Mixed Factorization for Collaborative Recommendation with Heterogeneous Explicit Feedbacks

Weike Pan, Shanchuan Xia, Zhuode Liu, Xiaogang Peng and Zhong Ming*
panweike@szu.edu.cn, shaun.xia@outlook.com, zero.lzd@gmail.com, patrickpeng@126.com, mingz@szu.edu.cn

College of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China
Collaborative Recommendation with Heterogeneous Explicit Feedbacks (CR-HEF)

Input:
- 5-star grade scores $\mathcal{R} = \{(u, i, r_{ui})\}$, where $r_{ui} \in \mathbb{G} = \{0.5, 1, 1.5, \ldots, 5\}$
- like/dislike binary ratings $\tilde{\mathcal{R}} = \{(u, i, \tilde{r}_{ui})\}$, where $\tilde{r}_{ui} \in \mathbb{B} = \{\text{like}, \text{dislike}\}$

Goal: rating prediction (of the missing 5-star grade scores)
Challenges

How can we integrate two different types of feedbacks in a principled way?

- **Collective factorization** such as CMF [Singh and Gordon, 2008]: two jointly conducted factorization tasks are loosely coupled, which may not fully leverage the binary ratings to the grade scores.

- **Integrative factorization** such as SVD++ [Koren, 2008]: leveraging the implicit feedbacks to the grade scores in such an integrative manner may not well capture the implicit-feedback-dependent effect.
Transfer by Mixed Factorization (TMF)

- We first take the CR-HEF problem from a transfer learning view, in which grade scores are taken as target data and binary ratings are taken as auxiliary data.

- We then propose a novel and generic mixed factorization based transfer learning framework, which consists of collective factorization and integrative factorization as different assembly components.
The following methods are special cases of our TMF

- **RSVD** [Koren, 2008]: \{e_1, e_2\}
- **CMF** [Singh and Gordon, 2008]: \{e_1, e_2, e_3, e_4\}
- **iTCF** [Pan and Ming, 2014]: \{e_1, e_2, e_3, e_4, e_5\}
Advantages of Our Solution

- TMF unifies collective factorization and integrative factorization in one single transfer learning framework, which enables both feature-based and instance-based preference learning and transfer in a principled way.

- TMF is expected to transfer more knowledge from binary ratings to grade scores than collective factorization, and to model binary-rating-dependent and -independent effect more accurately than integrative factorization.
**Notations (1/3)**

**Table:** Some notations.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>user number</td>
</tr>
<tr>
<td>$m$</td>
<td>item number</td>
</tr>
<tr>
<td>$u \in {1, 2, \ldots, n}$</td>
<td>user ID</td>
</tr>
<tr>
<td>$i, j \in {1, 2, \ldots, m}$</td>
<td>item ID</td>
</tr>
<tr>
<td>$r_{ui}$</td>
<td>observed grade score of user $u$ on item $i$</td>
</tr>
<tr>
<td>$\tilde{r}_{ui}$</td>
<td>observed binary rating of user $u$ on item $i$</td>
</tr>
<tr>
<td>$\mathcal{R} = {(u, i, r_{ui})}$</td>
<td>grade score records (training data)</td>
</tr>
<tr>
<td>$\tilde{\mathcal{R}} = {(u, i, \tilde{r}_{ui})}$</td>
<td>binary rating records (training data)</td>
</tr>
<tr>
<td>$p =</td>
<td>\mathcal{R}</td>
</tr>
<tr>
<td>$\tilde{p} =</td>
<td>\tilde{\mathcal{R}}</td>
</tr>
</tbody>
</table>
### Table: Some notations (cont.)

