FCMF: Federated Collective Matrix Factorization for Heterogeneous Collaborative Filtering

Enyue Yang#, Yunfeng Huang#, Feng Liang, Weike Pan*, Zhong Ming*

National Engineering Laboratory for Big Data System Computing Technology
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
College of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China
Outline

1. Introduction
2. Methods
3. Experiments
4. Conclusions and Future Work
5. Acknowledgement
Figure: Illustration of the studied problem of heterogeneous collaborative filtering (HCF), including a target matrix $\mathcal{R} = \{(u, i, r_{ui})\}$ decomposed into two low-rank matrices $U$ and $V$, and an auxiliary matrix $\tilde{\mathcal{R}} = \{(u, i, \tilde{r}_{ui})\}$ decomposed into two low-rank matrices $\tilde{Z}$ and $V$. 

\[ \mathcal{R} \sim UV^T \]

\[ \tilde{\mathcal{R}} \sim \tilde{Z}V^T \]
Problem Definition

- **Input:**
  - A target numerical rating matrix or a set of numerical rating records
    \[ R = \{ (u, i, r_{ui}) \} \]
  - An auxiliary binary rating matrix or a set of binary rating records
    \[ \tilde{R} = \{ (u, i, \tilde{r}_{ui}) \} \].

- **Goal:** Predict the unobserved rating score of a user \( u \) to an item \( i \) in the target matrix by leveraging the auxiliary binary rating matrix in a privacy-preserving manner.

- **Privacy:**
  - the original rating records in \( R \) and \( \tilde{R} \)
  - the learned user-specific latent feature vectors
  - the learned item-specific latent feature vectors (less sensitive)
Overall of Our Solution (1/3)

Step 1. Train $V_r$ at local

Step 2. Encryption via HE public key

Step 3. Train $[[V_r]]$ at local

Step 4. Interactions

Step 5. Decryption via HE private key
Overall of Our Solution (2/3)

Step 1  The auxiliary part randomly initializes the model parameters with some small random values $V_i, i = 1, 2, \cdots, m$ and $\tilde{Z}_u, u = 1, 2, \cdots, n$, and uses the local rating records to train $V_i$ and $\tilde{Z}_u$ fully.

Step 2  The auxiliary part uses homomorphic encryption to encrypt $V_i$ and then sends $[[V_i]]$ to the target part.

Step 3  The target part receives $[[V_i]]$, initializes the model parameters with some small random values $\mu, b_u, b_i$ and $U_u$, and then uses the local rating records to train the model parameters via Step 4 and Step 5.

Step 4  The target part sends the intermediate parameters $[[e_{ui}]] = r_{ui} - [[\tilde{r}_{ui}]]$ and $e_{ui}[[V_i]]$ to the auxiliary part and receives the decrypted value $e_{ui}$ and $e_{ui}V_i$.

Step 5  The auxiliary part receives a series of parameters, and then decrypts and sends them to the target part. Specially, the auxiliary part adds Laplace noise to the value of $e_{ui}V_i$ via differential privacy.
We choose to share the model parameters $V_i$, $i = 1, 2, \ldots, m$, which are less sensitive than $U_u$, $u = 1, 2, \ldots, n$.

- We make use of **homomorphic encryption** and **differential privacy** to protect the privacy.
## Notations (1/3)

