Powerful Graph Convolutioal Networks with Adaptive Propagation Mechanism for Homophily and Heterophily

Tao Wang[1,†], **Rui Wang**[1,†], **Di Jin**[1], **Dongxiao He**[1,*], **Yuxiao Huang**[2]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
[2] Data Scienc, George Washington University, Washington, D.C., USA
{2019216113, wr1895, jindi, hedongxiao}@tju.edu.cn, yuxiaohuang@gwu.edu

Abstract

Graph Convolutional Networks (GCNs) have been widely applied in various fields due to their significant power on processing graph-structured data. Typical GCN and its variants work under a homophily assumption (i.e., nodes with same class are prone to connect to each other), while ignoring the heterophily which exists in many real-world networks (i.e., nodes with different classes tend to form edges). Existing methods deal with heterophily by mainly aggregating higher-order neighborhoods or combing the immediate representations, which leads to noise and irrelevant information in the result. But these methods did not change the propagation mechanism which works under homophily assumption (that is a fundamental part of GCNs). This makes it difficult to distinguish the representation of nodes from different classes. To address this problem, in this paper we design a novel propagation mechanism, which can automatically change the propagation and aggregation process according to homophily or heterophily between node pairs. To adaptively learn the propagation process, we introduce two measurements of homophily degree between node pairs, which is learned based on topological and attribute information, respectively. Then we incorporate the learnable homophily degree into the graph convolution framework, which is trained in an end-to-end schema, enabling it to go beyond the assumption of homophily. More importantly, we theoretically prove that our model can constrain the similarity of representations between nodes according to their homophily degree. Experiments on seven real-world datasets demonstrate that this new approach outperforms the state-of-the-art methods under heterophily or low homophily, and gains competitive performance under homophily.

Introduction

Networks (such as social networks, citation networks, biological networks and traffic networks) are ubiquitous structures that can model relational data. Network analysis (Wang et al. 2016, 2019; Cao et al. 2021; Jin et al. 2021b; He et al. 2021) has been a hot research topic for decades and has been widely used in many scientific fields such as computer science, social science, biology and physics (Li and Goldwasser 2019; Yan et al. 2019). Recently, graph convolutional network (GCN) (Kipf and Welling 2017), which exhibits significant power on processing graph-structured data, has gained great success and already been adapted in various network analysis tasks, including node classification, community detection, anomaly detection and recommender system (Tang, Aggarwal, and Liu 2016; Tian et al. 2014; Gao, Denoyer, and Gallinari 2011; Gong et al. 2019; Yu and Han 2014; Jin et al. 2021a).

Although many GCN-based methods have been proposed in recent years, such as GraphSage (Hamilton, Ying, and Leskovec 2017), GAT (Velickovic et al. 2018), MixHop (Abu-El-Haija et al. 2019), and HIN (Xu et al. 2019), they implicitly assume that most connected nodes are from the same class or with similar attributes, which is typically called the homophily of network structure. GCN-based methods reflect the homophily assumption by feature propagation and aggregation within graph neighborhoods. And these methods have shown satisfactory performance in many downstream tasks on network with homophily. However, in real world, there are also many networks where most connected nodes are from different classes, which is called heterophily or low homophily. For example, in dating networks, most people tend to connect with people of the opposite gender. In protein networks, different types of amino are more likely to connect with each other. Under heterophily or low homophily, these methods suffer from poor performance, as the propagation mechanism within graph neighborhoods, which is the most fundamental part of GCN, is problematic and will mix irrelevant information from different classes. Therefore, existing GCN-based methods cannot adapt to the scenario of heterophily or low homophily.

Recently, some efforts have been dedicated to generalizing GCN to heterophilic networks. For example, Geom-GCN (Pei et al. 2020) proposes a novel geometric aggregation scheme which aggregates immediate neighborhoods and distant nodes that have a certain similarity with the target node in a continuous space. H2GCN (Zhu et al. 2020) applies some key designs, such as higher-order neighborhoods aggregation and combination of intermediate representations, to boost learning from graph with heterophily. GPR-GNN (Chien et al. 2021) deals with heterophily and oversmoothing by combining each step of feature propagation with a learnable weight. GGCN (Yan et al. 2021) allows negative message propagation between graph neighborhoods based on the similarity of representations in order

[†]Both authors contributed equally to this research
[*]Corresponding author

to decouple the heterophily and oversmoothing problems. CPGNN (Zhu et al. 2021) incorporates an interpretable compatibility matrix for modeling the heterophily or homophily level in graphs, enabling it to go beyond the assumption of strong homophily. However, these methods suffer from serious problems. They coarsely aggregate higher-order (distant neighbors), or combine the intermediate representations, to deal with heterophily. While doing so can fuse effective information to some extent, it will also result in introducing noise and irrelevant information that influence negatively the prediction of downstream tasks. Most importantly, these methods do not change the propagation mechanism which is the essential part of GCN and is problematic under heterophily. This also make the representation of nodes from different classes mixed and indistinguishable.

