Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 2, No. 3, 2022

Communication Efficient Federated Personalized Recommendation

Lingtao Wei *

College of Computer Science & Technology, Qingdao University, Qingdao Shandong, China
* **Corresponding author**: Email: qduwlt@163.com

Abstract: Recommendation systems that can correctly predict user preferences in the information age have become an important factor for business success. However, recommendation systems require users' personal information, and centralized collection and processing of user data may lead to serious privacy risks. Good progress has been made in recent years using federated learning techniques for privacy-preserving recommendations, but several key challenges remain to be addressed: most federated recommender systems ignore communication process optimization, inequities in aggregation of federated models, and lack of personalization to users. In this paper, we propose a communication efficient and fair personalized federated recommendation approach (CFFR) to address these challenges. CFFR uses adaptive client group selection to personalize models while accelerating the training process. A fair-aware model aggregation algorithm is proposed that adaptively captures the performance and data imbalance among different clients to address the unfairness problem. Extensive experimental results demonstrate the effectiveness and efficiency of our proposed method.

Keywords: Recommendation; Federated learning; Grouping.

1. Introduction

Users in the digital age are overwhelmed by the vast amount of information on the Internet and have difficulty getting the information they need. Recommender systems (RS) help users discover the most useful information or services at the right time and in the most appropriate way. A recommendation system that can correctly predict user preferences becomes a key element of business success. Recommender systems need to collect user data to perform analytical modeling to analyze user preferences in order to provide services to users. However, with introduction of data privacy laws, increased regulation, and growing public unease about how personal data is collected and used, it has become increasingly important for recommendation systems to find a better balance between personalization and privacy.

To address this issue, many studies on the protection of personal information in recommender systems have been conducted, and good progress has been made in recent years on privacy-preserving recommendations using federated learning techniques, but several key challenges remain to be addressed: most federated recommender systems only consider model performance and privacy-preserving capabilities and ignore communication process optimization, inequity problems in aggregation of federated models, and the systems, and insufficient research on personalization of federated recommender systems.

We aim to develop a more efficient way to train recommendation models faster in a federated environment so that users can enjoy accurate recommendations without spending unnecessary effort and suffering from high communication costs. The user in a federation learning scenario bears most of the costs involved in training the model. Specifically, users may experience performance degradation and increased communication load during training, especially if they have to wait for a long time to enjoy quality personalized recommendations, which can reduce their acceptance of the service.

In this paper, we propose a communication-efficient and fair personalized federated recommendation method (CFFR) to address these challenges. CFFR introduces a communication-efficient scheme that uses adaptive client group selection to achieve model personalization while accelerating the training process. A fair-aware model aggregation algorithm is proposed, which can adjust the parameter aggregation weights according to the imbalance state of performance and data among different clients to solve the unfairness problem. Extensive experimental results demonstrate the effectiveness and efficiency of our proposed method.

The remainder of this paper is organized as follows. Section 2 discusses related studies. Section 3 elaborates on our proposed CFFR method. Section 4 describes extensive experiments. Finally, we summarize our work in section 5.

2. Related Works

2.1. Federated Learning for Recommender System

Federated learning [1] is a machine learning technique that allows multiple clients to directly participate in model training without directly sharing data. Recommendation algorithms can be migrated to a federation learning scenario in order to achieve protection of users' private information, using users' local data for model training and implementing predictive inference. Typical federated learning-based recommender systems are FCF [2] and Fed-MVMF [3], but the protection of the gradient data uploaded by each client is not implemented in the two methods, since the user's rating information can be inferred if given the gradients of a user uploaded in two continuous steps [4]. The model updates sent to the server may contain enough information to discover the original data, FedRec [5] propose to give some (instead of all) unrated item to be randomly sampled and to assign with some virtual ratings. FedRec++ [6] extends the FedRec [5] to eliminate the nose in a privacy-aware manner by allocating

some denoising clients.

2.2. Local Differential Privacy

(ε-Local Differential Privacy, ε-LDP [7]) A randomized algorithm $f : \mathbb{T} \to \mathbb{Y}$ with domain $\mathbb{T}$ and range $\mathbb{Y}$ satisfies ϵ-LDP if and only if, for any inputs $t, t' \in \mathbb{T}$ and output $y \in \mathbb{Y}$:

$$\Pr[f(t) = y] \leq e^{\varepsilon} \Pr[f(t') = y] \quad (1)$$

where ε is the privacy budget that controls the balance between utility and privacy protection? We implement f with a Laplace mechanism by adding well designed noise to the data.