Table: Some Notations and explanations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>the number of users</td>
</tr>
<tr>
<td>$m$</td>
<td>the number of items</td>
</tr>
<tr>
<td>$u \in {1, 2, \ldots, n}$</td>
<td>user ID</td>
</tr>
<tr>
<td>$i \in {1, 2, \ldots, m}$</td>
<td>item ID</td>
</tr>
<tr>
<td>$r_{ui} \in {1, 2, \ldots, 5}$</td>
<td>the target grade score of user $u$ to item $i$</td>
</tr>
<tr>
<td>$\tilde{r}_{ui} \in {1, -1}$</td>
<td>the auxiliary binary rating of user $u$ to item $i$</td>
</tr>
<tr>
<td>$\mathcal{R} = {(u, i, r_{ui})}$</td>
<td>the set of target grade score records</td>
</tr>
<tr>
<td>$\tilde{\mathcal{R}} = {(u, i, \tilde{r}_{ui})}$</td>
<td>the set of auxiliary binary rating records</td>
</tr>
</tbody>
</table>
**Table:** Some Notations and explanations (cont.).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>the number of latent dimensions</td>
</tr>
<tr>
<td>$\mu \in \mathbb{R}$</td>
<td>the global average rating</td>
</tr>
<tr>
<td>$b_u \in \mathbb{R}$</td>
<td>the bias of user $u$</td>
</tr>
<tr>
<td>$b_i \in \mathbb{R}$</td>
<td>the bias of item $i$</td>
</tr>
<tr>
<td>$\hat{r}_{ui}$</td>
<td>the predicted target grade score of user $u$ to item $i$</td>
</tr>
<tr>
<td>$\tilde{r}_{ui}$</td>
<td>the predicted auxiliary binary rating of user $u$ to item $i$</td>
</tr>
<tr>
<td>$U_u. \in \mathbb{R}^{1 \times d}$</td>
<td>the latent feature vector of user $u$ in the target matrix</td>
</tr>
<tr>
<td>$\tilde{Z}_u. \in \mathbb{R}^{1 \times d}$</td>
<td>the latent feature vector of user $u$ in the auxiliary matrix</td>
</tr>
<tr>
<td>$V_i. \in \mathbb{R}^{1 \times d}$</td>
<td>the latent feature vector of item $i$</td>
</tr>
</tbody>
</table>
### Notations (3/3)

**Table:** Some Notations and explanations (cont.).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T, \tilde{T}$</td>
<td>the number of iterations in training</td>
</tr>
<tr>
<td>$\alpha, \beta$</td>
<td>the tradeoff parameters</td>
</tr>
<tr>
<td>$\gamma, \tilde{\gamma}$</td>
<td>the learning rate</td>
</tr>
<tr>
<td>$[[\ast]]$ or $E[\ast]$</td>
<td>encryption with a public key</td>
</tr>
<tr>
<td>$D[\ast]$</td>
<td>decryption with a private key</td>
</tr>
</tbody>
</table>
Methods

Prediction Rule of FCMF in the Target Matrix

The predicted grade score of user $u$ to item $i$ in the target matrix (in the training phase),

$$[[\hat{r}_{ui}]] = \mu + b_u + b_i + U_u.[[V_i.]]^T,$$

(1)

where $V_i. \in \mathbb{R}^{1 \times d}$ is the item-specific latent feature vector of item $i$, $U_u. \in \mathbb{R}^{1 \times d}$ is the user-specific latent feature vector of user $u$, and $\hat{r}_{ui}$ is the predicted grade score.

The predicted grade score of user $u$ to item $i$ in the target matrix (in the test phase),

$$\hat{r}_{ui} = \mu + b_u + b_i + U_u. V_i.\hat{T}.$$  

(2)
The predicted rating of user $u$ to item $i$ in the auxiliary matrix,

$$\hat{r}_{ui} = \tilde{Z}_u \cdot V_i^T,$$

(3)

where $V_i \in \mathbb{R}^{1 \times d}$ is the item-specific latent feature vector, and $\tilde{Z}_u \in \mathbb{R}^{1 \times d}$ is the user-specific latent feature vector.
Objective Function of FCMF in the Target Matrix

