John (Ziyuan) Zhou

Week 7 Summary

UW ID: 2150115

In the article “FedGNN: Federated Graph Neural Network for Privacy-Preserving Recommendation”, the authors point out the privacy issue of GNN-based recommendation system. The recent approaches on GNN-based recommendation method rely on centralized storage of user-item graphs and centralized model learning. However, user-item interaction data is highly privacy-sensitive, and its centralized storage can lead to the privacy concerns of users and the risk of data leakage. In addition, under the pressure of strict data protection regulations, online platforms may not be able to centrally store user-item interaction data to learn GNN models for recommendation in the future. There is an intuitive way to tackle the privacy issue of user-item interaction data is to locally store the raw data on user devices and learn local GNN models based on it. However, it faces the other challenges to train a GNN model with good performance. First, for most users, the volume of interaction data on their devices is too small which is not sufficient to train an accurate model locally. Second, the local GNN model trained on local user data may convey private information, and it is challenging to protect user privacy when synthesizing the global GNN model from the local ones. Last, local user data only contains first-order user-item interactions, and users’ interaction items cannot be directly exchanged due to privacy restrictions.

To solve these problems, the authors propose a federated framework named FedGNN for privacy-preserving GNN-based recommendation, which can effectively exploit high-order user-item interaction information by collectively training GNN models for recommendation in a privacy preserving way. Specifically, they locally train GNN model in each user client based on the user-item graph inferred from the local user-item interaction data. Each client uploads the local gradients of GNN to a server for aggregation, which are further sent to user clients for updating local GNN models. During this process, an embedding layer is first used to convert the user node, item nodes, and the N neighboring user nodes into their embeddings. Next, a graph neural network is applied to these embeddings to model the interactions between nodes on the local first-order subgraph. For this step, various kinds of GNN network can be used. Then, a rating predictor module is used to predict the ratings given by the user to his interacted items based on the embeddings of items and this user. The loss function for the user is . This loss function is used to derive the gradients of the models and embeddings, and these gradients will be further uploaded to the server for aggregation. The server aims to coordinate all user devices and compute the global gradients to update the model and embedding parameters in these devices. In each round, the server awakes a certain number of user clients to compute gradients locally and send them to the server. After the server receiving the gradients from these users, the aggregator in this server will aggregate these local gradients into a unified one. Then, the server sends the aggregated gradients to each client to conduct local parameter update. This process will be iteratively executed until the model converges. For the privacy-preserving model update, the authors propose two strategies to protect user privacy in the model update process which are pseudo interacted item sampling and local differential privacy. To find the neighbors of users and extend the local user-item graphs in a privacy-preserving way, they also propose a privacy-preserving user-item graph expansion method that finds the anonymous neighbors of users to enhance user and item representation learning, where user privacy does not leak. In the last section, extensive experiments on six benchmark datasets validate that FedGNN can achieve competitive results with existing centralized GNN-based recommendation methods and meanwhile effectively protect user privacy.