<table>
<thead>
<tr>
<th>$\mathcal{P}_u$</th>
<th>items liked by user $u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}_u$</td>
<td>items disliked by user $u$</td>
</tr>
</tbody>
</table>
### Table: Some notations (cont.).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu \in \mathbb{R} )</td>
<td>global average rating value</td>
</tr>
<tr>
<td>( b_u \in \mathbb{R} )</td>
<td>user bias</td>
</tr>
<tr>
<td>( b_i \in \mathbb{R} )</td>
<td>item bias</td>
</tr>
<tr>
<td>( d \in \mathbb{R} )</td>
<td>number of latent dimensions</td>
</tr>
<tr>
<td>( U_{u.<em>}, W_{u.</em>} \in \mathbb{R}^{1 \times d} )</td>
<td>user-specific latent feature vector</td>
</tr>
<tr>
<td>( U, W \in \mathbb{R}^{n \times d} )</td>
<td>user-specific latent feature matrix</td>
</tr>
<tr>
<td>( V_{i.<em>}, P_{j.</em>}, N_{j.*} \in \mathbb{R}^{1 \times d} )</td>
<td>item-specific latent feature vector</td>
</tr>
<tr>
<td>( V, P, N \in \mathbb{R}^{m \times d} )</td>
<td>item-specific latent feature matrix</td>
</tr>
<tr>
<td>( T_E = {(u, i, r_{ui})} )</td>
<td>grade score records of test data</td>
</tr>
<tr>
<td>( \hat{r}_{ui} )</td>
<td>predicted preference of user ( u ) on item ( i )</td>
</tr>
<tr>
<td>( T )</td>
<td>iteration number in the algorithm</td>
</tr>
</tbody>
</table>
Transfer Learning for Heterogeneous Feedbacks

Transfer learning approaches
- Model-based transfer
- Feature-based transfer
- Instance-based transfer

Transfer learning algorithm styles
- Adaptive knowledge transfer
- Collective knowledge transfer
- Integrative knowledge transfer
- Mixed knowledge transfer: Collective knowledge transfer + Integrative knowledge transfer

TMF: Feature-based transfer + Instance-based transfer + Mixed knowledge transfer
Related Work

Factorization for Collaborative Recommendation

Different problem settings

- Explicit feedbacks
- Implicit feedbacks
- Explicit feedbacks and implicit feedbacks
- Heterogeneous explicit feedbacks

**TMF**: for heterogeneous explicit feedbacks
Prediction Rule for Grade Scores

For grade scores, the prediction rule of user $u$ on item $i$,

$$\hat{r}_{ui} = U_u \cdot V_i^T + \bar{P}_u \cdot V_i^T + \bar{N}_u \cdot V_i^T + b_u + b_i + \mu,$$

where $\bar{P}_u$ and $\bar{N}_u$ are virtual user profiles from binary feedbacks:

$$\bar{P}_u = \delta_P w_p \frac{1}{\sqrt{|P_u|}} \sum_{j \in P_u} P_{j.},$$

$$\bar{N}_u = \delta_N w_n \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} N_{j.},$$

where $\delta_P, \delta_N \in \{1, 0\}$, and $w_p, w_n$ are weight.
Prediction Rule for Binary Ratings

For binary ratings, the prediction rule of user $u$ on item $i$, 

$$\hat{r}_{ui} = W_u \cdot V_i^T.$$  

(4)
Objective Function for Grade Scores

For grade scores, we have the objective function,

\[
\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} \left[ \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \text{reg}(U_u, V_i, b_u, b_i, P, N) \right]
\]  

(5)

where \( \text{reg}(U_u, V_i, b_u, b_i, P, N) = \frac{\alpha_u}{2} \| U_u \|^2 + \frac{\alpha_v}{2} \| V_i \|^2 + \frac{\beta_u}{2} \| b_u \|^2 + \frac{\beta_v}{2} \| b_i \|^2 + \delta_P \frac{\alpha_p}{2} \sum_{j \in P_u} \| P \|_F^2 + \delta_N \frac{\alpha_n}{2} \sum_{j \in N_u} \| N \|_F^2 \) is the regularization term used to avoid overfitting.
Objective Function for Binary Ratings

For binary ratings, we have the objective function,

$$\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} \tilde{y}_{ui} \left[ \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \text{reg}(W_u, V_i) \right]$$

(6)

where $\text{reg}(W_u, V_i) = \frac{\alpha_w}{2} ||W_u||^2 + \frac{\alpha_v}{2} ||V_i||^2$ is the regularization term used to avoid overfitting.
Overall Objective Function

- We have the overall objective function,

\[
\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} f_{ui} + \lambda \sum_{u=1}^{n} \sum_{i=1}^{m} \tilde{y}_{ui} \tilde{f}_{ui} \tag{7}
\]

where \( f_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \text{reg}(U_u, V_i, b_u, b_i, P, N) \) and \( \tilde{f}_{ui} = \frac{1}{2}(r_{ui} - \hat{\tilde{r}}_{ui})^2 + \text{reg}(W_u, V_i) \).
Gradient for Grade Scores (1/2)

