

Practical and Secure Federated Recommendation with Personalized Mask

Liu Yang^{1,2}, Junxue Zhang^{1,2}, Di Chai^{1,2}, Leye Wang³, Kun Guo⁴, Kai Chen¹ and Qiang Yang¹

¹Hong Kong University of Science and Technology

²Clustar

³Peking University

⁴Fuzhou University

{lyangau, jzhangcs, dchai, kaichen, qyang}@cse.ust.hk, leyewang@pku.edu.cn, gukn@fzu.edu.cn

Abstract

Federated recommendation addresses the data silo and privacy problems altogether for recommender systems. Current federated recommender systems mainly utilize cryptographic or obfuscation methods to protect the original ratings for leakage. However, the former comes with extra communication and computation costs, and the latter damages model accuracy. Neither of them could simultaneously satisfy the real-time feedback and accurate personalization requirements of recommender systems. In this paper, we proposed federated masked matrix factorization (FedMMF) to protect the data privacy in federated recommender systems without sacrificing efficiency and effectiveness. In more details, we introduce the new idea of personalized mask generated only from local data and apply it in FedMMF. On the one hand, personalized mask offers protection for participants' private data without effectiveness loss. On the other hand, combined with the adaptive secure aggregation protocol, personalized mask could further improve efficiency. Theoretically, we provide security analysis for personalized mask. Empirically, we also show the superiority of the designed model on different real-world data sets.

1 Introduction

Federated recommender system (FedRec) is an essential application of federated learning in the recommendation scenario [Yang *et al.*, 2020]. In recent years, federated learning has been a fast-growing research field, which keeps private data locally at multiple parties and trains models collaboratively in a secure and privacy-preserving way [McMahan and others, 2021; McMahan *et al.*, 2017; Yang *et al.*, 2019]. For example, [Ammad-Ud-Din *et al.*, 2019] proposed a federated matrix factorization algorithm, which distributes the training process at each local party and aggregates the computed gradients on the central server. Privacy-preserving is one of the major challenges in federated learning. Data decentralization does alleviate privacy risks compared with the conventional data-center training scheme. However, the gradients transmit-

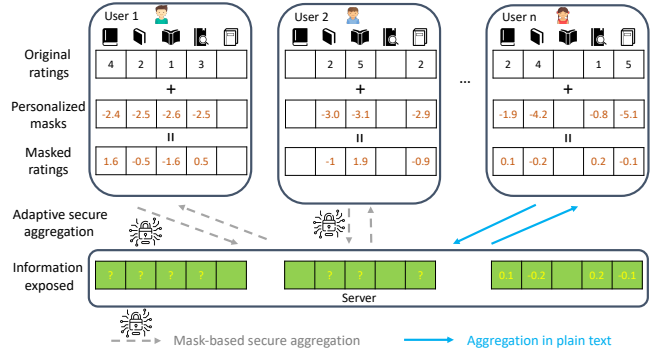


Figure 1: Illustration of the proposed FedMMF method. First, each party generates personalized masks via training a local model. Second, masked ratings are constructed via a combination of original ratings and personalized masks. Then, federated matrix factorization is performed on the masked ratings of all parties. An adaptive secure aggregation method is adopted. The parties with well-protected original data could share model updates via vanilla aggregation in plain-text format. And the other parties carry out a mask-based secure aggregation protocol. Finally, only the masked ratings with limited information are exposed to the server, leaking no data privacy.

ted among different parties could still leak user privacy [Aono *et al.*, 2017; Zhu *et al.*, 2019].

To address the privacy problem, current FedRec methods can be broadly divided into two categories. The first-kind solutions are based on cryptographic techniques such as homomorphic encryption (HE) [Gentry, 2009] or secure multi-party computation (SMC) [Yao, 1982]. For example, HE-based FedRec [Chai *et al.*, 2020] utilizes HE to protect the transmitted gradients. These methods could lead to lossless model performance. However, they produce extra computation and communication costs since federated learning needs a large amount of calculation and intermediate results exchange. The second-kind solutions utilize the obfuscation methods such as differential privacy (DP) [Dwork *et al.*, 2014]. For instance, DP-based FedRec [Hua *et al.*, 2015] has been designed to provide a recommendation service without leaking the data privacy of multiple sources. Although DP-based federated algorithms are efficient, they damage the accuracy of models. Therefore, the above solutions all have

difficulties when applying to practical problems. They cannot satisfy both the two requirements of recommender system (RecSys), *i.e.*, personalization and real-time.

