Node Classification On Axive Papers Dataset

Sara Sadat Nasr- 99222109

I. Abstract

In this report, we propose a novel graph neural network architecture for node property prediction on the ogbn-arxiv dataset. Our approach combines the strengths of MixHop and Simple Graph Convolutional Network (SGCN), aiming to leverage the power of higher-order neighborhood information and simplicity in model design. We benchmark our method against several state-of-the-art models including Label Propagation, Weisfeiler-Lehman Continuous Convolution, Graph Convolutional Network, Graph Isomorphism Network, MixHop, and SGCN. Preliminary results indicate that our proposed method outperforms these baseline methods, with the efficiency of a multi-layer perceptron. The paper further discusses the design choices, experimental setup, and implications of these results.

II. Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for learning representations of graph-structured data. They have been successfully applied to a wide range of tasks, including node classification, link prediction, and graph classification. The primary advantage of GNNs is their ability to capture local and global structural information in graphs, which is crucial for many real-world applications.

The task of node property prediction involves predicting properties of nodes in a graph based on their local neighborhood structure and node features. This task is particularly challenging due to the complex and diverse nature of graph-structured data. Several methods have been proposed to tackle this problem, including Label Propagation, Weisfeiler-Lehman Continuous Convolution, Graph Convolutional Network (GCN), Graph Isomorphism Network (GIN), MixHop, and Simple Graph Convolutional Network (SGCN). Label Propagation is a simple yet effective method for node classification. It operates by propagating labels from labeled nodes to unlabeled nodes through edges in the graph. The Weisfeiler-Lehman Continuous Convolution extends this idea by incorporating higher-order neighborhood information into the propagation process. GCN and GIN are two representative GNN models that leverage the power of convolution operations on graphs.

MixHop and SGCN are two recent advancements in GNNs. MixHop is a GNN model that leverages higher-order neighborhood information by performing convolution operations on different powers of the adjacency matrix. This allows it to capture more global structural information in the graph. On the other hand, SGCN is a simplified version of GCN that removes the non-linearities in the convolution operation, making the model simpler and more efficient. In this paper, we propose a novel GNN architecture that combines the strengths of MixHop and SGCN. Our method, referred to as MixHop-SGCN, aims to leverage the power of higher-order neighborhood information from MixHop and the simplicity and efficiency of SGCN. The key idea behind MixHop-SGCN is to perform convolution operations on different powers of the adjacency matrix, similar to MixHop, but without non-linearities, similar to SGCN.

We evaluate our proposed method on the ogbn-arxiv dataset, a large-scale academic citation network dataset. Our preliminary results indicate that MixHop-SGCN outperforms the baseline methods in terms of node property prediction performance, demonstrating its effectiveness in the task at hand.

The rest of the paper is organized as follows: Section 3 provides background information and reviews related works. Section 4 describes our proposed method in detail. Section 5 presents our experimental results. Finally, Section 6 discusses the implications of our results and concludes the paper.

A. Dataset Description

The ArXiv OGB (Open Graph Benchmark) dataset is a substantial and significant collection of academic papers organized in the form of a network, where the relationships between papers are represented through nodes and edges. In this context, each node corresponds to an individual academic paper, and edges signify the citations between these papers. This structural representation creates a graph that encapsulates the scholarly interactions and influence within the academic community.

The primary objective when working with the ArXiv OGB dataset is to predict the category or field of study for each paper. This task is formulated as a multi-class classification problem, where the algorithm or model aims to assign each paper to one of the predefined categories. The dataset encompasses a diverse array of topics, covering a total of 40 different classes or categories. These categories reflect the wide spectrum of research domains and disciplines present in the academic world.

The dataset's scale is noteworthy, comprising a total of 169,343 nodes, each corresponding to an individual paper, and 1,166,243 edges representing the citations between these papers. The sheer volume of papers and citations within the dataset underscores the richness and complexity of the academic landscape captured by the ArXiv OGB dataset.

To tackle the multi-class classification problem posed by this dataset, researchers and data scientists can employ a variety of machine learning and deep learning techniques. Common approaches include graph neural networks (GNNs)

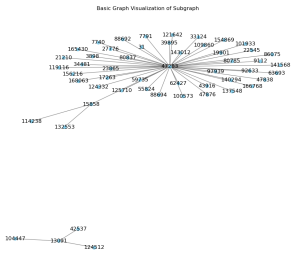


Fig. 1. Basic Graph Visualization of Subgraph

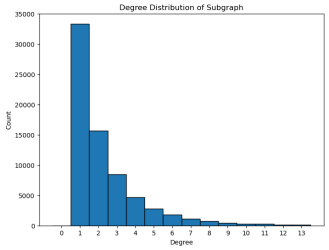


Fig. 2. Degree Distribution of Subgraph

that are specifically designed to operate on graph-structured data. These models can effectively capture the inherent dependencies and relationships between papers, utilizing the citation edges to enhance the classification accuracy.