The objective function for factorization of the target matrix,

$$\min_{\Theta} \sum_{(u,i,r_{ui}) \in \mathcal{R}} f_{ui},$$

where

$$f_{ui} = \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha}{2} (||U_u||^2 + ||[V_i]||^2 + ||b_u||^2 + ||b_i||^2),$$

and $\Theta$ is the set of model parameters for the target matrix, i.e., $U_u$, $[V_i]$, $b_u$, $b_i$ and $\mu$ with $u = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, m$. 
Objective Function of FCMF in the Auxiliary Matrix

The objective function for factorization of the auxiliary matrix,

$$\min_\Phi \sum_{(u,i,\tilde{r}_{ui}) \in \tilde{R}} \tilde{f}_{ui},$$

(5)

where $$\tilde{f}_{ui} = \frac{1}{2}(\tilde{r}_{ui} - \hat{r}_{ui})^2 + \frac{\beta}{2}(||\tilde{Z}_u||^2 + ||V_i||^2)$$, and $$\Phi$$ is the set of model parameters for the auxiliary matrix, i.e., $$\tilde{Z}_u$$ and $$V_i$$ with $$u = 1, 2, \ldots, n$$ and $$i = 1, 2, \ldots, m$$. 
Gradients of FCMF in the Target Matrix

\[ \nabla \mu = \frac{\partial f_{ui}}{\partial \mu} = -e_{ui}, \]  
(6)

\[ \nabla b_u = \frac{\partial f_{ui}}{\partial b_u} = -e_{ui} + \alpha b_u, \]  
(7)

\[ \nabla b_i = \frac{\partial f_{ui}}{\partial b_i} = -e_{ui} + \alpha b_i, \]  
(8)

\[ \nabla U_u. = \frac{\partial f_{ui}}{\partial U_u.} = -D[e_{ui}[[V_i.]]] + \alpha U_u., \]  
(9)

\[ \nabla[[V_i.]] = \frac{\partial f_{ui}}{\partial[[V_i.]]} = -e_{ui}U_u. + \alpha[[V_i.]], \]  
(10)

where \( e_{ui} = D[r_{ui} - [[\hat{r}_{ui}]]] \).
Gradients of FCMF in the Auxiliary Matrix

\[
\nabla \tilde{Z}_u = \frac{\partial \tilde{f}_{ui}}{\partial \tilde{Z}_u} = -\tilde{e}_{ui} V_i + \beta \tilde{Z}_u, \quad (11)
\]

\[
\nabla V_i = \frac{\partial \tilde{f}_{ui}}{\partial V_i} = -\tilde{e}_{ui} \tilde{Z}_u + \beta V_i, \quad (12)
\]

where \(\tilde{e}_{ui} = \tilde{r}_{ui} - \hat{r}_{ui}\).
The update rule for factorization of the target matrix,

$$\theta = \theta - \gamma \nabla \theta,$$  \hspace{1cm} (13)

where $\gamma$ is the learning rate and $\theta \in \Theta$ is the model parameter.
The update rule for factorization of the auxiliary matrix,

\[ \phi = \phi - \tilde{\gamma} \nabla \phi, \quad (14) \]

where \( \tilde{\gamma} \) is the learning rate and \( \phi \in \Phi \) is the model parameter.
The Learning Algorithm (1/3)

The process of our FCMF mainly contains two parts:

(i) **The auxiliary part**: it uses the original binary matrix to train and update the items’ latent matrix $V$, and then encrypts $V$ and obtains $[[V]]$. Finally, the auxiliary part sends $[[V]]$ to the target part.