In this paper, we propose federated masked matrix factorization (FedMMF) as a novel FedRec method. The designed FedMMF method could protect the data privacy of FedRec without sacrificing efficiency and effectiveness. Shown in Fig. 1, instead of using cryptographic or obfuscation methods, we introduce a new idea of protecting private data from leakage in FedRec, which is called personalized mask. Personalized mask is a locally generated mask that adds on the original data for preserving privacy without effectiveness loss. Gradients computed on the masked ratings of one participant could be secure enough to directly share with other parties. Moreover, combined with the adaptive secure aggregation protocol, personalized mask also further relieves the efficiency problem of FedRec. Theoretically and empirically, we show the superiority of FedMMF.

The paper is organized as follows, in Section 2, we first introduce the basic models and the privacy leakage problem; in Section 3, we explain the FedMMF algorithm, the training process, and the privacy guarantee; in Section 4, we show the performance of FedMMF in three real-world datasets.

2 Preliminaries

In this section, we first introduce the traditional matrix factorization for recommendation. Then, based on the current challenges of RecSys, we explain federated matrix factorization (FedMF). Although FedMF alleviates the privacy problem of FedRec, there still exists leakage in the training process. Finally, we talk about the current solutions of secure FedMF.

2.1 Matrix Factorization

Given a rating matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, the recommender system aims to fill in the missing values of the matrix. Matrix factorization (MF) is regarded as one of the most classic recommendation algorithm [Koren *et al.*, 2009]. It decouples the original matrix \mathbf{R} into two low-rank matrices. The rating r_{ui} that user u gives to the item i can be approximated as:

$$\hat{r}_{ui} = \mathbf{q}_i^T \mathbf{p}_u, \quad (1)$$

where $\mathbf{q}_i \in \mathbb{R}^{k \times 1}$ represents the latent factors of item i , $\mathbf{p}_u \in \mathbb{R}^{k \times 1}$ represents the latent factors of user u , and the latent dimension k can be regarded as the item's implicit characteristics. We could optimize the latent factors via minimizing the loss given below using the existing ratings:

$$\min_{\mathbf{q}_i, \mathbf{p}_u} \frac{1}{2} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda (\|\mathbf{q}_i\|_2^2 + \|\mathbf{p}_u\|_2^2), \quad (2)$$

where \mathcal{K} stands for the set of user-item pairs whose rating r_{ui} is already known and λ is the regularization coefficient. Stochastic gradient descent is utilized to update each parameter:

$$\mathbf{q}_i \leftarrow \mathbf{q}_i - \gamma \cdot (\lambda \cdot \mathbf{q}_i - e_{ui} \cdot \mathbf{p}_u), \quad (3)$$

$$\mathbf{p}_u \leftarrow \mathbf{p}_u - \gamma \cdot (\lambda \cdot \mathbf{p}_u - e_{ui} \cdot \mathbf{q}_i), \quad (4)$$

where $e_{ui} = r_{ui} - \mathbf{q}_i^T \mathbf{p}_u$ and γ is the learning rate. Conventional recommender systems centrally collect users' private data and train MF algorithm on the server, which leads to immense privacy risks.

2.2 Federated Matrix Factorization

With the development of federated learning, federated recommender system (FedRec) was proposed to address the privacy and data silo problems in the recommendation scenarios [Yang *et al.*, 2020]. In this paper, we focus on the horizontal FedRec, where each party only contains the rating information of one individual user and the user's private data is not allowed to leave the local device. Federated matrix factorization (FedMF) was designed to train recommendation models in such a naturally distributed situation. In the vanilla FedMF algorithm [Ammad-Ud-Din *et al.*, 2019], all the item latent factors $\{\mathbf{q}_i\}_{i \in \mathcal{I}}$ are maintained on the central server, while each user's latent factors \mathbf{p}_u is kept on the local party. The training process is as follows and loops until the convergence of model parameters: 1) party u downloads item i 's latent factors \mathbf{q}_i from the server; 2) party u updates user's latent factors \mathbf{p}_u using private local data r_u ; 3) party u computes the gradients of each item's latent factors $\boldsymbol{\eta}_{ui} = \lambda \cdot \mathbf{q}_i - e_{ui} \cdot \mathbf{p}_u$ with r_u and the updated \mathbf{p}_u ; 4) party u sends $\boldsymbol{\eta}_{ui}$ to server; 5) server aggregates the gradients $\sum_{u \in \mathcal{U}} \boldsymbol{\eta}_{ui}$ and updates \mathbf{q}_i .