3. Methodologies

3.1. System Overview

Some commercial behaviors of users, such as paid purchase records on movie and video websites are required to be kept by the server, i.e., user item interaction records cannot be hidden from the server, but the specific preferences of users for items, i.e., user item rating data, can still be stored locally on the user's device rather than uploaded to the server, thus reducing the privacy risk of users to some extent. We assume that there are n users $u_1, \ldots, u_n$ and m items $i_1, \ldots, i_m$. Firstly, we calculate user similarity based on the user item interaction records stored on the server, and generate a grouping of target user i based on the user similarity, and the users in the group collaborate to train the recommendation model, and we also propose a new model aggregation method to speed up the model training in the model training. Our method can serve small batches of users with similar preferences quickly and accurately.

3.2. The Recommendation Model

Any representation learning model can be used in our framework to obtain embeddings of users and items (e.g., MF [8], BPR [9], and GMF [10]). Here, we employ generalized matrix factorization (GMF) to learn the representation vector of users and items. We define the user embedding as $e_u \in R^k$ and the item embedding as $e_v \in R^k$, with k denoting the embedding dimension. The user embedding and item embedding are fed into the fully connected layer to obtain the prediction scores $\hat{r}_{uv}$.

$$\hat{r}_{uv} = w^T(e_u \odot e_v) \quad (2)$$

where $w \in R^{1\times k}$ are the parameters of the fully connected layer. GMF is used to train the model by minimizing the loss function between the predicted scores and their true values, and here we use the mean squared error for explicit recommendation. The optimization problem that minimizes the loss function can be expressed as follows:

$$\min \sum_{(u,\ v)\in R^{M\times N}} (r_{uv} - \hat{r}_{uv})^2 + \lambda(||e_u||^2 + ||e_v||^2) \quad (3)$$

During model training, we use LDP to add perturbations to the gradient to better protect user privacy since the gradient can be used to infer the real information of the user. The local model gradient is defined as g, which is perturbed by the stochastic algorithm M as follows:

$$M(g) = \mathrm{clip}(g,\ \delta) + \mathrm{La}(0,\ \lambda) \quad (4)$$

$\mathrm{clip}(g,\ \delta)$ denotes a gradient clipping technique that limits the gradient g to the δ scale to prevent gradient explosion and better train the model. $\mathrm{La}(0,\ \lambda)$ denotes Laplace noise with mean 0. λ controls the intensity of Laplace noise. The larger λ is, the higher the noise is and the higher the degree of privacy protection is. After randomizing the gradient operation using LDP technique, it is almost impossible to infer the original user data from the weights. The weights on the client are then uploaded to a central server for aggregation.

3.3. Grouping Method

The ideal user grouping should be a good representation of the user pattern, and the interaction records of user items within that grouping must have a high density. Therefore, user grouping should consider the following two factors. One is to estimate how similar the interaction records of other users are to those of the target user. Recommendation performance will be enhanced if the target user is grouped with other users having similar interaction patterns. The second is to evaluate the density of user records. The more user rating data there is, the denser the data within the grouping and the more useful information can be captured.

Therefore, we define a group scoring function as a weighted sum of user rating frequency scores and similarity scores.

For a group of size k, user i, other user j, $(1 \leq j \leq n)$, the group score function is defined as follows.

$$\mathrm{score}(i,\ j) = \begin{cases} 1, & i = j \\ l * \mathrm{freq}(j) + (1-l)\mathrm{sim}(s_i, s_j), & i \neq j \end{cases} \quad (5)$$

where $l = \frac{k}{n}$. Jaccard similarity [11] is used to evaluate the behavioral similarity of users, and s_i and s_j are the interaction records of users stored on the server as bit vectors.

$$\mathrm{sim}(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|} \quad (6)$$

$\mathrm{freq}(j)$ is the rating frequency function of user j, defined as the proportion of the number of interaction items of user j to the total number of items?

The grouping process is as follows, the server calculates the grouping scores with other users for the target user i, sorts the users in descending order of the grouping scores, and extracts the top $k-1$ users to form a grouping $U_i^{(k)}$ with grouping size k with the target user i.

