



Mixed Multi-channel Graph Convolution Network on Complex Relation Graph

Chengzong Li¹, Rui Zhai¹, Fang Zuo^{2,3}(✉), Junyang Yu¹, and Libo Zhang¹

¹ Henan Provincial Engineering Research Center of Intelligent Data Processing, Kaifeng, China

² Henan International Joint Laboratory of Intelligent Network Theory and Key Technology, Henan University, Kaifeng, China
zuoofang@henu.edu.cn

³ Subject Innovation and Intelligence Introduction Base of Henan Higher Educational Institution-Software Engineering Intelligent Information Processing Innovation and Intelligence Introduction, Base of Henan University, Kaifeng, China

Abstract. Graph Convolutional Networks (GCNs) have attracted increasing attention on network representation learning because they can generate node embeddings by aggregating and transforming information within node neighborhoods. However, the classification effect of GCNs is far from optimal on complex relationship graphs. The main reason is that classification rules on relationship graphs rely on node features rather than structure. Most of the existing GCNs do not preserve the feature similarity of the node pair, which may lead to the loss of critical information. To address these issues, this paper proposes a node classification network based on semi-supervised learning: Mif-GCN. Specifically, we propose a mixed graph convolution module to adaptively integrate the adjacency matrices of the initial graph and the feature graph in order to explore their hidden information. In addition, we use an attention mechanism to adaptively extract embeddings from the initial graph, feature graph, and mixed graph. The ultimate goal of the model is to extract relevant information for improving classification accuracy. We validate the effectiveness of Mif-GCN on multiple datasets, including paper citation networks and relational networks. The experimental results outperform the existing methods, which further explore the classification rules.

Keywords: Graph convolutional networks · Semi-supervised classification · Deep learning

1 Introduction

Graphs are common data structure that interpret the interrelationship of objects, and real-world data is increasingly represented as graph data. The purpose of graph analysis is to extract a large amount of hidden information from data, thus deriving many important tasks. The purpose of semi-supervised node classification task is to use a small number of labeled nodes to predict the labels of a large number of unlabeled nodes.

In recent years, graph neural networks have shown remarkable results in node representation learning, and GCN [9] being one of the most popular models. Existing GCNs

models often follow the message-passing manner to obtain the embedding of all nodes by aggregating the node features within the node neighborhood. However, recent studies have provided some new insights into the mechanisms of GCNs. Li et al. [5] prove that the convolution operation of GCNs is actually a kind of Laplacian smoothing, and explores the multi-layer structure. Wu et al. [6] show that graph structure as low-pass-type filter on node features. Zheng et al. [12] utilize the Louvain-variant algorithm and jump connection to obtain different granularity of node features and graph structure information for adequate feature propagation.

Most of the current GCNs rely on the structural information of graph, which can seriously affect the model performance when the structure of graph is not optimal. The reason may be that most of the current GCNs rely on the structural information of the graph to classify. In this work, we retain graph structure information and feature similarity, consider the influence of the mixed information of the two on the results, and introduce the attention mechanism for classification.

2 Related Work

The first representative work of applying deep learning models to graph structure data is network embedding, which learns fixed-length representations for each node by constraining the proximity of nodes, such as DeepWalk [1], Node2Vec [2]. After that, the research defines graph convolution in the spectral space based on convolution theorem. ChebNet [3] parameterize the convolution kernel in the spectral approach, which avoids the characteristic decomposition of the Laplace matrix and greatly reduces the complexity. Kipf [9] simplify the ChebNet parameters and proposes a first-order graph convolution neural network. SGC [4] put the first-order graph convolution neural introduces the first-order approximation of Chebyshev polynomials, which further simplifies the mechanism. With the increasing importance of attention mechanism, it has a considerable impact in various fields. GAT [7] leverage the attention mechanism to learn the relative weights between two connected nodes. GNNA [14] improve the embeddedness of users and projects by introducing attention mechanisms and high-level connectivity.

