Horizontal Cross-Silo Federated Recommender Systems

Saikishore Kalloori
ETH Zürich
Switzerland
ssaikishore@ethz.ch

Severin Klingler
ETH Zürich
Switzerland
severin.klingler@inf.ethz.ch

ABSTRACT

Recommender systems (RSs) completely rely on the knowledge of training information to generate recommendations. However, due to privacy, ownership, and protection of users' information, such training information is not easily accessible or shared with an RS. Moreover, with recent regulations in privacy laws (e.g, GDPR), collecting user preferences and perform centralized training may not be feasible. Federated Learning (FL) is a form of machine learning technique where the goal is to learn a high-quality recommendation model without never directly accessing raw training data. In this work, we specifically focus on situations where multiple stakeholders (referred to as corporate companies like e-commerce business partners, hospitals, banks, news media publishers) participate in federated learning to build a shared recommendation model. We performed offline experiments by simulating a real federated learning setup and investigated the benefits federated learning brings to stakeholders in terms of ranking compared to an RS model trained without participating in federated learning. Our experimental results reveal that stakeholders can significantly benefit from federated learning to generate accurate recommendations. Moreover, we also study the use and benefits of federated learning in situations when there are not enough preferences available for users.

ACM Reference Format:
Saikishore Kalloori and Severin Klingler. 2021. Horizontal Cross-Silo Federated Recommender Systems. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3460231.3478863

1 INTRODUCTION

Recommender systems (RSs) are intelligent systems that help users dealing with large volumes of information by providing personalized suggestions for items that are likely to be interesting and relevant to their needs [10]. Two widely explored RSs are: content-based and collaborative filtering systems. Content-based RSs [7] models user preferences by building a user profile based on the content of the items the user liked, and calculates recommendations for a user by comparing her profile with items profiles (i.e., descriptions of the content of items) and then returning the best matching items (content-based approach). On the other hand, Collaborative Filtering relies on a user-item rating matrix and generates recommendations by leveraging similarities between users or items based on available ratings [6].

It is worth noting that recommenders rely on the knowledge of users' personal preferences given for items to compute recommendations. The requirements to collect a high quantity and quality of data are increasing dramatically to train robust personalization models [3]. But, with the recent outbreak about the misuse of users' sensitive or personal information [2] and due to privacy and protection of users' information, such knowledge about user preferences is not easily accessible. Therefore it may not be feasible to collect user preferences and thus cannot perform centralized training. Recently, a few research works have been focusing on federated learning to avoid centralized training for various machine learning problems. Federated Learning (FL) is a form of machine learning technique that enable a recommender to train a recommendation model on a large corpus of decentralized training data servers by sharing and updating the model parameters. For instance, to build a federated matrix factorization model, the model parameters (item/user latent representations) are shared with stakeholders and updated till convergence, thus avoiding the need to gather training data to one location [5].

Most existing applications of FL in recommender systems focus on improving accuracy, privacy, and training a (shared) federated model from millions of mobile devices [1, 8]. Generally, each user's mobile device data correspond to one client and as a result, in such a scenario, there are millions of clients participate in FL (see figure 1, right). But, for RSs, there are many scenarios, where stakeholders such as hospitals, banks, restaurants, online video streaming, news media publishers who hold data from many users can boost their training information by collaboratively learning a federated model. There is a lot of previous FL research and applications for cross-edge devices scenarios (see figure 1, right) however there is very little previous work done towards building federated recommendation models for stakeholders. When stakeholders participate in FL, there are various key differences (see figure 1, left). Firstly, the amount of data, i.e., the number of users, the number of items, and the number of user-item interactions with each stakeholder largely vary. Moreover, the number of clients participating in FL is very few (e.g., in our setup, there are two to five clients). Secondly, the number of communication rounds and sampling clients for training differ significantly compared to traditional FL with mobile devices.

In this work, we consider such scenarios where the stakeholders like a small number of organizations collaborating to learn a shared recommendation model. Our goal is to understand the effectiveness of federated learning and the benefits for each stakeholder participating in the learning procedure. In our experiments, we simulate the federated learning environment (server-client design setup) and our results reveal that with the help of federated learning, any