3.4. Parameter Aggregation Method

The data on the client side is non-IID, the local dataset is usually heterogeneous between clients in terms of size and distribution, and the effect of model initialization, the local model performance of each client during model training may be less than optimal. Therefore, we propose a weighting strategy to handle the differences in model performance and data size between clients.

First, we denote the performance of client i as p_i, denoted as

$$p_i = \frac{1}{\mathrm{RMSE}(i)} \quad (7)$$

$\mathrm{RMSE}(i)$ is the error between the predicted score of the user i's local model and the true value, and a larger value indicates a larger local model error and a worse performance of its local model. Then we normalize it:

$$p_i' = \frac{p_i}{\sum_{i\in U_i^{(k)}} p_i} \quad (8)$$

Secondly, for the effect of data sample size imbalance, the user rating frequency score reflects the sample data size on the user's device, thus we define $q_i = \mathrm{freq}(i)$, which is normalized as follows:

$$q_i' = \frac{q_i}{\sum_{i \in U_i^{(k)}} q_i} \quad (9)$$

Finally, we get the final parameter aggregation function:

$$agg_i = \gamma \alpha^t p_i' + \beta q_i' \quad (10)$$

$$w^{t+1} = \sum_{i \in U_i^{(k)}} \frac{agg_i}{\sum_{i \in U_i^{(k)}} agg_i} w_i^{t+1} \quad (11)$$

where w^{t+1} is the global model updated by $t+1$ rounds of aggregation, α and β are two hyperparameters? As the model is updated by multiple rounds of aggregation, the performance of each local model will be improved, and then the main factor affecting the aggregation of the model will be shifted to the number of samples of each client, so we gradually weaken the influence of the performance of each local model when re-aggregating the parameters, i.e., the parameter α^{t+1} used in the $t+1$ round of aggregation is obtained by multiplying the parameter α^t used in the t round of aggregation by the discount factor γ, so that the value of α decreases with each round of the aggregation process.

4. Experiments

4.1. Data

Experiments are run on MovieLens 100K(ML-100K) and Movielens 1M(ML-1M). All the dataset have been widely used in recommender systems literature to evaluate collaborative filtering algorithms. These datasets are commonly used benchmark datasets in recommendation system research.

4.2. Evaluation

For the evaluation metric, we adopt the Root Mean Square Error (RMSE) as performance metrics, which is wide adopted in many related studies for performance evaluation. We group the users after randomly removing one local rating record, and the users within the group are trained for the federal model, and estimate the RMSE between the prediction score and the original score for recommendation performance evaluation. RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)} \quad (12)$$

Where n = Total number of rating data records for users in the current group in the test data, r_i = original rating information, $\hat{r}_i$ = predictive rating information.

We compare our proposed CFFR with the popular FedAvg method using GMF as recommendation model. The goal of our evaluation is to compare the recommendation quality and convergence speed in a federated setting, so we omit comparisons with other classical centralized algorithms. Users are grouped after randomly deleting a local rating record, and the users in the group are trained on the federation model and the error between the predicted and original scores is estimated for recommendation performance evaluation. The number of federation training rounds is set to 50, and the relevant parameters are set by default as follows: lr= 0.1, $\alpha = 1.0$, $\gamma = 0.98$, $\beta = 1 - \alpha$, $k = 200$, δ=0.01, λ=0.03.

4.3. Prediction Accuracy

In this section, we identify the improvement of recommendation performance by grouping. First, let us consider the effect of group size on recommendation performance. The smaller the group size, the higher the density. However, the information about the rating patterns is reduced. Conversely, if the group size increases, the information about the rating patterns increases, but the density decreases. Therefore, the appropriate group size k for grouping is important to improve the recommendation performance.

