MF-MPC [9] and SVD++ [7] and the proposed MF-DMPC is MF-DMPC > MF-MPC > SVD++, mainly because of the traversal during calculating $\bar{U}_{u\cdot}^{MPC}$ and $\bar{V}_{i\cdot}^{MPC}$ in MF-DMPC, $\bar{U}_{u\cdot}^{MPC}$ in MF-MPC, and $\bar{U}_{u\cdot}^{OPC}$ (oneclass preference context defined in [9]) in SVD++. As for space complexity, we can reckon from the size of dominating model parameters vectors shown in Table 2. In general, our MF-DMPC consume more time and memory than the closely related algorithms.

4 Experiments

In this section, we wonder what benefit does MF-DMPC bring with more resource consumption. We expect that it may lead to improvement of the accuracy, which is probably related to the effects of different kinds of MPC (including user-based MPC, item-based MPC and dual MPC).

```
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.(11 - 17)
        Update the parameters via Eq.(18)
    end for
    Decrease the learning rate $\gamma \longleftarrow \gamma \times 0.9$
end for
```

Fig. 3. The algorithm of MF-DMPC.

Table 2. The size of dominating model parameters vectors in different models.

Model	Dominating model parameters vectors	size ($\times\, d$)
SVD	$U_{u\cdot}, V_{i\cdot}$	$n + m$
SVD++	$U_{u\cdot}, V_{i\cdot}, M_{i\cdot}$	$n + m + n$
MF-MPC	$U_{u\cdot}, V_{i\cdot}, M^r_{i\cdot}$	$n + m + n\lvert\mathbb{M}\rvert$
MF-DMPC	$U_{u\cdot}, V_{i\cdot}, M^r_{i\cdot}, N^r_{u\cdot}$	$n + m + n\lvert\mathbb{M}\rvert + m\lvert\mathbb{M}\rvert$

4.1 Data Sets and Evaluation Metrics

For convenience, we choose the same data sets used in previous research about MF-MPC [9]. They are three public data sets from the Grouplens research lab, including MovieLens100K (ML100K), MovieLens1M (ML1M) and MovieLens10M (ML10M). Table 3 shows some information about them. Notice that now the ratio of user group size to item group size and the density of the rating matrix should be important factors to analyze the results of the experiments due to different characteristics of different neighborhood-based algorithms. We use five-fold cross validation in the empirical studies.

Table 3. Statistics of the data sets used in the experiments.

Data set	User number	Item number	Record number	Density of training set	n/m
ML100K	943	1,682	100,000	5.04%	0.56
ML1M	6,040	3,952	1,000,209	3.35%	1.53
ML10M	71,567	10,681	10,000,054	1.05%	6.70

We adopt mean absolute error (MAE) and root mean square error (RMSE) as evaluation metrics:

$$MAE = \sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} |r_{ui} - \hat{r}_{ui}| / |\mathcal{R}^{te}|$$

$$RMSE = \sqrt{\sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} (r_{ui} - \hat{r}_{ui})^2 / |\mathcal{R}^{te}|}$$

4.2 Baselines and Parameter Settings

In order to find out the effects of introducing different kinds of MPC into matrix factorization (MF) model, we compare the performance of SVD (see Eq. (1)) against that achieved by matrix factorization with user-based MPC (see Eq. (6)), item-based MPC (see Eq. (8)) and dual MPC (see Eq. (9)).

We configure the parameter settings of factorization-based methods as follows:

- For the learning rate γ, we set it to a commonly used default value, that is $\gamma = 0.01$.
- For the number of latent dimensions d, it is enough to show the advantages of introducing MPC when $d = 20$ (according to [9]).
- We set the iteration number $T = 50$, where results have reached steady state.
- The tradeoff parameters are searched through experiment using the first copy of each data and the RMSE metric, and follow the following conditions: $\alpha_u = \alpha_v = \beta_u = \beta_v = \alpha$, $\alpha \in \{0.001, 0.01, 0.1\}$; for user-based MF-MPC, $\alpha_m = \alpha$; for item-based MF-MPC, $\alpha_n = \alpha$; for dual MF-MPC, $\alpha_m, \alpha_n \in \{0.001, 0.01, 0.1\}$.

4.3 Results

The experimental results are shown in Table 4. Notice that the tradeoff parameters shown in the table are the searched best value for each method.

From the results in Table 4, we can have the following observations:

- The accuracy of factorization framework greatly improve when introducing multiclass preference context;
- Among all kinds of MPC, dual MPC contributes the most to the achievement of minimizing prediction error;
- In the MovieLens datasets, whether user-based or item-based MPC is more helpful depends on the ratio of user group size to item group size (n/m). Normally, item-based MF-MPC performs better when n/m is of a suitable size. As n/m getting larger, user-based MPC becomes even more important (may be affected by additional factors such as the density of rating matrix); and

Table 4. Recommendation performance of our MF-DMPC and other baseline methods on three MovieLens data sets.

