Matrix Factorization with Heterogeneous Multiclass Preference Context

Jing Lin[1,2,3], Weike Pan[1,2,3,*], Lin Li[1,2,3], Zixiang Chen[1,2,3] and Zhong Ming[1,2,3,*]

{linjing2018, lilin20171, chenzixiang2016}@email.szu.edu.cn, {panweike, mingz}@szu.edu.cn

[1]National Engineering Laboratory for Big Data System Computing Technology
Shenzhen University, China

[2]Guangdong Laboratory of Artficial Intelligence and Digital Economy (SZ)
Shenzhen University, China

[3]College of Computer Science and Software Engineering
Shenzhen University, China

Motivation

Making Recommendations with Internal Context Only

- With the growing awareness of personal privacy, using rating matrix only to discover more internal context (latent collaborative pattern) is a more reliable and perpetually efficient strategy.
- A recently proposed model called matrix factorization with multiclass preference context (MF-MPC) [Pan and Ming, 2017] is a unified method which combines the two major categories of collaborative filtering — neighborhood-based and model-based. However, it is lacking consideration on the orientations of the neighborhood information.

Overall of Our Solution

1. In this paper, we propose two MF models that contain not only user similarity but also item similarity, and collectively referred to as matrix factorization with heterogeneous multiclass preference context (MF-HMPC).
2. More specifically, MF-HMPC consists of matrix factorization with dual multiclass preference context (MF-DMPC) for concurrent structure and matrix factorization with pipelined multiclass preference context (MF-PMPC) for sequential structure.

Advantages of Our Solution

In general, our MF-HMPC that unifies MF-DMPC and MF-PMPC inherits both high accuracy of model-based recommendation algorithms and good explainability of neighborhood-based algorithms, and further strikes a good balance between user-oriented neighborhood information and item-oriented neighborhood information.

Notations (1/2)

Table: Some notations and explanations (1/2).

n	number of users
m	number of items
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
$\mathcal{I}_u$	items rated by user u
$\mathcal{I}_u^r, r \in \mathbb{M}$	items rated by user u with rating r
$\mathcal{U}_i$	users who rate item i
$\mathcal{U}_i^r, r \in \mathbb{M}$	users who rate item i with rating r

Notations (2/2)

Table: Some notations and explanations (2/2).

$\mu \in \mathbb{R}$	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
$V_{i\cdot} \in \mathbb{R}^{1\times d}$	item-specific latent feature vector
$N^r_{u\cdot} \in \mathbb{R}^{1\times d}$	user-specific latent feature vector w.r.t. rating r
$M^r_{i\cdot} \in \mathbb{R}^{1\times d}$	item-specific latent feature vector w.r.t. rating r
$\bar{U}^{\text{MPC}}_{u\cdot}$	aggregated user-specific latent preference vector
$\bar{V}^{\text{MPC}}_{i\cdot}$	aggregated item-specific latent preference vector
$\mathcal{R}^{te} = \{(u, i, r_{ui})\}$	rating records of test data
$\hat{r}_{ui}$	the final predicted rating of user u to item i
$\hat{r}^{1}_{ui}$	the first predicted rating (iff in residual based algorithm)
$\hat{r}^{2}_{ui}$	the second predicted rating (ditto)
r^{RES}_{ui}	the residual rating (ditto)
T	iteration number

Related Work

- Traditional collaborative filtering algorithms
  - Neighborhood-based methods
    User-oriented CF
    Item-oriented CF
  - Model-based methods
    SVD [Rendle, 2012]
    SVD++ [Koren, 2008]
    MF-MPC [Pan and Ming, 2017]
- Deep learning based collaborative filtering algorithms
  - Restricted Boltzmann machines (RBM) [Salakhutdinov et al., 2007]
  - Neural collaborative filtering (NCF) [He et al., 2017]

Problem Definition

- Input: An incomplete rating matrix represented by $\mathcal{R} = \{(u, i, r_{ui})\}$. Notice that u represents one of the ID numbers of n users (or rows in the rating matrix), i represents one of the ID numbers of m items (or columns), and $r_{ui} \in \mathbb{M}$ is the recorded rating of user u to item i, where $\mathbb{M}$ can be $\{1, 2, 3, 4, 5\}$, $\{0.5, 1, 1.5, \ldots, 5\}$ or other ranges.
- Goal: To predict the vacancies of the rating matrix.

