Personalized Federated Graph Learning

Karthik Pansetty[1], Yuhang Yao[1], Mohammad Mahdi Kamani[2], Carlee Joe-Wong[1]
[1]Carnegie Mellon University, [2]Wyze Labs

Background

Federated graph learning has recently raised great interest which trains graph neural networks at local clients to reduce privacy risks and computation cost, with applications ranging from fake news detection in social networks to anomaly detection in sensor networks.

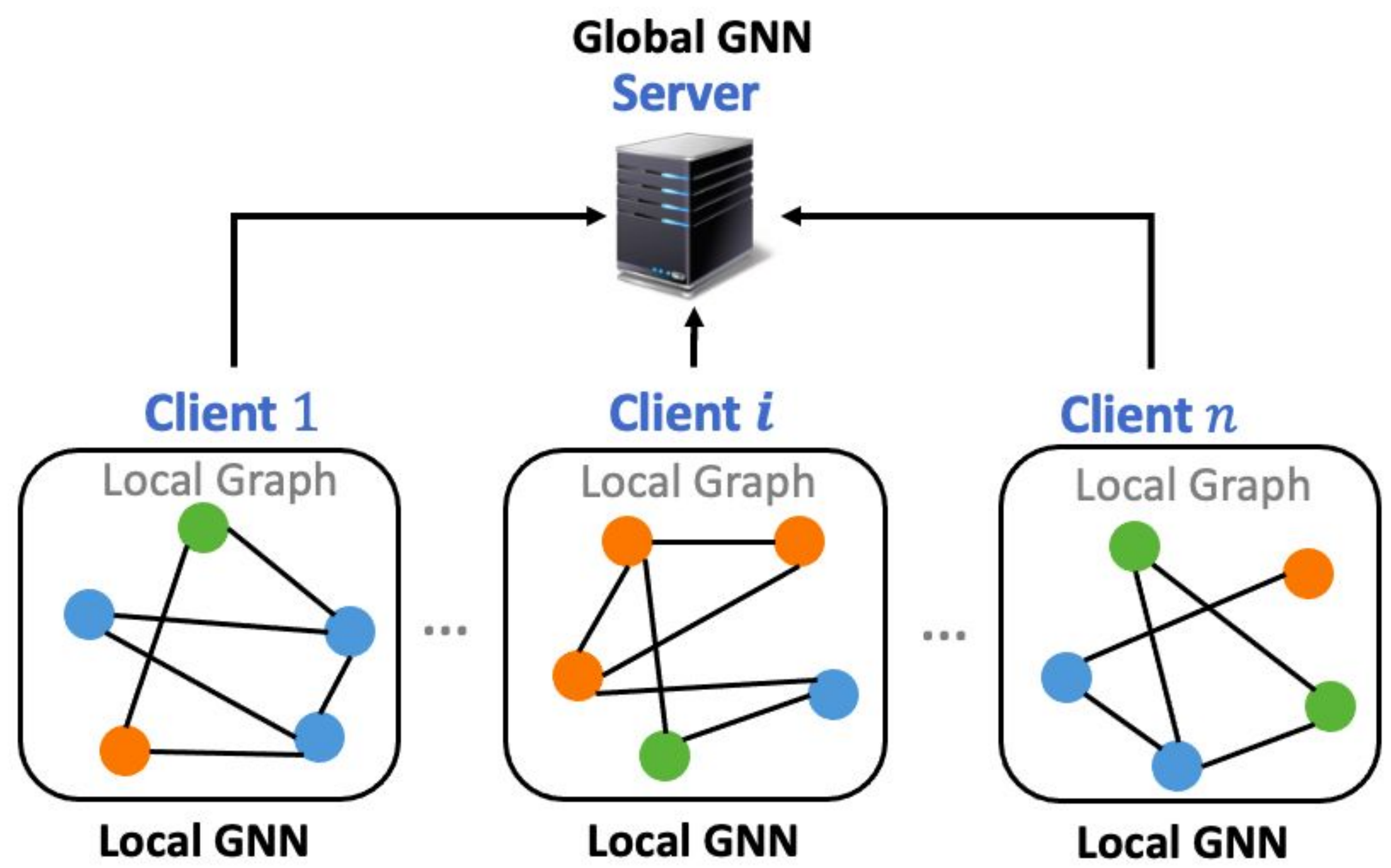

Figure 1: Example of the Federated learning flow for the graph classification and link prediction

However, one major challenge of federated training on graphs is that many clients have little local data, which makes statistical heterogeneity - clients' data is not identically and independently distributed (IID) a challenge.