Privacy Leakage from Gradients in FedMF

Vanilla FedMF makes sure that users' private data never leaves the local parties. However, the transmitted gradients could also lead to privacy leakage [Chai *et al.*, 2020]. From user u , the server continuously receives the gradients of the item i 's latent vector at step $t-1$ and step t :

$$\boldsymbol{\eta}_{ui}^{t-1} = \lambda \cdot \mathbf{q}_i^{t-1} - e_{ui}^{t-1} \cdot \mathbf{p}_u^{t-1}, \quad (5)$$

$$\boldsymbol{\eta}_{ui}^t = \lambda \cdot \mathbf{q}_i^t - e_{ui}^t \cdot \mathbf{p}_u^t, \quad (6)$$

where $e_{ui}^{t-1} = r_{ui} - \mathbf{q}_i^{t-1T} \mathbf{p}_u^{t-1}$ and $e_{ui}^t = r_{ui} - \mathbf{q}_i^{tT} \mathbf{p}_u^t$. Besides, the server also knows the update rule of the latent vector of user u :

$$\mathbf{p}_u^t = \mathbf{p}_u^{t-1} + \gamma \cdot \sum_{i \in \mathcal{K}_u} (\lambda \cdot \mathbf{p}_u^{t-1} - e_{ui}^t \cdot \mathbf{q}_i^t), \quad (7)$$

where \mathcal{K}_u stands for the set of items that user u has rated. Obviously, only \mathbf{p}_u^{t-1} , \mathbf{p}_u^t and r_{ui} are unknown to the server. Combining equations 5, 6, and 7, the server could solve the unknown variables [Lazard, 2009]. In this way, private raw ratings of each user are revealed.

Secure FedMF

To address the gradient leakage problem of vanilla FedMF, a few secure FedMF algorithms have been proposed. For example, HE-based FedMF [Chai *et al.*, 2020] and DP-based FedMF [Hua *et al.*, 2015], respectively, utilize HE and DP to further preserve privacy. HE-based FedMF encrypts gradients of item latent factors with HE before transmitting them to the server. Then, the server performs secure aggregation

on the encrypted gradients, updates item latent factors in ciphertext state, and distributes the new encrypted item latent factors to each user. In a similar way, DP-based FedMF adds noises to gradients before aggregation. However, the former one causes extra costs and the latter one results in accuracy losses.

3 Federated Masked Matrix Factorization

In this section, we explain the proposed FedMMF method. First, FedMMF adopts a new idea of the personalized mask and we analyze its security. Then, FedMMF applies an adaptive secure aggregation protocol according to different protection situations provided by personalized masks on various users.

3.1 Personalized Mask

We generate the personalized masks via private well-trained model separately at each party. As shown in Fig. 1, FedMMF applies the idea of personalized mask in the previous FedMF architecture. The whole training process is as follows. Firstly, before the federated training of latent factors, each local party u trains a private local model using only the user's own data. The corresponding loss function is shown below:

$$L_u = \frac{1}{|\mathcal{K}_u|} \sum_{i \in \mathcal{K}_u} (r_{ui} - f_u^{mask}(i))^2 \quad (8)$$

Without loss of generality, we define the private model of user u as f_u^{mask} . Then, the model is used to give prediction $f_u^{mask}(i)$ on each user-item pair u, i , where $i \in \mathcal{K}_u$. The opposite of the prediction is regarded as the personalized mask. Finally, all parties collaboratively train a matrix factorization model on the masked rating:

$$r_{u,i}^{masked} = r_{u,i} - f_u^{mask}(i). \quad (9)$$

The prediction of FedMMF algorithm for one specific user-item pair (u, i) is:

$$\hat{r}_{ui} = \mathbf{q}_i^T \mathbf{p}_u + f_u^{mask}(i). \quad (10)$$

The private model f_u^{mask} could be an arbitrary model which only trains on the local data. The well-behaved private model at each local party could protect the privacy of original ratings. Thus, parties with well-behaved private models are able to directly share their gradients computed on the masked ratings. Theorem 1 provides us with how much privacy could be protected by personalized masks.

Security Analysis

The private model f_u^{mask} aims to hide the information of $r_{ui} \in \mathcal{R}$, which is the rating that each user $u \in \mathcal{U}$ gives to item $i \in \mathcal{I}$. For user u , the training data of f_u^{mask} is denoted by $\mathcal{Z}^l = \{(i, r_{ui})\}_{i \in \{1, \dots, l\}}$. The training data is sampled from a joint distribution $P_{\mathcal{I}\mathcal{R}}$. We assume $\mathcal{R} \in [0, 1]$.

Definition 3.1 (Privacy indicator of personalized mask). *We define the private information exposed by one specific user u after applying personalized masks as:*

$$J(f_u^{mask}, P_{\mathcal{I}\mathcal{R}}) = E_{(\mathcal{I}, \mathcal{R}) \sim P_{\mathcal{I}\mathcal{R}}} [\|\mathcal{R} - f_u^{mask}(\mathcal{I})\|^2]. \quad (11)$$

With a smaller value of privacy indicator J , personalized mask could provide a better protection. If the local private model predicts more accurately, personalized masks will cover more information of the original ratings.

