2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-6283-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMLA55696.2022.00107

One-Shot Federated Group Collaborative Filtering

Maksim E. Eren
Analytics Division, LANL
Los Alamos, USA
maksim@lanl.gov

Manish Bhattarai
Theoretical Division, LANL
Los Alamos, USA
ceodspspectrum@lanl.gov

Nicholas Solovyev
Theoretical Division, LANL
Los Alamos, USA
nks@lanl.gov

Luke E. Richards
UMBC
Baltimore, USA
lurich1@umbc.edu

Roberto Yus
UMBC
Baltimore, USA
ryus@umbc.edu

Charles Nicholas
UMBC
Baltimore, USA
nicholas@umbc.edu

Boian S. Alexandrov
Theoretical Division, LANL
Los Alamos, USA
boian@lanl.gov

***Abstract*—Non-negative matrix factorization (NMF) with missing-value completion is a well-known effective Collaborative Filtering (CF) method used to provide personalized user recommendations. However, traditional CF relies on a privacy-invasive collection of user data to build a central recommender model. One-shot federated learning has recently emerged as a method to mitigate the privacy problem while addressing the traditional communication bottleneck of federated learning. In this paper, we present the first one-shot federated CF implementation, named One-FedCF, for groups of users or collaborating organizations. In our solution, the clients first apply local CF in-parallel to build distinct, client-specific recommenders. Then, the privacy-preserving local item patterns and biases from each client are shared with the processor to perform joint factorization in order to extract the global item patterns. Extracted patterns are then aggregated to each client to build the local models via information retrieval transfer. In our experiments, we demonstrate our approach with two MovieLens datasets and show results competitive with the state-of-the-art federated recommender systems at a substantial decrease in the number of communications.**

***Index Terms*—privacy, non-negative matrix factorization, one-shot, federated learning, recommendation system**

I. Introduction

Established machine learning (ML) recommender methods rely on a privacy-invasive collection of user data to train central models at the processors. As privacy awareness grows and stricter regulations are introduced, user expectations regarding data privacy are changing. For instance, the European General Data Protection Regulation (GDPR) has implications on how recommender systems collect and handle user data [1]. Future recommender systems will need to prioritize privacy policies to remain legally compliant. Federated learning-based recommender systems are a possible solution to this problem by performing part of the training process client-side, minimizing the amount of data sent to the processors.

Federated recommenders based on Collaborative Filtering (CF) have already been proposed [2], [3]. Such systems identify and filter potential user interests by collaboratively learning from the past preferences of many users [4], [5], performing multiple communications between the server and its clients at every iteration. Multi-round client participation can suffer from communication bottlenecks due to limited user data plans, potentially slow and unreliable network connections, and costs of cryptographic protocols [6]. The increased number of communications also increases the risk for attackers to intercept model updates. Thus, communication efficiency in federated CF has been an active field of study [7], [8]. While these approaches lower the communication complexity, they still expect multi-round client participation. In response to this set of desirables, an emerging field of *one-shot federated learning* addresses the communication problem by performing just a *single round of communication* between each client and the processor per training session [6], [9]–[14]. While the proposed one-shot federated methods significantly reduce the cost of communication, work thus far has only focused on supervised and semi-supervised classification problems.

In this paper, we introduce, what is to the best of our knowledge, the first implementation of one-shot federated learning for CF and recommenders. Our approach, named One-FedCF, is an unsupervised, communication-efficient (for the number of communications) federated method for providing privacy to a group of users or set of organizations (clients). In our one-shot federated setup, we achieve *single pair communication* after the initial small-sized communication to calculate the global mean between the server and its clients. One-FedCF first trains local client-specific CF in parallel. Specifically, our local CF is based on Non-negative Matrix Factorization (NMF) on user's explicit feedback data [15]. The extracted latent factors for items from each client along with the item biases are then shared with the processor to extract the global item patterns via transfer of information retrieval and joint factorization. Global item patterns and biases are then aggregated to each client to improve the recommendation capability of the distinct local models. Since the shared data between the clients and server is with respect to the items, the added benefit of One-FedCF is that the user-based information is abstracted, where the processor does not know which user is part of which group (client) or how many users are in the group. In our experiments, we evaluate our approach on the MovieLens 100K and MovieLens 1M datasets [16] and show that One-

978-1-6654-6283-9/22/$31.00 ©2022 IEEE
DOI 10.1109/ICMLA55696.2022.00107

FedCF achieves similar root-mean-square error (RMSE) with a single pair of communication as compared to the state-of-the-art federated CF with multi-round communication. In summary, our main contributions include:

- Introducing the first implementation of a one-shot federated collaborative filtering method.
- Demonstrating that NMF with joint factorization and transfer of information retrieval improves the recommendation capability of groups or organizations.
- Showcasing that our one-shot approach achieves similar RMSE as compared to the state-of-the-art federated CF.