Multiclass Preference Context (1/3)

Through investigations about combining neighborhood-based and factorization-based methods, [Pan and Ming, 2017] proposes a categorical internal context to encode the neighborhood information in a matrix factorization framework. Intuitively, the rating of user u to item i, i.e., r_{ui}, can be represented in a probabilistic way as follows,

$$\text{Prob}(r_{ui}|(u,i); (u,i',r_{ui'}), i' \in \cup_{r\in\mathbb{M}}\mathcal{I}_u^r\backslash\{i\}), \quad (1)$$

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 $r_{ui'} \in \mathbb{M}$ assigned to each item by user u. Here, the condition $(u,i',r_{ui'}), i' \in \cup_{r\in\mathbb{M}}\mathcal{I}_u^r\backslash\{i\}$ is given a name multiclass preference context (MPC) in contrast to oneclass preference context (OPC) without categorical scores.

Multiclass Preference Context (2/3)

In order to introduce MPC into an MF method, [Pan and Ming, 2017]

- defined a user-specific aggregated latent preference vector $\bar{U}_{u\cdot}^{\text{MPC}}$ for user u from the multiclass preference context,

$$\bar{U}_{u\cdot}^{\text{MPC}} = \sum_{r \in \mathbb{M}} \frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u^r \backslash \{i\}} M_{i'\cdot}^r. \qquad (2)$$

Notice that $M_{i\cdot}^r \in \mathbb{R}^{1 \times d}$ can be considered as a classified item-specific latent feature vector w.r.t. rating r, and $\frac{1}{\sqrt{|\mathcal{I}_u^r \backslash \{i\}|}}$ plays as a normalization term for the preference of class r. We believe that MPC can represent user similarity.

Multiclass Preference Context (3/3)

- then added the neighborhood information $\bar{U}_{u\cdot}^{\text{MPC}}$ to SVD model so as to get the MF-MPC prediction rule for the rating of user u to item i as follows,

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + \bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}^T + b_u + b_i + \mu, \tag{3}$$

  where $U_{u\cdot}$, $V_{i\cdot}$, 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 and SVD++, and also embraces them as special cases.

MF-DMPC

Inspired by the differences between user-oriented and item-oriented collaborative filtering, we can infer that item similarity (item-oriented MPC) can also be introduced to improve the performance of matrix factorization models. Furthermore, thanks to the extendibility of MF models, we can hopefully join both user-oriented MPC and item-oriented MPC into the prediction rule so as to obtain a hybrid model, i.e., matrix factorization with dual multiclass preference context (MF-DMPC).

Item-Oriented Multiclass Preference Context (1/3)

- Now we restate $\bar{U}_{u\cdot}^{\text{MPC}}$ as user-oriented multiclass preference context (user-oriented MPC). Similarly, we have a symmetrical form of MPC called item-oriented multiclass preference context (item-oriented MPC) $\bar{V}_{i\cdot}^{\text{MPC}}$ to represent item similarity, which is formulated as,

$$\bar{V}_{i\cdot}^{\text{MPC}} = \sum_{r \in \mathbb{M}} \frac{1}{\sqrt{|\mathcal{U}_i^r \backslash \{u\}|}} \sum_{u' \in \mathcal{U}_i^r \backslash \{u\}} N_{u'\cdot}^r, \quad (4)$$

where $N_{u\cdot}^r \in \mathbb{R}^{1 \times d}$ is a user-specific latent preference vector w.r.t. rating r.

- Likewise, we have the prediction rule of item-oriented MF-MPC,

$$\hat{r}_{ui} = U_{u\cdot} V_{i\cdot}^T + \bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}^T + b_u + b_i + \mu. \quad (5)$$

Item-Oriented Multiclass Preference Context (2/3)

The learning process of matrix factorization with user-oriented and item-oriented MPC respectively are quite similar. With different prediction rules, they have the same abbreviated optimization function as follows,

$$\arg\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui}[\frac{1}{2}(r_{ui} - \hat{r}_{ri})^2 + \text{reg}(u, i)]. \tag{6}$$

In particular, the regularization terms $\text{reg}(u, i)$ vary for specific cases, i.e., in user-oriented MF-MPC,