Denoting $f_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \text{reg}(U_. , V_. , b_u , b_i , P , N)$, we have,

\begin{align*}
\nabla \mu &= \frac{\partial f_{ui}}{\partial \mu} = -e_{ui}, \quad (8) \\
\nabla b_u &= \frac{\partial f_{ui}}{\partial b_u} = -e_{ui} + \beta_u b_u, \quad (9) \\
\nabla b_i &= \frac{\partial f_{ui}}{\partial b_i} = -e_{ui} + \beta_v b_i, \quad (10) \\
\nabla U_u . &= \frac{\partial f_{ui}}{\partial U_u .} = -e_{ui} V_. + \alpha_u U_u ., \quad (11) 
\end{align*}
Gradient for Grade Scores (2/2)

\[ \nabla V_i. = \frac{\partial f_{ui}}{\partial V_i.} = -e_{ui}(\rho U_u. + (1 - \rho) W_u. + \tilde{P}_u. + \tilde{N}_u.) + \alpha_v V_i., \quad (12) \]

\[ \nabla P_j. = \frac{\partial f_{ui}}{\partial P_j.} = \delta_P(-e_{ui} w_p \frac{1}{\sqrt{|P_u|}} V_i. + \alpha_p P_j.), j \in P_u, \quad (13) \]

\[ \nabla N_j. = \frac{\partial f_{ui}}{\partial N_j.} = \delta_N(-e_{ui} w_n \frac{1}{\sqrt{|N_u|}} V_i. + \alpha_n N_j.), j \in N_u, \quad (14) \]

where \( e_{ui} = (r_{ui} - \hat{r}_{ui}) \), and \( \rho U_u. + (1 - \rho) W_u. \) is used to introduce rich interactions as that in iTCF [Pan and Ming, 2014].
Method

Gradient for Binary Ratings

Denoting $\tilde{f}_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \text{reg}(W_u, V_i)$, we have,

$$\nabla W_u = \frac{\partial \tilde{f}_{ui}}{\partial W_u} = \lambda(-\tilde{e}_{ui} V_i + \alpha_w W_u), \quad (15)$$

$$\nabla V_i = \frac{\partial \tilde{f}_{ui}}{\partial V_i} = \lambda(-\tilde{e}_{ui}(\rho W_u + (1 - \rho) U_u) + \alpha_v V_i), \quad (16)$$

where $\tilde{e}_{ui} = (\tilde{r}_{ui} - \hat{r}_{ui})$. Note that “dislike” is converted as $\tilde{r}_{ui} = 1$ and “like” is converted as $\tilde{r}_{ui} = 5$. 

Pan et al., (CSSE, SZU)
We have the update rules,

\[ \theta = \theta - \gamma \nabla \theta, \]

(17)

where \( \gamma \) is the learning rate, and \( \theta \) can be \( \mu, b_u, b_i, U_u, V_i, W_u, P_j, N_j \).
Algorithm

1: Initialization of parameters $\Theta$.
2: for $t = 1, 2, \ldots, T$ do
3:     for iter $= 1, 2, \ldots, |R| + |\tilde{R}|$ do
4:         Randomly pick up a rating from $R$ and $|\tilde{R}|$.
5:           if $y_{ui} = 1$ then
6:             Calculate the gradients via Eq.(8-14).
7:           end if
8:           if $\tilde{y}_{ui} = 1$ then
9:             Calculate the gradients via Eq.(15-16).
10:          end if
11:     Update the parameters via Eq.(17).
12: end for
13: Decrease the learning rate $\gamma = \gamma \times 0.9$.
14: end for

Figure: The algorithm of transfer by mixed factorization (TMF).
(\lambda, \rho, \delta_P, \delta_N) = (0, 1, 0, 0): RSVD
(\lambda, \rho, \delta_P, \delta_N) = (1, 1, 0, 0): CMF
(\lambda, \rho, \delta_P, \delta_N) = (1, 0.5, 0, 0): iTCF
(\lambda, \rho, \delta_P, \delta_N) = (1, 0.5, 1, 0): TMF+
(\lambda, \rho, \delta_P, \delta_N) = (1, 0.5, 0, 1): TMF-
(\lambda, \rho, \delta_P, \delta_N) = (1, 0.5, 1, 1): TMF, default weight \(w_p = 2, w_n = 1\)

