DS603: Privacy-Preserving Recommendation Model

Aman Vishnoi

***Abstract*—Abstract—The main aim of this project is to create a privacy preserving federated news recommendation system. News recommendation aims to display news articles to users based on their personal interest. Existing news recommendation methods rely on centralized storage of user behavior data for model training, which may lead to privacy concerns and risks due to the privacy-sensitive nature of user behaviors. We will try to develop a privacy- preserving method for news recommendation model training based on federated learning, where the user behavior data is locally stored on user devices. The dataset that we will be using in the project is Mind Dataset. The MIND dataset for news recommendation was collected from anonymized behavior logs of Microsoft News website by randomly sampling 1 million users who had at least 5 news clicks during 6 weeks from October 12 to November 22, 2019. We will be trying out several new ideas in the paper as incorporating differential privacy, monitoring the model performance against federated attacks(model poisoning, Substitution-Based Profile Pollution Attacks), testing the model against several new benchmarks other than news as songs dataset, implementing of fairness-aware Federated Matrix Factorization**

I. Introduction

Many news recommendation methods still rely on centralized storage of user behavior data for model training, which raises some concerns regarding data privacy of user[1]. In this paper the user's data will be stored on the user's device.This technique can leverage the useful information in the behaviors of massive number users to train accurate news recommendation models and meanwhile remove the need of centralized storage of them. We keep a copy of small user model on each edge device and gradients are pushed to server where a large news model is being trained. Since we will be taking gradients from local devices we will be using techniques as Multiparty computation, Local Differential Privacy for privacy protection. The updated global model is then distributed to each user device for local model update. We repeat this process for multiple rounds. We will be monitoring the performance of our model against various attacks as substitution-based profile pollution attacks[2], and model poisoning[3]. As Fairness and robustness are two important concerns for federated learning systems[4] we have also incorporated several implementations that will make the model fair.

II. RelatedWork

Work have already been done in federated recommendation system by Tao Qi, Fangzhao Wu,Chuhan Wu, Yongfeng Huang, Xing Xie in his paper Privacy-Preserving News Recommendation Model Learning[1], and they are using multiparty computation, instead of diffrential privacy or homomorphic encryption. The model has not been tested to attacks and is not tested for fairness. Another approach has been taken up Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie in their paper Privacy-preserving news recommendation model learning[1]. However, the communication and computation cost of FedRec is unacceptable for user devices with limited resource due to the large size of news recommendation models, especially their news models.

III. SystemModel

The models have been trained on MIND and address data on P5000 GPU, 30 GB ram and 16 core CPU for a period of 8 hours and the logs have been published over wandb. Addressa dataset have to preprocessed into MIND dataset format to be used in the model.

IV. Algorithms

The first step is to properly format the news in dataset. We have dataset that contains user's history, the news he have clicked and the news he ignored. We will be denoting news clicked as a positive sample and rest all news as a negative sample. We will be creating a dataloder that will return an array containing a positive sample, all negative samples, history for a particular user. For the purpose of training we will be randomly sampling 50 users and aggregate their history, their positive and negative samples via Multiparty computation. We will be using BERT model pretrained model to generate news vector embeddings of 400 dimensions for the entire MIND/addressa dataset. We will be unfreezing some of pretrained layers in middle of BERT for fine tuning over news dataset. We have a user-encoder model in the server and one at each client devices. The user-model has multihead attention layer connected to two linear layers which when fed the encoding of user's history generates a user-embedding of 400. We then create a batch of news history from the randomly sampled users and generate user encoding for each user by the user's model stored at server. We then create encoding of news that we are trying to do the prediction over, i.e. positive samples and negative samples. We will then do dot product for these samples and user-vector to get the scores and then we will apply softmax, since we know the positive sample and negative sample we will do will do cross entropy loss for each of these users to do back-propagation to update BERT model and user models at the server. The weights of this user model at the server is sent back to each of the clients. Then again again randomly selects 50 user and again do backpropagation for the user model at the server for the news of these 50 people and send the gradients back to the user model at the client.