$\text{reg}(u, i) = \frac{\alpha_m}{2} \sum_{r \in \mathbb{M}} \sum_{i' \in \mathcal{I}_u^r \setminus \{i\}} ||M_{i'\cdot}^r||_F^2 + \frac{\alpha}{2}||U_{u\cdot}||^2 + \frac{\alpha}{2}||V_{i\cdot}||^2 + \frac{\alpha}{2}||b_u||^2 + \frac{\alpha}{2}||b_i||^2,$

and item-oriented MF-MPC,

$\text{reg}(u, i) = \frac{\alpha_n}{2} \sum_{r \in \mathbb{M}} \sum_{u' \in \mathcal{U}_i^r \setminus \{u\}} ||N_{u'\cdot}^r||_F^2 + \frac{\alpha}{2}||U_{u\cdot}||^2 + \frac{\alpha}{2}||V_{i\cdot}||^2 + \frac{\alpha}{2}||b_u||^2 + \frac{\alpha}{2}||b_i||^2.$

Item-Oriented Multiclass Preference Context (3/3)

Table: The gradients of the model parameters $\Theta = \{M^r_{i\cdot}, U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu\}$ in user-oriented MF-MPC and $\Theta = \{N^r_{u\cdot}, U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu\}$ in item-oriented MF-MPC, with $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$ in common.

User-oriented MF-MPC	Item-oriented MF-MPC
$\nabla U_{u\cdot} = -e_{ui} V_{i\cdot} + \alpha U_{u\cdot}$	$\nabla U_{u\cdot} = -e_{ui}(V_{i\cdot} + \bar{V}^{\text{MPC}}_{i\cdot}) + \alpha U_{u\cdot}$
$\nabla V_{i\cdot} = -e_{ui}(U_{u\cdot} + \bar{U}^{\text{MPC}}_{u\cdot}) + \alpha V_{i\cdot}$	$\nabla V_{i\cdot} = -e_{ui} U_{u\cdot} + \alpha V_{i\cdot}$
$\nabla M^r_{i'\cdot} = \frac{-e_{ui} V_{i\cdot}}{\sqrt{\lvert \mathcal{I}^r_u \backslash \{i\} \rvert}} + \alpha_m M^r_{i'\cdot},\ i' \in \mathcal{I}^r_u \backslash \{i\}$	$\nabla N^r_{u'\cdot} = \frac{-e_{ui} U_{u\cdot}}{\sqrt{\lvert \mathcal{U}^r_i \backslash \{u\} \rvert}} + \alpha_n N^r_{u'\cdot},\ u' \in \mathcal{U}^r_i \backslash \{u\}$
$\nabla b_u = -e_{ui} + \alpha b_u$	
$\nabla b_i = -e_{ui} + \alpha b_i$	
$\nabla \mu = -e_{ui}$	

Hence, with $e_{ui} = (r_{ui} - \hat{r}_{ui})$, the model parameters to be learned and the corresponding gradients are also different. With the gradients, we can update the model parameters Θ via the update rule,

$$\theta = \theta - \gamma \nabla \theta, \quad (7)$$

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

Dual Multiclass Preference Context (1/2)

- We can handle both user-oriented and item-oriented neighborhood information by joining both $\bar{U}_{u\cdot}^{\text{MPC}}$ and $\bar{V}_{i\cdot}^{\text{MPC}}$ simultaneously in the model, which is thus named as dual multiclass preference context (DMPC). To combine them together, we acquire $\bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}{}^{T} + \bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}{}^{T}$ as the DMPC term.
- In this way, we obtain our new model, i.e., matrix factorization with dual multiclass preference context (MF-DMPC). The prediction rule of our MF-DMPC for the rating of user u to item i is finally defined as follows,

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

with all notations described ahead.

Dual Multiclass Preference Context (2/2)

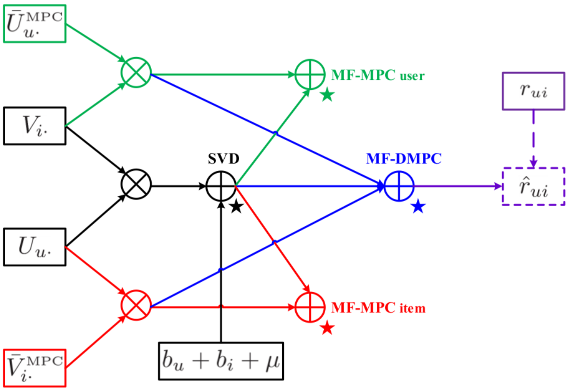


Figure: Illustration of SVD (in black), MF-MPC$_{\text{user}}$ (in green), MF-MPC$_{\text{item}}$ (in red) and our MF-DMPC (in blue). The stars mark the results of each method. Solid lines express value pass, and dashed line points from the true rating to the predicted one.