Theorem 1. *Personalized mask is (ϵ, δ) -private for user u if there exists a function $n_{\mathcal{F}_u} : (0, 1) \times (0, 1) \rightarrow \mathbb{N}$. For any $\epsilon, \delta \in (0, 1)$ and any distribution $P_{\mathcal{I}\mathcal{R}}$, if $n > n_{\mathcal{F}}$, then*

$$\begin{aligned} Pr_{\mathcal{Z}^n \sim P_{\mathcal{I}\mathcal{R}}} (J(f_u^{mask}, P_{\mathcal{I}\mathcal{R}}) \leq \min_{f_u \in \mathcal{F}_u} J(f_u, P_{\mathcal{I}\mathcal{R}}) + \epsilon) \\ \geq 1 - \delta. \end{aligned} \quad (12)$$

Proof. For any $f_u \in \mathcal{F}_u$, the privacy indicator of user u calculated on the training sample \mathcal{Z}^n is:

$$J(f_u, P_{\mathcal{I}\mathcal{R}}^n) = \frac{1}{n} \sum_{j=1}^n \|\mathcal{R}_j - f_u(\mathcal{I}_j)\|^2. \quad (13)$$

Each $\|\mathcal{R}_j - f_u(\mathcal{I}_j)\|^2$ is an independent random variable with mean $J(f_u, P_{\mathcal{I}\mathcal{R}})$. We further assume that $\|\mathcal{R}_j - f_u(\mathcal{I}_j)\|^2 \in [0, 1]$. According to Hoeffding's inequality¹, we obtain:

$$Pr_{\mathcal{Z}^n \sim P_{\mathcal{I}\mathcal{R}}} (|(f_u, P_{\mathcal{I}\mathcal{R}}^n) - J(f_u, P_{\mathcal{I}\mathcal{R}})| \geq \epsilon) \leq 2e^{-2n\epsilon^2}, \quad (14)$$

then we could get:

$$Pr_{\mathcal{Z}^n \sim P_{\mathcal{I}\mathcal{R}}} (\exists f_u \in \mathcal{F}_u, \text{s.t. } |(f_u, P_{\mathcal{I}\mathcal{R}}^n) - J(f_u, P_{\mathcal{I}\mathcal{R}})| \geq \epsilon) \leq 2|\mathcal{F}_u|e^{-2n\epsilon^2}. \quad (15)$$

This shows that if

$$n \geq \frac{\log(2|\mathcal{F}_u|/\delta)}{2\epsilon^2}, \quad (16)$$

then

$$\begin{aligned} Pr_{\mathcal{Z}^n \sim P_{\mathcal{I}\mathcal{R}}} (|(f_u, P_{\mathcal{I}\mathcal{R}}^n) - J(f_u, P_{\mathcal{I}\mathcal{R}})| \leq \epsilon, \\ \forall f_u \in \mathcal{F}_u) \geq 1 - \delta, \end{aligned} \quad (17)$$

which is equivalent to:

$$\begin{aligned} Pr_{\mathcal{Z}^n \sim P_{\mathcal{I}\mathcal{R}}} (J(f_u^{mask}, P_{\mathcal{I}\mathcal{R}}) \leq \min_{f_u \in \mathcal{F}} J(f_u, P_{\mathcal{I}\mathcal{R}}) + 2\epsilon) \\ \geq 1 - \delta. \end{aligned} \quad (18)$$

The reason is that, given

$$\forall f_u \in \mathcal{F}_u, |(f_u, P_{\mathcal{I}\mathcal{R}}^n) - J(f_u, P_{\mathcal{I}\mathcal{R}})| \leq \epsilon, \quad (19)$$

we could obtain step by step:

$$\begin{aligned} J(f_u^{mask}, P_{\mathcal{I}\mathcal{R}}) &\leq J(f_u^{mask}, P_{\mathcal{I}\mathcal{R}}^n) + \epsilon \\ &\leq \min_{f_u \in \mathcal{F}} J(f_u, P_{\mathcal{I}\mathcal{R}}^n) + \epsilon \\ &\leq \min_{f_u \in \mathcal{F}} J(f_u, P_{\mathcal{I}\mathcal{R}}) + \epsilon + \epsilon \\ &= \min_{f_u \in \mathcal{F}} J(f_u, P_{\mathcal{I}\mathcal{R}}) + 2\epsilon. \end{aligned} \quad (20)$$

¹https://en.wikipedia.org/wiki/Hoeffding's_inequality

Let $\epsilon = \frac{\epsilon}{2}$, we finally get

$$n_{\mathcal{F}_u}(\epsilon, \delta) \leq \frac{2 \log(2|\mathcal{F}_u|/\delta)}{2\epsilon^2}. \quad (21)$$

□

The function $n_{\mathcal{F}_u}$ determines the sample complexity of user u for training a FedMMF algorithm. It stands for how many samples at least are required by personalized masks to guarantee the privacy of user u . Besides, we assume the hypothesis class \mathcal{F}_u of local private model is finite. However, it is not a necessary condition, and Theorem 1 can be further generalized. From Theorem 1, we know that the privacy-preserving ability of personalized mask decides on the quality of local training data. The users with good enough local data could generate secure enough personalized masks, which successfully limit the exposed information from the masked ratings. The privacy indicator J can be used to judge if the personalized masks are secure enough. In addition, we should also try to find the most suitable hypothesis class \mathcal{F}_u on various data sets.