To solve this problem, we focus on designing an adaptive propagation mechanism for both heterophilic and homophilic networks, and giving a new **HO**mophily-**G**uided **G**raph **C**onvolutional **N**etwork called HOG-GCN. In this new approach, we introduce a homophily degree matrix into the graph convolution framework, which is used to model the homophily and heterophily of networks and further conduct the propagation process. This homophily degree matrix can be learned from the attribute and topology information via extracting class-aware information during the propagation process. As a result, the new graph convolution framework can automatically change the feature propagation process via modeling the homophily degree between node pairs using homophily degree matrix. The learning process of homophily degree matrix and the feature propagation process are trained jointly in an end-to-end fashion. Finally, we theoretically prove that the adaptive propagation process can constrain the similarity of representations between nodes according to homophily degree.

Preliminaries

We first give the notations and problem descriptions, then give the definition of homophily ratio.

Notations and Problem Descriptions

Given an undirected, unweighted and attributed network $G = (V, E, X)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of n nodes, $E = \{e_{ij}\} \subseteq V \times V$ is a set of edges, and $X \in \mathbb{R}^{n \times f}$ is a set of node attributes, where m represents the number of attributes. The i-th row of X represents the attributes of node v_i. The topological structure of G is represented by an adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, where $a_{ij} = 1$ if nodes v_i and v_j are connected, or $a_{ij} = 0$ otherwise. We focus on the semi-supervised node classification task in this paper. This is, assume each node belongs to one out of C classes and we have known the labels of a set of nodes V_L with $|V_L| \ll n$. Each node $v_i \in V_L$ is assigned to a label $y_i \in L = \{1, 2, ...C\}$. The objective of node classification task is to predict the labels of $V \backslash V_L$.

Definition 1. **Homophily Ratio.** Given a network $G = (V, E, X)$, the homophily ratio $h = \frac{|\{(u,v):(u,v) \in E \wedge y_u = y_v\}|}{|E|}$ is the fraction of edges which connect nodes that have the same class, i.e., intra-class edges.

The homophily ratio h measures the overall homophily level in the graph and thus we have $h \in [0, 1]$. To be specific, graphs with h closer to 1 tend to have more edges connecting nodes within the same class, or say stronger homophily; on the other hand, graphs with h closer to 0 tend to have more edges connecting nodes in different classes, or say a stronger heterophily.

The Framework

We first give a brief overview of our approach, and then introduce the proposed new method in specific including detailed descriptions of each component.

Overview

To let the propagation mechanism of GCN essentially suitable for both homophily and heterophily, we propose a novel homophily-guided graph convolution framework that can automatically learn the propagation process according to the homophily degree between node pairs, which is called HOG-GCN. In specific, we incorporate a homophily degree matrix into the graph convolution framework for modeling the homophily and heterophily and further use it to adaptively change the propagation process between neighborhoods. The whole structure of the proposed approach is shown in Fig. 1, which consists of two components: homophily degree matrix estimation and homophily-guided propagation. The homophily degree matrix is learned from the attribute and topology information during the propagation process, which is further used to conduct the feature propagation between neighborhoods. In return, the propagation process can help learn better homophily degree matrix through downstream semi-supervised task. Therefore, these two components can be enhanced by each other and trained jointly. For the homophily degree matrix, we extract class-aware information from node attributes and network topology, respectively, and then define the whole homophily degree based on these two types of information. From the perspective of attributes, we leverage multi-layer perceptron to extract class-aware information, since node attributes are not constrained by heterophily. Then, the homophily degree in attribute space can be further calculated. From the perspective of network topology, since it exhibits different degrees of heterophily, we consider making full use of the available label information to capture homophily degree in topology space. To this end, we propose a generalized label propagation technique with a learnable weight matrix, which can reflect homophily degree between node pairs in topology space. The intuition is that the influence between intra-class labels is greater than that between inter-class labels. For the homophily-guided propagation, we introduce the homophily degree matrix into the propagation process which can reveal the underlying distribution of homophily or heterophily in networks, enabling the framework to adaptively change the feature propagation weights according to homophily degree between neighborhoods.