Learning Algorithm of MF-DMPC (1/2)

- With the MF-DMPC prediction rule in Eq.(8), we can learn the model parameters in the following minimization problem,

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

where $\text{reg}^{DMPC}(u,i) = \frac{\alpha_m}{2} \sum_{r\in\mathbb{M}} \sum_{i'\in\mathcal{I}_u^r\backslash\{i\}} ||M_{i'\cdot}^r||_F^2 + \frac{\alpha_n}{2} \sum_{r\in\mathbb{M}} \sum_{u'\in\mathcal{U}_i^r\backslash\{u\}} ||N_{u'\cdot}^r||_F^2 + \frac{\alpha}{2}||U_{u\cdot}||^2 + \frac{\alpha}{2}||V_{i\cdot}||^2 + \frac{\alpha}{2}||b_u||^2 + \frac{\alpha}{2}||b_i||^2$ is the regularization term used to avoid overfitting, and $\Theta = \{U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu, M_{i\cdot}^r, N_{u\cdot}^r\}$ with $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$ are model parameters to be learned.

Learning Algorithm of MF-DMPC (2/2)

- Using the stochastic gradient descent (SGD) algorithm, the algorithm of MF-DMPC consists of three major steps.

```
Initialize model parameters Θ = {U_u·, V_i·, b_u, b_i, μ, M^r_i·, N^r_u·}, with u = 1, 2, ..., n, i = 1, 2, ..., m, r ∈ 𝕄
for t = 1, ..., T do
    for t_2 = 1, ..., |R| do
        Randomly pick up a rating from R
        Calculate the gradients
        Update the parameters via Eq.(7)
    end for
    Decrease the learning rate γ ← γ × 0.9
end for
```

Figure: The algorithm of MF-DMPC.

MF-PMPC

Inspired by the residual training strategy, we can obtain the predicted ratings through a pipelined process, which executes independent user-oriented MPC and item-oriented MPC methods one after another. For this reason, we name these kinds of methods as matrix factorization with pipelined multiclass preference context (MF-PMPC).

Residual Training

In residual training [Jahrer et al., 2010], the prediction rule for rating $\hat{r}_{ui}$ is divided into two parts, i.e., $\hat{r}^{\mathbb{1}}_{ui}$ and $\hat{r}^{\mathbb{2}}_{ui}$, as follows,

$$\hat{r}_{ui} = \hat{r}^{\mathbb{1}}_{ui} + \hat{r}^{\mathbb{2}}_{ui}. \quad (10)$$

There are two steps in series in an integrated residual training process of MF-PMPC as follows,

- Step 1: Train a user-oriented or item-oriented MF-MPC model to obtain $\hat{r}^{\mathbb{1}}_{ui}$, making it as close to the training data r_{ui} as possible, and get the residual rating,

$$r^{\text{RES}}_{ui} = r_{ui} - \hat{r}^{\mathbb{1}}_{ui}. \quad (11)$$

- Step 2: Train another MF-MPC model to obtain the prediction value $\hat{r}^{\mathbb{2}}_{ui}$, of which the target value is the residual data r^{RES}_{ui} instead of the original training data r_{ui}.

Pipelined Multiclass Preference Context (1/3)

As to assemble user-oriented MPC and item-oriented MPC into a pipelined process, we now arrange the <user-oriented MPC, item> interaction $\bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}^{T}$ and the <item-oriented MPC, user> interaction $\bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}^{T}$ in two steps. When introducing $\bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}^{T}$ in the first step and $\bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}^{T}$ in the residual step, this pipelined algorithm is denoted as MF-PMPC$_{\text{user}\rightarrow\text{item}}$. On the contrary, we denote a residual MF method which starts with item-oriented MPC as MF-PMPC$_{\text{item}\rightarrow\text{user}}$.

Pipelined Multiclass Preference Context (2/3)

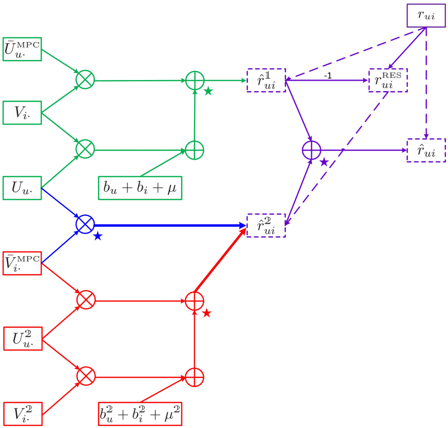


Figure: Illustration of MF-PMPC$_{\text{user}\rightarrow\text{item}}$. The left side shows possible components of one step's results, and the right side depicts the integration of two steps' results. The difference is that the two coarse lines denote two options for the residual result.