3.2 Adaptive Secure Aggregation

The data quality of different users varies in the real world. Therefore, not all users can generate perfect personalized masks for protection. We propose an adaptive secure aggregation protocol to address this problem. For a given privacy indicator threshold th_J , we could divide the users into two groups, *i.e.*, secure masked group \mathcal{U}_{secure} and insecure masked group $\mathcal{U}_{insecure}$. The privacy indicator J of user in \mathcal{U}_{secure} is larger than th_J , while the privacy indicator J of user in $\mathcal{U}_{insecure}$ is smaller than th_J .

For user $u \in \mathcal{U}_{secure}$ with secure enough personalized masks, the gradients η_{ui} could be directly shared with the central server for aggregation. And the server could get $\sum_{u \in \mathcal{U}_{secure}} \eta_{ui}$. However, for user $u \in \mathcal{U}_{insecure}$ with insecure personalized masks, sharing plain-text gradients will disclose the privacy of local rating data. Therefore, we adopt a mask-based secure aggregation method designed by [Bonawitz *et al.*, 2017]. For an arbitrary pair of users $u, v \in \mathcal{U}_{insecure}$ and $u < v$, they decide a random mask $s_{u,v} \in \mathbb{R}^{k \times 1}$ together. User u adds this random mask $s_{u,v}$ to its gradients, while user v subtracts $s_{u,v}$ from its gradients. Then, each user u could calculate:

$$\tilde{\eta}_{ui} = \eta_{ui} + \sum_{u < v} s_{u,v} - \sum_{u > v} s_{v,u} \quad \text{mod } l, \quad (22)$$

where l is a large prime number. Next, each user $u \in \mathcal{U}_{insecure}$ sends the computed $\tilde{\eta}_{ui}$ to the server. The server will calculate:

$$\begin{aligned} \sum_{u \in \mathcal{U}_{insecure}} \tilde{\eta}_{ui} &= \sum_{u \in \mathcal{U}_{insecure}} (\eta_{ui} + \sum_{u < v} s_{u,v} - \sum_{u > v} s_{v,u}) \\ &= \sum_{u \in \mathcal{U}_{insecure}} \eta_{ui} \quad \text{mod } l. \end{aligned} \quad (23)$$

The aggregated gradients could be obtained, and the gradients of one specific user are protected by the designed random

Algorithm 1 Federated Masked Matrix Factorization

- 1: **Input:** $r_{u \in \{1, \dots, n\}}, th_J$
 - 2: **Output:** $q_{i \in \{1, \dots, m\}}, p_{u \in \{1, \dots, n\}}, f_{u \in \{1, \dots, n\}}^{mask}$
 - 3: Server initializes $q_{i \in \{1, \dots, m\}}^0$, each party u initializes $p_{u \in \{1, \dots, n\}}^0$ and $f_{u \in \{1, \dots, n\}}^{mask}(\theta_u)$.
 - 4: **for** each party $u \in \{1, \dots, n\}$ in parallel **do**
 - 5: // run on each party u
 - 6: Train private model $f_u^{mask}(\theta_u)$ on local data r_u ;
 - 7: Compute personalized masked rating $r_{u,i}^{masked}$ according to Eq. 9 for each $i \in \mathcal{K}_u$;
 - 8: Grouped to \mathcal{U}_{secure} or $\mathcal{U}_{insecure}$ with th_J ;
 - 9: **end for**
 - 10: // run on the server
 - 11: **for** each $t = 1, 2, \dots, T$ **do**
 - 12: **for** each party $u \in \mathcal{U}_{secure}$ in parallel **do**
 - 13: Get gradients $\eta_{ui \in \mathcal{K}_u}^t = \text{MaskedUpdate}(q_{i \in \mathcal{K}_u}^{t-1})$;
 - 14: **end for**
 - 15: **for** each party $u \in \mathcal{U}_{insecure}$ in parallel **do**
 - 16: Get gradients $\tilde{\eta}_{ui \in \mathcal{K}_u}^t = \text{MaskedUpdate}(q_{i \in \mathcal{K}_u}^{t-1})$;
 - 17: **end for**
 - 18: Get the aggregated gradients $\sum_{u \in \mathcal{U}} \eta_{i \in \mathcal{I}}^t$ according to Eq. 24;
 - 19: Update item factors $q_i^t = q_i^{t-1} - \gamma \cdot \sum_{u \in \mathcal{U}} \eta_i^t$ for each $i \in \mathcal{I}$;
 - 20: **end for**
 - 21: // run on each party u
 - 22: **MaksedUpdate:**
 - 23: Compute $e_{ui}^t = r_{ui}^{masked} - q_i^{tT} p_u^t$ for each $i \in \mathcal{K}_u$;
 - 24: Update user factors p_u^t according to Eq. 7;
 - 25: Compute gradient η_{ui}^t according to Eq. 6 for each $i \in \mathcal{K}_u$;
 - 26: **Return** $\eta_{ui \in \mathcal{K}_u}^t$ or $\tilde{\eta}_{ui \in \mathcal{K}_u}^t$ to the server with adaptive secure aggregation protocol.
-