II. Background

A. Federated Learning

Federated learning formalized the concept of learning local models and then aggregating them to create a centralized global model in the pursuit of sharing knowledge derived from data. Further research has examined more efficient communication [17], [18], ensuring privacy of participants [19], [20], and extensions to recommendation systems [2], [3], [7], [20], [21]. A known drawback of these methods is the need for frequent communication of model updates. Our work is most similar to the concept of one-shot federated learning, where the goal is to limit communication to one iteration. [6]. A handful of approaches have examined methods for knowledge distillation [9], [10], data distribution modeling for synthetic data generation [9], [12], model selection [14], probabilistic aggregation [11] and addressing privacy-preserving aggregation [13]. Methods have been evaluated in the supervised and semi-supervised image and text classification domain and, to our knowledge, our work introduces the first one-shot federated learning method for unsupervised recommendation.

Frequent communication between devices and the server in federated learning is also a concern as the user data is left vulnerable to potential attacks [22]. Algorithms attempt to mitigate membership inference attacks [23], in-training data point identification, and inversion attacks [24]. The majority of federated learning methods use gradient information sent to the central server on the weights of the model. However, prior work has shown a method to recover private data from the exchanged gradients [25]. In response, works have examined how to integrate homomorphic encryption for data sharing [21], [26] and differential privacy algorithms [20]. Prior work in federated CF similarly evaluated the privacy of federated matrix factorization with gradient sharing and showed that original data can be reconstructed from gradient updates [22]. Within our work we forgo sharing gradients entirely.

Several prior works explore creating federated recommender systems using matrix factorization to solve problems similar to our work within CF [27], [28]. They focus on sharing gradient updates based on a *secret sharing* mechanism. Our work is similar to [29] in which NMF is used as a local privacy-filter within a federated learning paradigm. Their approach involves learning a local latent representation for a classification task. In our approach, we use latent representation for the utility of CF within a group setting for recommendation [30].

B. Non-negative Matrix Factorization (NMF)

NMF is an unsupervised dimensionality reduction method based on low-rank approximation. NMF approximates a given non-negative observation matrix $\mathbf{X} \in \mathbb{R}_{+}^{n \times m}$, as a product of two non-negative matrices, i.e., $\mathbf{X} \approx \mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}_{+}^{n \times k}$, and $\mathbf{H} \in \mathbb{R}_{+}^{k \times m}$, and usually $k \ll m, n$. Here, n is the number of samples, m is the number of features, and k is the low-rank of the approximation. We perform this factorization via a non-convex minimization with non-negativity constraint, utilizing the multiplicative updates algorithm [31], and Frobenius norm as distance metric to minimize $||\mathbf{X} - \mathbf{WH}||_F^2$. The factors $\mathbf{W}$ and $\mathbf{H}$ are estimated via alternative updates following the rules $\mathbf{W} = \mathbf{W}\frac{\mathbf{XH}^T}{\mathbf{WHH}^T}$ and $\mathbf{H} = \mathbf{H}\frac{\mathbf{W}^T\mathbf{X}}{\mathbf{W}^T\mathbf{WH}}$.

C. Collaborative Non-negative Matrix Factorization (CNMF)

The distance minimization between $\mathbf{X}$ and approximation $\mathbf{WH}$ in NMF includes the zero entries in $\mathbf{X}$. However, the standard NMF minimization does not work for CF because the zeros (the missing-values) in $\mathbf{X}$ are the future recommendations that we want to perform. To this end, our optimization needs to be done only with respect to the non-zero entries in $\mathbf{X}$ instead. Estimating the missing data value in this way, i.e. the user's future recommendations, is known as the matrix completion problem [32]. For this, we use the modified version of the Collaborative NMF algorithm presented in the *Surprise* package [15], and call it CNMF for simplicity. We modify the *Surprise* algorithm with non-negative projections to ensure the non-negativity of the latent factor matrices, and adopt its bias terms to the federated learning scheme (Section III-A).

