HL-Pretrained Filterbank GNNs

Wenhan Yang

June 11, 2022

Abstract

Graph Neural Network has achieved tremendeous success in recent years. Over all kinds of downstream tasks including node classifications, link predictions and graph classifications, GNNs, with Graph Convolutional Network being one of its most classic models, are able to achieve high accuracy on many real world datasets. However, using the aggregation scheme as its main tool, the current methods, though performing phenomenally in smooth datasets, fail to reach the same level of progress in hetergeneous datasets like molecule graphs and webpage datasets. There are many methods being proposed to solve such problems, with some of them focusing on alternating message passing filters,and the others on alternating the entire model structure. In this paper, we propose a new model based on the existing work which, via renormalized Laplacian matrix, completes the information lost during the message passing phase. Our work modifies the method and adds additional contrastive pre-training layers, so that the updated model can work in an semi-supervised learning scenario, which is more applicable in the real life.

1 Introduction

Graph is a powerful data structure that can include rich information in a variety of real-world applications. Its ability to encode complex relations in different data like social networks, biological compound structures, and citation networks help it gain popularity in different domains. However, it has always been a challenge to work with the graph data, which requires meaningful encoded representations of node features and edge relationships. Its atypical representation space in the non-Euclidean space also makes it difficult to transfer the success of neural networks directly to graphs. With the arise of Convolutional Neural Networks (CNNs) on images[5], Graph Neural Networks (GNNs)[4] are proposed in the graph domain. Different from their predecessors like DeepWalk[9] and node2vec[2] which can only learn shallow embedding from the data, they are able to recursively learn both the graph structures and the nodes' features via the information passed from every node's neighborhood, thus extracting a more complex meaningful representation from the data. However, GNNs' success is not without its limitations. Although their performances sets the state of the art for the smooth datasets like social network and citation network, the accuracy suffers for the datasets without such homogeneous nature. Across datasets of different homophily ratios, an indicator proposed in [17] that measures the smoothness of the graph data, GNNs' performances flactuate greatly. The reason behind this is that general GNN models, with GCN as its predecessors, rely heavily on the idea of homophily principle: to propagate the features across each node's neighbors in an aggregation manner. The essential reason behind such a instability is the use of aggregative-styled filters for message passing. GCN, for example, uses the normalized adjancency matrix solely to communicate among neighbors during the training, which limits its performance because of the information the process discards due to its incompleteness. To resolve the isssue, a two-channel message passing method is proposed, and in addition to the normalized adjancency matrix used in many classic GNN models, normalized laplacian matrix, which mathematically complements the former filter, is also used as one of the filtering channels[6]. The method surpasses state-of-the-art performance in heterogeneous datasets and is on par with popular GNN models in the homogeneous datasets. However, the two-channels structure is difficult to balance and the accuracy drops significantly when less labels are available, which is often an issue in the realistic settings. In fact, label availability has always been an concern for the GNNs, which relys heavily on the abundant labels and are mostly established in a supervised learning setting [19]. To make the models more adaptive and applicable to the real world, the models need to find a way to utilize unlabeled data as well, which are typically a less expensive training sources comparing to

Model	h = 0.1	h = 0.7
GCN	37.14	84.52
GAT	33.11	84.03
GCN-Cheby	64.10	84.92
GraphSAGE	72.89	85.06
MixHop	58.93	84.43
MLP	74.85	71.72

Table 1: Classification accuracy on datasets with different homophily ratio.As measured in [17], popular GNN methods do not generalize well in data with low homophily ratio.