masks. Furthermore, secret sharing [Shamir, 1979] is utilized to solve the dynamic user problem. With the adaptive secure aggregation protocol, the server could obtain

$$\sum_{u \in \mathcal{U}} \eta_{ui} = \sum_{u \in \mathcal{U}_{secure}} \eta_{ui} + \sum_{u \in \mathcal{U}_{insecure}} \tilde{\eta}_{ui} \quad (24)$$

The details of FedMMF are shown in Algorithm 1. Compared with only applying the original aggregation in [Bonawitz *et al.*, 2017], FedMMF utilizes the adaptive secure aggregation to further improve efficiency.

4 Experiments

In this section, we show that FedMMF could improve efficiency without the loss of privacy and model effectiveness. Firstly, we explain the data sets, baseline models, and other settings in the experiments. Then, we show the improvements of FedMMF on model efficiency. With the help of adaptive secure aggregation protocol based on personalized masks, FedMMF accelerates the training process. At last, we

discuss the model effectiveness of FedMMF with different kinds of personalized masks, compared to the baseline model.

4.1 Settings

We verify FedMMF on three real-world data sets. Two of them are MovieLens data sets [Harper and Konstan, 2015], *i.e.*, MovieLens 100K and MovieLens 10M. The other one is the LastFM data set [Cantador *et al.*, 2011]. In our experiment, each user is regarded as a participant in the collaborative training process. Therefore, the user’s own ratings are kept on the local party. Besides, we utilize the side information (*i.e.*, user profiles and item attributes) to train the local private model. To construct features from tags in the data set, we utilize TFIDF [Robertson, 2004] and PCA [Abdi and Williams, 2010] techniques. Besides, we set bins for the listening counts of music of the LastFM data set and convert them into ratings scaling from 1 to 5. In addition, the evaluation metrics of model efficacy are root mean square error (RMSE) and mean absolute error (MAE). They are averaged by each user-item pair but not each user, which is an alignment with most current works. Besides, we run each experiment ten times to obtain the mean and standard deviation values.

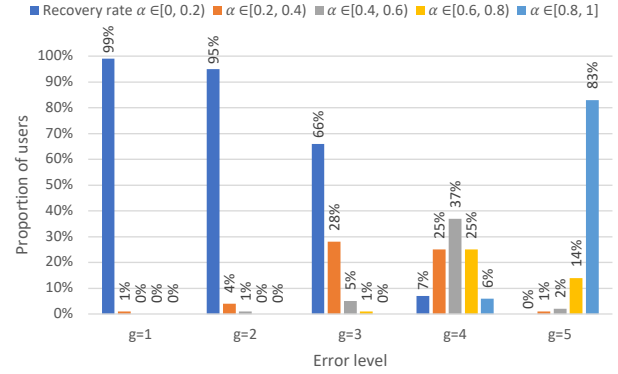
The compared models are: 1) **FedMF**: parties collaboratively train matrix factorization models via sharing the latent factors of common users, where neither HE nor DP is utilized; 2) **One-order FedMMF**: each party locally learns linear personalized masks to hide private rating information via a linear regression model [Montgomery *et al.*, 2012]. Then, all parties collaboratively train FedMF on the one-order masked ratings; 3) **Two-order FedMMF**: similarly, each party constructs two-order masks to protect private ratings via locally learning a factorization machine model [Rendle, 2010]; 4) **High-order FedMMF**: each party captures high-order and nonlinear feature interactions through a neural network model [Yegnanarayana, 2009]. We do not compare FedMMF with DP-based FedMF, because DP causes effectiveness loss while FedMMF does not. Besides, we also show the performance of various local context models and federated context models for reference.