Pipelined Multiclass Preference Context (3/3)

- Our preliminary studies show that it is more steady and effective to retrain the SVD parameters in the residual step, probably to ensure the coordination of different parameters. So we determine the prediction rules of MF-PMPC$_{\text{user}\rightarrow\text{item}}$ as follows,

$$\text{Step 1:} \quad \hat{r}_{ui}^{1} = U_{u\cdot} V_{i\cdot}^{T} + \bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}^{T} + b_u + b_i + \mu, \tag{12}$$

$$\text{Step 2:} \quad \hat{r}_{ui}^{2} = U_{u\cdot}^{2} V_{i\cdot}^{2\,T} + \bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}^{2\,T} + b_u^{2} + b_i^{2} + \mu^{2}. \tag{13}$$

- Meanwhile, we have MF-PMPC$_{\text{item}\rightarrow\text{user}}$ in a symmetrical way as follows,

$$\text{Step 1:} \quad \hat{r}_{ui}^{1} = U_{u\cdot} V_{i\cdot}^{T} + \bar{V}_{i\cdot}^{\text{MPC}} U_{u\cdot}^{T} + b_u + b_i + \mu, \tag{14}$$

$$\text{Step 2:} \quad \hat{r}_{ui}^{2} = U_{u\cdot}^{2} V_{i\cdot}^{2\,T} + \bar{U}_{u\cdot}^{\text{MPC}} V_{i\cdot}^{2\,T} + b_u^{2} + b_i^{2} + \mu^{2}. \tag{15}$$

Learning Algorithm of MF-PMPC (1/2)

Continuing with the MF-PMPC$_{\text{user}\rightarrow\text{item}}$ example, the optimization function, gradients calculation and update rule of Eq.(12) and Eq.(13) are similar to that of single user-oriented MF-MPC and single item-oriented MF-MPC, respectively. Only the target rating r_{ui} changes as,

$$\text{Step 1:} \quad r_{ui}^{\mathbb{1}} = r_{ui} \qquad \text{(the same as the original rating)}, \tag{16}$$

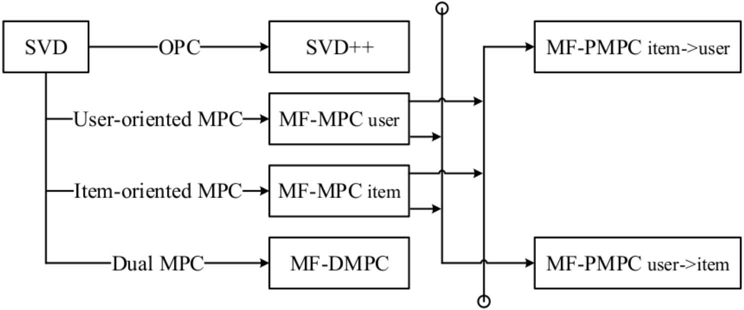
$$\text{Step 2:} \quad r_{ui}^{\mathbb{2}} = r_{ui}^{\text{RES}} = r_{ui} - \hat{r}_{ui}^{\mathbb{1}} \qquad \text{(change to the residual rating)}. \tag{17}$$

Learning Algorithm of MF-PMPC (2/2)