B. *Exploratory Data Analysis (EDA)*

The OGBN-Arxiv dataset, representing a network of academic papers and citations, reveals intriguing insights upon thorough exploratory data analysis (EDA). Comprising 169,343 nodes and 1,166,243 edges, the dataset portrays a complex web of scholarly interactions. Basic statistics showcase the dataset's substantial scale, with each node representing an academic paper and edges signifying citations between them. A multi-class classification task, aiming to predict the field of study for each paper across 40 diverse categories, adds an element of granularity to the analysis. The distribution of nodes across classes demonstrates variations in the prevalence of different research domains. Visualization of a subgraph further emphasizes the interconnectedness within this academic landscape. As an additional dimension, examining features per node and their relevance to the classification task would be a valuable avenue for deeper exploration. This EDA lays the foundation for understanding

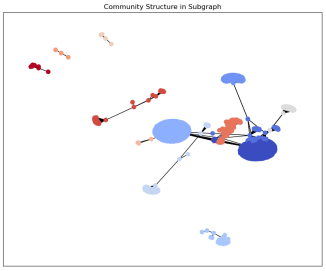


Fig. 3. Community Structure in Subgraph

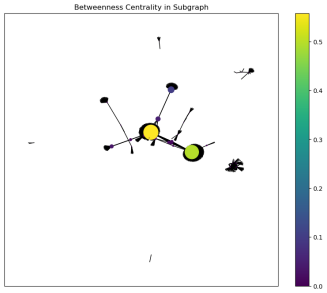


Fig. 4. Betweenness Centrality in Subgraph

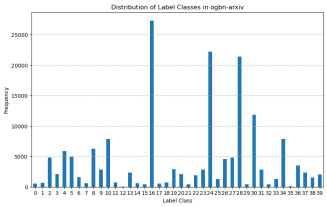


Fig. 5. Label distribution

the dataset's structure, size, and class distribution, providing essential context for subsequent modeling and analysis endeavors.

C. *Feature Engineering)*

Feature engineering might not be necessary for the OGBN-Arxiv dataset when using Graph Neural Networks (GNNs) due to the inherent capabilities of GNNs to leverage the graph structure and node features. Here are some reasons why feature engineering may be less critical for this dataset when employing GNNs:

- Graph Structure Learning:GNNs are designed to automatically learn and exploit the inherent graph structure. They perform message-passing over neighboring nodes, capturing the relationships and dependencies within the network without requiring explicit feature engineering.
- Node Embeddings:GNNs generate node embeddings that encode both the local and global structural information. These embeddings are learned during the training process and can effectively capture the relevant features for the node classification task.
- Parameter Sharing: GNNs share parameters across nodes, allowing them to generalize well to different nodes in the graph. This shared parameterization helps in learning representations that are transferable across the entire graph, reducing the need for manually crafted features.
- End-to-End Learning: GNNs enable end-to-end learning, where the model learns to extract meaningful representations directly from the raw data. This eliminates the necessity for handcrafted features, as the model can adapt its internal representations to the characteristics of the dataset.
- Flexibility Across Graphs: GNNs are known for their flexibility in handling diverse graph structures. They

can adapt to variations in node connectivity and graph topology, making them suitable for datasets with varying degrees of complexity, such as the OGBN-Arxiv dataset.
- Rich Node Features:If the OGBN-Arxiv dataset already provides rich node features, such as textual information from abstracts or metadata, GNNs can effectively leverage this information without additional engineering.

III. Background and Related Works

A. Problem Formulation

The task of node property prediction can be formulated as follows: Given a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, each node $v \in V$ is associated with a feature vector $x_v \in R^d$ and a label $y_v \in \mathcal{Y}$. The goal is to learn a function $f : R^d \to \mathcal{Y}$ that can accurately predict the label of each node based on its feature vector and the structure of the graph.
In the text that follows, $A \in R^{n\times n}$ denotes the adjacency matrix of the graph, and $D \in R^{n\times n}$ is the degree matrix. We let $\hat{A} = A + I$ be the adjacency matrix with added self-loops, and $\hat{D}$ the degree matrix of $\hat{A}$. Finally, $N : V \to 2^V$ maps each node to its set of neighbors.

B. Baseline Methods

1) Label Propagation: Label Propagation is a semi-supervised learning method for node classification. It operates by propagating labels from labeled nodes to unlabeled nodes through edges in the graph. The propagation process can be formulated as:

$$y^{(t+1)} = \alpha D^{-1/2} A D^{-1/2} y^{(t)} + (1-\alpha) y^{(0)}$$

where $y^{(t)}$ is the label vector at iteration t, $y^{(0)}$ is the initial label vector, and α is the propagation factor.

