Graph Learning for Heterophillic Data

Project report submitted
for the requirements of the course

ELL706 : Optimization

by

Vineeth Kumar Ponugoti 2019EE10545

Submitted to
Prof. Sandeep Kumar

Department of Electrical Engineering,
Indian Institute of Technology Delhi.

May 2023

Acknowledgments

We take this opportunity to express our gratitude to our Instructor **Professor Sandeep Kumar**, whose support and guidance have been invaluable to us during this whole project. His suggestions have helped us think of creative and different approaches towards the problems we faced during our research and come up with various ways to solve those problems. Furthermore, we are grateful to **Mr. Mohit Kataria, PhD Scholar**, for providing us with the right direction and assistance throughout the project. We also would like to thank the Electrical Engineering Department of IIT Delhi to give us the opportunity to do this research work in the incredible field of Optimization.

Vineeth Kumar Ponugoti

1 Topic

The project is based on Graph Learning. In graph learning, the goal is to learn a graph structure that accurately represents the relationships between the entities in the data. This graph structure can be used for various tasks, such as node classification, graph classification, link prediction, and clustering. This is done by combining the structure of graph that is formed by given data and the features of the nodes, forming new embedding of nodes. This works well when the given data is more homophilic in nature. Graph Convolutional Networks (GCNs) are a powerful framework for learning representations of graph-structured data.

2 Problem Statement

However, they face several challenges when dealing with heterophilic graphs, which are graphs that contain nodes that belong to different types or classes. Here are some of the problems that GCNs face when dealing with heterophilic data:

1. Difficulty in modeling node heterogeneity: GCNs assume that nodes in the graph are homogeneous, which means that they share the same features and labels. However, in heterophilic graphs, nodes can belong to different classes and have different features. GCNs struggle to capture the differences between these nodes, which can lead to poor performance.

2. Over-smoothing: GCNs apply a fixed number of convolutional layers to the entire graph, which can lead to over-smoothing of node representations. Over-smoothing occurs when the learned representations become too similar across all nodes, making it difficult to distinguish between different classes of nodes.

3. Limited expressiveness: GCNs are limited in their ability to capture complex relationships between nodes in heterophilic graphs. For example, they may struggle to capture relationships between nodes that belong to different classes or that have different connectivity patterns.

4. Lack of interpretability: GCNs are often criticized for their lack of interpretability, which can make it difficult to understand how they are making predictions. This is particularly problematic in heterophilic graphs, where there may be complex interactions between nodes that are difficult to interpret.

For this project, we want to learn the final weights of graph such that there is a clear weight edge between two nodes of same class even though if the two nodes are not connected in original graph structure or are surrounded by opposite class nodes.
In this project, we focus on using the smoothness property of graphs to learn representations for nodes in the graph. The smoothness property states that nodes that are close to each other in the graph should have similar representations. We develop algorithms that use this property to learn representations that capture the underlying structure of the graph.

3 How to solve the problem?

1. Vassilis Kalofolias in his research paper[1], has learnt graph from smooth signals using 2 methods

   (a) By imposing a log barrier on the degrees, this is done by solving

$\tilde{W} = \arg\min \|W \odot Z\|_{1,1} - \alpha 1^T \log W1 + \beta \|W\|_F^2$
s.t. $W \in \mathcal{W}_m$

where Z is a pairwise distance matrix, and, W_m is the set of valid symmetric weighted adjacency matrices.

(b) Learning graph by regularizing the l2-norm of the degrees
$\tilde{W} = \arg\min \|W \odot Z\|_{1,1} + \alpha \|W1\|^2 + \alpha \|W\|_F^2$
s.t. $W \in \mathcal{W}_m$
$\|W\|_{1,1} = s$
where Z is a pairwise distance matrix, and, W_m is the set of valid symmetric weighted adjacency matrices.

As mentioned in the above problem statement, we want to learn a graph without loosing the information of features of nodes, and we want an edge between 2 similar nodes, so we try to learn the graph based on learning graph on smooth signals. This helps us obtaining a considerable weight of the edge between 2 similar nodes and small weight between nodes of the graph which are not similar. The first term $|W \odot Z\|_{1,1}$ is the smoothness function of the graph, this allows us to learn the graph as per our requirement. The second term $\alpha 1^T \log W1$ imposes a barrier that each node is connected atleast by one edge.
Currently we have learnt graph for hetrophilic data by imposing log barrier on the degrees and next, we intend to see what compare results that are obtained by classification algorithms between ground truth graphs and learnt graphs.

We have tried two techniques to obtain the graph. Let W_{gt} be the ground truth graph, γ be the heterophilly index of the graph. The following are the techniques used:

1. We learn the graph by applying smoothing property as above

   $\tilde{W} = \arg\min \|W \odot Z\|_{1,1} - \alpha 1^T \log W1 + \beta \|W\|_F^2$
   s.t. $W \in \mathcal{W}_m$
   where Z is a pairwise distance matrix, and, W_m is the set of valid symmetric weighted adjacency matrices.