// the first user-oriented MPC step
Initialize model parameters $\Theta^{\mathbb{1}} = \{M^r_{i\cdot}, U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu\}$, with $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$.
Set the learning rate $\gamma = 0.01$.
for $t = 1, \ldots, T$ **do**
    for $t_2 = 1, \ldots, |\mathcal{R}|$ **do**
        Randomly pick up a rating record (u, i, r_{ui}) from $\mathcal{R}$
        Calculate the gradients $\nabla M^r_{i\cdot}, \nabla U_{u\cdot}, \nabla V_{i\cdot}, \nabla b_u, \nabla b_i, \nabla \mu$
        Update parameters in $\Theta^{\mathbb{1}}$ to make $\hat{r}^{\mathbb{1}}_{ui}$ approximate to r_{ui}
    end for
    Decrease the learning rate $\gamma \leftarrow \gamma \times 0.9$
end for
Obtain target residual rating r^{RES}_{ui} of each user to each item
// the residual item-oriented MPC step
Initialize model parameters $\Theta^{\mathbb{2}} = \{N^r_{u\cdot}, U^{\mathbb{2}}_{u\cdot}, V^{\mathbb{2}}_{i\cdot}, b^{\mathbb{2}}_u, b^{\mathbb{2}}_i, \mu^{\mathbb{2}}\}$, with $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, $r \in \mathbb{M}$
Reset the learning rate $\gamma = 0.01$
for $t = 1, \ldots, T$ **do**
    for $t_2 = 1, \ldots, |\mathcal{R}|$ **do**
        Randomly pick up a rating record (u, i, r_{ui}) from $\mathcal{R}$
        Calculate the gradients $\nabla N^r_{u\cdot}, \nabla U^{\mathbb{2}}_{u\cdot}, \nabla V^{\mathbb{2}}_{i\cdot}, \nabla b^{\mathbb{2}}_u, \nabla b^{\mathbb{2}}_i, \nabla \mu^{\mathbb{2}}$
        Update parameters in $\Theta^{\mathbb{2}}$ to make $\hat{r}^{\mathbb{2}}_{ui}$ approximate to $r^{\text{RES}}_{ui} = r_{ui} - \hat{r}^{\mathbb{1}}_{ui}$
    end for
    Decrease the learning rate $\gamma \leftarrow \gamma \times 0.9$
end for

Figure: The algorithm of MF-PMPC via user→item configuration.

MF-HMPC



Finally, we unify MF-DMPC and MF-PMPC into a generic factorization-based framework, i.e., matrix factorization with heterogeneous multiclass preference context (MF-HMPC). In this framework, the two specific variants of MF models with two types of MPC are of different structures, i.e, MF-DMPC for concurrent structure and MF-PMPC for sequential structure.

Datasets

Table: Statistics of the datasets used in the experiments, including the number of users (n), the number of items (m), the number of rating records in the whole dataset ($|\mathcal{R}| + |\mathcal{R}^{te}|$), the ratio of the number of users to the number of items (n/m), and the density of training data ($|\mathcal{R}|/nm$).

Dataset	n	m	$\|\mathcal{R}\| + \|\mathcal{R}^{te}\|$	n/m	$\|\mathcal{R}\|/nm$
ML100K	943	1,682	100,000	0.56	5.04%
ML1M	6,040	3,952	1,000,209	1.53	3.35%
ML10M	71,567	10,681	10,000,054	6.70	1.05%
NF10M	50,000	17,770	10,442,504	2.81	0.94%

Evaluation Metrics

- Mean absolute error:

$$MAE = \sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} |r_{ui} - \hat{r}_{ui}| / |\mathcal{R}^{te}|, \tag{18}$$

- Root mean square error:

$$RMSE = \sqrt{\sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} (r_{ui} - \hat{r}_{ui})^2 / |\mathcal{R}^{te}|}. \tag{19}$$

Baselines

- To find out the effect of single type of MPC, we have the foremost baselines:
  - **SVD** is the basic MF model for recommendations without MPC.
  - **MF-MPC$_{\text{user}}$** is an MF model with only a user-oriented MPC term.
  - **MF-MPC$_{\text{item}}$** is an MF model with only an item-oriented MPC term.
- We also include two deep learning CF models:
  - **RBM** utilizes a class of two-layer undirected graphical models called restricted Boltzmann machines to make rating prediction.
  - **NCF** utilizes neural networks with multi-layer perceptron layers to model implicit feedback in a non-linear way.

Our Methods

- **MF-DMPC** is an MF model which is combined with dual MPC. Structurally, DMPC is a parallel combination form.
- **MF-PMPC$_{\textbf{user}\rightarrow\textbf{item}}$** is a model which is divided into two MF steps – a user-oriented MF-MPC followed by an item-oriented MF-MPC. PMPC can be considered as a sequential modeling technique.
- **MF-PMPC$_{\textbf{item}\rightarrow\textbf{user}}$** is a reverse process of MF-PMPC$_{\text{user}\rightarrow\text{item}}$.

Parameter Configurations (1/3)

- We configure the parameters of the MF based methods as follows. For the learning rate γ, we set its initial value to a common default value, i.e., $\gamma = 0.01$. For the number of latent dimensions d, it is sufficient to show the advantages of each methods when $d = 20$ [Pan and Ming, 2017]. For the iteration number, we fix it as $T = 50$, where the results have reached a steady state. And for each baseline on each dataset, the tradeoff parameters $\alpha \in \{0.001, 0.01, 0.1\}$ are selected to be different values through parameter tuning experiments using the RMSE metric.