- We propose APFLGate, a personalized federated training method with variance reduction to avoid overfitting and have better generalization performance.

Personalized Federated Learning with Variance Reduction

Personalization

At global round t and local step r with R steps per round, the output of the personalized model for the i-th client is

$$\mathbf{h}_i^{t,r} = \alpha_i \mathbf{h}_{loc,i}^{t,r} + (1-\alpha_i)\mathbf{h}_{glob}^{t}$$

h_i^t, $h_{loc,i}^t$, h_{glob}^t are the output of personalized, local, global model at round t, respectively.

α_i is associated with the diversity of the local model and the global model. Higher α_i means more personalization on the i-th client.

Lack of data leads to poor generalization of personalized models due to overfitting to local training data.

Variance Reduction

Reducing the variance in the gradient updates at different clients, can force the local gradients to be more close to the global gradient to prevents overfitting and speed up training.

The variance reduction updates of client i are given by

$$\text{Device update: } \mathbf{d}_i^r = \mathbf{g}_i^r - \boldsymbol{\delta}_i^t, \quad \mathbf{g}_i^r \triangleq \nabla f_i(\mathbf{w}_i^r)$$

$$\mathbf{w}_i^{r+1} = \mathbf{w}_i^r - \eta \mathbf{d}_i^r$$

After R local steps

$$\mathbf{u}_i^t = \mathbf{w}^t - \mathbf{w}_i^{t,R} \text{ (to server, get } \mathbf{u}^t)$$

$$\bar{\mathbf{w}}^t = \mathbf{w}^t - \mathbf{u}^t$$

$$\boldsymbol{\delta}_i^{t+1} = \boldsymbol{\delta}_i^t + \frac{1}{\eta\tau}(\bar{\mathbf{w}}^t - \mathbf{w}_i^{t,R})$$

$$\text{Server update: } \mathbf{u}^t = \frac{1}{m}\sum_{i=1}^{m} \mathbf{u}_j^t$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \gamma \mathbf{u}^t,$$

where d_i^r is the local gradient of client i at step r, w^t is the weights of the global model and w_i^t is the model of client i at step t, δ is the gradient tracking parameter, τ is the variance reduction parameter, η and γ are learning rates.

The local gradient tracking term $\boldsymbol{\delta}_i$ ensures that each client i uses an estimate of the global gradient direction to locally update its model, which reduces the variance among clients. The variance reduction parameter τ controls the amount of gradient tracking. Lower τ means more gradient tracking, which leads to lower variance among clients.

Results and Analysis

Datasets

We use the molecule network dataset *BBBP* for graph classification and the recommender system dataset *ciao* for link prediction.

Training Speed and AUC-ROC

As shown in Figure 2, APFL overfits very easily and adding a variance reduction technique helps prevent overfitting and improves performance.