   If W^* is the solution obtained by solving the optimization problem, we incorporate the ground truth graph into the new graph according to the heterophily index of the given data. $W = (1-\gamma)W_{gt} + \gamma W^*$

   The γW^* term, adds newly learnt graph using optimization method above in the proportion of heterophilly of the data and $(1-\gamma)W_{gt}$ term adds the ground truth graph in proportion of homogenity of the data.

2. In another method, we incorporate the ground truth graph in the optimization problem as following
   $\tilde{W} = \arg\min \|W \odot Z\|_{1,1} - \alpha 1^T \log W1 + \beta \|W - W_{gt}\|_F^2$
   s.t. $W \in \mathcal{W}_m$

   where Z is a pairwise distance matrix, and, W_m is the set of valid symmetric weighted adjacency matrices and A is the given adjacency matrix.

   Through this we are incorporating W_{gt} in the sparsity term and are trying to deviate as much less as possible from the given graph. In human language, only learn or eliminate the edges if you are confident.

.

3.1 Chameleon Dataset

This dataset has heterophily index 0.8.

Following are the hyperparameters used for training:

1. Learning Rate = 0.001
2. Epochs = 1000

Results using 1st Technique are as follows for node classification:

1. Accuracy using Learnt Graph : 57-58%
2. Accuracy using Ground Truth Graph: 56-58%

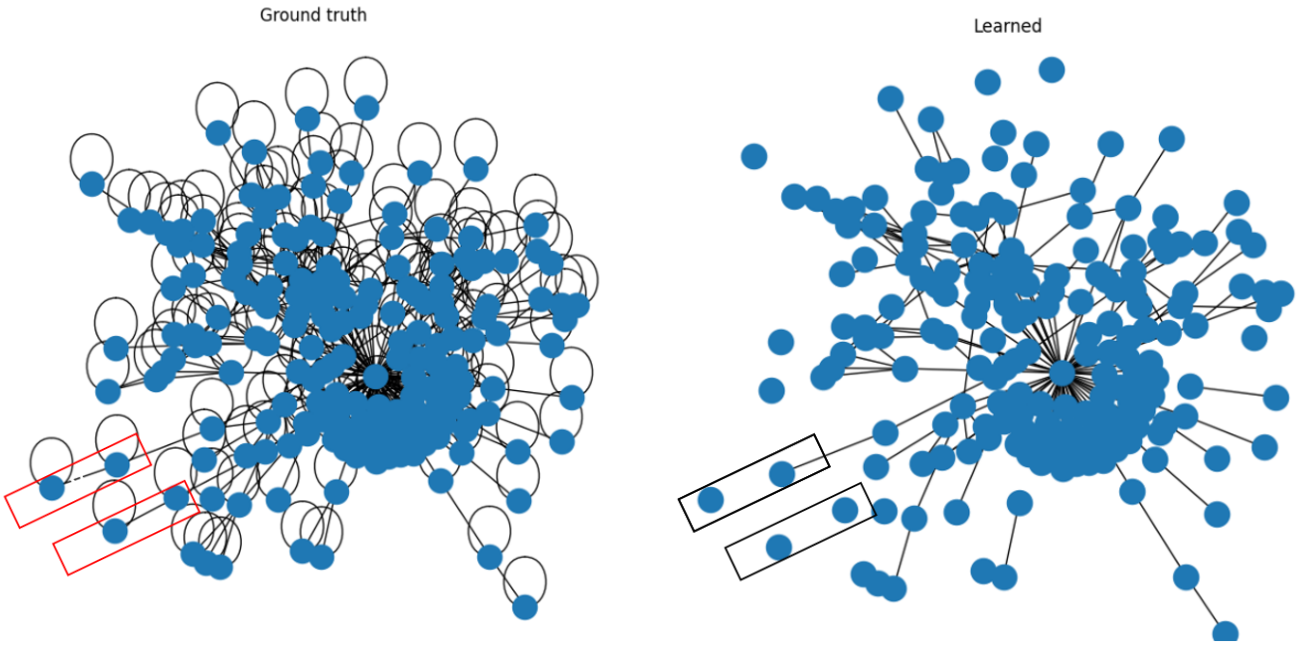
Results using 2nd Technique are as follows for node classification:

1. Accuracy using Learnt Graph : 60-62%
2. Accuracy using Ground Truth Graph: 56-58%

3.2 Cornell Dataset

Only 2nd technique was used here. The results are as follows for node classification:
Following are the hyperparameters used for training:

1. Learning Rate = 0.001
2. Epochs = 1000



(a) Ground Truth Graph (b) Learnt Graph

Figure 1

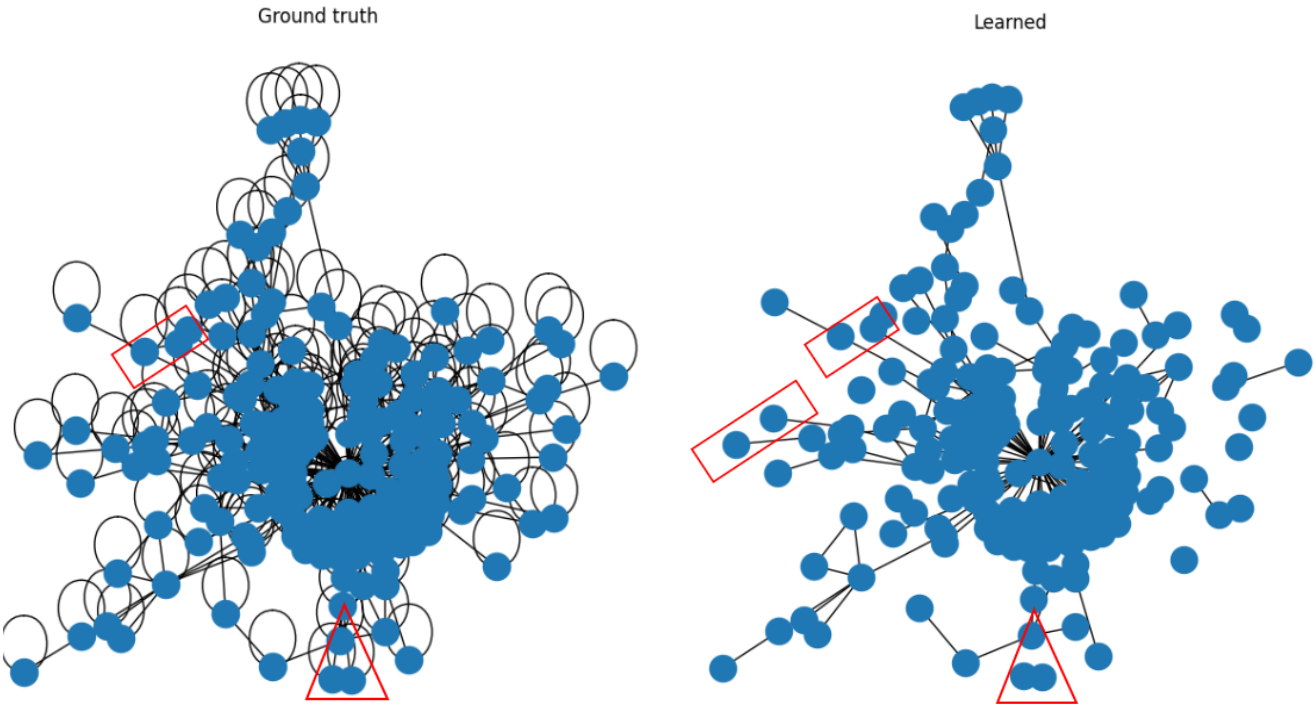
The following table shows the accuracy on test data for different values of α and β

α/β	0.2	0.6	1	1.4	1.8	2
0.2	0.51	0.51	0.48	0.44	0.48	0.51
0.6	0.34	0.51	0.48	0.51	0.48	0.48
1	0.24	0.44	0.51	0.48	0.44	0.48
1.4	0.24	0.51	0.51	0.51	0.44	0.48
1.8	0.24	0.2	0.44	0.48	0.48	0.44
2	0.24	0.2	0.44	0.48	0.48	0.51

1. Accuracy using Learnt Graph : 51%
2. Accuracy using Ground Truth Graph : 31%

3.3 Texas Dataset

Only 2nd technique was used here. The results are as follows for node classification:



(a) Ground Truth Graph (b) Learnt Graph

Figure 2: It can be seen that edges are removed in learnt graph corresponding to which were present in ground truth graph

Following are the hyperparameters used for training:

1. Learning Rate = 0.001
2. Epochs = 1000

α/β	0.2	0.6	1	1.4	1.8	2
0.2	0.31	0.28	0.31	0.31	0.31	0.31
0.6	0.38	0.31	0.31	0.28	0.31	0.31
1	0.24	0.28	0.28	0.28	0.28	0.28
1.4	0.24	0.28	0.31	0.28	0.31	0.28
1.8	0.24	0.2	0.31	0.28	0.31	0.31
2	0.24	0.2	0.28	0.28	0.31	0.31

1. Accuracy using Learnt Graph : 38%
2. Accuracy using Ground Truth Graph : 31%

4 Conclusion

For Chameleon Dataset, though the results are better(60-62%) than with compared to ground truth graph(56-58%), we could not achieve better results than current benchmark (68-70%)

For Cornell Dataset, the accuracy for test set using newly learnt graph(51%) is nearly 20% more than that obtained for ground truth graph(31%)

For Texas Dataset, the accuracy for test set using newly learnt graph(38%) is nearly 7% more than that obtained for ground truth graph(31%)

References

[1] Vassilis Kalofolias. How to learn a graph from smooth signals, 2016.