(ii) **The target part**: it receives the ciphertexts of the items’ latent feature vectors, and then uses its own rating matrix to train the model parameters.
## Methods

### The Learning Algorithm (2/3)

**Algorithm 1** The algorithm of FCMF in the auxiliary part.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td><strong>Input:</strong> A set of binary rating records $\tilde{R}$, the dimension of the latent feature vector $d$, the regularization parameter $\beta$, the number of iterations in training $\tilde{T}$, and the learning rate $\tilde{\gamma}$.</td>
</tr>
<tr>
<td>2:</td>
<td><strong>Output:</strong> The encrypted items' feature matrix $[[V_i]]$, $i = 1, 2, \ldots, m$.</td>
</tr>
<tr>
<td>3:</td>
<td>Initialize $\tilde{Z}_u$ and $V_i$, $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$.</td>
</tr>
<tr>
<td>4:</td>
<td>Create a public key and a private key of Paillier.</td>
</tr>
<tr>
<td>5:</td>
<td>for $t_1 = 1, \ldots, \tilde{T}$ do</td>
</tr>
<tr>
<td>6:</td>
<td>for $t_2 = 1, \ldots,</td>
</tr>
<tr>
<td>7:</td>
<td>Randomly pick up a record $(u, i, \tilde{r}_{ui}) \in \tilde{R}$.</td>
</tr>
<tr>
<td>8:</td>
<td>Calculate the gradients, i.e., $\nabla \tilde{Z}_u$ and $\nabla V_i$, via Eqs.(11-12).</td>
</tr>
<tr>
<td>9:</td>
<td>Update the model parameters, i.e., $\tilde{Z}_u$ and $\tilde{V}_i$, via Eq.(14).</td>
</tr>
<tr>
<td>10:</td>
<td>end for</td>
</tr>
<tr>
<td>11:</td>
<td>$\tilde{\gamma} = \tilde{\gamma} \times 0.9$.</td>
</tr>
<tr>
<td>12:</td>
<td>end for</td>
</tr>
<tr>
<td>13:</td>
<td>Encrypt $V_i$ and send $[[V_i]]$ to the target part, $i = 1, 2, \ldots, m$.</td>
</tr>
</tbody>
</table>
The Learning Algorithm (3/3)

Algorithm 2 The algorithm of FCMF in the target part.

1: **Input:** A set of grade score records $R$, the dimension of the latent feature vector $d$, the regularization parameter $\alpha$, the number of iterations in training $T$, and the learning rate $\gamma$.
2: **Output:** Predicted grade score of a user $u$ to any unrated item $i$, i.e., $\hat{r}_{ui}$, in the target matrix.
3: Receive $[[V_i.]]$, $i = 1, 2, \ldots, m$, from the auxiliary part.
4: Initialize $U_u \cdot$, $b_i$, $b_u$, $\mu$, $u = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$.
5: for $t_1 = 1, \ldots, T$ do
6: 
7:  for $t_2 = 1, \ldots, |R|$ do
8:  Randomly pick up a record $(u, i, r_{ui}) \in R$.
9:  Calculate $[[\hat{r}_{ui}]]$ via Eq.(1).
10:  Send $[[e_{ui}]] = r_{ui} - [[\hat{r}_{ui}]]$ to the auxiliary part for decryption and get back $e_{ui}$.
11:  Send $e_{ui}[[V_i.]]$ to the auxiliary part and get back the decrypted result with Laplace noise.
12:  Calculate the gradients, i.e., $\nabla \mu$, $\nabla b_u$, $\nabla b_i$, $\nabla U_u$, and $\nabla [[V_i.]]$, via Eqs.(6-10).
13:  Update the model parameters, i.e., $\mu$, $b_u$, $b_i$, $U_u$, and $[[V_i.]]$, via Eq.(13).
14: end for
15: $\tilde{\gamma} = \tilde{\gamma} \times 0.9$.
16: end for
17: Send $[[V_i.]]$ to the auxiliary part for decryption and obtain $V_i$, $i = 1, \ldots, m$. 
Security Analysis (1/4)

- **Step 1.** The **auxiliary part** trains the model via the local data, and all the calculation operations are performed locally in the auxiliary part without any information transmission or information leakage.