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

ACM ISBN 978-1-4503-8458-2/21/09.
https://doi.org/10.1145/3460231.3478863

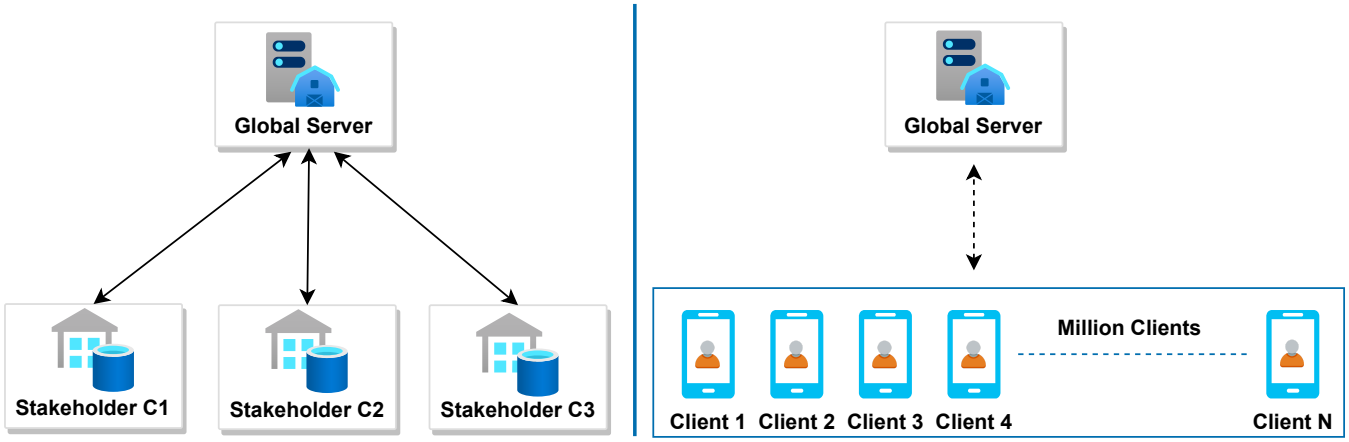

Figure 1: Left is our setup with stakeholders (only 3 clients but each client with large number of users) and right is traditional setup with millions of mobile devices

stakeholder participating in the FL can benefit in better modeling the user preferences to generate accurate recommendations and higher user satisfaction.

The rest of this paper is structured as follows. In the next section we illustrate the federated learning procedure in detail along with the recommendation techniques for computing personalized ranking of items. This is followed by the description of the evaluation strategy used in our experiments and a comprehensive discussion of the obtained results. Finally, we formulate our conclusions and discuss future work.

2 FEDERATED LEARNING AND COLLABORATIVE FILTERING

In our FL process, there is a central server and a small number of corporate companies or business partners referred to as stakeholders or (corporate) clients participate to collaboratively train a shared recommendation model. We will first review two state-of-the-art ranking recommendation algorithms that we use in our federated learning setup.

2.1 The Recommendation Models

Let U be the set of users and I be the set of items. Each user is described by a set of preferences over items in the form of explicit (1-5 star ratings) or implicit feedback (clicks or views). We denote the user u's preference on the item i with r_{ui} and with $\hat{r}_{ui}$ the predicted one. Let P denote the user-latent factor matrix, Q denote the item-latent factor matrix, p_u and q_i denote d-dimensional latent vectors for a user u and item i.

2.1.1 Bayesian Personalized Ranking. Bayesian Personalized Ranking is the state of the art ranking technique applied to implicit feedback data [9]. BPR models user preferences by implicitly assuming that users prefer a item with a click (or a rating) to not clicked ones. For a given a pair of clicked item i and not-clicked item j from each user u, BPR aims to find a personalized ranking $>_u$ such that $\hat{r}_{ui} > \hat{r}_{uj}$. The relative preference of the user u for pair (i, j) is calculated as $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$. To find optimal parameters b_i, p_u and q_i, BPR uses the following objective function:

$$\mathcal{L}_{BPR}(\theta) = \min_{\theta} \sum_{(u,i,j)} -ln(\sigma(\hat{r}_{uij})) + \mathcal{R}(\theta). \tag{1}$$

where $\sigma(\hat{r}_{uij}) = \frac{1}{1+e^{-\hat{r}_{uij}}}$, $\mathcal{R}(\theta)$ is the regularizing term and θ are the model parameters b_i, p_u and q_i to be learned. In our experiments we learn all the model parameters for BPR by using stochastic gradient descent (SGD) algorithm.

