SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks

Zepeng Zhang

This paper [1] proposed SLAPS (simultaneous learning of adjacency and GNN parameters with self-supervision), which identifies the supervision starvation problem that emerges for graph structure learning. The supervision starvation problem occurs when the edges between pairs of nodes are far from labeled nodes, i.e., the edges receive insufficient supervision. This results in learning poor structures away from labeled nodes and hence poor generalization. This paper proposed to supplement the classification task with a self-supervised task, which is based on the hypothesis that a graph structure that is suitable for predicting the node features is also suitable for predicting the node labels. It works by masking some input features (or adding noise to them) and training a separate GNN aiming at updating the adjacency matrix in such a way that it can recover the masked (or noisy) features. The architecture of SLAPS is as below.

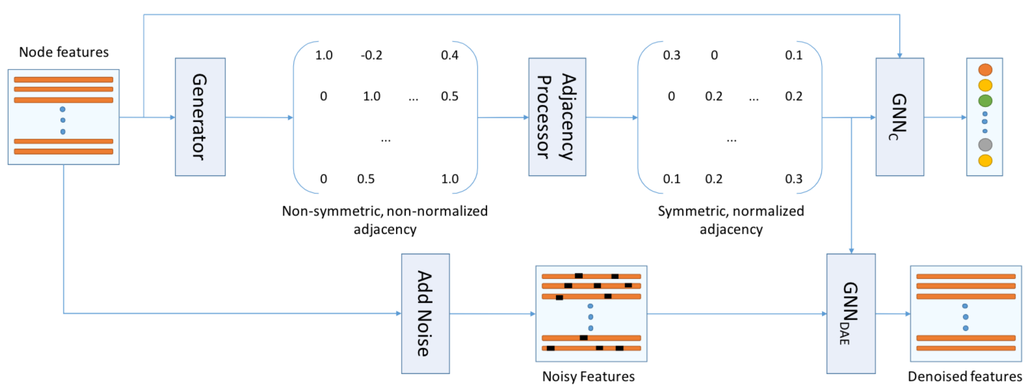

Figure 1: Overview of SLAPS.

Generator. The generator takes the node features as input and produces a adjacency matrix $\tilde{\mathbf{A}}$ as output. Two generators are considered in this paper: 1) Full parameterization: ignoring the node features and fully parameterize the adjacency matrix; 2) MLP-kNN: the node features passing through an MLP followed by a kNN which produces a sparse matrix.

Adjacency processor. The adjacency matrix generated by the generator have both positive and negative values, may be non-symmetric and non-normalized. Therefore it is further processed as $\mathbf{A} = \frac{1}{2}\mathbf{D}^{-\frac{1}{2}}\left(P(\tilde{\mathbf{A}}) + P(\tilde{\mathbf{A}})^T\right)\mathbf{D}^{-\frac{1}{2}}$ where P is a function with a non-negative range applied element-wise on its input (e.g. ReLU or ELU).

Classifier. The classifier can be any GNN models, e.g., a two layer-GCN.

Self-supervised loss. If we only use the above three components (with a two-layer GCN) to solve a node classification problem, some edges will not affect the training loss if the supervised nodes are not sufficient. Therefore, such starved edges may become problematic and lead to poor predictions at the test time. (The paper gave the exact probability of an edge being a starved edge with a two-layer GCN and made examples with several datasets.) To solve this problem, this paper proposed to add noise to the input features and train a separate GNN to denoise the node features, which takes

Preprint. Under review.

node features and a generated adjacency as input and provides updated node features with the same dimension as output.

The final model loss is the summation of the classification loss and the denoising autoencoder loss.

Some paper worth reading in the future:

A multi-task learning framework for pre-training strategies for GNNs [2].

References

[1] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021. (document)

[2] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. When does self-supervision help graph convolutional networks? In *international conference on machine learning*, pages 10871–10880. PMLR, 2020. (document)