$\mathbf{X} \in \mathbb{R}_{+}^{n \times m}$ is a matrix of movie ratings from n users for m movies where each user i rated a subset of m items, such that a non-zero entry $r^{i,j} = \mathbf{X}_{i,j}$ is the rating from user i for movie j. The user profile matrix is given by $\mathbf{W} \in \mathbb{R}_{+}^{n \times k}$ and the movie profile matrix is given by $\mathbf{H} \in \mathbb{R}_{+}^{k \times m}$. The future rating of user i for movie j is predicted with $\hat{\mathbf{X}}_{i,j} = \mathbf{W}_{i,:}\mathbf{H}_{:,j}$. For recommender systems, we also consider the bias terms $b_W \in \mathbb{R}^n$ and $b_H \in \mathbb{R}^m$ to remove the bias given by users or bias for an item. Here b_H describes how well an item is rated compared against the average across all of the items without accounting for the interaction between the item and a given user. Similarly, b_W describes the user's tendency to provide better/worse ratings compared to the average. We also account the group bias, where $nnz(\mathbf{X})$ represents a vector of non-zero entries (ratings) in $\mathbf{X}$, and group bias is:

$$\mu = \frac{1}{|nnz(\mathbf{X})|} \cdot \sum_{i}^{|nnz(\mathbf{X})|} nnz(\mathbf{X})_i \tag{1}$$

such that μ is simply the average of the ratings. The biases are included in predicting the rating from user i for movie j as shown in the Equation 2, and the performance of the prediction can be measured with RMSE $= \frac{||\mathbf{X} - \hat{\mathbf{X}}||_F^2}{\sqrt{mn}}$:

$$\hat{\mathbf{X}}_{i,j} = \mathbf{W}_{i,:}\mathbf{H}_{:,j} + b_{W_i} + b_{H_j} + \mu \tag{2}$$

Here i and j correspond only to the non-zero coordinates (ratings) such that $\mathbf{X}_{i,j} > 0$ for each i and j pair. The minimization in CNMF is based on Tikhonov regularization [33] for regularizing the ill-posed problems to obtain higher prediction accuracy. We perform this minimization with gradient descent updates to estimate the factors $\mathbf{W}$ and $\mathbf{H}$ with closed form alternating update expressions $\mathbf{W} = \mathbf{W}\frac{\mathbf{X}\mathbf{H}^T}{\hat{\mathbf{X}}\mathbf{H}^T + \alpha\mathbf{W}}$ and $\mathbf{H} = \mathbf{H}\frac{\mathbf{W}^T\mathbf{X}}{\mathbf{W}^T\hat{\mathbf{X}} + \beta\mathbf{H}}$ respectively. The bias term updates are $b_W = b_W + \eta_W \sum_{j=1}^{m}(\mathbf{err}_{:,j} - \gamma * b_W)$ and $b_H = b_H + \eta_H \sum_{i=1}^{n}(\mathbf{err}_{i,:} - \delta * b_H)$ where $\mathbf{err}_{i,j} = \mathbf{X}_{i,j} - \hat{\mathbf{X}}_{i,j}$.

III. Method

A. One-FedCF: One-shot Federated Collaborative Filtering

1) Step 1 - Clients (Local CF via Collaborative NMF): Let us consider a set of N clients each with their local data $\mathbf{X}^g \in \mathbb{R}_+^{n^g \times m}$, where g is in range $0 \leq g \leq N$, and non-zeros in $\mathbf{X}^g$ are the ratings for m items (e.g., movies) by n^g users belonging to the group g. The first step involves a pair of small-in-size communications between each client and the server. In order to put global bias into consideration of the local CNMF, each client sends the group bias μ^g (Equation 1) to the server and receives the global bias μ^{Global} (Equation 3).

$$\mu^{Global} = \frac{\mu^1 + \mu^2 + \cdots + \mu^g + \cdots + \mu^N}{N} \tag{3}$$

Next, using μ^{Global} and $\mathbf{X}^g$, each of the N clients trains local CF models in parallel using the Collaborative NMF approach (Section II-C) to obtain local patterns $\mathbf{W}^g \in \mathbb{R}_+^{n^g \times k^g}$ and $\mathbf{H}^g \in \mathbb{R}_+^{k^g \times m}$, and biases b_W^g and b_H^g. At this point, each client has a local CF model. However, we would like to utilize federated learning to improve the performance of these local models. Therefore, we proceed with our method where each client uploads their $(\mathbf{H}^g)^T$ and b_H^g to the server (the superscript T represents the matrix transpose).