Note that TMF++ can also be achieved in a similar way to SVD++ via a union of positive feedbacks and negative feedbacks without distinction.
We use two data sets from iTCF [Pan and Ming, 2014], including MovieLens10M (denoted as ML10M) and Flixter, each of which contains five copies of (i) target grade score records in the form of $(u, i, r_{ui})$ with $r_{ui} \in \mathbb{G} = \{0.5, 1, 1.5, \ldots, 5\}$, (ii) auxiliary binary rating records in the form of $(u, i, \tilde{r}_{ui})$ with $\tilde{r}_{ui} \in \mathbb{B} = \{\text{like}, \text{dislike}\}$, and test grade score records in the form of $(u, i, r_{ui})$ with $r_{ui} \in \mathbb{G}$.

Note that the auxiliary binary rating records are constructed via converting ratings larger than or equal to 4 as likes and the others as dislikes iTCF [Pan and Ming, 2014].
Datasets (2/2)

- The ML10M data set is associated with \( n = 71,567 \) users and \( m = 10,681 \) items. Each copy contains \( |\mathcal{R}| = 4,000,022 \) target records, \( |\tilde{\mathcal{R}}| = 4,000,022 \) auxiliary records and \( |\mathcal{T}_E| = 2,000,010 \) test records, where the ratio is \( |\mathcal{R}| : |\tilde{\mathcal{R}}| : |\mathcal{T}_E| = 2 : 2 : 1 \).

- The Flixter data set is associated with \( n = 147,612 \) users and \( m = 48,794 \) items. Each copy contains \( |\mathcal{R}| = 3,278,431 \) target records, \( |\tilde{\mathcal{R}}| = 3,278,431 \) auxiliary records and \( |\mathcal{T}_E| = 1,639,215 \) test records, where the ratio is also \( |\mathcal{R}| : |\tilde{\mathcal{R}}| : |\mathcal{T}_E| = 2 : 2 : 1 \).
Baselines

- **Average filling (AF)**: $\hat{r}_{ui} = b_u + b_i + \mu$;
- **RSVD** [Koren, 2008] approximates an observed target grade score via learning some latent variables of the corresponding user and item, which works well for data of grade scores;
- **CMF** [Singh and Gordon, 2008] extends RSVD via sharing items’ latent variables for the target grade scores and the auxiliary binary ratings, which can be applied to our collaborative recommendation with heterogeneous explicit feedbacks (CR-HEF) problem; and
- **iTCF** [Pan and Ming, 2014] further extends CMF via introducing interactions between users’ latent variables, which was reported to be more accurate than CMF.
Initialization of Model Parameters

We use the statistics of training data $\mathbf{R}$ to initialize the model parameters,

\[
\mu = \frac{\sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} r_{ui}}{\sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui}}\]

\[
b_u = \frac{\sum_{i=1}^{m} y_{ui} (r_{ui} - \mu)}{\sum_{i=1}^{m} y_{ui}}\]

\[
b_i = \frac{\sum_{u=1}^{n} y_{ui} (r_{ui} - \mu)}{\sum_{u=1}^{n} y_{ui}}\]

\[
U_{uk} = (r - 0.5) \times 0.01, \ k = 1, \ldots, d\]

\[
V_{ik} = (r - 0.5) \times 0.01, \ k = 1, \ldots, d\]

\[
P_{jk} = (r - 0.5) \times 0.01, \ k = 1, \ldots, d\]

\[
N_{jk} = (r - 0.5) \times 0.01, \ k = 1, \ldots, d, \ \text{if} \ \lambda \neq 0\]

\[
W_{jk} = (r - 0.5) \times 0.01, \ k = 1, \ldots, d, \ \text{if} \ \lambda \neq 0\]