V. Experiments and results

The prime dataset that I will be using are Mind Dataset for news articles, and addressa dataset. The result for MIND

Fig. 1: Addressa dataset

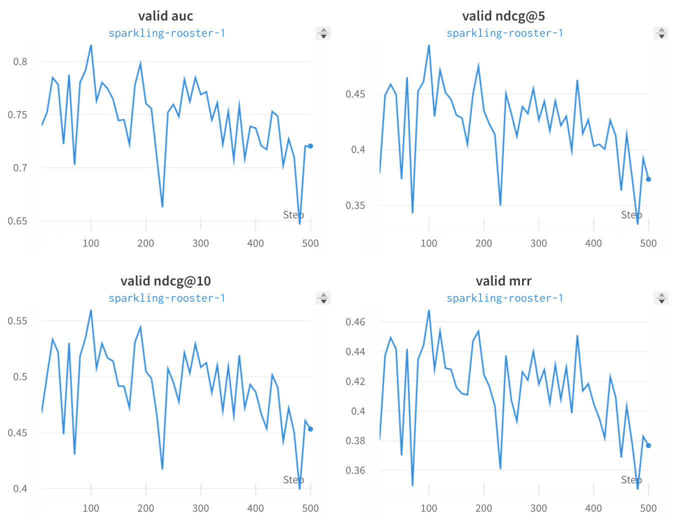
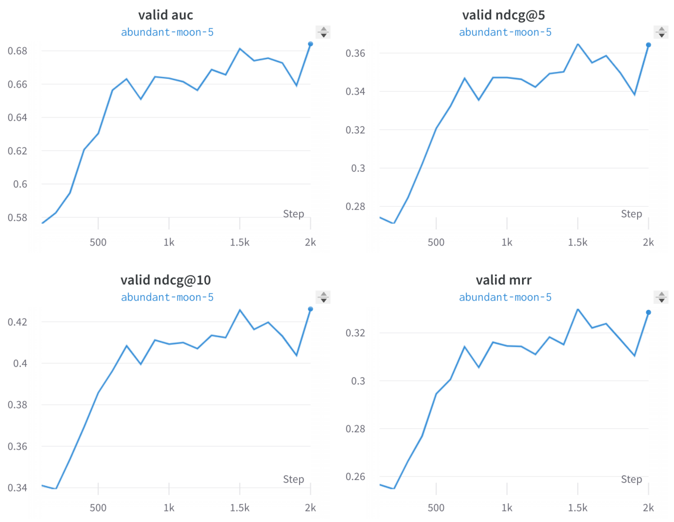


Fig. 2: Mind dataset



Dataset have been replicated, however the reported MRR and NDCG for addressa in the paper is not correct according to the experiments conducted by me.

1) **Mind Dataset**
2) **Addressa Dataset**

Experiment Results				
Dataset	MRR	AUC	NDCG@5	NDCG@10
MIND	32.86	68.42	36.43	42.62
Addressa	37.67	72.04	37.33	45.31

VI. Issues in the Paper

1) How user model at the client-end is being used. They have just reinitialized it at each training step. I am assuming that they are just distributing user model at the server to client but it is not clearly mentioned what they are doing
2) There is no clear way of generating news vector embeddings at the user's end while making a prediction. User model at the client requires news vectors but they have very vaugely created news vector at the server which is a breach of privacy.
3) Although they have mentioned that have reduced communication overhead between client and server, but there is no derivation of that in the entire paper of how they calculated it.
4) They have scaled training loss in code by a factor which is increasing, which is not very clear as to why they do it. This i think in incorrect.
5) I have retrained code over and over for addressa dataset and unable to replicate the result.
6) The way they are doing Multiparty computation in code is seriously violating privacy of user.

VII. Conclusion

[1] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning

[2] Defending Substitution-Based Profile Pollution Attacks on Sequential Recommenders by Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang and Dong Wang

[3] FedRecAttack: Model Poisoning Attack to Federated Recommendation by Dazhong Rong, Shuai Ye, Ruoyan Zhao, HonNing Yuen, Jianhai Chen and Qinming He

[4] Ditto: Fair and Robust Federated Learning Through Personalization by Tian Li, Shengyuan Hu, Ahmad Beirami 2, Virginia Smith 1