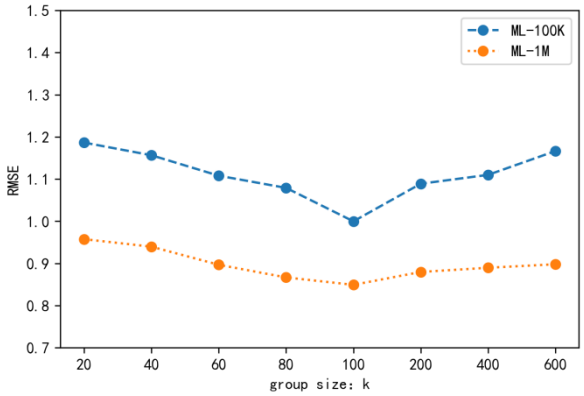

Figure 1. Recommender performance comparison by group size

In this section, we determine the improvement of grouping on recommendation performance. First, we analyze the effect of grouping size on recommendation performance. The smaller the group size, the higher the data density and the less information about user rating patterns, and conversely, if the group size increases, the information about user rating patterns increases, but the data density decreases.

Figure 1 shows that the performance of recommendation prediction is not satisfactory when the group size is very small or very large. The proposed method in this paper performs better in the ML100K dataset when the group size k is from 20 to 80. Among them, the best performance is achieved at k = 40. In the ML1M dataset, the recommended prediction performs better when k is 80. This result indicates that an appropriate grouping size k can effectively improve the recommendation performance.

4.4. Analysis of Model Convergence

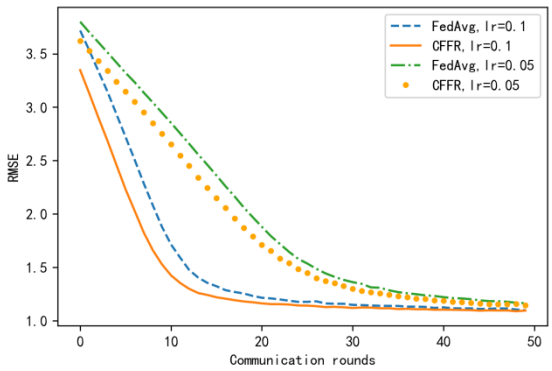

Figure 2. Convergence speed between CFFR and FedAvg

In addition to the comparison of the effect of group size on performance, this paper also analyzes the convergence speed difference between the proposed method and the comparison method FedAvg [12] by comparing the variation of RMSE metrics with the number of model aggregations under different learning rate settings. Since the Federated Recommender System needs to aggregate updates uploaded by multiple clients to perform global model updates, the convergence speed of the model is the focus of the scheme design. Therefore, this section compares the changes in model accuracy between the proposed method and FedAvg with different parameter settings as the number of aggregations

increases.

Set the group size k=200，$\alpha = 1.0$，$\gamma = 0.98$，$\beta = 1 - \alpha$,and the learning rate is set to 0.05 and 0.1. The experimental results are shown in Figure 2. Under different learning rate settings, this scheme has a faster convergence speed than the FedAvg method, and the effect is obvious at larger learning rates. This is due to the fact that in the early stage of federation model training, the performance of the locally trained model varies among clients, and the aggregated model is more affected by the performance of each client when the learning rate is larger. The proposed method can accelerate the convergence of the model by weakening the impact of updates of the poorly performing local models in the early stage of training.

4.5. Ablation Study

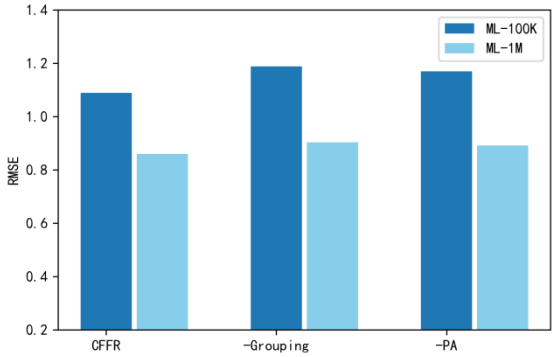

Figure 3. Contributions of the individual components of CFFR

The proposed federal recommendation algorithm in this paper contains two main modules, the first part is the grouping strategy, and the second part is the parameter aggregation mechanism that incorporates the performance differences and data differences among client models. In order to explore the performance impact of the two modules on the final solution, this section conducts ablation experiments on two datasets, ML100K and ML1M, and analyzes the final results, which are shown in Figure 3.