- **Step 2.** The auxiliary part transmits the **ciphertext of the latent feature vectors** $[[V_i]], i = 1, 2, \ldots, m$, to the target part. The target part cannot decrypt the items' latent feature vectors without the private key. Hence, there is no information leakage in this step.
Step 3. The **target part** receives the ciphertext of the items’ latent feature vectors $[\left[V_i.\right]]$, $i = 1, 2, \ldots, m$, and trains the model locally. In order to calculate the gradients, some intermediate results need to be sent back to the auxiliary part for decryption.

- **The error between the true rating score and the ciphertext of the predicted rating score** $r_{ui} - [\left[\hat{r}_{ui}\right]]$. When $r_{ui} - [\left[\hat{r}_{ui}\right]]$ is sent to the auxiliary part, it is a pure number without any information about user ID and item ID. Therefore, even if the auxiliary part can calculate the error between the true rating score and the predicted rating score, it could not obtain the user ID and item ID corresponding to this value. Thus, private information of the target part cannot be obtained by the auxiliary part.
ii. **The ciphertext of the intermediate result** $e_{ui}[[V_i.]]$. The decryption of the auxiliary part $e_{ui} V_i$ is still a numeric vector without user ID and item ID, so the auxiliary part could not get any private information from the target part. Also, the auxiliary part adds a random noise to the decrypted numerical vector, which prevents the target part from calculating the latent feature vector $V_i$ by $e_{ui} V_i$ and $e_{ui}$ that are decrypted previously.

In addition, all the original rating data and model parameters are kept locally at the target part, so there is no privacy leak.
Security Analysis (4/4)

- **Step 4.** The target part transmits the ciphertext of the item’s latent feature vector $[[V_i,]]$ to the auxiliary part. Different from Step 2, because $[[V_i,]]$ at this time has already been combined with the item information of the target part and the auxiliary part, even if the auxiliary part decrypts the latent feature vector $V_i$, it still cannot extract the items’ latent feature vectors at the target part. Therefore, it does not involve the disclosure of the private information.

- **Step 5.** The auxiliary part transmits the decryption result $V_i$ to the target part. Similarly, the target part cannot extract the items’ latent feature vectors at the auxiliary part, so there is no leakage of private information.
### Datasets

**Table:** Statistics of the first copy of the four datasets used in our experiments. Notice that the target part and the auxiliary part are equal in the number of users and items.

|          | n   | m   | |R|   | |R̂|   | |R^le|   | |R|/(nm) |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ML100K   | 943 | 1,682| 40000 | 40000 | 20000 | 2.522% |
| ML1M     | 6,040 | 3,952| 400084 | 400084 | 200042 | 1.676% |
| ML10M    | 71,567 | 10,681| 4000022 | 4000022 | 2000011 | 0.523% |
| NF       | 50,000 | 17,770| 4177000 | 4177002 | 2088500 | 0.470% |

Moreover, the scoring range of ML100K, ML1M and NF is \{1, 2, 3, 4, 5\}, and that of ML10M is \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}. 
Parameter Configurations

- The learning rate: $\gamma = \tilde{\gamma} = 0.01$;
- The iteration number: $T = \tilde{T} = 50$;
- The number of latent dimensions: $d = 20$ for ML100K, ML1M and ML10M, $d = 10$ for Flixster;
- The tradeoff parameter: we search the best value of $\alpha, \beta$ from $\{0.001, 0.01, 0.1\}$ using the first copy of each dataset and the RMSE metric.
For the original binary rating auxiliary matrix in training:
  if $\tilde{r}_{ui} = 1$, $\tilde{r}_{ui} = 5$,
  if $\tilde{r}_{ui} = -1$, $\tilde{r}_{ui} = 1$.