the labeled data. In the computer vision domain, pseudo-label is a popular method used when labels are scarce. Specifically, the FixMatch method proposed for semi-supervised image classifcation tasks innovatively use weak and strong augmentations, and by comparing classification model's decisions on differently augmented images, train the original model[11]. By treating the weakly augmented images as the more trusting labels, FixMatch, with this contrastive learning structure is able to surpass the state-of-the-art methods with less labels. This method is soon transferred to the graph domain, with the similar goal of reducing the labels needed for the training while not losing too much accuracy scores[16] [19]. Our method also uses the contrastive learning structure, but not like its predecessors which use mostly simple augmentations on the graph data, our model uses the two-channels, which aggregate and diversify the information separately, as the two augmentaion channels, and by maximizing the mutual information between the output representations of the two channels, we seek a balance point for the model's training channels in the pre-training scheme in a self-supervised leaning manner. Our method is able to achieve the state-of-art performances on both homogeneous and heterogeneous datasets without using excessive amount of labeled data.

2 Related Work

2.1 Homophily Ratio

In this paper, we use the notation of homophily ratio defined in [17]. It measures the graph homophily level, or, the likelihood of the linked nodes belonging to the same classes or having similar features [7]. Mathematically, it is defined as the fraction of edges in a graph that connect nodes that have the same class labels. The homophily ratio is ranged from 0 to 1, and a larger homophily ratio indicates a stronger homophpily in the graph.

2.2 Contrastive learning

Firsly proposed and used in the visual representation learning domain, the contrastive method uses positive and negative samples to learn discriminative representations. While in the image domain, models use distortion, cropping, rotations, and other simple augmentation methods to generate different postive and negative samples, in the graph domain, typical methods used include node dropping, edge perturbation, attribute masking, and subgraph [16]

3 Methods

3.1 Preliminaries

Denote an undirected graph of dimension $m \times n$ as $G = (V, E)$, with V and E represent the node set, $\{v_1, v_2, \ldots, v_m\}$, and the edge set, $V \times V$, respectively. Denote its symmetric Adjancency matrix as A. $A_{ij} = 1$ if and only if $(v_i, v_j) \in E$, and equals to 0 otherwise. Denote the Feature matrix as X, with $x_i \in R^m$ represents the feautre vector of the i_{th} node. D is the Degree matrix of the graph, with $D_{ii} = \sum_j A_{ij}$, and L is the Laplacian matrix of the graph, defined as $L = D - A$. The normalized version of the Laplacian matrix is denoted as $L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, and the normalized Adjancency matrix is defined in a similar fashion: $A_{sym} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ In the paper, we use the renormalized

version of the Adjancency matrix $\hat{A}_{sym}$ as introduced in (GCN PAPER), where $\tilde{A} = A+I$, $\tilde{D} = D+I$, and $\hat{A}_{sym} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$. The renormalized Laplacian matrix is defined as $\hat{L}_{sym} = I - \hat{A}_{sym}$.

3.2 Background

As introduced in [6], to adapet to the different discrepancies of smoothness between the input features and the labels, we utilize high-pass (L_{HP}) and low-pass (L_{LP}) filter to learn both the smooth and non-smooth components of the graph simultaneously. Generally, the Laplacian matrices can be regarded as L_{HP}, which serves to filter out the smooth components, and Adjancency matrix as L_{LP} (Maehara 2019), aiming at filtering out non-smooth component. Applying filtering operation is a linear projection that projects the feature matrix into a fixed subspace. Since $L_{HP} + L_{LP} = I$, we do not lose the expressive power of the features and both the smooth and nonsmooth components are preserved and convoluted separately. To contrust the downstream learning framework, we define the message passing procedure as such:

$$\begin{aligned} H_L^l &= f(L_{LP}H^{l-1}W_L^{l-1}), \quad H_H^l = f(L_{HP}H^{l-1}W_H^{l-1}) \\ H^l &= \alpha_L^l \cdot H_L^l + \alpha_H^l \cdot H_H^l, l = 1, \ldots, n \end{aligned} \tag{1}$$

where $H^0 = X$. f is an activation function. Note that different from the original proposed framework, we put the filtering matrix inside the activation function, so that the low-pass channel aligns with the typicall GCN message passing. $W_L^{l-1}, W_H^{l-1} \in R^{M_l \times M_{l-1}}$ are learnable parameter matrices for the high-pass and low-pass channels. $\alpha_L^l, \alpha_H^l \in [0, 1]$ are learnable scalar parameters that aim at keeping the balance between the smooth and non-smooth components learned by H^l.

