Graph neural network guided by feature selection and centrality measures for Node Classification on Homophilic and Heterophily graphs

Asmaa M. Mahmoud  
*Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt.*  
asmaadaoud.1959@azhar.edu.eg

Heba F. Eid  
*Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt.*

Abeer S. Desuky  
*Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt.*

Hoda A. Ali  
*Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt.*

Follow this and additional works at: [https://absb.researchcommons.org/journal](https://absb.researchcommons.org/journal)  
Part of the *Artificial Intelligence and Robotics Commons, and the Data Science Commons*

**How to Cite This Article**  
Mahmoud, Asmaa M.; Eid, Heba F.; Desuky, Abeer S.; and Ali, Hoda A. (2024) "Graph neural network guided by feature selection and centrality measures for Node Classification on Homophilic and Heterophily graphs," *Al-Azhar Bulletin of Science*: Vol. 35: Iss. 1, Article 2.  
DOI: [https://doi.org/10.58675/2636-3305.1663](https://doi.org/10.58675/2636-3305.1663)

This Original Article is brought to you for free and open access by Al-Azhar Bulletin of Science. It has been accepted for inclusion in Al-Azhar Bulletin of Science by an authorized editor of Al-Azhar Bulletin of Science. For more information, please contact kh_Mekheimer@azhar.edu.eg.
Graph Neural Network Guided by Feature Selection and Centrality Measures for Node Classification on Homophily and Heterophily Graphs

Asmaa M. Mahmoud*, Heba F. Eid, Abeer S. Desuky, Hoda A. Ali

Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, 11765, Egypt

Abstract

One of the most recent developments in the fields of deep learning and machine learning is graph neural networks (GNNs). The core task of GNNs is the feature aggregation stage, which is carried out over the node's neighbors without taking into account whether the features are relevant or not. The majority of these existing node representation techniques only consider the network's topology structure while completely ignoring the centrality information. In this paper, a new technique for explaining graph features depending on four different feature selection approaches and centrality measures to identify the important nodes and relevant node features is proposed. A significant design approach for GNNs is also presented. For this, batch renormalization is adopted to normalize over the GNN layers. In this study, we primarily focused on homogeneous graph datasets focusing specifically on homophily and heterophily characteristics. The simulation results show that selecting specific subsets of all features and adding additional features based on centrality measures, which are then sent to modify the GNN layer, can lead to better performance across a variety of homophily and heterophily datasets. The outcomes demonstrate the proposed method's superiority to recent methods with accuracy values of 90.65, 81.23, 88.27, 85.93, and 81.23 % for our datasets, respectively.

Keywords: Centrality measures, Feature selection, Graph neural network, Homophily and heterophily datasets, Node classification

1. Introduction

The use of graph neural networks (GNNs) in practical applications across a variety of fields has grown rapidly in recent years. These applications cover a wide range of tasks, including node ranking [1], molecular property predictions [2], natural language processing [3], vision-related tasks [4], recommendation systems [5], and graph data mining [6]. GNNs are made to work with dataset-derived graph structures and learn from node or edge features. The feature aggregation process is the main step of GNNs that sets them apart from other neural network architectures. The graph structure is used to analyze the neighbors of the nodes, and their features are integrated with the node's features. The fundamental principle is that nodes are interconnected with one another. As a result, integrating information from neighbors improves the signal, which aids in the learning process. Recent research [12] has demonstrated that certain datasets have connections between dissimilar nodes, which may call for various aggregation strategies when designing a GNN.

In this study, our focus is on the task of node classification using GNN. Researchers have repeatedly proposed a wide range of variations to solve its different inadequacies in model training and to enhance the predictive abilities due to the popularity of early GNN models like GCN [1]. Neighbor sampling [12], attention mechanism [5], use of a personalized PageRank matrix rather than an adjacency matrix [8], utilizing proximity in feature space [9], and simplified model design [10] are some of the techniques used in these variants. In addition,
there has been an increasing trend in deepening models by stacking more layers and using residual connections to improve the expressiveness of the model [11]. Incorporating up to 64 layers with scaled residual weights, for instance, is what GCNII does [6]. However, the majority of these models are, by their very nature, better suited to homophily datasets, wherein linked nodes have more potential to share a class. Because nodes with various labels are more likely to be connected in heterophily datasets, these GNNs might not perform well with them. Zhu et al. [7] drew attention to this issue and suggested node’s ego-embedding and neighbor-embedding separation to enhance efficiency on heterophily datasets. Other recent research takes a different approach to this issue, for instance, the study by Velickovic et al. [13] makes use of belief propagation approaches to address this problem by putting forth adaptive gating on edges.

