FR-FMSS: Federated Recommendation via Fake Marks and Secret Sharing

Zhaohao Lin
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
2017112126@email.szu.edu.cn

Weike Pan*
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
panweike@szu.edu.cn

Zhong Ming*
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

ABSTRACT

With the implementation of privacy protection laws such as GDPR, it is increasingly difficult for organizations to legally collect user data. However, a typical recommendation algorithm based on machine learning requires user data to learn user preferences. In order to protect user privacy, a lot of recent works turn to develop federated learning-based recommendation algorithms. However, some of these works can only protect the users' rating values, some can only protect the users' rating behavior (i.e., the engaged items), and only a few works can protect the both types of privacy at the same time. Moreover, most of them can only be applied to a specific algorithm or a class of similar algorithms. In this paper, we propose a generic cross-user federated recommendation framework called FR-FMSS. Our FR-FMSS can not only protect the two types of user privacy, but can also be applied to most recommendation algorithms for rating prediction, item ranking, and sequential recommendation. Specifically, we use fake marks and secret sharing to modify the data uploaded by the clients to the server, which protects user privacy without loss of model accuracy. We take three representative recommendation algorithms, i.e., MF-MPC, eALS, and Fossil, as examples to show how to apply our FR-FMSS to a specific algorithm.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Federated recommendation, Secret sharing, Rating prediction, Item ranking, Sequential recommendation

ACM Reference Format:
Zhaohao Lin, Weike Pan, and Zhong Ming. 2021. FR-FMSS: Federated Recommendation via Fake Marks and Secret Sharing. In *Fifthteenth ACM Conference on Recommender Systems (RecSys '21), September 27–October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3460231.3478855

*co-corresponding authors

RecSys '21, September 27–October 1, 2021, Amsterdam, Netherlands

ACM ISBN 978-1-4503-8458-2/21/09.
https://doi.org/10.1145/3460231.3478855

1 INTRODUCTION

With the development of online shopping, more and more people prefer to shop on the Internet. As a result, many shopping sites have developed rapidly, such as Amazon and Taobao. On these shopping sites, the users can choose far more commodities than in offline stores. However, the excessive number of commodities causes another problem. It is difficult for the users to select the best one from a large number of commodities. To solve this problem, many recommendation systems are proposed. A recommendation system can learn the preferences of the users and then provide recommended commodities. It helps the users to make a rough selection of commodities. As a result, the users only need to choose the most satisfactory one from a small number of commodities.

Traditional recommendation systems need to collect the historical data of the users, such as browses, purchases, and ratings. However, these are the users' privacy. They may not want others to know this information, including the servers of the shopping sites. In 2016, the European Union issued the General Data Protection Regulation (GDPR) [23] prohibiting commercial companies from collecting, processing or exchanging user data without the permission of the corresponding user. Later, the United States and China also issued similar legal provisions [3, 4]. Obviously, the traditional recommendation systems do not satisfy these laws, so we need new recommendation systems that can protect user privacy.

In 2016, Google proposed a machine learning framework called federated learning [11]. Federated learning distributes a model to multiple participants. The original data is kept by its owner and is not sent to other participants. All the participants jointly train the model. Recently, many federated recommendation works have been published, such as FCF [1], FedRec++ [16], and FedGNN [24], which apply federated learning to existing recommendation algorithms. However, most of these works are only designed for a specific recommendation algorithm or a class of similar algorithms. It is thus difficult to apply the proposed method to a wide spectrum of recommendation algorithms.

In this paper, we propose a generic federated recommendation framework called FR-FMSS. It uses fake marks and secret sharing to protect user privacy without sacrificing the accuracy. Its biggest advantage is that it can be applied to most recommendation algorithms, such as PMF [13], SVD++ [12], and MF-MPC [18] for rating prediction, eALS [6], BPR [19], and Mult-VAE [15] for item ranking, and FPMC [20], Fossil [5], and SASRec [10] for sequential recommendation, etc. Note that we do not include empirical studies because we have shown that a federated algorithm via our

FR-FMSS is lossless compared with the corresponding un-federated one, which is independent of the datasets.

2 RELATED WORK

2.1 Cross-Organization Federated Recommendation

Federated recommendation with participants of two or more organizations is called cross-organization federated recommendation. The organizations' private information is protected, such as the number of sold commodities and the number of times the commodities have been browsed.