Homophily Degree Matrix Estimation

To adaptively change the propagation process for homophily and heterophily, we learn the homophily degree between

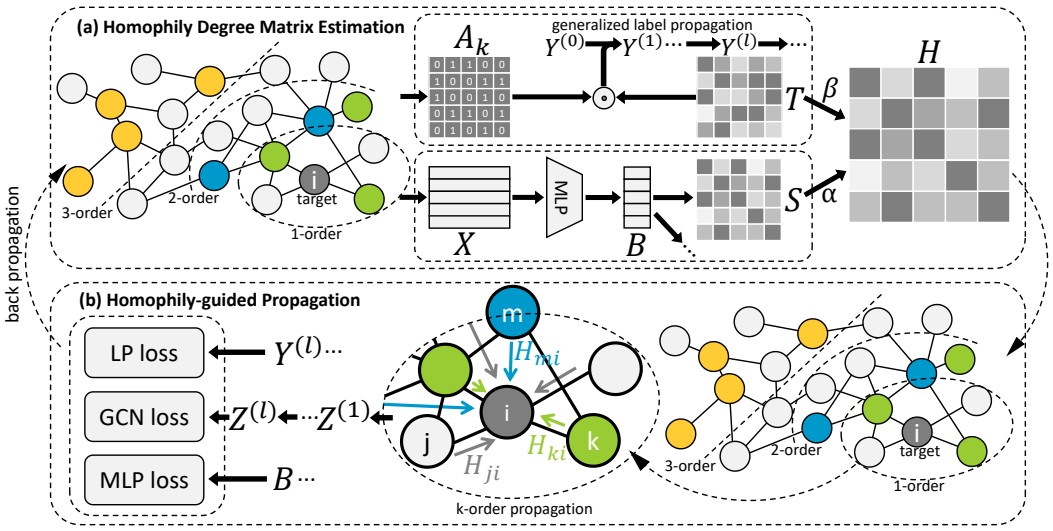


Figure 1: The structure of HOG-GCN, which consists of two components, including (a) homophily degree matrix estimation and (b) homophily-guided propagation.

node pairs during the propagation process. The homophily degree describes the extent to which two nodes belong to the same class. However, it is difficult to calculate the homophily degree directly from node labels, since only part of label information is available under the semi-supervised task. To this end, we consider estimating the homophily degree between node pairs from attribute space and topology space, respectively, and then combine them with adjustable parameters.

On one hand, from the perspective of attribute space, we apply graph-agnostic multi-layer perceptron (MLP) to extract class-aware information from original node attributes. The l-th layer of MLP is defined as:

$$Z_m^{(l)} = \sigma(Z_m^{(l-1)} W_m^{(l)}) \tag{1}$$

where $Z_m^{(0)} = X$, $W_m^{(l)}$ is the learnable weight matrix for MLP, and σ the activation function. Denote the output of the final layer of MLP as Z_m, then we can obtain the soft assignment matrix $B \in \mathbb{R}^{n \times C}$ as follows :

$$B = \text{softmax}(Z_m) \tag{2}$$

Each element B_{ic} denotes the probability that node v_i belongs to class c. Let all parameters of MLP be Θ_m , then the optimal Θ_m^* is obtained by minimizing the following loss of predicted labels by MLP:

$$\Theta_m^* = \underset{\Theta_m}{\text{argmin}}\, \mathcal{L}_{mlp} = \underset{\Theta_m}{\text{argmin}}\, \frac{1}{|V_L|} \sum_{v_a \in V_L} J(\hat{b}_a^{mlp}, y_a) \tag{3}$$

where $\hat{b}_a^{mlp}$ is the predicted labels of v_a by MLP. Since soft assignment matrix B is learned under the guidance of semi-supervised classification, it can capture class-aware information in attribute space. Then based on the matrix B we can calculate the extent that two nodes belong to the same class, called homophily degree matrix, which is defined as:

$$S = BB^T \tag{4}$$

where $S_{ij} = b_i b_j^T$ denotes the extent to which node v_i and node v_j belong to the same class.

It is worth noting that the homophily degree matrix S is estimated based on original node attributes, which are not constrained by the heterophily of networks. Therefore, it also holds in networks with heterophily or low homophily.