4.2 Efficiency Promotion and Privacy Discussion

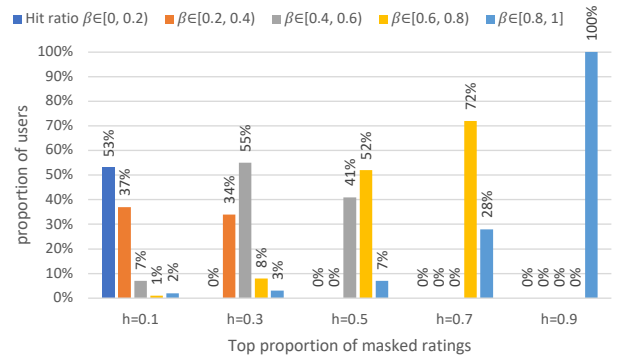
Compared with HE-based FedMF [Chai *et al.*, 2020], FedMMF with all users in the insecure user group could largely speed up the training process [Bonawitz *et al.*, 2017]. Then, the personalized mask technique could further improve the efficiency of the secure aggregation process via sharing plaintext gradients of parties with well-protected ratings. We provide two attack methods, *i.e.*, recovery attack and ranking attack, for analyzing how much the personalized mask technique could further promote model efficiency. Taking two-order FedMMF on MovieLens 10M data set as an example, we conduct the attack experiments. The rating range of the MovieLens 10M data set is from 0.5 to 5.0. And the rating interval is 0.5.

Recovery Attack

Against the masked ratings, an adversary could conduct an intuitive attack to recover the original ratings. However, the



(a) The results of recovery attack under different error levels. On the one hand, when error level g is small, the recovery attack could hardly reveal the original rating information. On the other hand, when error level g grows, the recovery attack becomes more accurate. However, the utility of recovered ratings also decreases.



(b) The results of ranking attack under different top proportions. With the masked ratings of each party, the adversary wants to choose the actual high-rated items. When the adversary utilizes a small top proportion h , the attacks performs on most parties achieve a poor hit ratio, which is less than 0.5. Although the hit ratio grows as h increases, a large h results in a useless ranking attack.

Figure 2: Experiments results on recovery and ranking attacks.

attack could be difficult if only the masked ratings are exposed. Therefore, for each party, we assume that the adversary knows the minimum and maximum values of the original ratings. Then, the adversary could scale the masked ratings to the range of original ratings for recovery. We define g as the error level. If the difference between one recovered value and the corresponding original rating is less than α , the recovery is considered successful. Thus, there exists a recovery rate α for each party’s masked rating. In Fig. 2a, we show the proportion of parties whose recovery rate is in a certain range under different error levels. As we can see, when the error level is small, *e.g.*, $g = 1$ and $g = 2$, the adversary could nearly reveal no party’s privacy with a recovery rate larger than 0.5. And as the error level increases, the recovery rate begins to grow. However, a higher error level means a more inaccurate recovery, and the utility of the recovered ratings is poorer.

Models	MovienLens 100K		MovienLens 10M		LastFM	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
FedMF	0.9491 ± 0.0040	0.7412 ± 0.0027	0.7753 ± 0.0034	0.5827 ± 0.0015	1.2235 ± 0.0068	0.8780 ± 0.0047
LocalLR	1.0107 ± 0.0025	0.8040 ± 0.0022	0.8818 ± 0.0023	0.6766 ± 0.0011	1.1081 ± 0.0099	0.8163 ± 0.0085
FedLR	1.0796 ± 0.0081	0.8844 ± 0.0058	0.9703 ± 0.0020	0.7497 ± 0.0012	1.5448 ± 0.0110	1.3538 ± 0.0137
One-order FedMMF	0.9340 ± 0.0043	0.7340 ± 0.0035	0.7695 ± 0.0013	0.5808 ± 0.0008	1.0886 ± 0.0109	0.8066 ± 0.0092
LocalFM	1.0083 ± 0.0019	0.8054 ± 0.0019	0.8938 ± 0.0023	0.6862 ± 0.0012	1.0845 ± 0.0130	0.7988 ± 0.0035
FedFM	1.0628 ± 0.0070	0.8644 ± 0.0053	0.9639 ± 0.0022	0.7445 ± 0.0015	1.5301 ± 0.0133	1.3369 ± 0.0117
Two-order FedMMF	0.9218 ± 0.0037	0.7250 ± 0.0030	0.7720 ± 0.0013	0.5827 ± 0.0007	1.0842 ± 0.0090	0.7964 ± 0.0031
LocalNN	1.0114 ± 0.0021	0.8087 ± 0.0017	0.8819 ± 0.0020	0.6816 ± 0.0009	1.1007 ± 0.0068	0.8007 ± 0.0123
FedNN	1.0945 ± 0.0074	0.9176 ± 0.0060	0.9756 ± 0.0024	0.7689 ± 0.0028	1.5461 ± 0.0060	1.3598 ± 0.0075
High-order FedMMF	0.9319 ± 0.0025	0.7317 ± 0.0018	0.7648 ± 0.0016	0.5772 ± 0.0008	1.0860 ± 0.0055	0.7933 ± 0.0070

Table 1: Performance of FedMMF compared with baseline models on different data sets. FedMMF models with different personalized masks have no effectiveness loss compared with FedMF in all data sets. Besides the comparison between FedMMF and FedMF, we also show that FedMMF outperforms local context models and federated context models.