PrivNet [7] uses an adversarial learning method to increase the difficulty for external attackers to infer users' privacy from the shared information. It uses a neural network to simulate an external attacker to infer the users' privacy from the shared information. A transfer learning model learns to protect user privacy in alternative training between the method and the attacker. However, PrivNet requires some users to share their privacy for the training of the attacker, which is almost impossible in reality. FCMF [25] uses differential privacy and homomorphic encryption to encrypt the shared parameters. Homomorphic encryption protects the shared parameters, but encryption and decryption cause a lot of computation. Differential privacy also introduces noise to the model training, resulting in a decrease in recommendation performance.

2.2 Cross-User Federated Recommendation

Federated recommendation with participants of individual users within an organization is called cross-user federated recommendation. The users save their historical data locally, and only upload some intermediate data to the server for model training. The users' privacy is thus protected, such as the commodities that the users have browsed or purchased and the corresponding ratings. Unlike cross-organization federated recommendation, the privacy of the organization is not considered because there is only one organization.

The algorithms of cross-user federated recommendation may expose two types of user privacy. First, the algorithms may expose a user's rating behavior, i.e., the items that the user has interacted with. We call some parameters ID-sensitive parameters, such as the item-specific latent feature vectors in the matrix factorization models, and the item embeddings in the neural network models, because the values of these parameters may be not sensitive, but whether the users update them may expose the users' rating behavior. Second, the algorithms may expose the values of a user's ratings. A recent work [2] shows that the server can obtain the user-specific latent feature matrix and the ratings of the rated items as long as it knows the gradients of the item-specific latent feature matrix uploaded by the same user in two continuous iterations. Therefore, there is a risk of privacy leakage when uploading the real gradients to the server. It thus proposes a method called FedMF to protect the users' rating values using homomorphic encryption [2]. However, this method leaks a user's rating behavior, i.e., which items the user has rated.

Federated collaborative filtering (FCF) [1] divides PMF [21] into two parts. It uses Alternating Least Squares (ALS) on the clients (i.e., users) to update the user-specific latent feature vectors, and gradient descent on the server to update the item-specific latent feature vectors. However, FCF is only suitable for matrix factorization models with one-class feedback. FedRec [17] samples some unrated items and assigns some fake ratings to prevent the server from inferring the users' rating behavior. However, the generated fake ratings will bring noise to the recommendation system and reduce its accuracy. FedRec++ [16] is a lossless method, which divides the clients into two types, including ordinary clients and denoising clients. The ordinary clients send the noise anonymously to the denoising clients, and then the denoising clients aggregate the received noise and send them to the server to eliminate the influence of the noise. Note that FedRec and FedRec++ do not modify the gradients of the rated items, which may lead to the leakage of the users' rating values [2]. Another work [14] uses secret sharing to calculate the similarity between every two rated items, thereby protecting users' rating values. However, this method cannot protect users' rating behavior and is only suitable for item-based collaborative filtering. SharedMF [26] is a generic method that uses secret sharing to protect user privacy. It protects the user's rating values and rating behavior from being inferred by the server. However, when a client receives the data sent by other clients, it will know the rated items of the senders.

3 FR-FMSS

In this section, we describe our proposed federated recommendation framework FR-FMSS in detail. Specifically, in our studied problem, we have a set of users (i.e., clients) $\mathcal{U}$ and a set of items $\mathcal{I}$. Each user u has interacted with a set of items $\mathcal{I}_u$ resulting in some records $\mathcal{R}_u = \{(u, i, r_{ui}) | i \in \mathcal{I}_u\}$. Our goal is to predict the rating of a user u to each item in $\mathcal{I} \backslash \mathcal{I}_u$ or to rank the items in $\mathcal{I} \backslash \mathcal{I}_u$ for a user u without leaking the user's privacy. A typical un-federated recommendation algorithm has parameters $\Theta = \{\Theta^p, \Theta^s\}$. Θ^p are the private user-specific parameters, which are saved on the clients. Θ^s are the shared parameters, which are saved on the server. We put some commonly used notations in Table 1.

