Improved Machine Learning Methods for Graph Data

Aseem Baranwal, Cheriton School of Computer Science, University of Waterloo

1 Background and Motivation

A large number of interesting challenges in machine learning are defined on datasets where attributes are accompanied by a graph structure. Several leaders in the industry employ a variety of semi-supervised and unsupervised learning methods for applications in knowledge graphs, community detection, recommender systems, financial forensics, and drug discovery. This has led to the insurgence of Graph Neural Networks (GNNs) [5] which are highly efficient methods for machine learning on graphs. Despite a lot of research on the development, efficiency, and scale of different GNN architectures, there has been limited progress on a comprehensive framework that can identify regimes concerning properties of the data, where GNNs have a clear advantage over traditional learning methods for certain tasks, e.g., node-classification and link-prediction. A thorough understanding of the performance and capabilities of existing architectures is required to help build superior learning models that work in regimes where traditional methods fail, in addition to being extensively scalable.

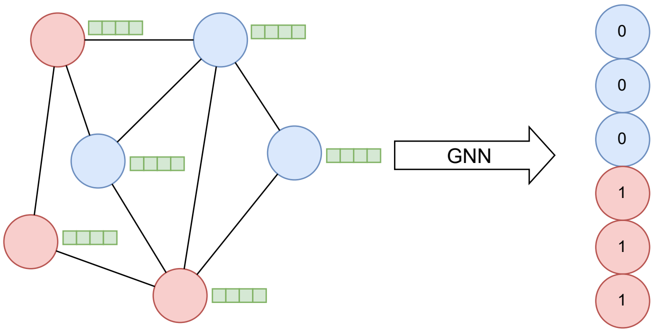


Figure 1: Formulation of a simple node-level binary classification problem. The input is a graph with features for every node, and the output is the class label for each node.

The goal of my research is two-fold: first, to provide a rigorous theoretical framework that helps us analyze the performance and generalization capacity of several families of GNN architectures in the context of graph-based machine learning tasks, e.g., node-classification, graph classification and link-prediction; and second, to use insights from these analyses to design faster and scalable learning models that utilize the benefits of both the sources of information: (1) relational information as the graph, and (2) feature information as the attributes of nodes and edges.

2 Summary of past research

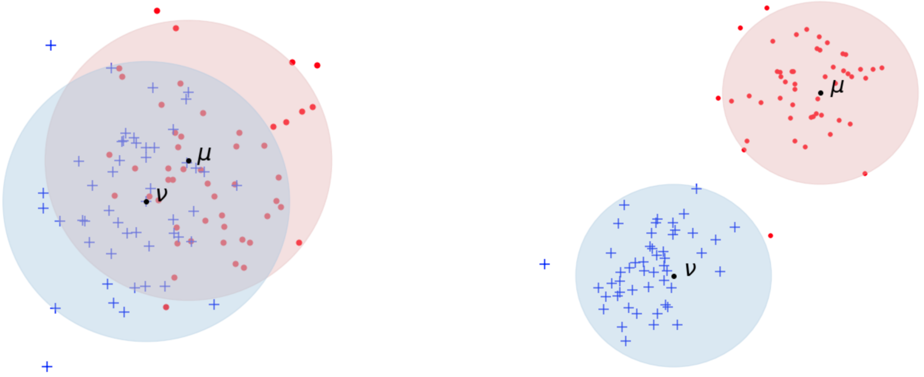
My research has identified the precise quantities associated with the *signal* from the two sources of information in the data, i.e., the graph (relational information) and the features (attribute information). The theoretical results are developed on top of the CSBM (Contextual Stochastic Block Model) which is a Gaussian Mixture Model coupled with a Stochastic Block Model. CSBM is a commonly studied data model for methods that work with both sources of information. For the node-classification problem, it is characterized by Gaussian node features and a graph that exhibits a community structure corresponding to the classes. My results state that when the magnitude of the signals is larger than a threshold, then architectures like GCN [6] and GAT [8] can correctly classify all the nodes, as opposed to traditional methods in that regime, like an MLP.