Some experiments show that the graph neural network is inefficient when the graph structure is noisy and incomplete. kNN-GCN [8] take the k-nearest neighbor graph created from the node features as input. DEMO-Net [13] formulate the feature aggregation into a multi-task learning problem according to nodes' degree values. AM-GCN [10] use consistency constraints and disparity constraints to obtain information and improves the ability to fuse network topology and node features. SimP-GCN [11] make use of self-supervised methods for node classification and explore the different effects of GCNs on assortative and disassortative graphs. Hence, it motivates us to develop a new approach that combines graph structure and feature similarity on the node classification task.

3 The Proposed Model

The key point is that Mif-GCN license node features to propagate in both structure space and feature space, because we consider that the classification rules of datasets depend on graph structure or feature similarity. We propose a mixed graph convolution

module to adaptively balance the mixed information from initial graph and feature graph, which may contain hidden information. This module is designed to explore depth-related information.

3.1 Multi-Channel Convolution Module

Feature Graph Construction. We first calculate the cosine similarity between the features of each node pair and construct the similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$. Then, we select predetermined number of nodes to establish the connection relationship, in order to obtain a k-nearest neighbor (kNN) graph $G_f = (\mathbf{A}_f, \mathbf{X})$.

The cosine similarity can be calculated from the cosine of the angle between two vectors:

$$S_{jk} = \frac{\mathbf{x}_j \cdot \mathbf{x}_k}{|\mathbf{x}_j| |\mathbf{x}_k|} \tag{1}$$

Where \mathbf{X}_j and \mathbf{X}_k are the feature vectors of nodes j and k Fig. 1.

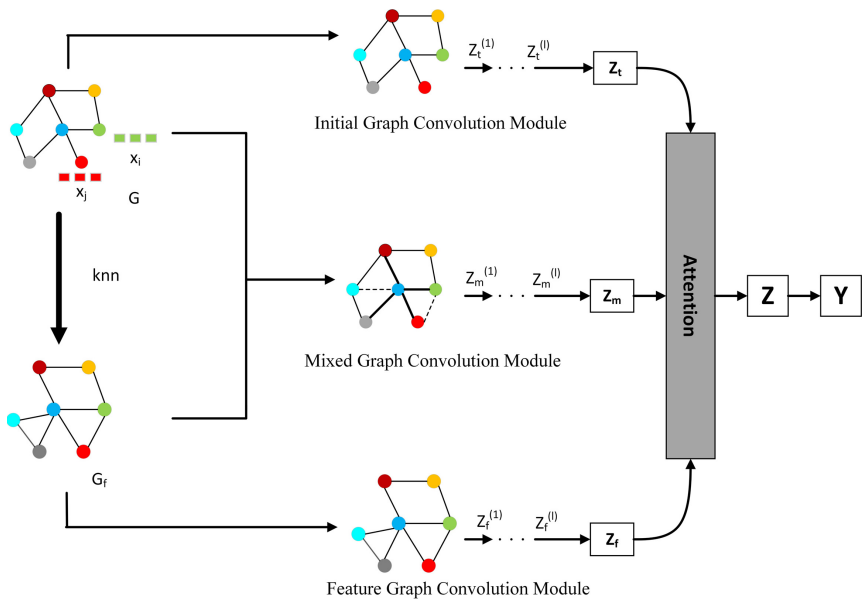


Fig. 1. The framework of the proposed Mif-GCN model.

Initial Graph Convolution Module and Feature Graph Convolution Module. In this module, node features are propagated in structure space and feature space separately so that we can distinguish correlations. Both modules use the typical GCN to get the node representation.