The node’s neighbors may not all share the same feature information as it does. As a result, any feature aggregation performed by a GNN with the rigorous homophily assumption may frequently result in the addition of noisy features, which could weaken the signal needed for effective learning in the node features. Researchers have investigated numerous approaches to address this problem by putting forth numerous model design enhancements, such as deeper models, using attention layers [13], residual layers [16], etc. However, the majority of these strategies try to successfully adapt the model to the noisy data [17]. We suggest treating the issue with a feature selection strategy rather than creating a GNN model to fit the noisy data. By deleting unimportant and noisy components, feature selection algorithms seek to select a small subset of the appropriate features from the entire set of input features [18]. When there are noisy characteristics, it may be necessary to fit a model to those noisy components, which might result in less-than-ideal learning and limited generalization. The model can perform better on the prediction task and learn more efficiently by choosing informative features that are important to the task at hand. Centrality features and feature selection are combined in our strategy. GNN performs well, as per Duong et al. [15], if the relationship between node labels and node features is significant. So, in light of centrality metrics, we presented new features in our work and then selected features that closely correlated with node labels. The main advantage of this work is increasing node classification task accuracy by merging:

(1) Centrality measures: these are features that are obtained on the central measurements used to capture significant information about the graph and assign each node a value that indicates the significance of that node within the graph, which aids in enhancing the solution to the problem of node classification.

(2) Selecting features: using four different feature selection methods to select the subset of a dataset’s most crucial features will reduce memory use and reduce high dimensionality, making the model run faster.

(3) GNN architecture: we recommend a simpler GNN design that uses batch renormalization (BR) to accelerate and stabilize the learning process.

2. Preliminaries

2.1. Graph neural networks

Assume that the undirected graph \( G = (V, E) \) has \( n \) vertices (nodes) and \( m \) links (edges) between them. To gather neighborhood data for a node, GNNS use the feature propagation method [2] and nonlinear transformation with matrices of trainable weight to obtain the nodes’ final embeddings. A standard definition of a simple GNN layer is

\[
H^{(i+1)} = \sigma \left( \tilde{A}_{sym} H^{(i)} W^{(i)} \right)
\]

where \( \tilde{A}_{sym} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \) is a symmetric normalized adjacency matrix with additional self-loops. \( H^i \) stands for features from the preceding layer, where \( H^{(0)} = X \), \( X \in \mathbb{R}^{n \times d} \) is the feature matrix for all nodes for each node has a \( d \)-dimensional feature vector associated with it. \( W^i \) is the learnable weight matrix and is a nonlinear activation function, often ReLU in common GNN implementations. However, because properties are cumulatively aggregated in this formulation — that is, the node’s features are added to those of its neighbors — it is appropriate for homophily datasets. By combining a node’s features with those of its neighbors, the signal associated with the label is made stronger, which also serves to increase prediction accuracy. Contrarily, in the case of heterophily, nodes are supposed to have features and labels that are different from those of their neighbors.

H2GCN [26] suggests a propagation technique for heterophily datasets, which separates the features of a node’s neighbors from its features. So, for the GNN layer, we use the following formulation:

\[
H^{(i+1)} = \sigma \left( A_{sym} H^{(i)} W^{(i)} \right)
\]
where $A_{sym} = D^{-1/2}AD^{-1/2}$ a symmetric normalized adjacency matrix without additional self-loops. A concatenation operator can be applied before the final layer to merge features from many hops. Using the standard GNN architecture a simple two-layered GNN may be described as:

$$Z = \hat{A}_{sym} \sigma(\hat{A}_{sym}XW^{(0)})W^{(1)}$$

(3)

2.2. Centrality measures

According to Cui et al. [19], there are numerous ways to generate artificial features; therefore, in our model GNN using feature selection-based centrality measures on homophily and heterophily datasets (GNNFCH), we produce several features based on centrality measures. Finding significant nodes in a network using the essential idea of centrality is used to evaluate the relative importance of the network's nodes. Now, each node might be relevant from a different angle depending on how ‘importance’ is defined [20]. There are numerous metrics for measuring centrality, each of which portrays the significance of a node from a different angle and provides crucial analytical data about the network and its nodes.

2.3. Homophily and heterophily graphs

The term homophily refers to the tendency of nodes belonging to the same group or label to cluster together as shown in Fig. 1. Hence, for instance, in a social network, graduates from the same institution tend to interact with one another, and the network is homophilous in terms of the university’s identifier. The opposite of homophily is called heterophily when connectivity across groups is sparse, while nodes of distinct groups prefer to be connected. For instance, fraudsters are more likely to have connections to accomplices than to other fraudsters in transaction networks. In this instance, fraudsters establish heterophilic relationships with the nodes of their accomplices [21].