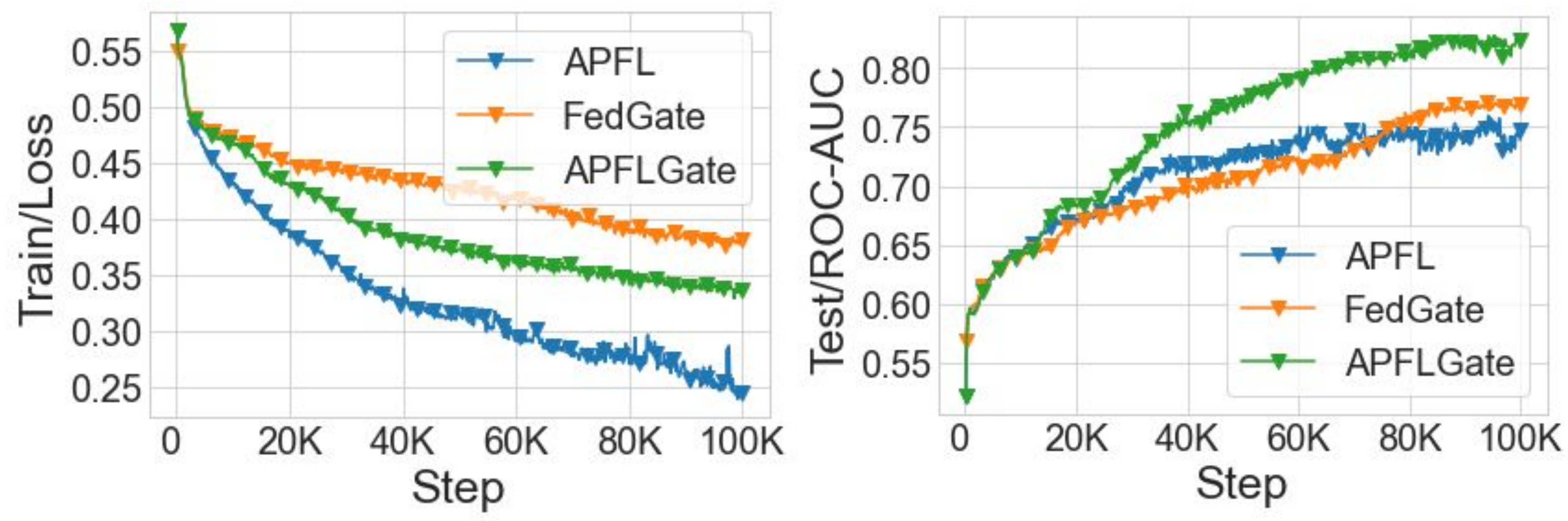

Figure 2: Train Loss (left) and Test ROC-AUC (right) on the Graph Classification Task for 16 Clients

Tradeoff between Personalization and Variance Reduction

- The variance reduction parameter τ controls the convergence of the algorithm and the personalization parameter α controls the performance of the model.
- The tradeoff between personalization and variance reduction has a large impact on when the model converges and its performance, as shown in Table 1 and Table 2.
- Experiments with α=1 converge faster but the experiments with τ = 10 have higher performance.

Approach	ROC-AUC
APFL ($\alpha = 0.1$)	0.7492
FedGate ($\tau = 100$)	0.7703
APFLGate ($\alpha = 0.1, \tau = 0.1$)	0.8342
APFLGate ($\alpha = 0.1, \tau = 100$)	0.8347
APFLGate ($\alpha = 0.1, \tau = 1$)	0.8349
APFLGate ($\alpha = 0.9, \tau = 10$)	0.8429
APFLGate ($\alpha = 0.75, \tau = 10$)	0.844
APFLGate ($\alpha = 0.5, \tau = 10$)	0.8458
APFLGate ($\alpha = 0.25, \tau = 10$)	0.8472
APFLGate ($\alpha = 0.1, \tau = 10$)	**0.848**

Table 1: Test ROC-AUC of Methods on the Graph Classification Task for 16 Clients

Approach	MAE
APFL ($\alpha = 0.25$)	0.8054
FedGate ($\tau = 10$)	0.8055
APFLGate ($\alpha = 0.1, \tau = 10$)	0.7921
APFLGate ($\alpha = 0.25, \tau = 10$)	0.7895
APFLGate ($\alpha = 0.5, 0.75, 0.9, \tau = 10$)	**0.7891**

Table 2: Test MAE of Methods on the Link Prediction task for 28 Clients

By balancing the personalization and variance reduction, APFLGate performs better than both APFL and FedGate on graph classification and link prediction.

References:
[1] Li, T. (2019, November 12). Federated Learning: Challenges, Methods, and Future Directions. https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions.
[2] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2):1–210, 2021.
[3] Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461, 2020.
[4] Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In International Conference on Artificial Intelligence and Statistics, pp. 2350–2358. PMLR, 2021.