Then with the initial graph $G_i = (\mathbf{A}_i, \mathbf{X})$, the l -th layer output $\mathbf{Z}_i^{(l)}$ can be represented as:

$$\mathbf{Z}_i^{(l)} = \text{ReLU} \left(\tilde{\mathbf{D}}_i^{-\frac{1}{2}} \tilde{\mathbf{A}}_i \tilde{\mathbf{D}}_i^{-\frac{1}{2}} \mathbf{Z}_i^{(l-1)} \mathbf{W}_i^{(l)} \right) \quad (2)$$

Where $\mathbf{W}_i^{(l)}$ is the weight matrix of the l -th layer, $\tilde{\mathbf{A}}_i$ is the adjacency matrix of the initial graph of the added self-loop and $\tilde{\mathbf{D}}_i$ is the diagonal matrix of $\tilde{\mathbf{A}}_i$. We have $\tilde{\mathbf{A}}_i = \mathbf{A}_i + \mathbf{I}$, \mathbf{A}_i is the adjacency matrix of the initial graph, \mathbf{I} is the identity matrix. ReLU is the ReLU activation function and $\mathbf{Z}_i^{(0)} = \mathbf{X}$.

Then with the feature graph $G_f = (\mathbf{A}_f, \mathbf{X})$, the l -th layer output $\mathbf{Z}_f^{(l)}$ can be represented as:

$$\mathbf{Z}_f^{(l)} = \text{ReLU} \left(\tilde{\mathbf{D}}_f^{-\frac{1}{2}} \tilde{\mathbf{A}}_f \tilde{\mathbf{D}}_f^{-\frac{1}{2}} \mathbf{Z}_f^{(l-1)} \mathbf{W}_f^{(l)} \right) \quad (3)$$

Where \mathbf{A}_f is the adjacency matrix of the feature graph, $\mathbf{Z}_f^{(0)} = \mathbf{X}$, other parameters are the same as Eq. (2).

Mixed Graph Convolution Module. This module can mix information from the initial and feature graphs. We define a goal vector to balance the degree of contribution from the two graphs. Then, this module gets the node embedding.

Then with the initial graph and the feature graph, the l -th layer output of the mixed propagation matrix $\mathbf{P}^{(l)}$ can be represented as:

$$\mathbf{P}^{(l)} = \mathbf{g}^{(l)} * \tilde{\mathbf{D}}_i^{-\frac{1}{2}} \tilde{\mathbf{A}}_i \tilde{\mathbf{D}}_i^{-\frac{1}{2}} + (1 - \mathbf{g}^{(l)}) * \tilde{\mathbf{D}}_f^{-\frac{1}{2}} \tilde{\mathbf{A}}_f \tilde{\mathbf{D}}_f^{-\frac{1}{2}} \quad (4)$$

Where $\mathbf{g}^{(l)} \in \mathbb{R}^n$ is the goal vector that balances the information from the initial and feature graphs, $'*'$ denotes the operation of multiplying the j -th element of a vector with the j -th row of a matrix. $\mathbf{g}^{(l)}$ can combine information from two graphs to different degrees.

Then the l -th layer output of goal vector $\mathbf{g}^{(l)}$ can be represented as:

$$\mathbf{g}^{(l)} = \sigma \left(\mathbf{Z}_i^{(l-1)} \mathbf{W}_g^{(l)} + \mathbf{b}_g^{(l)} \right) \quad (5)$$

Where $\mathbf{Z}_i^{(l-1)} \in \mathbb{R}^{n \times d^{(l-1)}}$ is the input hidden representation of the previous layer, and $\mathbf{Z}_i^{(0)} = \mathbf{X}$. $\mathbf{W}_g^{(l)} \in \mathbb{R}^{d^{(l-1)} \times 1}$ and $\mathbf{b}_g^{(l)}$ are parameters in order to calculate $\mathbf{g}^{(l)}$. σ denotes the sigmoid activation function.

Then with the mixed propagation matrix $\mathbf{P}^{(l)}$, the l -th layer output $\mathbf{Z}_m^{(l)}$ can be represented as:

$$\mathbf{Z}_m^{(l)} = \sigma \left(\mathbf{P}^{(l)} \mathbf{Z}_m^{(l-1)} \mathbf{W}_m^{(l)} \right) \quad (6)$$

Where $\mathbf{P}^{(l)}$ denotes the mixed propagation matrix and $\mathbf{Z}_m^{(0)} = \mathbf{X}$, $\mathbf{W}_m^{(l)}$ is the parameter, here σ denotes the sigmoid activation function.