where $r$ (0 $\leq$ $r$ $<$ 1) is a random variable.
**Parameter Settings**

- The weight on auxiliary data: $\lambda = 1$
- The weight for interactions: $\rho = 0.5$ [Pan and Ming, 2014]
- Boolean variables: $\delta_p, \delta_u \in \{0, 1\}$, $\delta_p = 1$, $\delta_u = 1$ for TMF
- The weight on positive feedbacks and negative feedbacks: $w_p = 2$, $w_n = 1$
- The tradeoff parameters: $\alpha_u = \alpha_v = \alpha_w = \alpha_p = \alpha_n = 0.01$, $\beta_u = \beta_v = 0.01$ [Pan and Ming, 2014]
- The learning rate $\gamma$ is decreased every iteration $t$: $\gamma = \gamma \times 0.9$, $\gamma$ is initialized as 0.01 [Pan and Ming, 2014]
- The number of latent dimensions: $d = 20$ for ML10M, $d = 10$ for Flixter [Pan and Ming, 2014]
- Iteration number: $T = 50$ [Pan and Ming, 2014]
Data Structure

- Grade scores
  - $(u, i)$ pairs in a 2-D array

- Binary ratings
  - $u \rightarrow \mathcal{P}_u$
  - $u \rightarrow \mathcal{N}_u$
Post-Processing

- For the rating range (grade score set) \( G = \{1, 2, 3, 4, 5\} \)
  - If \( \hat{r}_{ui} > 5 \), \( \hat{r}_{ui} = 5 \)
  - If \( \hat{r}_{ui} < 1 \), \( \hat{r}_{ui} = 1 \)

- For the rating range (grade score set) \( G = \{0.5, 1, 1.5, \ldots, 5\} \)
  - If \( \hat{r}_{ui} > 5 \), \( \hat{r}_{ui} = 5 \)
  - If \( \hat{r}_{ui} < 0.5 \), \( \hat{r}_{ui} = 0.5 \)

Pan et al., (CSSE, SZU)
Transfer by Mixed Factorization (TMF)
Evaluation Metrics

- Mean Absolute Error (MAE)

\[
MAE = \frac{\sum_{(u,i,r_{ui}) \in \mathcal{T}_E} \left| r_{ui} - \hat{r}_{ui} \right|}{|\mathcal{T}_E|}
\]

- Root Mean Square Error (RMSE)

\[
RMSE = \sqrt{\frac{\sum_{(u,i,r_{ui}) \in \mathcal{T}_E} (r_{ui} - \hat{r}_{ui})^2}{|\mathcal{T}_E|}}
\]

- Performance: the smaller the better.
Table: Prediction performance on MAE and RMSE (d = 20 for ML10M and d = 10 for Flixter).

<table>
<thead>
<tr>
<th>Data</th>
<th>Algorithm</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML10M</td>
<td>AF</td>
<td>0.6766 ± 0.0006</td>
<td>0.8735 ± 0.0007</td>
</tr>
<tr>
<td></td>
<td>RSVD</td>
<td>0.6438 ± 0.0011</td>
<td>0.8364 ± 0.0012</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.6334 ± 0.0012</td>
<td>0.8273 ± 0.0013</td>
</tr>
<tr>
<td></td>
<td>iTCF</td>
<td>0.6197 ± 0.0006</td>
<td>0.8091 ± 0.0008</td>
</tr>
<tr>
<td></td>
<td>TMF</td>
<td><strong>0.6124 ± 0.0007</strong></td>
<td><strong>0.8005 ± 0.0008</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flixter</td>
<td>AF</td>
<td>0.6867 ± 0.0005</td>
<td>0.9128 ± 0.0007</td>
</tr>
<tr>
<td></td>
<td>RSVD</td>
<td>0.6561 ± 0.0007</td>
<td>0.8814 ± 0.0010</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.6423 ± 0.0009</td>
<td>0.8710 ± 0.0012</td>
</tr>
<tr>
<td></td>
<td>iTCF</td>
<td>0.6373 ± 0.0005</td>
<td>0.8636 ± 0.0010</td>
</tr>
<tr>
<td></td>
<td>TMF</td>
<td><strong>0.6348 ± 0.0007</strong></td>
<td><strong>0.8615 ± 0.0012</strong></td>
</tr>
</tbody>
</table>
Observations

- The overall performance ordering of the studied methods is $AF < RSVD < CMF < iTCF < TMF$, which demonstrates the effectiveness of the factorization-based methods (as compared with the average filling baseline), in particular of the designed mixed factorization approach.