In Fig. 3, (-Grouping) is the removal of user grouping module and (-PA) is the removal of parameter aggregation module. From Fig. 3, it can be observed that the scheme in this paper achieves the best results on both datasets, which verifies the importance of the two key modules. Among them, the model performance degrades significantly in the experiments where the user grouping module is removed, which proves that the lack of personalized grouping significantly reduces the learning capability of the framework. The absence of the parameter aggregation method still significantly degrades the model performance for the same number of aggregation rounds, demonstrating the importance of the impact of local model performance on the aggregated global model when the model is aggregated. Each of the modules demonstrated by the ablation experiments improves the model performance in a different way and significantly.

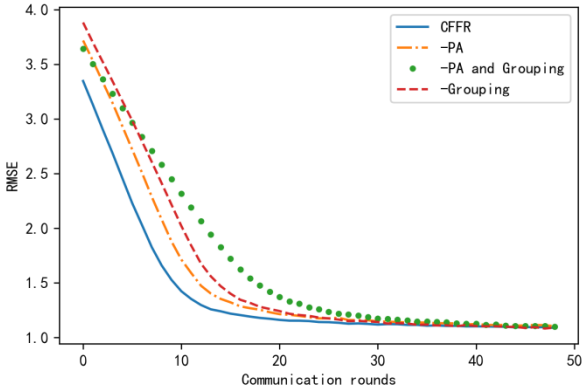

Figure 4. Comparison of model convergence performance

Figure 4 shows the change of model convergence after removing each module on the ML-100K dataset. When the parameter aggregation module (-PA) proposed in this scheme is removed, the convergence of the model slows down slightly, but when the user grouping module (-Grouping) is removed, i.e., the same number of users are randomly selected to collaborate in model training, the convergence of the model slows down significantly, and at this time there are large differences in the preferences and scoring patterns of the users participating in the federal training, resulting in the inability to get the maximum benefit from the training of other users, thus affecting the global model This has an impact on the convergence speed of the global model. Compared with traditional training methods, i.e., removing two modules (-grouping and parameter aggregation), the proposed method has significant advantages in model convergence speed and prediction accuracy

5. Conclusion

In this paper, we propose a communication-efficient personalized federation recommendation method that uses adaptive client grouping to enable users with the same interests to collaborate in training the model, and assigns different aggregation weights to the current round of weighted aggregation of each client's updated model according to the amount of data and performance status among different clients in training, which reduces the communication burden and accelerates the model training at the same time. We have conducted experiments on real datasets, and the experimental results demonstrate the effectiveness of the proposed method in this paper.

References

[1] J. Koneˇcn`y, H.B. McMahan, et al.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492, 2016.

[2] M. Ammad-Ud-Din, E. Ivannikova, et al.: Federated collaborative filtering for privacy-preserving personalized recommendation system. arXiv preprint arXiv:1901.09888, 2019.

[3] A. Flanagan, W. Oyomno, et a.l: Federated multi-view matrix factorization for personalized recommendations. arXiv preprint arXiv:2004.04256, 2020.

[4] D. Chai, L. Wang, K. Chen, and Q. Yang: Secure federated matrix factorization. IEEE Intelligent Systems, vol. 36 (2020) No. 5 p.11-20.

[5] G. Lin, F. Liang, W. Pan, and Z. Ming: Fedrec: Federated recommendation with explicit feedback. IEEE Intelligent Systems, vol. 36 (2021) No. 5 p.21-30.

[6] F. Liang, W. Pan, and Z. Ming. Fedrec++: Lossless federated recommendation with explicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence (online February 2-9, 2021). vol.35, p. 4224.

[7] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J.A. McCann, S.Y. Philip: High-dimensional crowdsourced data publication with local differential privacy, IEEE Trans. Inf. Forensics Secur. Vol. 13 (2018) No.9 p.2151– 2166.

[8] Y. Koren, R. Bell, C. Volinsky: Matrix factorization techniques for recommender systems, Computer , Vol.42 (2009) No.8 p.30– 37.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme: BPR: bayesian personalized ranking from implicit feedback: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (Montreal, Canada, June 18-21, 2009). p. 452.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua: Neural collaborative filtering, Proceedings of the 26th International Conference on World Wide Web (Perth, Australia, April 3-7, 2017). p.173.

[11] S. Niwattanakul, et al.: Using of Jaccard coefficient for keywords similarity, Proc. International Multi Conference of Engineers and Computer Scientists (Hong Kong, China, March 13-15, 2013). Vol.1, p.380.

[12] Mcmahan, H. B , et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data." 2016.