2) Step 2 - Server (Joint Factorization via NMF): Once the processor receives the local item factors and the item biases from each client, it first forms the global $\mathbf{X} \in \mathbb{R}_+^{m \times (k^1 + k^2 + \cdots + k^g + \cdots + k^N)}$ (Equation 4) and calculates the global item bias (Equation 5):

$$\mathbf{X} = [(\mathbf{H}^1)^T, (\mathbf{H}^2)^T, \cdots, (\mathbf{H}^g)^T, \cdots, (\mathbf{H}^N)^T] \tag{4}$$

$$b_H^{Global} = \frac{b_H^1 + b_H^2 + \cdots + b_H^g + \cdots + b_H^N}{N} \tag{5}$$

It can easily be seen that concatenation is equivalent to a joint factorization since Frobenius norms are sums of squares of residuals, and sums of squares can be re-arranged to be summed in arbitrary order. Because $\mathbf{X}$ is the concatenation of each $(\mathbf{H}^g)^T$ from the clients, factorization of $\mathbf{X}$ is joint factorization, which serves to identify the common global item patterns. Here, we utilize our method *SPLIT* [34], [35], which excludes patterns that are non-negative linear combinations of other patterns. Since the joint factor matrix $\mathbf{X}$ is dense, we apply standard NMF as described in Section II-B to obtain the factor matrices $\mathbf{W}^{Global} \in \mathbb{R}_+^{m \times K}$ and the transfer matrix, $\mathbf{H}^{Global} \in \mathbb{R}_+^{K \times (k^1 + k^2 + \cdots + k^g + \cdots + k^N)}$. Here, the objective is to estimate common item patterns across the N groups. We let $\mathbf{M}^g$ represent a slice from $\mathbf{H}^{Global}$ belonging to group g:

$$\mathbf{M}^g = \mathbf{H}^{Global}_{:,k^1+k^2+\cdots+k^{g-1}:k^g+k^{g+1}\cdots+k^N} \tag{6}$$

Note that since the server does not have access to clients' latent feature matrix for user patterns $\mathbf{W}^g$ and user biases b_W^g, the server cannot directly re-construct the private data $\mathbf{X}^g$ of a group g. Hence, providing the server with only $(\mathbf{H}^g)^T$, b_H^g, and μ^g minimizes the shared data and provides a level of privacy to the groups. Finally, the server broadcasts $\mathbf{M}^g$ to each client g along with $\mathbf{W}^{Global}$ and b_H^{Global}.

3) Step 3 - Clients (Information Retrieval Transfer): The next step is to transfer the global information retrieval, obtained via NMF [36], to the local clients, g. This is done by transferring to each client the transformations of the local coordinates, $\mathbf{M}^g$, needed to adjust all local patterns to the common global patterns, $\mathbf{W}^g$. Once the client g receives both the local transfer matrix $\mathbf{M}^g$ and the common global patterns $\mathbf{W}^{Global}$, the step of the transfer of the information retrieval is completed. Here, the goal is to improve the recommendation capability of the local CF model. This is simply done with a single step using $\mathbf{M}^g$ as follows:

$$\mathbf{W}^{*g} = \mathbf{W}^g(\mathbf{M}^g)^T \tag{7}$$

4) Step 4 - Clients (Local Recommendations): At this point each of the N clients has an improved client-specific CF model. Using these local models each client can perform rating estimation for recommendations. A rating of movie j for user i in group/client g is now estimated as follows:

$$\hat{\mathbf{X}}^g_{i,j} = \mathbf{W}^{*g}_{i,:}(\mathbf{W}^{Global})^T_{:,j} + b^g_{W_i} + b^{Global}_{H_j} + \mu^{Global} \tag{8}$$

B. Privacy by Design and Implementation Considerations

1) Control and Flow of Data: One-FedCF utilize NMF and CNMF; therefore, it is designed for NMF-based applications such as recommendation systems. Users have local control of their data $\mathbf{X}^g$ within a group which extends to the context of collaborating organizations [30], [37], [38]. In addition to their local data, groups also have access to the derived data obtained from the processor, which includes $\mathbf{W}^{Global}$, b_H^{Global}, $\mathbf{M}^g$, and μ^{Global}. The central aggregator has access to the latent feature matrices $\mathbf{H}^g$ and bias vectors b_H^g for items, and the average ratings μ^g from each group. This means that the server does not have access to the raw data of the groups.