Parameter Configurations (2/3)

- For RBM, we use the hidden layer of 20 units, mini-batches of size 16, CD learning with 1 step Gibbs sampling, and the learning rate of 0.05. Taking consideration of efficiency and effectiveness, we adopt a momentum of 0.9 and search the optimal weight decay value from $\{0.0005, 0.001, 0.005, 0.01\}$ according to the RMSE results. We also apply early stop strategy to avoid overfitting, i.e., model training stops when the RMSE no longer decrease within 20 epoches or the number of training iterations reach to a ceiling value of 200.

Parameter Configurations (3/3)

- For NCF, we use mean-square-loss and linear function for the prediction layer. We use mini-batch Adam for optimization. Specifically, we fix the MLP hidden layers as $64 \rightarrow 32 \rightarrow 16 \rightarrow 8$, which corresponds to the best performance from the paper, and search the best value of the learning rate from $\{0.0001, 0.0005, 0.001, 0.005\}$ and the batch size from $\{128, 256, 512, 1024\}$.

Main Results (1/4)

Table: Recommendation performance of SVD, MF-MPC, MF-DMPC, MF-PMPC and two deep learning methods RBM and NCF on ML100K and ML1M. The best result of each dataset is highlighted in bold and the suboptimal result is marked with “△”.

Data	Method	MAE	RMSE	$(\alpha, \alpha_m, \alpha_n)$
ML100K	SVD	0.7446±0.0033	0.9445±0.0035	(0.01, N/A, N/A)
	MF-MPC$_{user}$	0.7129±0.0032	0.9098±0.0028	(0.01, 0.01, N/A)
	MF-MPC$_{item}$	0.7025±0.0025	0.8980±0.0020	(0.01, N/A, 0.01)
	MF-DMPC	0.7008±0.0027	0.8972±0.0029	(0.01, 0.01, 0.01)
	MF-PMPC$_{user \to item}$	0.7002 ± 0.0025△	0.8959 ± 0.0032△	(0.01, 0.01, N/A)
	MF-PMPC$_{item \to user}$	**0.6991**±0.0029	**0.8941**±0.0025	(0.01, N/A, 0.01)
	RBM	0.7734±0.0028	0.9693±0.0022	$\alpha_w = 0.01$
	NCF	0.7189±0.0042	0.9169±0.0041	$\gamma_{NCF} = 0.0001$, $s_{batch} = 128$
ML1M	SVD	0.7017±0.0016	0.8899±0.0023	(0.01, N/A, N/A)
	MF-MPC$_{user}$	0.6605± 0.0013	0.8442±0.0017	(0.01, 0.01, N/A)
	MF-MPC$_{item}$	0.6568±0.0013	0.8411±0.0016	(0.01, N/A, 0.01)
	MF-DMPC	0.6552 ± 0.0013△	0.8409 ± 0.0017△	(0.01, 0.01, 0.01)
	MF-PMPC$_{user \to item}$	0.6572±0.0014	0.8410±0.0018	(0.01, 0.01, N/A)
	MF-PMPC$_{item \to user}$	**0.6549**±0.0012	**0.8396**±0.0052	(0.01, N/A, 0.01)
	RBM	0.7071±0.0009	0.8979±0.0008	$\alpha_w = 0.001$
	NCF	0.6746±0.0034	0.8634±0.0034	$\gamma_{NCF} = 0.0001$, $s_{batch} = 256$

Main Results (2/4)

Table: Recommendation performance of SVD, MF-MPC, MF-DMPC, MF-PMPC and two deep learning methods RBM and NCF on ML10M and NF10M. The best result of each dataset is highlighted in bold and the suboptimal result is marked with "△".