There are several commonly used homophily measures in the literature: edge homophily (homophily ratio $H_{edge}$) denotes the fraction of edges that connect two similar nodes. Similarity can be considered in terms of node features or node labels:

$$H_{edge} = \frac{|\{(u,v) \in E : y_u = y_v\}|}{|E|}$$

(4)

Higher values (closer to) suggest strong homophily in the dataset, whereas lower values (near to 0) denote strong heterophily.

Node homophily computes the fraction of neighbors that have the same class for all nodes and then averages these values across the nodes:

$$h_{node} = \frac{1}{n} \sum_{v \in V} \frac{|\{ u \in N(v) : y_u = y_v \}|}{d(v)}$$

(5)

where $d(v) = N(v)$ is the neighbor of the node $v$ and $y$ is the class label.

3. The proposed model: graph neural network using feature selection-based centrality measures

In the proposed model, the optimal subset of node feature matrices is input to the multilayer perceptron (MLP) to create initial node representations. Graph convolution layers are added to capture information from the further level of neighbors. In this step, aggregated feature matrix is linearly transformed

Fig. 1. Homophily and heterophily graph.
and normalized using BR. Then, these features, passed through nonlinearity (ReLU), are finally mapped to the softmax layer to determine the node class.

3.1. Feature aggregation

GNNs’ main function is the feature aggregation stage, which is performed over the node’s neighbors without taking into account whether the features are valuable or not. The effect of network-captured and selected features on the node classification problem will be experimentally investigated. We use five benchmark datasets for node classification that have different structural and homophily properties (Table 1).

3.2. Features based on centrality measures

To choose the most insightful features, a new technique based on node centrality ideas is suggested. A significant problem in network analysis has been the centrality of vertices. A significant problem in network analysis has been the identification of vertices that are more ‘central’ than others. The results of a few important vertices with strong centralities to describe the network’s features are important.

Most GNN models use node representation techniques that concentrate on the topology structure of the networks but completely neglect centrality data. There is some evidence suggesting that additional attribute values derived from the network’s graph structure in the data, such as betweenness centrality, may improve the classification task’s accuracy. To collect lots of information and identify the most important node in a network, we thus provide novel approaches for explaining graph features that are based on graph centrality measurements. There are numerous centrality metrics in use, including eigenvector, degree, closeness, and betweenness metrics.

3.3. Feature selection methods

It becomes more computationally expensive as the number of hops rises as the number of input feature combinations increases exponentially. It is necessary to enhance the model’s predictive accuracy in addition to feature selection. As a result, the feature selection approach may be used to build a GNN model. GNNFCH tends to determine significant features while minimizing the effect of insignificant features utilizing all characteristics as input at first. \( \chi^2 \), principal component analysis, variance threshold, analysis of variance, and mutual information algorithms all operate directly on feature matrices, thus we evaluated these five feature selection methods.

A 16 GB computer was used to perform feature selection algorithms because of the high dimensional data’s intense memory needs. After the algorithm has finished evaluating features from the graph (based on centrality measures), it merges them with the chosen features that were derived from the input features. The result feature matrix is then input to the MLP layer as indicated in Fig. 2 to generate the initial node representation. The merging procedure may be expressed as:

\[
X = \text{Concat}(\text{SelectedF}, \text{CentralityF})
\]

where SelectedF denotes the features chosen using the mentioned selection methods and CentralityF denotes the features computed using the degree, closeness, betweenness, and eigenvector centrality measures.

3.4. Neighborhood normalization

The new features are input to GNN which includes four steps:

1. A preprocessing layer that uses MLP to generate initial node representations.
(2) A GCN-based message passing layer.
(3) A postprocessing layer that creates final node embeddings using MLP.
(4) Feed the node embeddings into a SoftMax layer to determine the node class.

The original node representation is processed using an MLP to create a message in the first step $h_v^0 = X_v$. Equation (7) is the simplest neighborhood aggregation method that only adds up the neighbor embeddings. The representation of a node is then repeatedly updated by equation (8) in a message passing step by aggregating the representations of its neighbors with its own representation in the case of a homophily dataset. The aggregation method has a few drawbacks, including the potential for instability and extreme sensitivity to node degrees. Just normalizing the aggregate operation depending on the nodes' degrees is one way to solve this issue. Hence, the GNN model establishes the message-passing function as follows:

$$a_v^{(k+1)} = \text{SUM} \left( \{ \text{ReLU} \left( \text{BR} \left( W^{(k)} h_u^{(k)} + b^{(k)} \right) \right) \} \Big| u \in N(v) \right) \right) \quad \text{(aggregate technique)}$$

$$h_v^{(k)} = \text{COMBINE} \left( h_v^{(k-1)}, a_v^{(k)} \right) \quad \text{(update technique)}$$

where $h_v^{(k)}$ is the $k^{th}$ layer embedding of node, $W^{(k)}$ and $b^{(k)}$ are trainable weights, and $N(v)$ is the local neighborhood of $v$.