For an example of the identified signals, consider the node-classification problem on the CSBM data model with a binary GMM coupled with a two-block SBM. Let n be the number of nodes and d be the number of attributes per node. Denote by $\mathbf{X} \in \mathbb{R}^{n\times d}$ the feature matrix sampled from a Gaussian mixture, $\mathbf{X}_i \sim \mathcal{N}(y_i\boldsymbol{\mu}, \sigma^2\mathbf{I}_d)$ where node i has label $y_i \in \{-1, 1\}$. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be the adjacency matrix of the graph, where $a_{ij} = 1$ with probability p and 0 with probability $1 - p$ if nodes i and j are in the same class, and $a_{ij} = 1$ with probability q and 0 with probability $1 - q$ if they are in different classes. We denote the degree of a node by $\mathbf{deg}(i) = \sum_j a_{ij}$. In this case, the signal from the Gaussian mixture is given by the quantity $\frac{\|\boldsymbol{\mu}-\boldsymbol{\nu}\|_2}{\sigma}$, that is, the ratio of the distance between the means of the clusters to the standard deviation. On the other hand, the signal from the graph is given by the quantity $\left|\frac{p-q}{p+q}\right|$, which quantifies the distinguishability of the nodes based on the relational information. With the identification of these signals, I have obtained results for perfect node-classification for MLP, GCN and GAT in a series of papers [2, 3, 4], explained briefly below.

2.1 Benefits and limitations of graph convolutions (ICML 2021)

A graph convolution transforms each node feature as $\mathbf{X}_i \to \frac{1}{\mathbf{deg}(i)}\sum_j a_{ij}\mathbf{X}_j$. Then for a classification task with k classes, a node embedding function f is constructed that takes as input $\mathbf{X}$ and $\mathbf{A}$ and produces k output variables per node $f(\mathbf{X}, \mathbf{A}) \in \mathbb{R}^{n\times k}$ where k is the number of classes.

Variance reduction and improvement in classification. In the papers [2, 3], we identified that the key effect of a graph convolution is the reduction in the variance (noise) of the data (as seen in Fig. 2). The variance imposes a classification regime characterized by a threshold for the product of signals from the two sources of information. We showed that when node features are accompanied by a graph, a single convolution enables a multi-layer network to classify the nodes in a wider regime as compared to methods that do not utilize the graph, improving the threshold for the distance between the means of the features by a factor of up to $1/\sqrt{\mathbb{E}\,\mathbf{deg}}$, while two convolutions further improve it by up to $1/\sqrt{n}$, where n is the number of nodes and $\mathbb{E}\,\mathbf{deg}$ denotes the expected degree of a node.

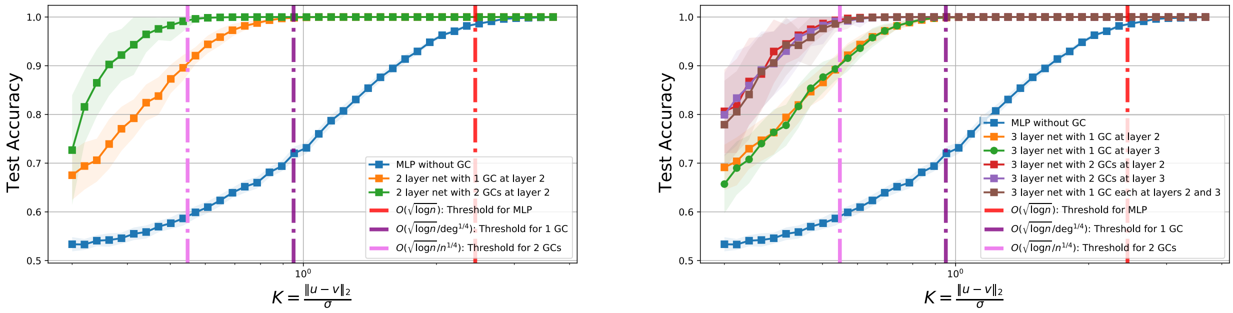


(a) Original data

(b) After graph convolution

Figure 2: A simplified depiction of the reduction in the variance of data due to a graph convolution operation. It reduces the overlap between the two classes, making classification easier.