Data	Method	MAE	RMSE	Parameter $(\alpha, \alpha_m, \alpha_n)$
ML100K	SVD	0.7446 ± 0.0033	0.9445 ± 0.0035	(0.01, N/A, N/A)
	User-based MF-MPC	0.7123 ± 0.0028	0.9102 ± 0.0029	(0.01, 0.01, N/A)
	Item-based MF-MPC	0.7038 ± 0.0021	0.9008 ± 0.0025	(0.01, N/A, 0.01)
	MF-DMPC	**0.7011** ± 0.0025	**0.8991** ± 0.0024	(0.01, 0.01, 0.01)
ML1M	SVD	0.7017 ± 0.0016	0.8899 ± 0.0023	(0.01, N/A, N/A)
	User-based MF-MPC	0.6613 ± 0.0015	0.8465 ± 0.0017	(0.01, 0.01, N/A)
	Item-based MF-MPC	0.6587 ± 0.0009	0.8439 ± 0.0013	(0.01, N/A, 0.01)
	MF-DMPC	**0.6564** ± 0.0016	**0.8434** ± 0.0017	(0.01, 0.01, 0.01)
ML10M	SVD	0.6067 ± 0.0007	0.7913 ± 0.0009	(0.01, N/A, N/A)
	User-based MF-MPC	0.5965 ± 0.0006	0.78135 ± 0.0007	(0.01, 0.01, N/A)
	Item-based MF-MPC	0.6024 ± 0.0006	0.7900 ± 0.0008	(0.01, N/A, 0.01)
	MF-DMPC	**0.5955** ± 0.0005	**0.78133** ± 0.0007	(0.01, 0.001, 0.1)

- The performance of MF-DMPC is in a way restrained by the better result between user-based and item-based MF-MPC – just slightly better than the better result. The improvement shows that MF-DMPC strikes a good balance between user-based MPC and item-based MPC.

Last but not the least, MF-DMPC not only inherits high accuracy of model-based algorithm, but also inherits good explainability of neighborhood-based algorithm.

5 Related Work

5.1 Recommendation with Internal Context

Recommendation with internal context refers primarily to collaborative filtering, which consider only rating data given by users. Collaborative filtering (or simply "CF") methods can be broadly separated into two categories: neighborhood-based and model-based (see Fig. 4). Neighborhood-based CF, which is based on the assumption that users with similar interests have similar preferences for an item, makes predictions by calculating user similarities (represented by user-based CF method) or item similarities (represented by item-based CF method) [11]. Model-based CF (represented by SVD methods) can obtain latent feature vectors of users and items through matrix factorization model, and further obtain predictive scores. SVD++ method proposed by Koren [7], which is regarded as attempt to combine neighborhood-based and model-based methods by adding a latent feature vector (which is actually a latent form of user similarities), gets a better result. After analyzing the merits and demerits of SVD++, Pan et al. proposed the MF-MPC method [9]. Instead of ignoring categorical scores, this method makes full use of multiclass preference context and treats SVD++ as an exception, so as to achieve better results. As an upgraded method of MF-MPC, our MF-DMPC model is also a typical example of using internal context. There are also some different views in modeling the rating scores such as categorical, numerical and ordinal [6], from which we may derive some rich preference context.

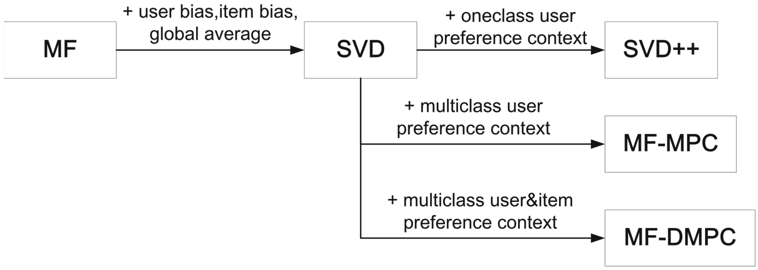

MF
+ user bias,item bias, global average
SVD
+ oneclass user preference context
SVD++
+ multiclass user preference context
MF-MPC
+ multiclass user&item preference context
MF-DMPC

Fig. 4. Illustration of related model-based algorithms.