2.1.2 Neural Collaborative Filtering. The Neural Collaborative Filtering (NCF) model for implicit feedback was proposed in [4]. NCF replaces the traditional user-item inner product with a neural architecture using a multi-layer perceptron (MLP) to learn the user–item interaction function. Each user and item is represented as sparse data (one-hot identifier) and they are converted into a dense information by means of an embedding layer. NCF uses multi-layer perceptrons to model the two-way interaction between users and items, which is meant to capture the non-linear relationship between users and items. The user embedding and item embedding are then fed to a multi-layer neural architecture to map the latent vectors to prediction scores. The final output layer is the predicted score and a standard log loss is used for the optimization. The scoring function is defined as:

$$\hat{r}_{ui} = f(P^T * s_u^{user}, Q^T * s_i^{item} | P, Q, \theta) \tag{2}$$

where s_u^{user} and s_i^{item} denote one-hot identifier of user u and item i. The function $f(\cdot)$ represents the multilayer perceptron, and θ is the parameters of this network. We note that once the optimal parameter θ are learned, we predict the missing preference for NCF using equation 2 and for BPR as $\hat{r}_{ui} = b_i + p_u * q_i$.

2.2 Horizontal Federated Collaborative Filtering

We have a global server G along with a set of n stakeholders $C = \{C_1, C_2, ..., C_{n-1}, C_n\}$ in our FL. Let D_{C_k} denote training data (either in the form of ratings or clicks) for stakeholder C_k. During the learning process, the global server G acts as a monitor and coordinates with the available stakeholders to learn a single recommendation model. Usually, federated learning can be grouped into three categories [11]: horizontal federated learning, vertical federated learning, and federated transfer learning. In this work, we focus on horizontal federated learning such as scenarios where companies that are conglomerate of multiple brands legally sharing data between business brands is prohibited. In such horizontal FL setup, one assumes that all stakeholders share the same item catalog i.e., every item is present with each participating stakeholder. However, the number of users, the amount of user-item interactions vary from stakeholder to stakeholder. Since each stakeholder shares the same items, the item-latent factor matrix Q is the same for all the stakeholders, and we need to learn optimal values for Q during the federated learning process.

Algorithm 1 describes the pseudo-code of the federated learning and training procedure. During the training, two major steps to be performed. The first one is *global aggregation* which is performed on the global server and the second one is local *stakeholder training* which is performed on the client. The global server kick starts the training and learning process with the available stakeholders. As the first step, the global server initializes (line 2-6) the model parameters of the federated model (e.g., for BPR, the model architecture, the number of latent dimensions, initialization of item-latent factor matrix Q with random weights). Each stakeholder receives $\mathbf{Q}$ from the global server and runs over their local privately-held training data (line 10-16) and calculates the error gradient, and updates the parameters to get new model weights $\mathbf{Q}_{C_k}$. Each stakeholder then returns the updated $\mathbf{Q}_{C_k}$ to the global server. Finally, the global server aggregates[1] by averaging all the updates from the stakeholder [11] (line 7). We note that when we use the NCF algorithm, the model parameters θ are the neural network weights that will be shared between the global server and stakeholders.

It is important to notice that each stakeholder has their own user-latent matrix $\mathbf{P}_{C_k}$ but shares same item-latent matrix $\mathbf{Q}$ across stakeholders. The above learning and training process continues until (pre-defined) max-rounds between server and stakeholders has been reached. During the complete training, one could easily notice that user data is never shared across stakeholders. Such learning enables a recommender to train a robust model from large data source.

3 EXPERIMENTS

This section describes our experimental setup and datasets used to investigate the usefulness of federated learning. We conduct our experiments to address the following research question: can each stakeholder benefit from federated learning to produce accurate recommendations?

[1]We use the popular FedAvg [5] algorithm to implement the aggregator

Algorithm 1 Horizontal Federated Learning

1: **function** GLOBALAGGREGATION(Stakeholder set C)
2: Random initialize Q ▷ done on the global server
3: **repeat**
4: **for** $C_k \in C$ **do** ▷ local client training step
5: θ_{C_k} = ClientTraining(C_k, θ)
6: **end for**
7: $\theta = \theta + \frac{\gamma}{n} \sum_{C_k \in C} (\theta_{C_k} - \theta)$ ▷ global aggregation step
8: **until** *max-rounds between server and stakeholders has been reached;*
9: **end function**

10: **function** CLIENTTRAINING(Stakeholder C_k, parameter θ_{C_k})
11: Random initialize P_{C_k}
12: **repeat**
13: **for** $r_{ui} \in D_{C_k}$ **do**
14: update P_{C_k} and update θ_{C_k}
15: **end for**
16: **until** *convergence or max-iteration has been reached;*
17: **return** θ_{C_k}
18: **end function**

Our experiment setup is as follows: we use a large dataset and simulate an FL setup by distributing the users and their ratings among a set of a pre-defined number of stakeholders and collaboratively train a global model. For our experiments, to set up stakeholders and data for each stakeholder, we vary the number of stakeholders participating in the federated learning and we divide a given dataset into roughly equal-sized subsets. We first fix the number of stakeholders $n = \{3, 5\}$ participating in the FL, and for each n we randomly split the users into n subsets. We repeat the above procedure five times with randomly splitting data, and the experimental results are averages of five runs.