BR is used for normalization during the message-passing process. When training node characteristics in (7), BR is utilized.

Hamilton [22] states that normalization is particularly helpful in tasks when the range of node degrees is large or where information obtained from node features is much more essential than structural information.

The BR is particularly effective at stabilizing the value in GNNs, especially for deep GNNs. As a result, normalization is being used by more and more GNNs. Deep GNN models typically produce poor performance as a result. One option is to gather the inputs and outputs of the GNN using skip connections; in this case, we examine dense connections named SKIP-CAT, which concatenate embeddings from all preceding layers.

4. Experimental results and discussion

This section discusses the thorough examination of the proposed model. The outcomes are also extensively reported. The numerous subheadings have covered the entire result analysis. Two categorical citation network benchmark datasets, homophily and heterophily, which are common citation networks with identical training/testing divisions [23], are selected to check and verify the effectiveness of the experiment on GNNFCH.

4.1. Datasets and settings

With five benchmark datasets, as mentioned in Table 1, which are often used in GNN literature, we perform experiments on the fully supervised node classification problem. The citation network-based datasets Cora and Citeseer [23] are generally

![Fig. 2. An illustration of the proposed method, GNNFCH.](image-url)
regarded as homophily datasets. Links between websites are shown by Cornell graph, whereas Chameleon and Squirrel [24] indicate Wikipedia articles on related topics. We refer to these datasets as heterophily datasets. We use publicly available data splits to give a fair comparison. We compared our suggested model’s experimental outcomes on this particular split with those of other GNN models.

Each dataset is divided into 20 and 80 % for training and testing for a supervised node classification problem [23]. As the mean classification accuracy over 10 random splits, we present the performance. We study the accuracy results with those of other GNN models to provide a thorough performance evaluation of our model.

4.2. Performance of GNNFCH versus traditional graph neural network models

In our experiments, two proposed novel tricks are combined with already-known ones. The effects of various collecting methods in various datasets are explored. According to Kong et al. [25] the distribution of the data determines whether or not the suggested tricks are successful. As a result, it is necessary to use several datasets for tricks and their collections.

Through research using these datasets, we find that GNN with BR in the homophily and heterophily datasets successfully resolves the sensitivity to node degrees problem in the aggregation step. This technique may also be applied with other methods due to its adaptability and great generality. Normalization is frequently utilized to stabilize the GNN gradient.

The representations for each node in the graph get less informative and more alike after a significant amount of message-passing iterations. Among other things, this can result in the problem of over-smoothing. However, with a feature selection method and features based on centrality measures, these representations will vary and uninformative features can be discarded.

So, in our second trick, each node in our model is represented by several kinds of features such as features of the node itself, features of its neighbors, and feature-based centrality, that is combining graph topology with node feature-based centrality in homophily and heterophily datasets can significantly improve GNN in the majority of cases. As a result, using these techniques with GNN is highly recommended. The average accuracy for these datasets was increased by combining the two suggested design strategies, as shown in Table 2.

We find that our model overcomes the effect of noisy features and significantly improves the performance of Cora (3.75 %), Citeseer (5.19 %), Cornell (12.32 %), Squirrel (44.32 %), and Chameleon (22.5 %) when compared with Graph SAGE and GAT that use all features as input.

We find that our method outperforms the Dual-Net GNN method, even though both methods use feature selection techniques. But our method uses the centrality measures, which identify the features of nodes that differ from one to another and overcome the smoothing problem. Therefore, we find that our method achieves an increase for Cora (2.88 %), Citeseer (4.08 %), Cornell (0.16 %), Squirrel (11.96 %), and Chameleon (2.77 %) over the Dual-Net GNN method.

When comparing the results of all settings with other models, for homophily datasets, the difference for Cora (2.16 %) and Citeseer (3.24 %) is significantly smaller compared with heterophily datasets: Cornell (0.16 %), Squirrel (11.96 %), and Chameleon (2.77 %).