On the other hand, from the perspective of topology space, network topology contains many useful information even if it exhibits strong heterophily. However, it is also difficult to estimate the homophily degree directly based on the topology, as we may not know the distribution of heterophily of network in advance. To address this problem, we further apply label propagation technique to estimate the homophily degree matrix in topology space.

The classic label propagation typically assumes that two connected nodes are more likely to have the same class, and thus propagates labels iteratively between neighborhoods. Let $Y^{(l)} = [y_1^{(l)}, y_2^{(l)}, ..., y_n^{(l)}]^T \in \mathbb{R}^{n \times C}$ denotes the soft label matrix in iteration $l > 0$, where $y_i^{(l)}$ represents predicted label distribution for node v_i in iteration l. When $l = 0$, the initial label matrix $Y^{(0)}$ is initialized by the labels of training data, i.e., $Y^{(0)} = [y_1^{(0)}, y_2^{(0)}, ..., y_n^{(0)}]^T$, which consists of one-hot label indicator vectors $y_i^{(0)}$ for $v_i \in V_L$ or zero vectors otherwise (i.e., unlabeled nodes). Then the label propagation in iteration l is defined as follows:

$$\begin{aligned} Y^{(l)} &= D^{-1} A Y^{(l-1)}, \\ y_i^{(l)} &= y_i^{(0)}, \forall v_i \in V_L \end{aligned} \tag{5}$$

where D is the degree matrix with entries $D_{ii} = \sum_j A_{ij}$. In this equation, all nodes propagate labels to their direct

neighborhoods first, and then the labels of all labeled nodes are reset to their initial labels.

However, classic label propagation technique aims to capture the assumption of homophily, which can not adapt to networks with heterophily directly. To capture the homophily degree between node pairs, we generalize classic label propagation with a learnable weight matrix, which is trained under the guidance of labeled data. The key intuition is that the influence between intra-class label is greater than that between inter-class label. Thus the learned weight matrix can be used to represent the extent to which two nodes belong to same class. Since networks exhibit different degrees of heterophily, we perform label propagation over k-order structure of network to capture more homophilic nodes (e.g., $k = 2$). The k-order structure is defined as:

$$A_k = A + A^2 + ... + A^k \tag{6}$$

Then the generalized label propagation in iteration l is defined as:

$$Y^{(l)} = D_k^{-1}(A_k \odot T)Y^{(l-1)} \tag{7}$$

where D_k^{-1} is the diagonal degree matrix for matrix $A_k \odot T$. In the equation above, all nodes propagate labels to their k-order neighborhoods according the learnable edge weights. Then we can learn the optimal edge weights T^* by minimizing the loss of predicted labels by generalized label propagation:

$$T^* = \underset{T}{\operatorname{argmin}}\, \mathcal{L}_{lp} = \underset{T}{\operatorname{argmin}} \frac{1}{|V_L|} \sum_{v_a \in V_L} J(\hat{y}_a^{lp}, y_a) \tag{8}$$

where J is the cross-entropy loss, $\hat{y}_a^{lp}$ and y_a are the predicted label distribution of v_a by generalized label propagation and the true one-hot label of v_a, respectively. Note that we do not add to or remove edges from graphs, but only learn the weights of existing k-order neighborhoods. The optimal T^* maximizes the probability that each node is correctly classified by generalized label propagation, thus also increases the intra-class influence. This reflects the extent to which two nodes belongs to the same class. In order to form an end-to-end pattern, here we take the weight matrix T as the homophily degree matrix estimated from topology space.

At last, to make the model more effective and robust, we combine the hompily degree matrix estimated from attribute space and topology space with adjustable parameters as follows:

$$H = \alpha S + \beta T \tag{9}$$

where α and β are hyper-parameters. Also of note, matrix S contains homophily degree between any node pairs while T learns homophily degree within k-order neighborhoods. It does not influence the performance, as we will filter the entries that do not involve in propagation process.