Efficiency and scalability. The theoretical results state that three or more graph convolutions offer the same improvement in classification as two graph convolutions, concluding that for $L > 2$, it is unnecessary to use L graph convolutions if we have an L-layer network as the learning model. Our results are verified by extensive experiments on benchmark datasets (OGB, Cora, Pubmed, CiteSeer) where we observe that multi-layer networks with two graph convolutions perform better than those with one convolution; and networks with three convolutions perform similar to those with two convolutions irrespective of which layers contain the convolutions. Naturally, the performance of learning methods that use the same number of graph convolutions is mutually similar, as shown in Fig. 3.



(a) Two-layer networks with $(p, q) = (0.2, 0.02)$.

(b) Three-layer networks with $(p, q) = (0.2, 0.02)$.

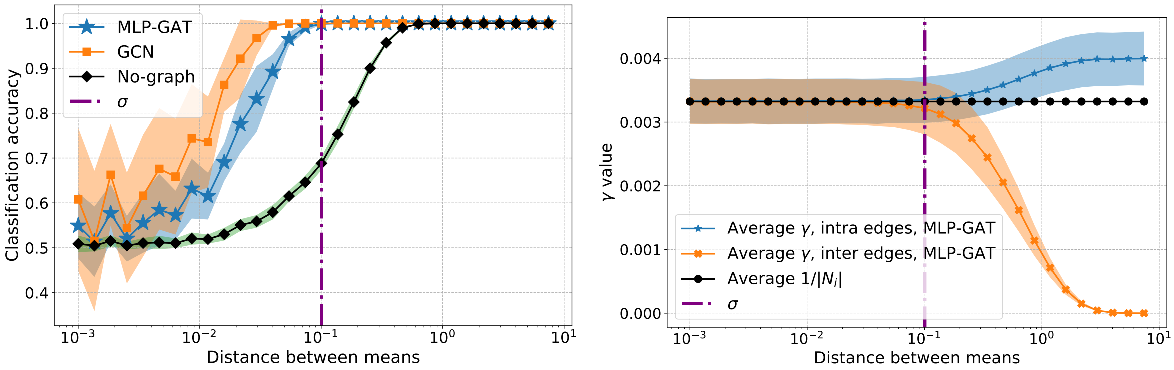
Figure 3: Learning methods with the same number of graph convolutions (GCs) achieve similar performance. Two graph convolutions obtain better performance than a single convolution.

2.2 The capacity of graph attention mechanisms (ICLR 2022)

Graph attention extends the idea of graph convolutions. The coefficients γ_{ij} (which were a_{ij} in the case of GCNs) are implicitly defined as a byproduct of an attention mechanism $\psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as follows:

$$\gamma_{ij} = \frac{\exp(\psi(\mathbf{X}_i, \mathbf{X}_j))}{\sum_{k\in[n]} \exp(\psi(\mathbf{X}_i, \mathbf{X}_k))}$$

In [4], we analyzed the GAT mechanism and showed two results: first, in the regime where the distance between the means of the mixture model is large enough, the graph is not needed at all, however, GAT performs strictly better than a GCN in terms of prediction accuracy; and second, in the regime where the distance between the means is very small, GAT fails to learn the correct weights of the edges and performs similar to a GCN. This proves a fundamental limitation of the attention mechanism. Fig. 4a shows that for a node-classification task, GAT is not better than GCN when the distance between the means is smaller than σ, the standard deviation. Fig. 4b shows that beyond this threshold, GAT can reduce the weights of the edges between nodes of different classes (orange) and boost the weights of the edges between nodes of the same class (blue), hence, it outperforms a GCN in this regime.



(a) Node-classification accuracy.

(b) Implicitly learned edge-weights.

Figure 4: Characteristics of GAT.

3 Progress on current research

My previous results rely heavily on probability theory and concentration arguments, hence, they require an assumption on the graph density. More precisely, the results assume that the average degree of a node is $\omega(\log n)$ where n is the total number of nodes. My current research aims at extending the previous results to very sparse regimes, where the average degree is $O(1)$. This problem is hard because it is known [1] that solely based on relational information, it is impossible to obtain the perfect classification of *all* nodes if the degrees are not at least logarithmic; however, it is still possible to get practically useful results by relaxing the requirement of successful classification to a fraction of nodes rather than all of the nodes. I am working on new GNN architectures that can be analyzed for very sparse regimes. In this regard, there are two directions of research that I am exploring.

