POSTER: PERSONALIZED FEDERATED GRAPH LEARNING

Karthik Pansetty [1] **Yuhang Yao** [1] **Mohammad Mahdi Kamani** [2] **Carlee Joe-Wong** [1]

ABSTRACT

We explore the personalization of graph neural networks for graph classification and link prediction in a federated learning setting with massive clients and little local data. Our experiments indicate that using both personalization and variance reduction can significantly improve the performance, while using only personalization can easily lead to overfitting especially with limited local data. We then propose a new federated learning algorithm that leverages both personalization and variance reduction techniques. We further show that there is a trade-off between personalization and variance reduction (as reducing the variance of models on the client side can hamper personalization) and conduct experiments that demonstrate how these factors affect the training and performance of different graph learning tasks. Our algorithm can be easily extended to the general FL with similar settings.

1 INTRODUCTION

Graph neural networks has been widely used for applications from behavior classification in social networks to anomaly detection in the Internet of Things. The data, however, is highly sensitive with private information (e.g. credit record and in-house data). Federated learning is proposed as a distributed learning approach that reduces privacy risks and communication costs for training machine learning models on data located at multiple clients, which has been widely adopted for privacy-preserved training of non-graph data.

However, one major challenge of federated training on graphs is that many clients have little local data (for example, in IoT settings, millions of clients only have access to a small number of device graphs for each client), which makes statistical heterogeneity - clients' data is not identically and independently distributed (IID) (Kairouz et al., 2021) a challenge. A strand of solutions to deal with the heterogeneity is by training a shared global model *and* a different local model at every client (e.g. Adaptive Personalized Federated Learning (APFL) (Deng et al., 2020)).

Learning personalized models, however, leads to overfitting to the limited amount of data at each client. We attempt to solve this challenge by personalization with *variance reduction* techniques, in which clients' gradients are modified to be more aligned with each other to guarantee the generalization ability of the personalized models, which is previously used to improve convergence speed (e.g. FedGate (Haddadpour et al., 2021)) for non-personalized FL training.

[1]Carnegie Mellon University [2]Wyze Labs. Correspondence to: Yuhang Yao <yuhangya@andrew.cmu.edu>.

Proceedings of the 5^{th} MLSys Conference, Santa Clara, CA, USA, 2022.

We then propose a new federated learning algorithm, called APFLGate, that leverages both personalization and variance reduction techniques. Extensive experiments on graph classification and link prediction show that there is a trade-off between personalization and variance reduction to have the best performance. We see an improvement of 10.08% AUC when compared to state-of-the-art algorithms. Although we mainly focus on federated graph training, the algorithm can be easily extended to other neural network structures.

2 PERSONALIZED FEDERATED LEARNING WITH VARIANCE REDUCTION

Lack of data at federated learning clients can lead to poor generalization of personalized models due to overfitting to local training data. Reducing the variance in the gradient updates at different clients, can be done to accelerate convergence of a global model by forcing the local gradients to be more similar to the global gradient. Thus, it prevents overfitting by forcing the local gradients away from those calculated on local data. Our approach, APFLGate, is to use APFL for personalization and FedGate's local gradient tracking scheme for variance reduction. 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

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

where $\mathbf{h}_i$ is the output of personalized model, α_i is the personalization parameter and $\mathbf{h}_{loc,i}$ is the output of the local model at the i-th client and $\mathbf{h}_{glob}^t$ is the output of global model at round t. 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.

The variance reduction updates of client i are given by

$$\text{Device update: } \mathbf{d}_i^r = \mathbf{g}_i^r - \boldsymbol{\delta}_i^t, \quad \mathbf{g}_i^r \triangleq \nabla f_i(\mathbf{w}_i^r) \tag{2}$$

$$\mathbf{w}_i^{r+1} = \mathbf{w}_i^r - \eta \mathbf{d}_i^r \tag{3}$$

$$\text{After } R \text{ local steps} \tag{4}$$

$$\mathbf{u}_i^t = \mathbf{w}^t - \mathbf{w}_i^{t,R} \text{ (to server, get } \mathbf{u}^t) \tag{5}$$

$$\bar{\mathbf{w}}^t = \mathbf{w}^t - \mathbf{u}^t \tag{6}$$

$$\boldsymbol{\delta}_i^{t+1} = \boldsymbol{\delta}_i^t + \frac{1}{\eta\tau}(\bar{\mathbf{w}}^t - \mathbf{w}_i^{t,R}) \tag{7}$$

$$\text{Server update: } \mathbf{u}^t = \frac{1}{m}\sum_{i=1}^{m} \mathbf{u}_j^t \tag{8}$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \gamma \mathbf{u}^t, \tag{9}$$

where $\mathbf{d}_i^r$ is the local gradient of client i at step r, $\mathbf{w}^t$ is the weights of the global model and $\mathbf{w}_i^t$ is the model of client i at step t, $\boldsymbol{\delta}_i$ is the gradient tracking term, 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.

3 Results and Analysis

We use the molecule network dataset *BBBP* for graph classification and a recommender system dataset *ciao* for link prediction. We performance both tasks by using Graph Convolutional Network (GCN) models.

With massive clients and little local data, APFLGate has higher test accuracy than both APFL and FedGate. Due to the small amount of data at each client, personalization helps improving the training performance and variance reduction reduces overfitting for generalization. We do not see similar results for few clients with enough data at each client, because the model does not overfit easily for APFL.

To understand the trade-off between the personalization and variance reduction methods, we perform a series of experiments with the personalization parameter α which controls the amount of interpolation between the global and the local models and the variance reduction parameter τ which controls the amount of gradient tracking.

Our experiments indicate that the variance reduction parameter τ controls the convergence of the algorithm and the personalization parameter α controls the performance of the model. So, the trade-off between personalization and variance reduction has a large impact on when the model converges and its performance and this can be seen from Table 1 and Table 2. We see that experiments with $\alpha = 1$ converge faster but the experiments with $\tau = 10$ have higher performance. In conclusion, by balancing the personalization and variance reduction, APFLGate performs better than both APFL and FedGate as seen in Table 2 and Table 3 on graph classification and link prediction respectively.

Approach	Steps
APFLGate ($\alpha = 0.1, \tau = 0.1, 1, 100$)	**53.1K**
APFLGate ($\alpha = 0.1, \tau = 10$)	63.1K
APFLGate ($\alpha = 0.25, \tau = 10$)	63.9K
APFLGate ($\alpha = 0.5, \tau = 10$)	66.1K
APFLGate ($\alpha = 0.75, \tau = 10$)	67.2K
APFLGate ($\alpha = 0.9, \tau = 10$)	68.6K

Table 1. Steps to reach a Test ROC-AUC of 0.8 for Graph Classification task for 16 clients (Lower steps means faster convergence).

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 2. Test performance for different approaches on the Graph Classification task for 16 clients (Higher is better for ROC-AUC).

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 3. Test performance for different approaches on the Link Prediction task for 28 clients (Lower is better for MAE).

References

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In *Proc. of AISTATS*, pp. 2350–2358. PMLR, 2021.

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.