3.3 HL-Pretrained Filterbank GNNs

3.3.1 Motivation

The originally proposed filtre bank assisted GNNs achieved state-of-the-art performance across homogeneous and heterogeneous datasets under the supervised learning. However, when the labels are scarce, the two-channel filters fail to extract balanced smooth and non-smooth components in the final representation, and the preformance becomes unstable and highly depends on the initialization of the α_L^l, α_H^l. This paper aims at adapting the method in the semi-supervised learning, which is a more realistic setting given the high cost and technical difficulties to obtain labels for all the nodes in large and specialized datasets.

3.3.2 Contrastive Learning Framework

The proposed HL pretraining framework follows the common graph CL paradigm where the model seeks to maximize the agreement of representations between different views. We first generate two graph views by performing high-pass and low-pass filtering on the input. Comparing to the typical stochastic graph augmentation, we treat L_H and L_L as innovated learnable augmentation matrix that diverify and aggregate the graph representations respectively. Then, we employ a contrastive objective that enforces the encoded embeddings of each node in the two different views to agree with each other and can be discriminated from embedding of other nodes. Intuitively, by enforcing the agreeement between aggregated and divrsified views, we check on the relative strength of the filtering operation and balance the smooth and non-smooth components free from the influence of label scarcity (the pretraining scheme is under unsupervised learning setting). We denote the two views as $U^l = f(H_L^l, \tilde{A})$ and $V^l = f(H_H^l, \tilde{A})$, where H_L^l, H_H^l are the low-pass filtered and high-pass filtered representations respectively. For any node, its embedding generated in one view, u_i^l, is treated as the anchor, and the one generated in the other view, v_i^l, form the positive sample. The other embeddings in the two views are treated as the negative samples. For the objective contrastive loss function, we use InfoNCE loss proposed in [12]. Mathematically, it is defined as

$$l(u_i, v_i) = log \frac{e^{\theta(u_i, v_i)/\tau}}{e^{\theta(u_i, v_i)/\tau} + \sum_{k \neq i} e^{\theta(u_i, v_k)/\tau} + \sum_{k \neq i} e^{\theta(u_i, u_k)/\tau}} \tag{2}$$

The $\theta(u, v)$ is defined as the cosine similarity between $f(u)$ and $f(v)$, where $f(\cdot)$ is a nonlinear projection function too enhance the expression power. In the model, it is implemented as a two-layer

Datasets	Cora	CiteSeer	Chameleon	Squirrel	Texas
Hom.ratio	0.81	0.74	0.23	0.22	0.11
Nodes	2708	3327	2277	5201	183
Edges	5278	4676	31421	198493	295
Classes	6	7	5	5	5

Table 2: Datasets used in the paper

Datasets	Cora	CiteSeer	Chameleon	Squirrel	Texas
HL-FBGCN	**86.87**	**78.67**	**63.86**	**49.63**	70.12
CPGNN	83.64	72.06	56.93	37.03	70.42
H2GCN	83.43	71.76	48.12	29.50	**71.39**
GraphSAGE	81.60	71.74	45.45	34.35	67.36
GCN-Cheby	83.29	72.04	36.66	26.52	58.96
MixHop	85.34	73.23	46.84	36.42	62.15
GCN	83.56	72.27	52.00	33.31	55.90
GAT	79.57	72.63	50.54	31.20	55.83
MLP	64.81	66.52	37.36	25.50	64.65