Dataset: We used MovieLens 1M dataset which is time stamped for our experiments. Furthermore, every stakeholder then splits up his data into a test set and a training set. Each stakeholder uses first 80% of the data as training data, and the test data contains remaining 20% of the data. From the training data, we randomly took 20% as validation set to obtained the best model parameters.

Metrics: To evaluate the quality of personalized ranked list of each user, we used three widely-adopted ranking metrics: Mean Reciprocal Rank (MRR), Recall and Mean Average Precision (MAP)

Baseline Algorithm: It is worth noting that each stakeholder already possesses enough training information to build a recommendation model and the major goal for a stakeholder to participate in FL is to have a recommendation model which performs better than a model that is trained using their local private data. If the federated model does not perform better than the local baseline model, then it does not bring any benefit for stakeholders to participate in federated learning. Second, since each stakeholder poses a large number of users, articles, and interactions, the federated recommendation model should be able to model interactions across all the participating stakeholders. We, therefore, have two baseline algorithms. We evaluated a federated model (FM) against the following baselines:

- **Local model (LM)** is the recommendation algorithm built by each client's local private training data alone without

participating in the federated learning setup (the number of local models equals the number of clients). This is a strong baseline to understand the effectiveness of federated learning for stakeholders.

- **Centralised model (CM)** is the recommendation algorithm built by gathering all the training information to one location. This would allow us to understand if the federated learning training process can capture and models user preferences across stakeholders.

4 EVALUATION RESULTS

To address our research question, we conducted two experiments. In our first experiment, we wanted to understand if federated learning helping the participating stakeholder to generate better recommendations when compared to the local model. Table 1 and 2 show the ranking performance of the federated model against the local model and the centralized model for three clients setup. For all the metrics higher values indicate better performance. We observed similar results for five clients setup, hence we do not show these results. As it can be noted from table 1 and 2, each stakeholder participating in the FL can better generate recommendations when compared to individual local models. The ranking performance of the federated model (FM) is significantly better ($p < 0.05$ for MAP@10) than local models (LM). This is because each stakeholder with the help of federated learning exploits additional users' information present with other stakeholders to build a better recommendation model. Additionally, our FM has a similar ranking performance to the centralized model (CM). This shows that our proposed design for federated learning can capture complex highly spare user-item interaction across stakeholders. Our results indicate that federated learning can help each stakeholder to exploit other stakeholder's large user's preferences without never exchanging their user information.

In our second experiment, we wanted to understand how federated learning help stakeholder to generate recommendations for users with a low number of rating information. Since RSs usually have a large portion of users with a low number of rating information, we, therefore, investigated the ranking performance of the federated model against local models for users with a low number of rating information. To identify for each stakeholder, users with a low number of rating information, we took the distribution of rating information (number of ratings per user) and considered the first 1/3 quantile of this distribution as users with low number of rating information.

Table 3 shows the ranking performance of the local model against the federated model for users with low number of ratings under three stakeholders setup and BPR algorithm. The results demonstrate that the federated model is able to better generate recommendations for users with low rating information than local models for all three stakeholders. For instance, for stakeholder two, the ranking performance of the federated model is significantly better than its local model for all the metrics. The federated model has the advantage of better capturing/modeling the users' low number of rating information with the help of other stakeholders' information, whereas the local model is limited to learn from its own local private user information. Overall, our results show that with the help of FL, any stakeholder participating in the learning of the federated model can benefit in better modeling the user preferences to generate better recommendations.

4.0.1 ***Discussion***. We want to emphasize that our focus is to understand the gain or loss in the ranking performance observed by stakeholders by participating in federated learning. Although our results are from one movie dataset, we believe with multiple datasets our results will be similar. This is because we created synthetic stakeholders by distributing (splitting) the data. As our next step, we focus to collaborate with local news media publishers to train a federated model with our proposed design. We note that comparing with any other federated learning baselines is not within the scope of this paper because of the general experiment setup where any advanced algorithms (see section 2.1) such as deep neural nets can easily be adapted to our setup. Finally, we used only two baseline models: local model and centralised model. But as previously mentioned, these two models are important for any stakeholder to understand whether to participate in the FL or not. We also believe that local models play a crucial role and exploring different types of models will be left for our future work.