Our FR-FMSS is a cross-user federated recommendation framework. To protect the users' rating behavior and rating values, it uses two techniques, i.e., fake marks and secret sharing. It can be applied to many un-federated recommendation algorithms, as long as the model parameters of an un-federated recommendation algorithm on all the clients are the same. The justification for this restriction is that if a client wants to use secret sharing to modify the uploaded data, it must have the ability to upload the received data. For example, a client u_1 whose model parameter is Θ_1^s sends the secret $\nabla\Theta_1^s(u_2, u_1)$ to a client u_2, hoping that client u_2 can send the secret to the server to update Θ_1^s. However, the model parameter of client u_2 is different, i.e., Θ_2^s, so it can not update Θ_1^s. At this time, secret sharing cannot work properly.

Fake marks means that when a client uploads the gradients of ID-sensitive parameters, it not only uploads the gradients of the parameters it actually calculated, but also uploads some that the client has not calculated. There are two types of fake marks. The first one is called fake items. Each client generates the fake gradients of some randomly sampled unrated items, and sends them to the server. By doing this, the clients can prevent the server from inferring their rating behavior. FedRec [17], SDCF [8], and FedRec++ [16] have

Table 1: Some notations and explanations used in the paper.

Notations	Explanations
$\mathcal{R} = \{(u, i, r_{ui})\}$	rating records in training data
$\mathcal{R}_u$	rating records w.r.t. user u in $\mathcal{R}$
$\mathcal{U}$	the whole set of users
$\mathcal{I}$	the whole set of items
$\mathcal{M}$	the whole set of marks of the shared parameters
$\mathcal{M}_u$	the set of marks of the shared parameters that the user u wants to update
$\mathcal{M}'_u$	the set of fake marks of the shared parameters that the user u samples
$\bar{\mathcal{M}}_u$	the set of marks of the shared parameters that the user u receives from other clients
$\mathcal{N}_u$	the set of users who receive data sent by the user u in secret sharing
Θ^p_u	the private parameters w.r.t. user u
Θ^s_m	the shared parameters w.r.t. mark m
$\nabla\Theta^s_m(u)$	the gradient of Θ^s_m calculated by the user u, which will be sent to the server
$\nabla\Theta^s_m(\tilde{u}, u)$	the random number of $\nabla\Theta^s_m(u)$, which will be sent to the user $\tilde{u}$
$\nabla\Theta^p(u)$	the sum of the gradients of Θ^p_u calculated by the user u using $\mathcal{R}_u$
Y_m	the number of users who wants to update Θ^s_m
γ	the learning rate
ρ	the sampling parameter

already used this technique. The second one is called fake ratings, which can be used in federating a more sophisticated method such as MF-MPC [18]. The parameter $M^r_{i\cdot}$ in MF-MPC is related to the rating value r and the item i. The server can infer the specific rating value of item i by observing the value of r of the parameter $M^r_{i\cdot}$ uploaded by the clients. Therefore, the parameter $M^r_{i\cdot}$ is ID-sensitive. Similar to the fake items, the clients randomly sample some fake ratings for rated items and sampled items, generate fake gradients, and then send them to the server.

Secret sharing is an important technology of cryptography [22]. First, each client divides a value into several random numbers. Second, each client keeps a copy of the random number, and sends the others to other clients. Third, each client adds the received random numbers to that it keeps. Finally, each client sends the summation to the server. The server accumulates all the received values. The result obtained is the same as that obtained without using secret sharing.

We describe our FR-FMSS in Algorithm 1 and Algorithm 2. First, the server initializes all the shared parameters $\Theta^s_m, m \in \mathcal{M}$, and each client initializes its private parameters Θ^p_u. Second, the server sends the shared parameters to all the clients, i.e., step 1 in Figure 1. Third, each client uses its local data to calculate the parameter gradients. Fourth, each client randomly samples a set of fake marks $\mathcal{M}'_u$, and assigns $\nabla\Theta^s_m(u) \leftarrow \mathbf{0}, m \in \mathcal{M}'_u$. Fifth, each client divides the value of every dimension of $\nabla\Theta^s_m(u), m \in \mathcal{M}_u \cup \mathcal{M}'_u$ into several random numbers, where sum of the random numbers must be equal to the value of the corresponding dimension. Note that $\nabla\Theta^s_m(u) = \mathbf{0}, m \in \mathcal{M}'_u$ is also divided into several positive and negative numbers. Sixth, each client keeps a random number and sends the rest to other clients, i.e., step 2 in Figure 1. Seventh, each client adds the received random numbers and that it keeps to get the modified gradients. Eighth, each client uploads the modified gradients to the server, i.e., step 3 in Figure 1. Ninth, the server accumulates all the gradients and divides them by $Y_m, m \in \mathcal{M}$ to get the average gradients. Finally, the server updates the shared parameters, and each client updates its private parameters. We can see that our FR-FMSS not only protects user privacy well, but also does not reduce the accuracy of the model because it does not introduce any noise.