3.2 Attention Mechanism

We use the attention mechanism to combine to get the final node embedding. The ultimate goal is to explore the most relevant information in the three embeddings. $\mathbf{Z}_I, \mathbf{Z}_F, \mathbf{Z}_M$ contain the embedding of n nodes, $\alpha_i, \alpha_f, \alpha_m$ represent the respective attention values.

For node $j, \mathbf{z}_F^j \in \mathbb{R}^{1 \times h}$ denotes the embedding of node j in \mathbf{Z}_F . Then use a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{h \times 1}$ to obtain the attention values ω_F^j :

$$\omega_F^j = \vec{\mathbf{a}}^T \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_F^j) \quad (7)$$

Where $\mathbf{W} \in \mathbb{R}^{h' \times h}$ are parameters. Similarly, we get the attention values ω_I^j and ω_M^j for node j in \mathbf{Z}_I and \mathbf{Z}_M .

Then we use the *softmax* activation function to obtain α_F^j :

$$\alpha_F^j = \text{softmax}(\omega_F^j) = \frac{\exp(\omega_F^j)}{\exp(\omega_I^j) + \exp(\omega_F^j) + \exp(\omega_M^j)} \quad (8)$$

Where α_F^j is the percentage of embedding. Similarly, we can get α_I^j and α_M^j . For all n nodes, there are $\alpha_I = \text{diag}(\alpha_I^1, \alpha_I^2, \dots, \alpha_I^n), \alpha_F = \text{diag}(\alpha_F^1, \alpha_F^2, \dots, \alpha_F^n)$ and $\alpha_M = \text{diag}(\alpha_M^1, \alpha_M^2, \dots, \alpha_M^n)$.

Finally, we combine the three embeddings to obtain \mathbf{Z} :

$$\mathbf{Z} = (\alpha_I \cdot \mathbf{Z}_I + \alpha_F \cdot \mathbf{Z}_F + \alpha_M \cdot \mathbf{Z}_M) \quad (9)$$

3.3 Objective Function

Our model has shown good results without adding other loss functions. We use the cross-entropy error as our overall objective function.

Then with the final embedding \mathbf{Z} , We can get the class predictions for n nodes as $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = \text{softmax}(\mathbf{W} \cdot \mathbf{Z} + \mathbf{b}) \quad (10)$$

Where *softmax* is a commonly used activation function for classification.

If the training set is L , for each $l \in L$, \mathbf{Y}_l denotes the real label and $\hat{\mathbf{Y}}_l$ denotes the predicted label. Then, our classification loss can be expressed as:

$$\mathcal{L}_{class} = - \sum_{l \in L} \sum_{j=1}^C \mathbf{Y}_{lj} \ln \hat{\mathbf{Y}}_{lj} \quad (11)$$

4 Experiments

4.1 Experimental Settings

Datasets. We select a representative dataset and four complex relational datasets for our experiments, which include paper citation networks (Citeseer and ACM) and social networks (UAI2010, BlogCatalog and Flickr). The statistics of these datasets are shown in Table 1.

Table 1. Dataset statistics.

Datasets	Nodes	Edges	Classes	Features	Training	Test
Citeseer	3327	4732	6	3703	120	1000
ACM	3025	13128	3	1870	60	1000
UAI2010	3067	28311	19	4973	380	1000
BlogCatalog	5196	171743	6	8189	120	1000
Flickr	7575	239738	9	12047	180	1000

Parameters Setting. To validate our experiments, we choose 20 nodes per class as the training set and 1000 nodes as the test set. We set hidden layer units $nhid1 \in \{512, 768\}$ and $nhid2 \in \{32, 128, 256\}$. We set learning rate $0.0002 \sim 0.0005$, dropout rate is 0.5, weight decay $\in \{5e - 3, 5e - 4\}$ and $k \in \{2 \dots 8\}$. For other baseline methods, we refer to the default parameter settings in the authors’ implementation. For all experiments, we run 10 times and recorded the average results. And we use accuracy (ACC) and macro F1 score (F1) to evaluate the performance of the model.