We believe the research combination of stakeholders and FL can lead to many interesting research directions in RSs. Preference elicitation and active learning are the major components for RSs and with the help of FL, one could understand which type of items preferences one should elicit for a stakeholder based on how the user's preferences are observed with other participating stakeholders. Session-based recommendations will be a novel research direction involving designing a model which can take into account asynchronous users' space, session time, and session lengths across participating stakeholders.

5 CONCLUSION AND FUTURE WORK

In this paper, we exploited the usefulness of federated learning for stakeholders and recommender systems. We presented a federated learning technique based on averaging the model updates to collaboratively learn a federated recommendation model with the help of training data from other stakeholders/clients. We applied FL to two state of the art algorithms: Bayesian Personalized Ranking and Neural Collaborative Filtering. We conducted our experiments using a large movie dataset and various ranking metrics. Our results show that FL can be useful to effectively build RSs for stakeholders and the recommender will have access to a large data source (across stakeholders), thus helping system to better understand and capture user interests to compute accurate recommendations.

In our future, we focus on exploring different methods to aggregate the model updates and apply to deep learning based recommendation algorithms. We further would like to develop a industry level prototype framework for stakeholders and conduct online experiments to understand the efficacy of FL towards user satisfaction and recommendation quality.

REFERENCES

[1] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876* (2018).

[2] Arik Friedman, Bart P Knijnenburg, Kris Vanhecke, Luc Martens, and Shlomo Berkovsky. 2015. Privacy aspects of recommender systems. In *Recommender*

Table 1: Recommendation performance for 3 client setup and BPR algorithm. A single asterisk ($p < 0.05$) and double asterisk ($p < 0.01$) mean that federated model is significantly better than local model.

	Client C1			Client C2			Client C3		
	LM	FM	CM	LM	FM	CM	LM	FM	CM
Recall@5	0.0210	**0.0232***	0.0223	0.0270	0.0275	0.0262	0.0252	0.0295	0.0283
Recall@10	0.0432	0.0444	0.0461	0.0493	0.0515	0.0522	0.0459	0.0480	0.0473
MAP@5	0.1496	**0.1661****	0.1510	0.1121	0.1199	0.1131	0.1131	0.1214	0.1173
MAP@10	0.1233	**0.1367****	0.1253	0.0886	**0.0945***	0.0925	0.0885	**0.0957****	0.0922
MRR@5	0.0937	0.0947	0.0952	0.0530	**0.0580****	0.0551	0.0439	0.0451	0.0466
MRR@10	0.0980	0.0989	0.0999	0.0565	**0.0614****	0.0590	0.0459	0.0475	0.0494

Table 2: Recommendation performance for 3 client setup and NCF algorithm A single asterisk ($p < 0.05$) and double asterisk ($p < 0.01$) mean that federated model is significantly better than local model.

	Client C1			Client C2			Client C3		
	LM	FM	CM	LM	FM	CM	LM	FM	CM
Recall@5	0.0176	**0.0238***	0.0196	0.0276	**0.0319***	**0.0305***	0.0275	0.0270	0.0287
Recall@10	0.0351	**0.0417***	**0.0406***	0.0502	0.0577	0.0583	0.0506	0.0458	0.0522
MAP@5	0.1763	**0.1955***	0.1872	0.1867	**0.2173***	0.1833	0.1481	0.1641	0.1622
MAP@10	0.1511	0.1570	0.1592	0.1549	**0.1784***	0.1566	0.1291	0.1290	0.1305
MRR@5	0.0669	**0.0726***	**0.0728***	0.0458	0.0486	0.0455	0.0301	0.0299	0.0298
MRR@10	0.0693	**0.0751***	**0.0759***	0.0473	0.0497	0.0470	0.0315	0.0311	0.0317

Table 3: Recommendation performance for users with low information under 3 client setup and BPR algorithm. A single asterisk ($p < 0.05$) and double asterisk ($p < 0.01$) mean that federated model is significantly better than local model.

	Client C1		Client C2		Client C3	
	LM	FM	LM	FM	LM	FM
Recall@5	0.0170	**0.0201****	0.0233	**0.0273***	0.0216	**0.0280****
Recall@10	0.0347	0.0365	0.0457	**0.0519****	0.0373	**0.0476****
MAP@5	0.1628	**0.1731***	0.1484	**0.1950****	0.1491	0.1518
MAP@10	0.1391	0.1412	0.1225	**0.1567****	0.1214	0.1185

Systems Handbook. Springer, 649–688.

[3] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*. 169–176.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[5] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[6] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. In *Recommender systems handbook*. Springer, 77–118.

[7] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[8] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. Fedfast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1234–1242.

[9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

[10] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. Springer.

[11] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.