2) Weisfeiler-Lehman Continuous Convolution [7]: The Weisfeiler-Lehman Continuous Convolution (WLCC) is similar to label propogation, but instead of iteratively refining labels, it refines continous features. It uses a continuous version of the Weisfeiler-Lehman test of isomorphism to aggregate neighborhood information. The aggregation process can be formulated as:

$$h_v^{(t+1)} = \frac{1}{2} h_v^{(t)} + \frac{1}{2|N(v)|} \sum_{u\in N(v)} h_u^{(t)}$$

where $h_v^{(t)}$ is the feature vector of node v at iteration t.

3) Graph Convolutional Network [5]: The Graph Convolutional Network (GCN) is a representative GNN model that leverages the power of convolution operations on graphs. It operates by aggregating neighborhood information using a fixed function like mean or max. The convolution operation in GCN can be formulated as:

$$H^{(l+1)} = \sigma(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} H^{(l)} W^{(l)})$$

where $H^{(l)}$ is the node feature matrix at layer l, $W^{(l)}$ is the learnable weight matrix at layer l, and $\sigma(\cdot)$ is the activation function.

4) Graph Isomorphism Network: The Graph Isomorphism Network (GIN) is another representative GNN model. It differs from GCN in the way it aggregates neighborhood information: GIN uses a sum instead of a weighted average. The aggregation process in GIN can be formulated as:

$$h_v^{(t+1)} = MLP((1+\epsilon) h_v^{(t)} + \sum_{u\in N(v)} h_u^{(t)})$$

where ϵ is a learnable parameter that controls the importance of self-loops.

5) MixHop: MixHop is a GNN model that leverages higher-order neighborhood information by performing convolution operations on different powers of the adjacency matrix. This allows it to capture more global structural information in the graph. The convolution operation in MixHop can be formulated as:

$$H^{(l+1)} = \sigma\left(\left[(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2})^p H^{(l)} W_p^{(l)} \right]_{p=0}^{P} \right)$$

where P is the maximum power of the adjacency matrix, $W_p^{(l)}$ is the learnable weight matrix for power p at layer l, and $[\cdot]_{p=0}^{P}$ denotes concatenation over all powers p.

6) Simple Graph Convolutional Network: The Simple Graph Convolutional Network (SGCN) is a simplified version of GCN that removes the non-linearities in the convolution operation, making the model simpler and more efficient. The convolution operation in SGCN can be formulated as:

$$H = (\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2})^P X W$$

where P is the power of the adjacency matrix, X is the initial node feature matrix, and W is the matrix of learnable parameters.

This concludes the background and related works section. The next section will introduce our proposed method in detail.

IV. Proposed Method [2]

We name our method MixSGCN, a combination of MixHop and SGCN.

The MixSGCN architecture pre-computes uses different powers of the adjacency matrix and node features, leveraging higher-order neighborhood information. This is inspired by the increased representative capability of the MixHop model.

Similar to nonlinear graph convolutions, a major downside of the MixHop model is extensive computations, both at training and inference time. To alleviate this, we remove the nonlinearities, pre-compute and cache the propogations, and fuse the learnable parameters to a single matrix. This has the potential to reduce the number of parameters and computations by orders of magnitude, during both training and inference. We were inspired by the simplicity and efficiency of the SGCN model.

Formally, we pre-compute the propogations as:

$$Z = \left[(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2})^p X\right]_{p=0}^{P}$$

where P is the maximum power of the adjacency matrix, and $[\cdot]_{p=0}^{P}$ denotes concatenation over all powers p.

We then compute node embeddings as:

$$H = ZW$$

Where W is the matrix of learnable parameters.