4.2 Node Classification

In this subsection, we discuss the classification effects of each model on different datasets. The experimental results are shown in Table 2.

Table 2. Node classification accuracy (%). The best performance is highlighted in bold.

Datasets	Metrics	GCN	kNN-GCN	GAT	DEMO-Net	AM-GCN	If-GCN	Mif-GCN
Citeseer	ACC	70.50	61.40	72.40	69.40	73.10	71.48	72.96
	F1	67.62	58.96	68.34	67.42	68.42	67.16	68.64
ACM	ACC	87.95	78.53	87.25	83.29	90.40	90.28	90.70
	F1	87.87	78.14	87.38	83.17	90.43	90.20	90.64
UAI2010	ACC	50.12	66.09	58.64	25.41	70.10	69.24	70.98
	F1	32.90	50.45	41.48	19.82	55.61	55.30	56.13
BlogCatalog	ACC	69.48	76.54	64.32	61.09	81.98	81.82	83.00
	F1	68.53	75.59	63.43	58.79	81.36	81.20	82.38
Flickr	ACC	42.20	71.28	38.54	55.26	75.26	74.98	76.94
	F1	39.98	71.49	37.10	54.13	74.63	74.32	76.56

In the paper citation networks, the existing methods have achieved relatively excellent results, which shows the consistency of our method. In the social relationship networks, our method, kNN-GCN and AM-GCN have all achieved good results. This phenomenon

may be that they retain the feature similarity between nodes. Other models showed varying degrees of fluctuation and even failure. The reason may be that they depend on graph structure for classification. The importance of node similarity for GCN is further confirmed.

The experimental results show that our model Mif-GCN outperforms other comparison baselines. Our model removes the mixed graph convolution module as If-GCN. Comparing If-GCN and Mif-GCN, the results show the effectiveness of our model. We visualize the attention trend graph. Among the three types of social networks, the feature graph convolution module and the mixed graph convolution module are more valued. They make the classification results biased toward the favorable side.

5 Conclusion

We propose Mif-GCN, a framework that simultaneously preserves graph structure and features similarity. We consider classification rules on different datasets including simple and complex graphs, our method is more generally applicable to networks with complex node relationships.

Acknowledgement. This work is supported by the Henan Province Science and Technology R&D Project(212102210078); the Major Public Welfare Project of Henan Province(201300210400); Key Scientific Research Project of Universities in Henan Province under Grant No.19A520016; Henan Higher Education Teaching Reform Research and Practice Project(Graduate Education) under Grant No.2019SJGLX080Y.

References

1. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
2. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint [arXiv:1606.09375](https://arxiv.org/abs/1606.09375) (2016)
4. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: graph neural networks meet personalized pagerank. arXiv preprint [arXiv:1810.05997](https://arxiv.org/abs/1810.05997) (2018)
5. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
6. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning, pp. 6861–6871. PMLR (2019)
7. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
8. Franceschi, L., Niepert, M., Pontil, M., He, X.: Learning discrete structures for graph neural networks. In: International Conference on Machine Learning, pp. 1972–1982. PMLR (2019)

9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
10. Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., Pei, J.: Am-gcn: adaptive multi-channel graph convolutional networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1243–1253 (2020)
11. Jin, W., Derr, T., Wang, Y., Ma, Y., Liu, Z., Tang, J.: Node Similarity Preserving Graph Convolutional Networks. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 148–156 (2021)
12. Zheng, W., Qian, F., Zhao, S., Zhang, Y.: Multi-granularity graph wavelet neural networks for semi-supervised node classification, *Neurocomputing* (2020)
13. Wu, J., He, J., Xu, J.: Net: degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 406–415 (2019)
14. Guo, Y., Yan, Z.: Collaborative filtering: graph neural network with attention. In: Wang, G., Lin, X., Hendler, J., Song, W., Xu, Z., Liu, G. (eds.) WISA 2020. LNCS, vol. 12432, pp. 428–438. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60029-7_39