Poster: Towards Privacy-preserving Mobile Applications with Federated Learning – The Case of Matrix Factorization

Koustabh Dolui
imec-DistriNet, KU Leuven
koustabh.dolui@cs.kuleuven.be

Illapha Cuba Gyllensten
Philips Research, Eindhoven
illapha.cuba.gyllensten@philips.com

Dietwig Lowet
Philips Research, Eindhoven
dietwig.lowet@philips.com

Sam Michiels
imec-DistriNet, KU Leuven
sam.michiels@cs.kuleuven.be

Hans Hallez
imec-DistriNet, KU Leuven
hans.hallez@kuleuven.be

Danny Hughes
imec-DistriNet, KU Leuven
danny.hughes@cs.kuleuven.be

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; Supervised learning.

1 INTRODUCTION

Recommender systems have facilitated decision making among users to view content from the vast repository of information that the internet is today. However, these recommender systems leverage a plethora of personal information collected from the users to improve their performance. This leads to a trade-off between the personalization of recommender systems and the protection of privacy of its users.

Recommender systems are built upon various machine learning methods and one such method is matrix factorization. The principle of matrix factorization is to comprehend the underlying characteristics of user ratings for articles, by expressing the user-article rating matrix into two lower ranked matrices called affinity matrices. These affinity matrices implicitly express the characteristics of users and articles respectively in terms of 'k' latent factors[1]. However, application of matrix factorization requires aggregation of both the user and article affinity matrices on a centralized server. This results in the following concerns (i) large datasets entail use of fairly large training models which makes data accumulation and processing on centralized servers a tedious and expensive task, (ii) privacy concerns arise over processing personal information of the user and (iii) the storage of user data and the model leads to an exposure to a central point of attack.

Google proposed the federated learning paradigm to tackle problems along these lines and to enable privacy-preserving learning relying on (i) training models on-device using locally stored data, (ii) transmission of ephemeral updates to the central server and (iii) aggregation of these updates to train a central model[2]. Federated learning has been deemed suitable for applications where data is distributed over number of devices greater than the average number of data points per device. It has been applied by Google for text prediction in the GBoard. We extend the application of the federated learning paradigm to recommender systems and in particular matrix factorization to answer the following research questions (i) to what extent can the user data sharing be minimized while maintaining high levels of accuracy and (ii) how does optimizing on-device training affect the performance of the system. To this end, we propose a federated matrix factorization algorithm where (i) the user-article matrix for each user is stored on device and only the updates for affinity matrices are transmitted to a centralized server and (ii) varying levels of privacy for transmission and aggregation of the updates are supported.

2 APPROACH

The matrix factorization method is initiated with a user-rating matrix R_{mxn}, where m = number of users and n = number of articles. Here, each element r_{ij} represents the rating or relevance of user i to article j. This matrix is factorized into (i) user affinity matrix U_{mxk} and (ii) article affinity matrix A_{nxk}, which expresses the affinity of the user and article towards k common latent factors. For example, a particular movie can be considered a composition of k genres while a user can have different preferences for watching these k genres of movies. The goal is to optimize the matrices U and A in such a way that the missing values of r_{ij} can be predicted correctly. This is performed by the minimizing the error between the predicted value and the user rating using the following equation:

$$f(u,a)^{(t)} = \sum_{(r_{ij} \in R)} (r_{ij} - u_i.a_j)^2 + \lambda(||U||^2 + ||A||^2) \tag{1}$$

where, λ is a regularization parameter to prevent overfitting of training data.

In the centralized approach, the matrices R, U and A are stored on the server. The method of Alternating Least Squares (ALS) is used to find the optimal solution, with equations 2 and 3 computed alternately in each iteration over the data. However, in this approach both the user affinity matrix U and the rating matrix R, which qualify as personal information, are stored and processed in the centralized server leading to privacy concerns.

$$U_i = (A^T A + \lambda I)^{-1} A^T R_i \tag{2}$$

$$A_j = (U^T U + \lambda I)^{-1} U^T R_j \tag{3}$$

In our proposed federated approach, the ratings for each user are stored locally on the user device as a vector r_i. The affinity vector of the user U_i and article affinity matrix A are stored as well. The entire matrix A is required in contrast to only the vector U, since affinity

MobiSys '19, June 17–21, 2019, Seoul, Republic of Korea

ACM ISBN 978-1-4503-6661-8/19/06.
https://doi.org/10.1145/3307334.3328657

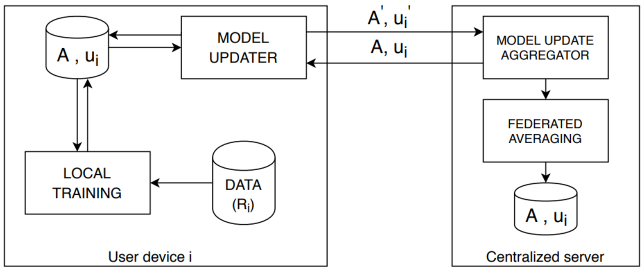

Figure 1: Data flow in the allocation view of the architecture for federated matrix factorization algorithm