This outcome shows that some feature subsets are more helpful than others for enhancing classification performance among all the feature subsets. Features that rely on centrality measures are among

<table>
<thead>
<tr>
<th>Feature selection technique</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Cornell</th>
<th>Squirrel</th>
<th>Chameleon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method (GNNFCH)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chi (our previous method)</td>
<td>89.5</td>
<td>80.1</td>
<td>78.12</td>
<td>37.72</td>
<td>50.81</td>
</tr>
<tr>
<td>Principal component analysis</td>
<td>82.67</td>
<td>71.83</td>
<td>62.3</td>
<td>37.31</td>
<td>45.424</td>
</tr>
<tr>
<td>Analysis of variance</td>
<td>90.65</td>
<td>81.23</td>
<td>88.27</td>
<td>85.93</td>
<td>81.23</td>
</tr>
<tr>
<td>Variance threshold</td>
<td>86.56</td>
<td>76.66</td>
<td>82.63</td>
<td>36.50</td>
<td>47.84</td>
</tr>
<tr>
<td>Mutual</td>
<td>87.12</td>
<td>77.21</td>
<td>81.18</td>
<td>35.78</td>
<td>46.58</td>
</tr>
<tr>
<td>GCN (2017) [26]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph SAGE (2017) [12]</td>
<td>86.90</td>
<td>76.04</td>
<td>75.95</td>
<td>41.61</td>
<td>58.73</td>
</tr>
<tr>
<td>GAT (2018) [12]</td>
<td>82.68</td>
<td>75.46</td>
<td>58.92</td>
<td>30.62</td>
<td>54.69</td>
</tr>
<tr>
<td>GEOM-GCN (2020) [12]</td>
<td>85.27</td>
<td>77.99</td>
<td>60.81</td>
<td>38.14</td>
<td>60.90</td>
</tr>
<tr>
<td>WRGAT (2021) [26]</td>
<td>88.20</td>
<td>76.81</td>
<td>81.62</td>
<td>48.85</td>
<td>65.24</td>
</tr>
<tr>
<td>GPRGNN (2021) [26]</td>
<td>88.49</td>
<td>77.08</td>
<td>81.89</td>
<td>49.03</td>
<td>66.47</td>
</tr>
<tr>
<td>Dual-Net GNN(2022) [27]</td>
<td>87.77</td>
<td>77.15</td>
<td>88.11</td>
<td>73.97</td>
<td>78.46</td>
</tr>
<tr>
<td>UDGN (2022) [28]</td>
<td>87.28</td>
<td>76.68</td>
<td>84.32</td>
<td>70.05</td>
<td>76.79</td>
</tr>
<tr>
<td>NSD (2023) [29]</td>
<td>87.30</td>
<td>77.14</td>
<td>86.49</td>
<td>56.34</td>
<td>68.68</td>
</tr>
</tbody>
</table>
Fig. 3. Convergence of loss function over testing data.
these features. If the nonrelevant features are present in the input, they may also act as noise and affect the performance of the model whether the datasets are homophilic or heterophilic.

4.3. Analysis of validation loss during testing

In this part, we talk about the classifier’s testing dynamics in terms of validation loss. After 200 iterations of running the code, the validation loss converged to a steady value. This is demonstrated in Fig. 3. A binary cross-entropy loss function was used to alter the weights of the neural network.

A random subset of the node feature matrices is chosen for every epoch. Across all subsets, a subset's quality in relation to the label may differ. Therefore, we can see large fluctuations at the top of the curve in Fig. 3. But when the ideal input feature subset is found, we see a sharp decline in the loss value and fewer variations in the validation loss as shown in the bottom of the curve. Validation loss decreases quickly and is noticeably less than at the top of the curve.

4.4. Conclusion

The GNN model's performance can be negatively affected by feature aggregation procedures that result in noisy features for nodes. This study addresses this issue by putting an adjusted GNN architecture to discover the ideal subset of features to perform better on the node classification task. For this, a general design space for GNNs has emerged based on the centrality-based feature selection and five different feature selection methods. The proposed model performance has been analyzed on homogeneous graph datasets, with an emphasis on homophily and heterophily features. Simulation results and comparison demonstrate through extensive trials that the proposed model outperforms feature selection approaches and GNN models and can reduce the effects of noisy features by choosing the best feature subset. The basic datasets utilized in our research consist of unweighted graphs with several thousand nodes. In addition, real data gathered from various applications can be depicted as a weighted graph. In a weighted graph, the edge can represent the strength of the relationship between nodes, this is a benefit in the case of a heterogeneous graph to take some features not all from this node to use in message representation where nodes in heterogeneous graphs often connect to the nodes of the opposite classes. In our future research, we will modify the feature aggregation step specific to the dataset to extend the explanation to real and weighted graphs, and we will use the proposed model to learn from informative features while avoiding noisy ones.

Conflicts of interest

There are no conflicts of interest.

References