Algorithm 1 The algorithm of FR-FMSS in the server.

Initialize the shared parameters $\Theta^s_m, m \in \mathcal{M}$
for $t = 1, 2, \ldots, T$ **do**
  for each client u in parallel **do**
    ClientTrain($\Theta^s_m, m \in \mathcal{M}$;t;$\gamma$), i.e., Algorithm 2
  end for
  for $m \in \mathcal{M}$ **do**
    Receive $Y_m(u), u \in \mathcal{U}$
    $Y_m \leftarrow \sum_{u\in\mathcal{U}} Y_m(u)$
    Receive $\nabla\Theta^s_m(u), u \in \mathcal{U}$
    $\nabla\Theta^s_m = \sum_{u\in\mathcal{U}} \nabla\Theta^s_m(u)$
    $\Theta^s_m \leftarrow \Theta^s_m - \gamma \frac{\nabla\Theta^s_m}{Y_m}$
  end for
  Decrease the learning rate $\gamma \leftarrow 0.9\gamma$
end for

4 FR-FMSS SPECIALIZATION

Our FR-FMSS can be applied to a variety of un-federated recommendation algorithms. First, we judge whether each parameter of the un-federated algorithm is a shared parameter or a private parameter. Second, we find out the ID-sensitive parameters of the un-federated algorithm. Third, we design the intermediate values that the clients upload to the server according to the parameter update rules of the un-federated algorithm. Fourth, we specialize

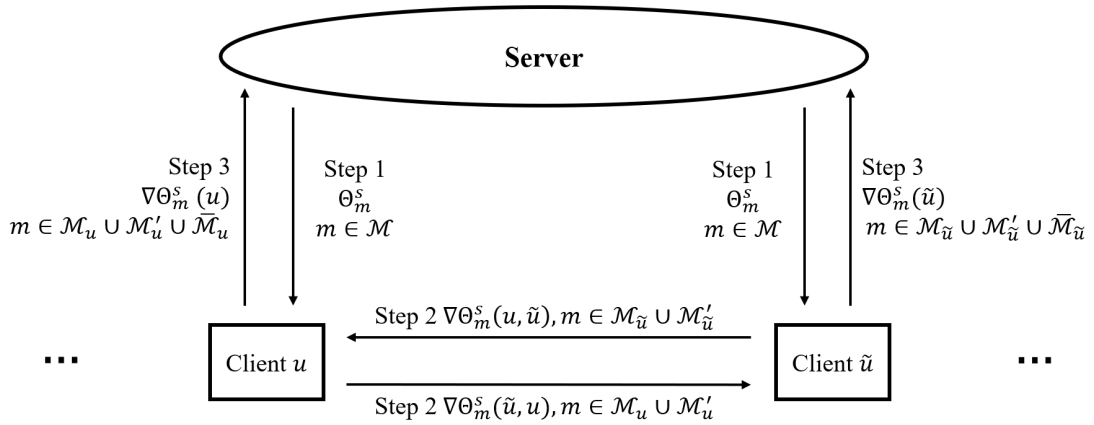

Figure 1: Illustration of the interactions between the server and each client in FR-FMSS.

Algorithm 1 and Algorithm 2 according to the shared parameters, private parameters, ID-sensitive parameters, and intermediate values. We take MF-MPC, eALS, and Fossil as examples to show how to apply our FR-FMSS to a specific algorithm.

MF-MPC [18] integrates the multiclasss preference context of users for rating prediction. First, we set U_u. and b_u in MF-MPC as private parameters Θ^p_u in our FR-FMSS, and set V_i., b_i, $M^r_{i\cdot}$, and μ as shared parameters Θ^s_m in our FR-FMSS. Second, we find that V_i., b_i, and $M^r_{i\cdot}$ are ID-sensitive parameters. Note that the mark of V_i. and b_i is item ID i, and the mark of $M^r_{i\cdot}$ is a combination of item ID i and rating r, so we must use both fake items and fake ratings to federate MF-MPC. Third, we use gradient descent to update the parameters of MF-MPC, so we set $\nabla V_i.(u)$, $\nabla b_i(u)$, $\nabla M^r_{i\cdot}(u)$, and $\nabla\mu(u)$ in MF-MPC as $\nabla\Theta^s_m(u)$ in our FR-FMSS. Finally, we specialize Algorithm 1 and Algorithm 2 based on the above settings. In addition, MF-MPC degenerates to PMF [21] when it does not consider the multiclass preference context. Hence, we can implement a federated version of PMF using our FR-FMSS regardless of whether its optimization method is gradient descent or alternating least squares. SVD++ [12] can also be seen as a special case of MF-MPC with one-class preference context [18]. Therefore, we can also use our FR-FMSS to implement a federated version of SVD++. In the above algorithms, Y_m does not change in all the iterations, so Y_m only needs to be calculated in the first iteration.