2) Implications of group based design: Our method is designed to federate data from groups of users or a set of organizations. The group-based design has implications for privacy and communication of the data among the group members. If the group members use a single device (household scenario), that device can be the client participating in federated learning. When the members of the group use multiple devices we need to consider the potential ways of data sharing between the members. One approach is where each user's device becomes a client containing data from the other users that are in the same group. This approach would require the members of a group to first aggregate their data among themselves. Another option is to use hierarchical federated learning [39], where the members of the group send their data to an intermediate server which becomes the client participating in federated learning. While the second option provides better privacy among the group members, the control of the user data is relinquished to the intermediate server. The group approach also extends to the scenario where a set of organizations collaborate without sharing their user's data. In the case of collaborating organizations, each participant becomes a client similar to the household scenario. Finally, the benefit of group-based design is that the user information is further abstracted; the server does not need to know which user belongs to which group or how many users are members of a given group.

IV. Experiments

A. *Dataset and Experimental Setup*

TABLE I: MovieLens 100K and 1M datasets statistics.

Dataset	Users	Items	Num. Ratings	Mean Num. Groups
MovieLens 100K	610	9,724	100,836	36.70 (± 2.61)
MovieLens 1M	6,040	3,706	1,000,209	355.50 (± 11.80)

We evaluate our method on the MovieLens 100K and 1M datasets: benchmarking datasets for evaluating recommender methods [16]. We use the explicit feedback of users for movie ratings, which are between 1 and 5 (Table I). When evaluating One-FedCF, we randomly assign users to be members of groups of randomly selected size between 3 and 30. To show that our results are statistically significant, we use distinct random seeds to group the users 10 times for the MovieLens 100k dataset and 4 times for the MovieLens 1M dataset. For the non-private CF model baseline (baselines will be explained in Section IV-C), which is not based on groups, each experiment with a distinct random seed is used to select a different test set. We report our mean results with a 95% Confidence Interval (CI). Table I also includes the average number of groups we generate per experiment.

We test the performance of our method on a held-out test set of 20% of the entire dataset. For the non-private CF models and each group's local CF models, the hyper-parameter tuning is performed using a validation set sized at 20% of the training-set with 5 fold cross-validation (CV) per tuning trial using a popular package named *Optuna* [40]. We run 100 tuning trials for the MovieLens 100K dataset and 50 tuning trials for the larger MovieLens 1M dataset. The latent factor matrices for the non-private CF models and each group's local CF are initialized randomly and the following hyper-parameters are tuned by running the models for a maximum of 100 iterations (ranges are given in parentheses in log scale): α and β (0.04-0.08), γ and δ (0.01-0.04), η_W and η_H (0.002-0.009). We also tune k (2-$[min(n^g, 21) - 1]$). To tune the NMF performed by the server (Section III-A2), we test for $k \in [2, 4, 6, \cdots, 30]$ with 5 fold CV for each k (again 20% validation-set size) with a maximum 1,000 iterations. The latent factor matrices are initialized with Non-negative Double Singular Value Decomposition (NNDSVD) [41]. After the tuning, we run the ultimate CF models for a maximum of 500 iterations using the found optimal hyper-parameters.