1. **Non-backtracking matrix.** For a graph with m edges, this is an $m \times m$ matrix B where $B_{ef} = 1$ for edges $e = (u, v)$ and $f = (u', v')$ if $v = u'$ and $u \neq v'$, and $B_{ef} = 0$ otherwise. This matrix is known to have better spectral properties as compared to the adjacency or the Laplacian matrix [7] and can perform weak recovery on a community detection problem [1] even when the average degree is constant. A comprehensive study of this matrix will help in the design of new GNN architectures that perform well in very sparse regimes with theoretically backed guarantees. The intuition behind why this matrix is better is that it simulates a non-backtracking random walk on the graph, which enables it to gather information from a larger number of nodes as compared to the adjacency matrix that looks only at direct neighbours.
2. **Belief propagation.** The BP algorithm is a dynamic programming approach to solve inference problems and answer probability queries on a graph, based on passing local messages among the nodes. Although it is well studied and adopted in various settings [1, 9], there are a lot of open questions about the regimes in which BP performs better than other algorithms on graphs, and about the convergence guarantees of the algorithm. A thorough understanding of the properties of BP will enable the design of new learning methods that work in very sparse regimes, backed by sharp theoretical results.

4 Objectives for future research

My previous research has completely closed the problem for perfect node-classification in the regime where the average degree of the graph is at least logarithmic. My current research focuses on extending these results for extremely sparse graphs with constant expected degree. Following several key insights from previous and current research, I consider the following broad-spectrum goals for future work.

- **Information-theoretic limits and generalization.** My current results hold for several fixed network architectures that are most popularly used in practice. However, to develop new learning models, it will be useful to understand the information-theoretic limits for learning problems on graph data. These limits will provide us with a glimpse of the characteristics of learning models that can achieve the best possible performance, fostering the design of new learning methods. In addition, I aim to obtain a theoretical understanding of the generalization potential of various GNN models on unseen data. Formally, this involves results that provide tight upper bounds on the generalization error. We already have tight bounds for regimes where the data is completely classifiable with high probability; however, extension of these results to partial classification is non-trivial and a promising direction for research.
- **Additive learning methods.** My results indicate that the threshold for perfect classification is strongly correlated with the architecture of the learning model. For example, every layer of a multi-layer GCN is multiplicative ($\mathbf{X}' = \mathbf{D}^{-1}\mathbf{A}\mathbf{X}$), hence, the total signal of the data that a GCN

model can utilize is the product of the signals from the two sources of information (features $\mathbf{X}$ and graph $\mathbf{A}$). Having a product of two signals is undesirable, because if the signal from any of the two sources of information is feeble, then the model will fail. This motivates the design of architectures that can provide a similar variance reduction but with an additive guarantee, which will be able to effectively *ignore* an unhelpful piece of information.

- **Efficiency and scalability.** As shown in previous work, if a learning method requires an L-layer network for optimal performance, then it is not necessary to have L graph convolutions. Current state of the art architectures place one graph convolution in each layer, however, we obtain the same performance guarantee by placing multiple graph convolutions in only one layer of a deep network. Based on this fact, I aim to develop new models that will reduce the amount of resources used during training, since graph operations are computationally expensive.

- **Applications beyond node-classification.** I aim to extend my previous work to develop a unified theory for various other machine learning tasks on graph data, including link-prediction and graph classification for more general and practically useful data models.

References

[1] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[2] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proc. of Mach. Learn. Res.*, pages 684–693. PMLR, 18–24 Jul 2021.

[3] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Effects of graph convolutions in multi-layer networks. *arXiv preprint arXiv:2204.09297*, 2022.

[4] Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. *Workshop on Anchoring Machine Learning in Classical Algorithmic Theory, 10th International Conference on Learning Representations (ICLR)*, 2022.

[5] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[6] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[7] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[8] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[9] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8(236-239):0018–9448, 2003.