For predictions:
  For the predicted rating in ML100K and ML1M:
    if $\hat{r}_{ui} < 1$, $\hat{r}_{ui} = 1$,
    if $\hat{r}_{ui} > 5$, $\hat{r}_{ui} = 5$.
  For the predicted rating in ML10M and Flixster:
    if $\hat{r}_{ui} < 0.5$, $\hat{r}_{ui} = 0.5$,
    if $\hat{r}_{ui} > 5$, $\hat{r}_{ui} = 5$. 
Evaluation Metrics

- Mean Absolute Error (MAE):
  \[
  MAE = \frac{\sum_{(u,i,r_{ui}) \in \mathcal{R}_{te}} |r_{ui} - \hat{r}_{ui}|}{|\mathcal{R}_{te}|} \tag{15}
  \]

- Root Mean Square Error (RMSE):
  \[
  RMSE = \sqrt{\frac{\sum_{(u,i,r_{ui}) \in \mathcal{R}_{te}} (r_{ui} - \hat{r}_{ui})^2}{|\mathcal{R}_{te}|}} \tag{16}
  \]
In this subsection, we study whether Paillier homomorphic encryption (HE) would affect the recommendation accuracy. In particular, we compare the experimental results of FCMF with and without HE.

- **FCMF with HE.** The shared feature vector $V_i$ is encrypted by Paillier homomorphic encryption.

- **FCMF without HE.** Without Paillier homomorphic encryption, the target part can directly calculate the gradients, and does not need to exchange any information in the training process. Therefore, we simplify the algorithmic process that the auxiliary part fully trains the latent feature vector $V_i$, and then transmits them directly to the target part for further training.
Impact of Homomorphic Encryption (2/2)

Table: Recommendation performance with and without homomorphic encryption on ML100K.

<table>
<thead>
<tr>
<th>Baseline</th>
<th>$\alpha, \beta$</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCMF with HE</td>
<td>0.001</td>
<td>0.7459±0.0043</td>
<td>0.9508±0.0051</td>
</tr>
<tr>
<td>FCMF without HE</td>
<td>0.001</td>
<td>0.7481±0.0045</td>
<td>0.9523±0.0053</td>
</tr>
</tbody>
</table>

Observations:
- The influence of Paillier homomorphic encryption in FCMF on the recommendation results are within the normal error fluctuation range. In other words, the experimental results of FCMF with HE are equivalent to the one without HE.
- This also indicates that Paillier homomorphic encryption is a reliable and effective security technology, which ensures that the private information of the target part and the auxiliary part are not disclosed and also the recommendation accuracy is not affected.
In this subsection, we study the influence of Laplace noise on the prediction accuracy of FCMF.

We study the trend of the recommendation accuracy with the change of the value of the privacy budget $\epsilon$.

In order to avoid the impact of homomorphic encryption on the experimental results and to improve the efficiency of the experiments, we choose the version without HE.

And we add Laplace noise after calculating the gradients of the users’ latent feature vectors in the target part:

- $\mu = 0$
- $b = \frac{T \Delta f}{\epsilon}$, where

$$\Delta f = \max_{Data, Data'} \sum_{f=1}^{d} |\nabla U_{uf}^{Data} - \nabla U_{uf}^{Data'}| \times \gamma = 0.01 \text{ and } T = 50$$
Impact of Differential Privacy (2/3)

(a) ML100K

(b) ML1M

(c) ML10M

(d) NF
Observations:

- With the increased values of the privacy budget $\epsilon$, the results become better.

- The value of $\epsilon = 2$ is a good balance between confidentiality and recommendation accuracy, which is suggested in real applications.
In this subsection, we compare the experimental results of our FCMF with RSVD and CMF:

- RSVD only makes use of the target rating data.
- CMF shares the latent feature vector $V_i$ between the target part and the auxiliary part.
- For our FCMF, because we have verified that the results of FCMF with HE are equivalent to that without HE, and the impact of a large value of the privacy budget $\epsilon$ can also be ignored, we use FCMF without HE and differential privacy for efficiency consideration.
Comparison with Baseline Methods (2/3)