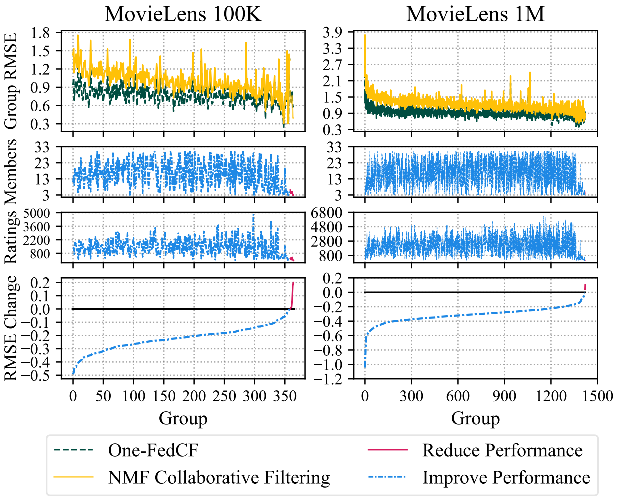

Fig. 1: Comparison of RMSE scores of groups for One-FedCF and standard CF and display of how much One-FedCF improves group's RMSE. We also show the number of ratings and members of each group.

B. *Performance Analysis*

The goal of federated learning is to utilize data from all users to build better performing ML models while also preserving the privacy of users. Therefore, we begin the analysis of One-FedCF by measuring its improvement of recommendation performance for each group's model. To do so, we compare the RMSE of distinct local models obtained with One-FedCF to traditional CF with CNMF for each group across all experiments. In Figure 1, we show the RMSE for each group obtained using One-FedCF and CNMF, the change in group RMSE after One-FedCF, and the number of members and ratings belonging to the group. The displayed results are for the groups from all runs of the experiments.

The first row of Figure 1 shows that One-FedCF improves the RMSE scores for the majority of the groups across both of the datasets. This can also be seen in the last row of the figure which shows the difference in RMSE between standard CF and One-FedCF. A small number of the groups (6) do see a drop in performance in the MovieLens 100K dataset. However, the remaining majority of the 365 groups see an average of 0.22 RMSE reduction. In the MovieLens 1M dataset, all but two groups see an average of 0.31 reduction in RMSE. Also, the number of group members and ratings have almost no correlation to the improvement in performance, which is pointed out by fluctuating numbers in rows 2 and 3 of Figure 1. In summary, One-FedCF can improve the recommendation capability of local models using federated learning.

C. Baseline Comparisons

Due to the novelty of one-shot federated learning recommenders, our baselines are federated recommenders with iterative updates. These federated baseline models are FedRec with SVD++ [8] (factorization method via batching and stochastic updates), Two-order FedMMF [28] (masked matrix factorization via secret sharing), FedGNN [44] (graph neural network), FedMF [26] (matrix factorization with homomorphic encryption, scores obtained from [44]), FCF [2] (gradient sharing, scores obtained from [44]), FCMF [27] (collective matrix factorization with differential privacy and homomorphic encryption), CLFM-VFL [21] (clustering), another homomorphic encryption based method [42], and FedRecon [43] (meta learning). We also provide results for non-private CF using CNMF (single matrix with all users), and the results from each group's local CF model prior to the transfer of information retrieval. The baseline comparison results are provided in Table II. In this table, we report the averaged RMSE scores for the initial local CF models and the One-FedCF improvements across all groups and experiments. Note that for three of our baselines (CLFM-VFL [21], FedRec [8], and Homomorphic Encryption [42]) we have included their score at the lowest reported iteration (communication round). Since these federated methods are not designed to be one-shot, their reported best score requires more rounds of iterations.

Table II show that the group's recommendation performance improves with One-FedCF. For MovieLens 100K, before the local models are updated with One-FedCF, the average RMSE for the groups is 1.00. We can also see that One-FedCF outperforms the baseline federated recommender systems for MovieLens 100K. On the MovieLens 1M dataset, One-FedCF performs slightly worse than the baselines. This can be attributed to noisier groups with higher RMSE. Since the reported RMSE for our method is the average of all groups, the groups with high RMSE bring up the average score. For example, 30.45% of the groups achieve an RMSE equal to or better than 0.84, the best baseline on this dataset. Furthermore, we can attain a mean RMSE of 0.84 if we remove groups with 0.96 RMSE or higher (31.72 % of the groups).

Our RMSE for the MovieLens 1M is close to the baselines, and we achieve this score with a one-shot method while the baselines require multi-round client participation. FedRec (SVD++) needed 100 rounds of communication to achieve RMSE of 0.84 [8]. The performance of their model at 10 rounds of communication is $\sim$0.95 and $\sim$0.90 RMSE for MovieLens 100K and 1M, respectively [8]. The homomorphic encryption-based approach reports RMSE of $\sim$3.40 at the 10th communication [42] and the clustering approach reports $\sim$3.8 RMSE at the 1st communication [21] for MovieLens 100K. Finally, standard CNMF performs better than the federated approaches. However, this method, which relies on gathering raw user data, compromises privacy for this level of utility.