Table 3: Accuracy on graphs comparing to other models

perceptron model. $\sum_{k\neq i} e^{\theta(u_i,v_k)}$ represents the inter-view negative pairs, a.k.a between view U and view V, and $\sum_{k\neq i} e^{\theta(u_i,u_k)}$ represents the intra-view negative pairs, a.k.a within the view V.Since two views are symmetric, the loss for another view, $l(v_i, u_i)$, is defined in a similar fashion. The overall objective to be maximized is then defined as the average over all positive pairs. Mathematically, it is denotes as

$$J = \frac{1}{2N}\sum_{i=1}^{N}[l(u_i, v_i) + l(v_i, u_i)] \tag{3}$$

4 Experiment

In our method, we mostly combine our filter bank channels with the popular GCN [4] structure. We design our experiments to evaluate the performances of the proposed model on graphs with different levels of homophily ratios.

4.1 Datasets

Citation Network: The citation network datasets "Cora" and "CiteSeer" from [15]. Nodes represent papers and edges represent the citation relationship between different papers.
Wikipedia Network: The Wikipideia network datasets "Chameleon" and "Squirrel" from [10]. Nodes represent websites and edges represent the hyperlink between different websites.
WebKB Network: The webpage datasets "Texas" from [8]. Collected from the various unverisities' CS departments' webpages. Nodes represent websites and edges represent the hyperlink between different websites.

4.2 Experimental Setup

For the specific data split, we use the same setup as used in [18]:we generate 10 random splits, with 10% of the data for training, 10% for validation, and 80% for testing. We report the average classificsation accuracy as the final performance score of each model. We compare our methods to other popular GNN models. Our baselines include MLP, GCN[4], GAT[13], GCN-Cheby[4], GraphSAGE[3], MixHop[1], and newly proposed methods like H2GCN [17] and CPGNN [18].

4.3 Summary

As shown in the table, our method surpasses other baselines models' performances on most of the heterogeneous and homogeneous datasets.

5 Future Work

According to the study conducted by][14], there is no single "best" filters for all the datasets. Thus, even though our method achieves high performances on different datasets, there are still improving spaces if we can use a variety of filters or adaptive filters like proposed in the paper mentioned above to modify our channel structures. In addition, more datasets and more rigorous theoretical justifications are needed to further confirm our results.

6 Conclusion

We propose HL-FBGCN, a model modifed on the existing work that aims at refining GNNs' performances on classifications tasks across graph datasets of different homophily levels. Our model innovatively adds the contrastive pretraining structure that utilizes the two-channel methods as the augmentation techniques. The resulting method are able to achieve the state-of-the-art performance with less labeled samples.

References

[1] Sami Abu-El-Haija et al. "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing". In: *international conference on machine learning*. PMLR. 2019, pp. 21–29.

[2] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[4] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[6] Sitao Luan et al. "Complete the missing half: Augmenting aggregation filtering with diversification for graph convolutional networks". In: *arXiv preprint arXiv:2008.08844* (2020).

[7] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[8] Hongbin Pei et al. "Geom-gcn: Geometric graph convolutional networks". In: *arXiv preprint arXiv:2002.05287* (2020).

[9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[10] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. "Multi-scale attributed node embedding". In: *Journal of Complex Networks* 9.2 (2021), cnab014.

[11] Kihyuk Sohn et al. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 596–608.

[12] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv e-prints* (2018), arXiv–1807.

[13] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[14] Yewen Wang et al. "Demystifying graph neural network via graph filter assessment". In: (2019).

[15] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. "Revisiting semi-supervised learning with graph embeddings". In: *International conference on machine learning*. PMLR. 2016, pp. 40–48.

[16] Yuning You et al. "Graph contrastive learning with augmentations". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5812–5823.

[17] Jiong Zhu et al. “Beyond homophily in graph neural networks: Current limitations and effective designs”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7793–7804.

[18] Jiong Zhu et al. “Graph neural networks with heterophily”. In: *arXiv preprint arXiv:2009.13566* (2020), pp. 11168–11176.

[19] Yanqiao Zhu et al. “Deep graph contrastive representation learning”. In: *arXiv preprint arXiv:2006.04131* (2020).