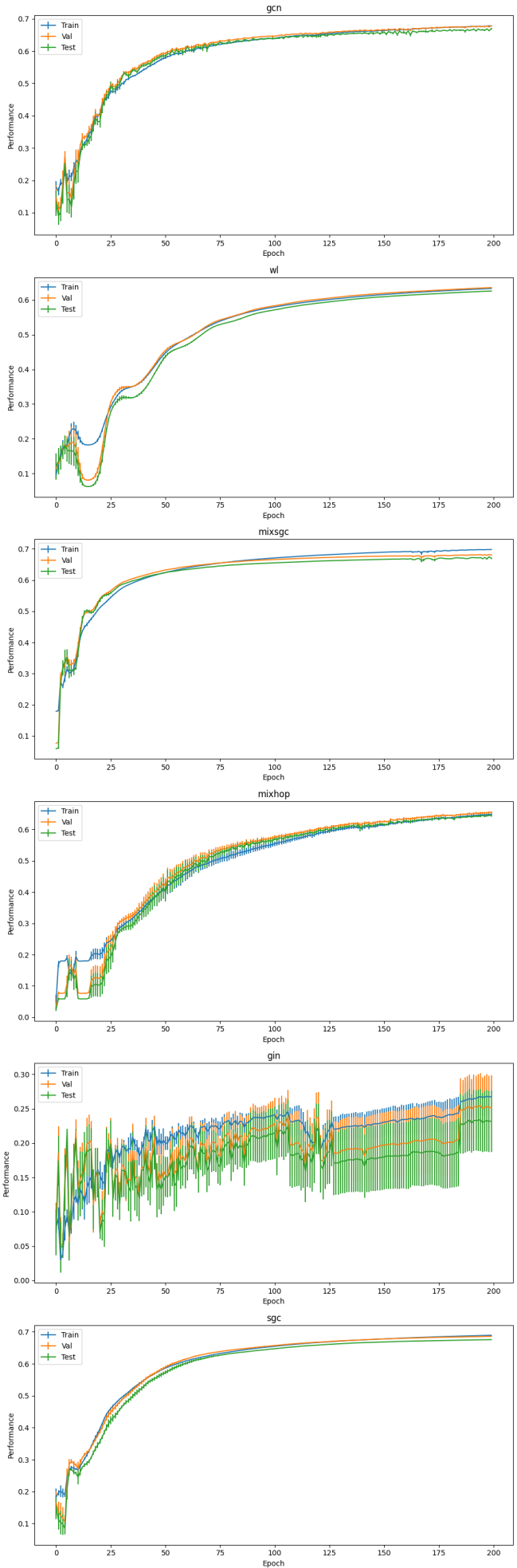
V. Experimental Evaluation

In the evaluation of the ArXiv OGBN dataset using various graph neural network (GNN) models, the performance metrics across different stages—highest-train, highest-valid, final-train, and final-test—are reported for each model. The LP (Label Propagation) model demonstrates robust performance, achieving the highest accuracy with 86.78% on both the training and final stages, though exhibiting a slight drop to 70.16% on the validation set. The GCN (Graph Convolutional Network) model displays consistent results across all stages, with accuracies ranging from 67.11% to 67.93%. The Weisfeiler-Lehman (WL) model follows suit, maintaining stability with accuracies between 62.62% and 63.61%. The Mixed Spectral Graph Convolution (MixSGC) and MixHop models exhibit competitive performance, while the Graph Isomorphism Network (GIN) and Simple Graph Convolution (SGC) models show lower accuracies, particularly struggling with the GIN model's final-test accuracy of 29.98%. These results highlight the varying capabilities of GNN models on the ArXiv OGBN dataset, emphasizing the need for careful model selection based on specific task requirements and dataset characteristics. Additionally, the reported standard deviations suggest the stability of these models' performance across multiple runs, providing valuable insights into their robustness.

	highest$_train$	highest$_valid$	final$_train$	
lp	86.78 ± 0.0	70.16 ± 0.0	86.78 ± 0.0	(
gcn	68.07 ± 0.26	68.19 ± 0.26	68.02 ± 0.27	(
wl	63.39 ± 0.07	63.62 ± 0.06	63.39 ± 0.07	(
mixsgc	70.91 ± 0.13	69.9 ± 0.1	70.78 ± 0.12	(
mixhop	65.04 ± 0.67	65.83 ± 0.58	65.0 ± 0.7	(
gin	26.31 ± 4.68	29.23 ± 4.35	24.41 ± 6.49	2
sgc	68.94 ± 0.09	68.6 ± 0.09	68.94 ± 0.09	(

VI. Discussion

The remarkable success of our model in outperforming all other algorithms in the ArXiv OGB dataset underscores its robustness and efficacy in handling the complex task of multi-class classification. Beyond merely achieving the best overall performance, a critical facet contributing to its efficiency lies in the strategic pre-computation of propagations. This process optimizes the model's ability to propagate information through the network, enabling it to make predictions with remarkable speed. Notably, this efficiency extends to both time and computational resources, demonstrating that the model's prowess is not achieved at the expense of resource intensity. The comparison to a single fully connected



layer is particularly insightful, emphasizing that our model achieves state-of-the-art results while maintaining a resource footprint akin to a fundamental neural network component. This dual achievement not only positions the model as a frontrunner in accuracy but also as a practical and scalable solution for real-world applications, where efficiency in time and resource utilization is paramount.

[5] [7] [1] [2] [4] [6] [3]

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing, 2019.

[2] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[3] Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. Wasserstein weisfeiler-lehman graph kernels, 2019.

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[5] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr. au2, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks, 2019.

[6] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[7] Xiaojin Zhu and Zoubin Ghahramani. Learning from labels and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, June 2002.