V. Conclusion

One-FedCF is the first one-shot federated learning approach for collaborative filtering. We obtain client-specific recommendations with only a single pair of communications between the server and its clients after a small initial communication. Our approach utilizes joint non-negative matrix factorization and transfer of information retrival. We evaluated our method on two popular recommendation benchmark datasets. Our results show that One-FedCF improves the predictive capabilities of local standard CF models, and obtains similar or better scores with a one-shot approach compared to the state-of-the-art federated recommendation methods that require multi-round client participation. Future work includes using local model selection to filter the groups that can participate in federated learning, which can improve the overall performance [6].

VI. Acknowledgment

This manuscript has been approved for unlimited release and has been assigned LA-UR-22-26537. This research was partially funded by the Los Alamos National Laboratory (LANL) Laboratory Directed Research and Development (LDRD) grant 20190020DR, Analytics, Intelligence and Technology Division, and LANL Institutional Computing Program, supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001.

References

[1] L. M. Krebs, O. L. Alvarado Rodriguez, P. Dewitte, J. Ausloos, D. Geerts, L. Naudts, and K. Verbert, "Tell me what you know: Gdpr implications on designing transparency and accountability for news recommender systems," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[2] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.

[3] A. Flanagan, W. Oyomno, A. Grigorievskiy, K. E. Tan, S. A. Khan, and M. Ammad-Ud-Din, "Federated multi-view matrix factorization for personalized recommendations," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020, pp. 324–347.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[5] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[6] N. Guha, A. S. Talwalkar, and V. Smith, "One-shot federated learning," *ArXiv*, vol. abs/1902.11175, 2019.

[7] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, J. Rush, and S. Prakash, "Federated reconstruction: Partially local federated learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[8] G. Lin, F. Liang, W. Pan, and Z. Ming, "Fedrec: Federated recommendation with explicit feedback," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 21–30, 2020.

[9] Y. Zhou, G. Pu, X. Ma, X. Li, and D. O. Wu, "Distilled one-shot federated learning," *ArXiv*, vol. abs/2009.07999, 2020.

[10] Q. Li, B. He, and D. X. Song, "Practical one-shot federated learning for cross-silo setting," in *IJCAI*, 2021.

[11] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International Conference on Machine Learning*, 2019, pp. 7252–7261.

[12] A. Kasturi, A. R. Ellore, and C. Hota, "Fusion learning: A one shot federated learning," in *International Conference on Computational Science*, 2020, pp. 424–436.

[13] M. Shin, C. Hwang, J. Kim, J. Park, M. Bennis, and S.-L. Kim, "Xor mixup: Privacy-preserving data augmentation for one-shot federated learning," *arXiv preprint arXiv:2006.05148*, 2020.

[14] N. Guha and V. Smith, "Model aggregation via good-enough model spaces," *arXiv preprint arXiv:1805.07782*, 2018.

TABLE II: Comparison of RMSE scores from both datasets for our One-FedCF, baseline federated learning models (multi-round communication, sorted by the rounds of communications), and standard CF. Results of our approach, and best performing methods per section are in bold. Not Available (*NA*) is used if the prior work did not provide the particular information. Symbol $\sim$ is used when the information was not numerically reported, and we report it to the best of our understanding.