Data	Method	MAE	RMSE	$(\alpha, \alpha_m, \alpha_n)$
ML10M	SVD	0.6067±0.0007	0.7913±0.0009	(0.01, N/A, N/A)
	MF-MPC$_{user}$	0.5950±0.0005	0.7782±0.0006	(0.01, 0.01, N/A)
	MF-MPC$_{item}$	0.5983±0.0005	0.7833±0.0005	(0.01, N/A, 0.01)
	MF-DMPC	0.5941 ± 0.0005△	0.7782 ± 0.0007△	(0.01, 0.001, 0.1)
	MF-PMPC$_{user \to item}$	**0.5933**±0.0006	**0.7766**±0.0006	(0.01, 0.01, N/A)
	MF-PMPC$_{item \to user}$	0.5952±0.0007	0.7796±0.0006	(0.01, N/A, 0.01)
	RBM	0.6550±0.0008	0.8500±0.0005	$\alpha_w = 0.0005$
	NCF	0.6108±0.0006	0.7994±0.0009	$\gamma_{NCF} = 0.0005$, $s_{batch} = 512$
NF10M	SVD	0.6506±0.0003	0.8402±0.0004	(0.01, N/A, N/A)
	MF-MPC$_{user}$	0.6415±0.0004	0.8314±0.0005	(0.01, 0.01, N/A)
	MF-MPC$_{item}$	0.6435±0.0005	0.8342±0.0003	(0.01, N/A, 0.01)
	MF-DMPC	0.6390 ± 0.0005△	0.8303±0.0004	(0.01, 0.001, 0.01)
	MF-PMPC$_{user \to item}$	0.6392±0.0003	**0.8298**±0.0005	(0.01, 0.01, N/A)
	MF-PMPC$_{item \to user}$	**0.6389**±0.0004	0.8300 ± 0.0005△	(0.01, N/A, 0.01)
	RBM	0.7000±0.0007	0.8948±0.0005	$\alpha_w = 0.0005$
	NCF	0.6526±0.0021	0.8477±0.0004	$\gamma_{NCF} = 0.001$, $s_{batch} = 128$

Main Results (3/4)

Observations:

- The recommendation performance improves with richer preference context, i.e., from void preference context in SVD to simple and complex preference context in MF-MPC, MF-DMPC and MF-PMPC.
- For modeling user-oriented preference context and item-oriented preference context, the performance depends on the ratio of user group size to item group size, i.e., n/m. Specifically, item-oriented MF-MPC performs better when n/m is of a suitable size (as on ML100K and ML1M). And as n/m becomes larger (as on ML10M and NF10M), user-oriented MPC behaves more reliable because of the increasing probability to find similar users.
- For modeling dual preference context, MF-DMPC outperforms SVD and MF-MPC in all cases. However, the performance of MF-DMPC is in a way restrained by the better result of user-oriented MF-MPC and item-oriented MF-MPC. Gratifyingly, this improvement reveals that MF-DMPC strikes a good balance between user-oriented MPC and item-oriented MPC ...

Main Results (4/4)

Observations:

- For modeling sequential preference context, MF-PMPC is the best approach in all cases, which showcases the effectiveness of our residual-based sequential modeling approach. The relative performance between the two types of MF-PMPC is similar to that of MF-MPC$_{\text{user}}$ and MF-MPC$_{\text{item}}$, which shows the importance of the first step in MF-PMPC. There is one exception that it is hard to tell whether MF-MPC$_{\text{user}}$ or MF-MPC$_{\text{item}}$ is better on NF10M, where n/m is small and the records number is large.
- ...

Conclusions

- We study a classical recommendation problem, i.e., rating prediction in a user-item matrix, and develop a generic factorization-based framework, i.e., matrix factorization with heterogeneous multiclass preference context (MF-HMPC).
- We design two specific variants with different structures, including MF with dual MPC (MF-DMPC) for concurrent structure and MF with pipelined MPC (MF-PMPC) for sequential structure.
- Empirical studies on four public datasets clearly showcase the advantages of our methods over the very state-of-the-art methods.

Future Works

For future works, we are interested in studying the issues of robustness of factorization-based algorithms with internal preference context. We also plan to study some advanced strategies such as adversarial sampling, denoising and multilayer perception in our proposed factorization framework.

Thank you!

- We thank the handling Associate Editor and Reviewers for their efforts and constructive expert comments
- We thank Mr. Yunfeng Huang for assistance in code review and helpful discussions.
- We thank the support of National Natural Science Foundation of China Nos. 61872249, 61836005 and 61672358.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017).
Neural collaborative filtering.
In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182.

Jahrer, M., Töscher, A., and Legenstein, R. (2010).
Combining predictions for accurate recommender systems.
In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 693–702.

Koren, Y. (2008).
Factorization meets the neighborhood: A multifaceted collaborative filtering model.
In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434.

Pan, W. and Ming, Z. (2017).
Collaborative recommendation with multiclass preference context.
IEEE Intelligent Systems, 32(2):45–51.

Rendle, S. (2012).
Factorization machines with libfm.
ACM Trans. Intell. Syst. Technol., 3(3):57:1–57:22.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007).
Restricted boltzmann machines for collaborative filtering.
In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798.