eALS [6] is designed for item ranking. First, we set p_u. in eALS as Θ^p_u in our FR-FMSS, and set q_i., S^q, and S^p as Θ^s_m in our FR-FMSS. Second, we find that only q_i. is ID-sensitive parameter. Third, because eALS does not use gradient descent to update its parameters, we set $A_{if}(u) = [\omega_{ui}r_{ui} - (\omega_{ui} - c_i)\hat{r}^f_{ui}]p_{uf}$, $B_{if}(u) = (\omega_{ui} - c_i)p^2_{uf}$, and $\mathrm{S}^p(u)$ in eALS as $\nabla\Theta^s_m(u)$ in our FR-FMSS. Then we rewrite the update rule of q_{if} as $q_{if} = \frac{\sum_{u\in\mathcal{R}_i} A_{if}(u) - c_i \sum_{k\neq f} q_{ik} s^p_{kf}}{\sum_{u\in\mathcal{R}_i} B_{if}(u) + c_i s^p_{ff} + \lambda}$ so that the server can use the received $A_{if}(u)$, $B_{if}(u)$, and $\mathrm{S}^p(u)$ to update q_{if}. Finally, we specialize Algorithm 1 and Algorithm 2 based on the above settings. Other algorithms for item ranking can also be federated via our FR-FMSS, including BPR [19], FISM [9], and Mult-VAE [15]. Since the algorithms for item ranking using one-class feedback always need to sample some negative samples, their federated versions only use secret sharing and do not use fake marks.

Fossil [5] uses FISM [9] to capture the long-term preference of the users and high-order Markov Chains to capture the short-term preference of the users. First, we set η^u_k in Fossil as Θ^p_u in our FR-FMSS, and set Q_j, β_j, P_j, and η_k in Fossil as Θ^s_m in our FR-FMSS. Second, we find that Q_j, β_j, and P_j are ID-sensitive parameters. Third, we use gradient descent to update the parameters of Fossil, so we set $\nabla Q_j(u)$, $\nabla\beta_j(u)$, $\nabla P_j(u)$, and $\nabla\eta_k(u)$ in Fossil as $\nabla\Theta^s_m(u)$ in our FR-FMSS. Finally, we specialize Algorithm 1 and Algorithm 2 based on the above settings. Because we use batch gradient decent to update Fossil, all the items in a user's rated item sequence will be used to calculate the gradients in one iteration, which means that the order information of the rated item sequence is transparent to the server. Therefore, the steps for applying FR-FMSS to Fossil are very similar to those of MF-MPC. Similar to Fossil, a federated version of FPMC [20] can also be implemented using our FR-FMSS. As for SASRec [10] using neural networks, its models on all the clients are also the same, which means that our FR-FMSS can also be applied to it.

5 DISCUSSIONS

Because our FR-FMSS uses fake marks and secret sharing, the server and all the clients can infer no private information from their received data. Moreover, a client will send random numbers to several randomly sampled clients. Therefore, unless the server colludes with all the clients, the privacy of each client is hard to be inferred. This means that our FR-FMSS is more secure than the algorithms that require third-party servers or denoising clients such as FedGNN [24] and FedRec++ [16].