			RMSE Results on Datasets	
Method	**Reference**	**# of Comm. Rounds**	**MovieLens 100K**	**MovieLens 1M**
Standard CF				
Non-Private/Standard CF (CNMF)	-	-	**0.71 (± 0.006)**	**0.76 (± 0.006)**
Groups' Local CF (CNMF)	-	-	1.00 (± 0.022)	1.22 (± 0.013)
Iterative Federated Baselines				
CLFM-VFL	[21]	1−175	$\sim$3.80 (*NA*) − $\sim$1.00 (*NA*)	*NA*
FedRec (SVD++)	[8]	10−100	$\sim$0.95 (*NA*) − **0.92 (± 0.005)**	$\sim$0.90 (*NA*) − **0.84 (± 0.001)**
Homomorphic Encryption	[42]	10−100	$\sim$3.40 (*NA*) − 1.03 (*NA*)	*NA*
FCMF	[27]	50	0.95 (± 0.005)	0.88 (± 0.001)
FedRecon	[43]	500	*NA*	0.90 (*NA*)
FedGNN	[44]	*NA* (> 1)	**0.92 (*NA*)**	**0.84 (*NA*)**
Two-order FedMMF	[28]	*NA* (> 1)	**0.92 (± 0.003)**	*NA*
FedMF	[26], [44]	*NA* (> 1)	0.94 (*NA*)	0.87 (*NA*)
FCF	[2], [44]	*NA* (> 1)	0.95 (*NA*)	0.87 (*NA*)
One-shot Federated CF				
One-FedCF	(ours)	**1**	**0.78 (± 0.016)**	**0.91 (± 0.016)**

[15] N. Hug, "Surprise: A python library for recommender systems," *Journal of Open Source Software*, vol. 5, no. 52, p. 2174, 2020. [Online]. Available: https://doi.org/10.21105/joss.02174

[16] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[17] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *6th Indian Control Conference (ICC)*, 2019, pp. 299–300.

[18] W. Luping, W. Wei, and L. Bo, "Cmfl: Mitigating communication overhead for federated learning," in *IEEE 39th international conference on distributed computing systems (ICDCS)*, 2019, pp. 954–964.

[19] R. Basu, "Privacy-preserving recommendation system using federated learning," 2020.

[20] L. Minto, M. Haller, B. Livshits, and H. Haddadi, *Stronger Privacy for Federated Collaborative Filtering With Implicit Feedback*, 2021, pp. 342–350.

[21] J. Zhang and Y. Jiang, "A vertical federation recommendation method based on clustering and latent factor model," in *International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 362–366.

[22] D. Gao, B. Tan, C. Ju, V. W. Zheng, and Q. Yang, "Privacy threats against federated matrix factorization," *arXiv preprint arXiv:2007.01587*, 2020.

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE symposium on security and privacy (SP)*, 2017, pp. 3–18.

[24] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[25] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 11–20, 2020.

[27] E. Yang, Y. Huang, F. Liang, W. Pan, and Z. Ming, "Fcmf: Federated collective matrix factorization for heterogeneous collaborative filtering," *Knowledge-Based Systems*, vol. 220, p. 106946, 2021.

[28] L. Yang, B. Tan, B. Liu, V. W. Zheng, K. Chen, and Q. Yang, "Practical and secure federated recommendation with personalized masks," *arXiv preprint arXiv:2109.02464*, 2021.

[29] Z. Alsulaimawi, "A non-negative matrix factorization framework for privacy-preserving and federated learning," in *IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[30] M. O'connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: A recommender system for groups of users," in *ECSCW 2001*, 2001, pp. 199–218.

[31] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[32] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, 2012.

[33] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM journal on matrix analysis and applications*, vol. 21, no. 1, pp. 185–194, 1999.

[34] M. E. Eren, N. Solovyev, M. Bhattarai, K. Rasmussen, C. Nicholas, and B. Alexandrov, "Senmfk-split: Large corpora topic modeling by semantic non-negative matrix factorization with automatic model selection," in *Proceedings of the ACM Symposium on Document Engineering 2022*, ser. DocEng '22. New York, NY, USA: Association for Computing Machinery, 2022.

[35] V. Stanev, E. Skau, I. Takeuchi, and B. S. Alexandrov, "Topic analysis of superconductivity literature by semantic non-negative matrix factorization," in *International Conference on Large-Scale Scientific Computing*. Springer, 2021, pp. 359–366.

[36] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 267–273.

[37] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *4th ACM conference on Recommender systems*, 2010, pp. 119–126.

[38] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 586–19 597, 2020.

[39] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[40] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[41] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[42] Y. Du, D. Zhou, Y. Xie, J. Shi, and M. Gong, "Federated matrix factorization for privacy-preserving recommender systems," *Applied Soft Computing*, vol. 111, p. 107700, 2021.

[43] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, K. Rush, and S. Prakash, "Federated reconstruction: Partially local federated learning," *ArXiv*, vol. abs/2102.03448, 2021.

[44] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," *arXiv preprint arXiv:2102.04925*, 2021.