Table: Recommendation performance of RSVD, CMF and our FCMF.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>$\alpha$, $\beta$</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML100K</td>
<td>RSVD</td>
<td>0.1</td>
<td>0.7600 ± 0.0051</td>
<td>0.9623 ± 0.0060</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.001</td>
<td>0.7528 ± 0.0049</td>
<td>0.9579 ± 0.0062</td>
</tr>
<tr>
<td></td>
<td>FCMF</td>
<td>0.001</td>
<td><strong>0.7481 ± 0.0045</strong></td>
<td><strong>0.9523 ± 0.0053</strong></td>
</tr>
<tr>
<td>ML1M</td>
<td>RSVD</td>
<td>0.001</td>
<td>0.7214 ± 0.0013</td>
<td>0.9143 ± 0.0016</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.01</td>
<td>0.6918 ± 0.0014</td>
<td>0.8820 ± 0.0016</td>
</tr>
<tr>
<td></td>
<td>FCMF</td>
<td>0.01</td>
<td><strong>0.6886 ± 0.0013</strong></td>
<td><strong>0.8803 ± 0.0016</strong></td>
</tr>
<tr>
<td>ML10M</td>
<td>RSVD</td>
<td>0.001</td>
<td>0.6386 ± 0.0005</td>
<td>0.8323 ± 0.0005</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.01</td>
<td>0.6330 ± 0.0007</td>
<td>0.8269 ± 0.0006</td>
</tr>
<tr>
<td></td>
<td>FCMF</td>
<td>0.01</td>
<td><strong>0.6287 ± 0.0006</strong></td>
<td><strong>0.8225 ± 0.0009</strong></td>
</tr>
<tr>
<td>NF</td>
<td>RSVD</td>
<td>0.001</td>
<td>0.6502 ± 0.0007</td>
<td>0.8769 ± 0.0005</td>
</tr>
<tr>
<td></td>
<td>CMF</td>
<td>0.01</td>
<td>0.6422 ± 0.0010</td>
<td>0.8703 ± 0.0013</td>
</tr>
<tr>
<td></td>
<td>FCMF</td>
<td>0.01</td>
<td><strong>0.6394 ± 0.0006</strong></td>
<td><strong>0.8678 ± 0.0008</strong></td>
</tr>
</tbody>
</table>
Observations:

- On all the four datasets, the recommendation performance of CMF and our FCMF are significantly better than that of RSVD, which shows that aggregating the two types of heterogeneous feedback is helpful.

- Compared with CMF, the recommendation performance of our FCMF is better in most cases. The reason is that during the entire training process, CMF will iteratively fit the score values of the auxiliary part, while our FCMF focuses more on fitting the true rating scores of the target part, which is more consistent with our goal of rating prediction in the target part.
Conclusions and Future Work

Conclusions (1/2)

- We study an important collaborative filtering problem with heterogeneous feedback and design a novel federated learning solution, i.e., federated collective matrix factorization (FCMF).
- In our FCMF, both parts keep the sensitive original data and the users’ latent feature vectors locally, and only share the low sensitive items’ latent feature vectors.
- We use the Paillier homomorphic encryption to encrypt the shared items’ latent feature vectors and add Laplace random noise to further ensure the security of the auxiliary part’s item information.
- We conduct a detailed security analysis in each step of our FCMF to verify the security of the private information during the entire training process.
Conclusions (2/2)

- Experimental results on different real-world datasets show that the introduction of the Paillier homomorphic encryption does not reduce the accuracy of our FCMF.
- We analyse the effectiveness of different values of the privacy budget $\epsilon$ in the Laplace mechanism.
Future Work

- We are interested in studying how to achieve knowledge transfer among different parts with small overlaps, as well as privacy-aware joint modeling in terms of both part-level and user-level.

- We are interested in generalizing the problem setting from two sets of heterogeneous feedback to two general totally-ordered sets.
Thank you!

We thank the handling editors and reviewers for their effort and constructive expert comments, and the support of National Natural Science Foundation of China Nos. 61872249 and 61836005.