If we just conduct distributed training without using fake marks and secret sharing, the communication cost in one iteration is $\sum_{u\in\mathcal{U}} |\mathcal{M}_u||\nabla\Theta^s_m(u)|$, where $|\nabla\Theta^s_m(u)|$ denotes the dimension of $\nabla\Theta^s_m(u)$. As for our FR-FMSS, the communication cost of secret sharing is $\sum_{u\in\mathcal{U}} |\mathcal{N}_u||\mathcal{M}_u \cup \mathcal{M}'_u||\nabla\Theta^s_m(\tilde{u},u)|$, and that when all the clients upload data to the server is $\sum_{u\in\mathcal{U}} |\mathcal{M}_u \cup \mathcal{M}'_u \cup \bar{\mathcal{M}}_u||\nabla\Theta^s_m(u)|$. Therefore, the increased communication cost is $\sum_{u\in\mathcal{U}} |\mathcal{N}_u||\mathcal{M}_u \cup \mathcal{M}'_u||\nabla\Theta^s_m(\tilde{u},u)| + \sum_{u\in\mathcal{U}} |\mathcal{M}'_u \cup \bar{\mathcal{M}}_u \backslash \mathcal{M}_u||\nabla\Theta^s_m(u)|$. Because a client only needs to send $\nabla\Theta^s_m(\tilde{u},u)$ to a small number of clients

Algorithm 2 ClientTrain($\Theta^s_m, m \in \mathcal{M}$;t;$\gamma$), the algorithm of FR-FMSS in the client u

$\nabla\Theta^p(u) \leftarrow 0$
for $m \in \mathcal{M}_u$ **do**
  $\nabla\Theta^s_m(u) \leftarrow 0$
  $Y_m(u) \leftarrow 0$
end for
for $m \in \mathcal{M}_u$ **do**
  Calculate $\nabla\Theta^s_m$ and $\nabla\Theta^p_u$ with the recommendation algorithm
  $\nabla\Theta^s_m(u) \leftarrow \nabla\Theta^s_m(u) + \nabla\Theta^s_m$
  $Y_m(u) \leftarrow 1$
  $\nabla\Theta^p(u) \leftarrow \nabla\Theta^p(u) + \nabla\Theta^p_u$
end for
Sample fake marks $\mathcal{M}'_u$ from $\mathcal{M}\backslash\mathcal{M}_u$ with $|\mathcal{M}'_u| = \rho|\mathcal{M}_u|$
for $m \in \mathcal{M}'_u$ **do**
  $\nabla\Theta^s_m(u) \leftarrow 0$
  $Y_m(u) \leftarrow 0$
end for
Sample $\mathcal{N}_u$ from $\mathcal{U}\backslash\{u\}$
for $m \in \mathcal{M}'_u \cup \mathcal{M}_u$ **do**
  Generate $|\mathcal{N}_u| + 1$ random numbers that meet $\nabla\Theta^s_m(u) = \sum_{\tilde{u}\in\mathcal{N}_u\cup\{u\}} \nabla\Theta^s_m(\tilde{u}, u)$
  Keep $\nabla\Theta^s_m(u, u)$ and send $\nabla\Theta^s_m(\tilde{u}, u), \tilde{u} \in \mathcal{N}_u$ to the client $\tilde{u}$
  Generate $|\mathcal{N}_u| + 1$ random numbers that meet $Y_m(u) = \sum_{\tilde{u}\in\mathcal{N}_u\cup\{u\}} Y_m(\tilde{u}, u)$
  Keep $Y_m(u, u)$ and send $Y_m(\tilde{u}, u), \tilde{u} \in \mathcal{N}_u$ to the client $\tilde{u}$
end for
Synchronize()
$\bar{\mathcal{M}}_u \leftarrow \emptyset$
for $m \in \mathcal{M}$ **do**
  if receive a set of $\nabla\Theta^s_m(u, \tilde{u}), u \in \mathcal{N}_{\tilde{u}}$ **then**
    $\bar{\mathcal{M}}_u \leftarrow \bar{\mathcal{M}}_u \cup \{m\}$
    $\nabla\Theta^s_m(u) \leftarrow \nabla\Theta^s_m(u, u) + \sum_{\tilde{u}} \nabla\Theta^s_m(u, \tilde{u})$
    $Y_m(u) \leftarrow Y_m(u, u) + \sum_{\tilde{u}} Y_m(u, \tilde{u})$
  end if
end for
Upload $\nabla\Theta^s_m(u), Y_m(u), m \in \mathcal{M}_u \cup \mathcal{M}'_u \cup \bar{\mathcal{M}}_u$ to the server
$\Theta^p_u \leftarrow \Theta^p_u - \gamma \frac{\nabla\Theta^p(u)}{|\mathcal{M}_u|}$

and $\rho = \frac{|\mathcal{M}'_u|}{|\mathcal{M}_u|}$ is also very small, the increased communication cost is not high.