- TMF is **significantly better** than all other methods on both data sets (the p-value\(^1\) is smaller than 0.01).

\(^1\)http://www.mathworks.com/help/stats/ttest2.html
Table: Prediction performance of algorithm variants of the SVD family and TMF family on ML10M and Flixter, which are associated with different configurations, including (i) “++” for a union set of both positive and negative feedbacks without distinction, i.e., \( \mathcal{P}_u \cup \mathcal{N}_u \) for user \( u \), (ii) “−” for a set of negative feedbacks only, (iii) “+” for a set of positive feedbacks only, and (iv) “+-” for one set of positive feedbacks and one set of negative feedbacks.

<table>
<thead>
<tr>
<th>Data</th>
<th>Metric</th>
<th>Conf.</th>
<th>SVD family</th>
<th>TMF family</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \text{MAE} )</td>
<td>( \text{RMSE} )</td>
</tr>
<tr>
<td>ML10M</td>
<td></td>
<td>++</td>
<td>0.6285 ± 0.0006</td>
<td>0.6169 ± 0.0006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.6305 ± 0.0006</td>
<td>0.6176 ± 0.0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>0.6233 ± 0.0005</td>
<td>0.6152 ± 0.0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>++</td>
<td>0.6206 ± 0.0006</td>
<td>\textbf{0.6124} ± 0.0007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.8187 ± 0.0007</td>
<td>0.8058 ± 0.0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>0.8211 ± 0.0009</td>
<td>0.8066 ± 0.0006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>++</td>
<td>0.8123 ± 0.0007</td>
<td>0.8036 ± 0.0007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.8093 ± 0.0008</td>
<td>\textbf{0.8005} ± 0.0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>0.6494 ± 0.0008</td>
<td>0.6373 ± 0.0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.6479 ± 0.0008</td>
<td>0.6373 ± 0.0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>0.6456 ± 0.0012</td>
<td>0.6366 ± 0.0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>++</td>
<td>0.6422 ± 0.0011</td>
<td>\textbf{0.6348} ± 0.0007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.8747 ± 0.0009</td>
<td>0.8635 ± 0.0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>0.8734 ± 0.0008</td>
<td>0.8633 ± 0.0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>++</td>
<td>0.8709 ± 0.0013</td>
<td>0.8626 ± 0.0012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−</td>
<td>0.8680 ± 0.0011</td>
<td>\textbf{0.8615} ± 0.0012</td>
</tr>
</tbody>
</table>
Observations

- The overall performance ordering of the algorithms with different configurations from either the SVD family or the TMF family is 
  \[ "++" \approx "-" < "+" < "+-" ]; \]
- A simple combination of positive feedbacks and negative feedbacks without distinction is harmful since the configuration 
  \[ "++" \] does not perform well;
- Positive feedbacks are more useful than negative feedbacks in modeling users’ preferences, which can be explained by the fact 
  that users usually prefer to assign grade scores to liked items than to disliked items, e.g., the global average preference scores of 
  ML10M and Flixter are 3.51 and 3.61, respectively; and
- Positive feedbacks and negative feedbacks are complementary for the prediction performance because the algorithm variant with 
  configuration \[ "+-" \] is better than that with either \[ "+" \] or \[ "-" \] on both ML10M and Flixter.
Conclusions

- We propose a novel method, i.e., transfer by mixed factorization (TMF), for collaborative recommendation with heterogeneous explicit feedbacks (CR-HEF).
- TMF is able to model users’ preferences more accurately via transferring more feedback-independent knowledge than either collective factorization or integrative factorization alone.
- TMF can leverage each part of auxiliary feedbacks significantly better than the state-of-the-art method.
Future Work

- We are mainly interested in generalizing our TMF with temporal context in a multi-objective oriented optimization manner.
Thank you!

- We thank the support of Natural Science Foundation of Guangdong Province No. 2014A030310268, National Natural Science Foundation of China (NSFC) No. 61502307, 61170077, 61272303, and Natural Science Foundation of SZU No. 201436.

- We also thank the editors and reviewers for their constructive and expert comments, and our colleague George Basker for his help on linguistic quality improvement.