5.2 Recommendation with External Context

For a long time, researchers are searching for ways to make prediction with only user's explicit feedback in a form of (user, item, rating) etc., due to the limited access to get more information from users in web services. However, with rapid spread of big data, we are now able to make use of more context in recommendation algorithm. Context-aware recommendation have been developed and applied, thanks to the findings in behavioral research on consumer decision. For example, temporal context is likely to have an impact on a travel recommendation system. In contrast with traditional recommendation, context-aware recommendation typically deal with data records of the form (user, item, rating, context), which means that it takes more contextual information (mainly relevant external context) into consideration. According to [4], there are four major approaches to model contextual information in context-aware recommendation systems, distinct by change (static, dynamic) and knowledge (fully observable, partially observable, unobservable) of/about contextual factors. And [4] also introduces the three main algorithmic paradigms for incorporating contextual information into rating-based recommender systems: contextual pre-filtering, post-filtering, and modeling.

5.3 Discussions

From single to hybrid CF recommendation methods, we have achieved great progress on recommendation with internal context. However, we are still looking forward to more improvement because the two-dimensional (2D) data (user, item) will also be included in recommendation with external context and the way of applying the 2D data in traditional method can be illuminating to many other methods such as the prediction rule of timeSVD++ [8].

Different from ours, timeSVD++ is a representative method of containing external context, which also adopts matrix factorization model. TimeSVD++ takes temporal effects on baseline predictors and user preferences into consideration. The prediction rule of timeSVD++ is as follows [8]:

$$\hat{r}_{ui} = U_{u\cdot}(t_{ui})V_{i\cdot}^T + \bar{U}_{u\cdot}^{OPC}V_{i\cdot}^T + b_u(t_{ui}) + b_i(t_{ui}) + \mu, \tag{19}$$

where the user bias b_u, item bias b_i and user-specific latent feature vector $U_{u\cdot}$ in SVD++ model are replaced by time changing part $b_u(t_{ui})$, $b_i(t_{ui})$ and $U_{u\cdot}(t_{ui})$, respectively. Notice that $\bar{U}_{u\cdot}^{OPC}$ is the oneclass form of MPC ($\mathbb{M} = \{0, 1\}$). It is said that a timeSVD++ model of dimension 10 is already more accurate than an SVD model of dimension 200, which can be an evidence of the importance of capturing proper external context (temporal dynamics, in this case). It is understandable that multiclass preference context (defined in our MF-DMPC) can also be introduced into timeSVD++ because of the same factorization framework. Notice that the improvement on timeSVD++ is not illustrated any further since we mention it here only to address the significance of recommendation with internal context in the field of recommendation system.

Notably, approaches mentioned above are all applied in isolated services and ignoring services sociability. Lately, however, a methodology to construct a global social service network for social influence-aware service recommendation approach, which provides recommend-as-you-go, has been proposed [3]. From one perspective, we can say that this work provides a kind of thought to connect recommendation approaches used in different services and therefore achieves better results.

6 Conclusions and Future Work

In this paper, we present a novel collaborative filtering method that joins neighborhood information to factorization model for rating prediction. Specifically, we extend multiclass preference context (MPC) to include two types, i.e., user-based and item-based, and combine them in one single prediction rule in order to achieve better recommendation performance than the reference models.

For future works, as discussed, the inefficiency of our model is an inevitable problem and what we must firstly solve. In the second place, we are interested in studying the issues of robustness of factorization-based algorithms with preference context. We also expect some advanced strategy such as adversarial sampling [13], denoising [12] or multilayer perception [5] to be used.

Acknowledgements. We thank Mr. Yunfeng Huang for assistance in code review and helpful discussions. We thank the support of National Natural Science Foundation of China No. 61502307, No. 61672358 and U1636202, and Natural Science Foundation of Guangdong Province No. 2016A030313038. Weike Pan and Zhong Ming are the corresponding authors for this work.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adap. Inter. **24**(1–2), 67–119 (2014)
3. Chen, W., Paik, I., Tanaka, T., Kumara, B.T.G.S.: Awareness of social influence for service recommendation. In: 2013 IEEE International Conference on Services Computing, pp. 767–768 (2013)
4. Shapira, B., Ricci, F., Rokach, L.: Recommender Systems Handbook, 2nd edn. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
6. Hu, J., Li, P.: Collaborative filtering via additive ordinal regression. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 243–251 (2018)
7. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
8. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447–456 (2009)
9. Pan, W., Ming, Z.: Collaborative recommendation with multiclass preference context. IEEE Intell. Syst. **32**(2), 45–51 (2017)
10. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**(3), 57:1–57:22 (2012)
11. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
12. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 1096–1103 (2008)
13. Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., Zhang, D.: IRGAN: a minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 515–524 (2017)
14. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017)