6 CONCLUSIONS AND FUTURE WORK

We propose a generic cross-user federated recommendation framework called FR-FMSS, which mainly makes use of two techniques, i.e., fake marks and secret sharing. As long as the model parameters of an un-federated recommendation algorithm are consistent on all the clients, our FR-FMSS can be applied to it, i.e., converting a traditional un-federated recommendation algorithm to a privacy-aware federated one without sacrificing the accuracy. We take MF-MPC, eALS and Fossil as examples to show how to apply our FR-FMSS to a specific algorithm. For future works, we are interested in extending our FR-FMSS to cross-organization federated recommendation. Moreover, we are also interested in studying efficient interactions methods among the server and clients, in terms of both computation and communication complexity.

ACKNOWLEDGMENTS

We thank the support of National Natural Science Foundation of China Nos. 61836005 and 61872249, and Shenzhen Basic Research Fund No. JCYJ20200813091134001.

REFERENCES

[1] Muhammad Ammad-ud-din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. *CoRR* abs/1901.09888 (2019). arXiv:1901.09888 http://arxiv.org/abs/1901.09888
[2] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2019. Secure Federated Matrix Factorization. *CoRR* abs/1906.05108 (2019), 1–1. arXiv:1906.05108 http://arxiv.org/abs/1906.05108
[3] Jinting Deng and Pinxin Liu. 2017. Consultative Authoritarianism: the Drafting of China's Internet Security Law and E-commerce Law. *Journal of Contemporary China* 26, 107 (2017), 679–695.
[4] Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. 2019. Understanding the Scope and Impact of the California Consumer Privacy Act of 2018. *Journal of Data Protection & Privacy* 2, 3 (2019), 234–253.
[5] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain.* 191–200.
[6] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016.* 549–558.
[7] Guangneng Hu and Qiang Yang. 2020. PrivNet: Safeguarding Private Attributes in Transfer Learning for Recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online, 4506–4516.
[8] Jia-Yun Jiang, Cheng-Te Li, and Shou-De Lin. 2019. Towards a More Reliable Privacy-Preserving Recommender System. *Inf. Sci.* 482 (2019), 248–265.
[9] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored Item Similarity Models for Top-N Recommender Systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013.* 659–667.
[10] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018.* 197–206.
[11] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492
[12] Yehuda Koren. 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, Las Vegas, Nevada, USA, August 24-27, 2008.* 426–434.
[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
[14] Dongsheng Li, Chao Chen, Qin Lv, Li Shang, Yingying Zhao, Tun Lu, and Ning Gu. 2016. An Algorithm for Efficient Privacy-Preserving Item-Based Collaborative Filtering. *Future Generation Computer Systems* 55 (2016), 311–320.
[15] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. Republic and Canton of Geneva, CHE, 689–698.
[16] Feng Liang, Weike Pan, and Zhong Ming. 2021. FedRec++: Lossless Federated Recommendation with Explicit Feedback. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021.* 4224–4231.
[17] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2020. FedRec: Federated Recommendation with Explicit Feedback. *IEEE Intelligent Systems* (2020), 1–1.
[18] Weike Pan and Zhong Ming. 2017. Collaborative Recommendation with Multiclass Preference Context. *IEEE Intelligent Systems* 32, 2 (2017), 45–51.
[19] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009.* 452–461.

[20] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-Basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*. 811–820.

[21] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*. 1257–1264.

[22] Adi Shamir. 1979. How to Share a Secret. *Commun. ACM* 22, 11 (1979), 612–613.

[23] Paul Voigt and Axel Von dem Bussche. 2017. The EU General Data Protection Regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10 (2017), 3152676.

[24] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. FedGNN: Federated Graph Neural Network for Privacy-Preserving Recommendation. *CoRR* abs/2102.04925 (2021). arXiv:2102.04925 https://arxiv.org/abs/2102.04925

[25] Enyue Yang, Yunfeng Huang, Feng Liang, Weike Pan, and Zhong Ming. 2021. FCMF: Federated Collective Matrix Factorization for Heterogeneous Collaborative Filtering. *Knowledge-Based Systems* 220 (2021), 106946.

[26] Senci Ying. 2020. Shared MF: A Privacy-Preserving Recommendation System. *CoRR* abs/2008.07759 (2020). arXiv:2008.07759 https://arxiv.org/abs/2008.07759