Ranking Attack

Since the intuitive recovery attack seems not successful enough, we introduce another method named ranking attack. Instead of recovering the original concrete ratings, ranking attack tries to find the high-rating items from their masked ratings. First, for each party, the adversary ranks the rated items according to their masked ratings. Then, items in the top h proportion of masked ratings are selected as the high-rating items. Similarly, given h , we also sort these items with regard to their original ratings as the true high-rating item set. Thus, we could evaluate the ranking attack with hit ratio β , which is calculated as the ratio that items selected using masked ratings are in the true high-rating item set. Fig. 2b shows that, under different top proportion h , the ranking attack could reveal the rating ranking privacy of parties. If the selected top proportion is small, *e.g.*, $h = 1$ and $h = 2$, the attacks performed on most parties’ masked ratings obtain a hit ratio less than 0.5. It means that more than half of the selected items do not have high ratings. When the adversary tunes h larger, the attack becomes more effective. However, a large h is relatively meaningless because the adversary does not want to choose all items to be high-rating in reality.

According to the experiment results of the above two attack methods, we find that a considerable number of users get their rating privacy well-protected with the help of personalized masks. These users can be put in the secure group and transfer their gradients in plain text. Therefore, the personalized mask could further accelerate the training process of federated recommendation. Besides federated learning, the personalized masked ratings of users in the secure group could also be centrally collected and used for training without privacy leakage. This operation is able to reduce the communication and computation costs once again.

4.3 Discussion on Model Effectiveness

In this section, we verify the effectiveness of FedMMF on three real-world data sets. We implement three private models to construct personalized masks with different properties: the one-order mask, two-order mask, and high-order mask. The performances of FedMMF with these three masks are shown in Tab. 1. RMSE and MAE are both regression evaluation

metrics. Smaller value stands for better model efficacy. As we can see, FedMMF models with different personalized masks have no effectiveness loss compared with FedMF in all data sets.

Moreover, FedMMF even outperforms FedMF. The effectiveness improvements could be divided into two parts. The first part benefits from the ensemble training scheme of FedMMF. The incorporation of personalized masks utilizes the idea of ensemble learning to combine weak learners for a better generalization ability. The second part takes advantage of the side information utilized in the private model of FedMMF. In the recommendation scenario, feature interactions are important information to capture. Another observation is that, on all three data sets, two-order FedMMF and high-order FedMMF dominate alternatively. It means we should utilize cross features to construct personalized masks in the recommendation scenarios. We also compare FedMMF models with corresponding local context and federated context models, shown in Tab. 1. Comparing FedMMF with different local context models and federated context models, we could see that FedMMF also outperforms both of them. This observation verifies the main contribution to the effectiveness improvement is the incorporation of ensemble learning. On the other hand of the shield, FedMMF can also be regarded as an excellent way to combine collaborative information and feature information.

5 Conclusion

In this paper, we provide a new idea of personalized masks to protect data privacy in federated learning, which neither slows the training process down nor damages model performance. Taking the recommendation scenario as an example, we apply it in the FedMMF algorithm. Combining with the adaptive secure aggregation protocol, FedMMF shows superiority theoretically and empirically. In our future work, we would like to extend personalized masks to more general federated learning tasks besides recommender systems and try to combine personalized masks with differential privacy theory.

References

- [Abdi and Williams, 2010] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
- [Ammad-Ud-Din *et al.*, 2019] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.
- [Aono *et al.*, 2017] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2017.
- [Bonawitz *et al.*, 2017] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.
- [Cantador *et al.*, 2011] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.
- [Chai *et al.*, 2020] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 2020.
- [Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- [Gentry, 2009] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.
- [Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (TiIS)*, 5(4):1–19, 2015.
- [Hua *et al.*, 2015] Jingyu Hua, Chang Xia, and Sheng Zhong. Differentially private matrix factorization. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1763–1770. AAAI Press, 2015.
- [Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [Lazard, 2009] Daniel Lazard. Thirty years of polynomial system solving, and now? *Journal of symbolic computation*, 44(3):222–231, 2009.
- [McMahan and others, 2021] H Brendan McMahan et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021.
- [McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [Montgomery *et al.*, 2012] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.
- [Rendle, 2010] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.
- [Robertson, 2004] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 2004.
- [Shamir, 1979] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.
- [Yang *et al.*, 2020] Liu Yang, Ben Tan, Vincent W Zheng, Kai Chen, and Qiang Yang. Federated recommendation systems. In *Federated Learning*, pages 225–239. Springer, 2020.
- [Yao, 1982] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.
- [Yegnanarayana, 2009] Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
- [Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14774–14784, 2019.