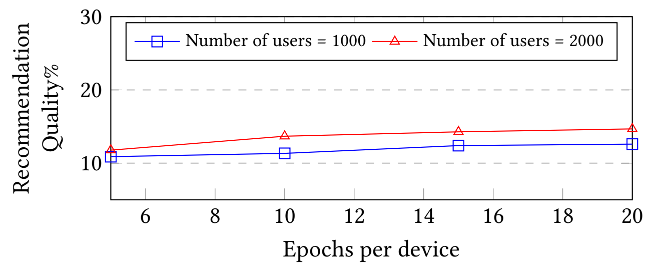

Figure 2: Effect of per device iteration on performance

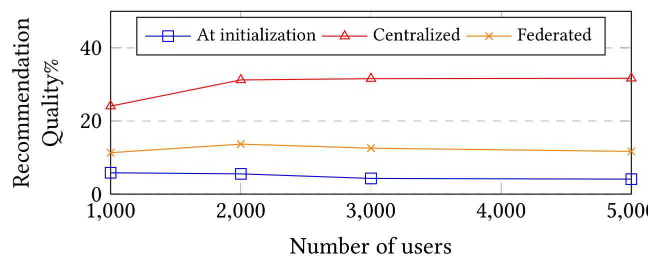

Figure 3: Comparison of federated and centralized approaches for varying number of users

of other users are independent of user i. We learn the parameters U and A locally on the device from each user using Gradient Descent as illustrated in Algorithm 1.

Phase I: Choose d devices to transmit updated models A and u_i
Phase II: On each device,
while *iterations<epoch* **do**
  $u_i \leftarrow u_i + \alpha.a_j(r_{ij} - u_i.a_j) - \lambda.a_j$
  for $(j = 0; j < n)$
  $a_j \leftarrow a_j + \alpha.u_i(r_{ij} - u_i.a_j) - \lambda.u_i$
end
Send A_i to the centralized server
Phase III: Update model on the central server A = $(\sum A_i)/d$

Algorithm 1: On device learning of u_i and A

We propose two approaches of performing the update aggregation from user devices. In the most privacy preserving approach, the user affinity matrix is not sent to the centralized server. This does not affect the performance of the above algorithm. However, in certain cases, the recommendation provider requires the affinity of users to create or modulate content. In such scenarios, we transmit the user affinity matrices to the centralized server by applying privacy measures like data anonymization and differential privacy [3] as illustrated in Figure 1.

3 PRELIMINARY RESULTS

We have implemented our federated matrix factorization algorithm on an in-house dataset collected from a mobile application in Philips Research, Eindhoven. The application offers a platform for users to read articles, log their personal information while offering health tips and article recommendations to the user.

The dataset comprises of implicit rating in terms of the reading times for articles read by the user and user opinion, 'like' and 'dislike'. Since the user article rating matrix is not available readily, we normalize the reading times for each user and compute a weighted user-article relevance value as follows:

$$r_{ij} = w.t\_norm(i,j) + (1-w).u\_opinion(i,j) \tag{4}$$

For this pilot study, we have chosen a part of the dataset comprising of 914,768 data points from 5260 users and 570 articles. For our experiment we have used the following hyper-parameters, $\alpha = 0.05$, $\lambda = 0.1$, $k = 7$, $w = 0.5$, $u\_opinion \in \{-10, 10\}$, epochs = 10, learned over cross-validation data for the centralized approach. The performance is measured by the quality of recommendation as follows, the percentage of users for whom the article predicted by the algorithm with the highest score, is in the top 5 articles for the same user in the test dataset. This metric is chosen since we want to recommend the article with the highest score to each user.

In Figure 2, we compare the performance between the two approaches with varying number of users participating in the training in each step. The quality of recommendation improved upon after initialization both the centralized and federated approaches. However, the performance for the federated approach does not improve with the number of users. In addressing research question (i), minimizing the sharing of data leads to significantly lower performance with the federated approach. In Figure 3, we measure the performance for varying number of iterations on the entire dataset. We observe that performance slowly improves with the number of iterations. Hence, in addressing research question (ii), optimizing the resource usage by reducing the amount of processing performed on the devices, does not significantly degrade performance.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 766139.

REFERENCES

[1] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
[2] H. B. McMahan and D. Ramage. 2017. Federated Learning: Collaborative Machine Learning without Centralized Training Data. Retrieved April 12, 2019 from https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
[3] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963* (2017).