Homophily-guided Propagation

The *core* of our approach is adaptively learning the propagation process for homophily and heterophily. The homophily degree matrix which represents the extent to which two nodes belong to the same class can reflect the underlying distribution of homophily or heterophily of networks. Therefore, we incorporate the learnable homophily degree matrix into graph convolution framework to automatically change the propagation weights between neighborhoods according to homophily degree. To be specific, during the propagation process, we aim to increase the feature influence between underlying intra-class nodes and reducing feature influence between underlying inter-class nodes according to the homophily degree matrix. To achieve this, we assign different weights to different neighborhoods according to the homophily degree, which can distinguish the homophily or heterophily between neighborhoods. Similarly, as we may not know the underlying distribution of heterophily in advance, we perform feature propagation over the k-order neighborhoods, to capture more homophilic nodes. In addition, we deal with ego-representation and neighborhood-representation separately to preserve more personalized information. Then, the feature propagation process of the proposed method HOG-GCN in iteration l is given by:

$$Z^{(l)} = \sigma(\mu Z^{(l-1)} W_e^{(l)} + \xi \hat{D}^{-1} A_k \odot H Z^{(l-1)} W_n^{(l)}) \tag{10}$$

where μ and ξ denote the weights of ego-representation and neighborhood-representation respectively, $\hat{D}$ is the diagonal degree matrix, $Z^{(0)} = X$ is the original node attributes. σ is the activation function.

Also of note, our proposed graph convolution framework can be taken as learning proper attention weights. The most significant difference between our approach and those feature-based attention methods is that, the attention weights are learned based on feature similarity alone, while our proposed method measures the edge weights according to the underlying homophily degree, which is more task-oriented. In addition, it is often believed that the 2-hop neighborhoods of a node v is always homophily-dominant in expectation (Zhu et al. 2020). So, we also set $k = 2$ in this work since it yields best performance in experiments and has relatively lower complexity meanwhile.

Optimization Objective

The whole framework consists of two components: homophily degree matrix estimation and homophily-guided propagation. The first component contains MLP and the generalized label propagation. The objective functions of MLP and generalized label propagation are give by Eq. (3) and Eq. (8). In the second component, we incorporate the homophily degree matrix into the graph convolution framework. Denote all parameters of graph convolution operation as Θ_g, then we can get the optimal Θ_g^* according the final output Z of HOG-GCN:

$$R = \operatorname{softmax}(Z)$$
$$\Theta_g^* = \underset{\Theta_g}{\operatorname{argmin}} \mathcal{L}_{gcn} = \underset{\Theta_g}{\operatorname{argmin}} \frac{1}{|V_L|} \sum_{v_a \in V_L} J(r_a^{gcn}, y_a) \tag{11}$$

In this model, the homophily degree matrix is learned from attribute and topology information during the propagation process and further used to conduct feature propagation. In return, the propagation process can help learn better homophily degree matrix. That is, these two components are

enhanced by each other. Therefore, we combine these objectives to train the whole process in an end-to-end fashion as:

$$\Theta_g^*, \Theta_m^*, T^* = \underset{\Theta_g, \Theta_m, T}{\operatorname{argmin}} \mathcal{L}_{gcn} + \lambda \mathcal{L}_{mlp} + \gamma \mathcal{L}_{lp} \tag{12}$$

where λ, γ are balance hyper-parameters. In this way, the feature propagation process is guided by current homophily degree matrix, and the result of propagation can conduct the learning of homophily degree matrix through semi-supervised classification.

Theoretical Analysis

In this section we prove that our new approach can constrain the similarity of representations between nodes within k-order neighborhoods according to homophily degree between them. In other words, nodes with higher homophily degree have more similar representation (Theorem 1).

Theorem 1. Denote the output of HOG-GCN as Z, then its propagation process can be taken as the minimization of the objective:

$$O = \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in N_k(v_i)} H_{ij} \|Z_i - Z_j\|^2 \tag{13}$$

where $N_k(v_i)$ denotes the k-order neighborhoods of node v_i, Z_i and Z_j are representations for node v_i and v_j respectively.

Proof: We can rewrite the above equation as:

$$\begin{aligned} O =& \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in N_k(v_i)} H_{ij} \|Z_i - Z_j\|^2 \\ =& \operatorname{tr}(Z^T(\hat{D} - A_k \odot H)Z) \end{aligned} \tag{14}$$

where $\hat{D}$ is the degree matrix for $A_k \odot H$. By setting derivative of Eq. (14) with respect to Z to zero (assume H is a symmetric matrix), we have:

$$\begin{aligned} \frac{\partial O}{\partial Z} =& 2(\hat{D} - A_k \odot H)Z = 0 \\ \Downarrow& \\ Z =& \hat{D}^{-1}(A_k \odot H)Z \end{aligned} \tag{15}$$

Then, it can be explained as a limit distribution where

$$Z_{\text{limit}} = \hat{D}^{-1}(A_k \odot H)Z_{\text{limit}} \tag{16}$$

We use the following iterative form to approximate the limit with $l \to \infty$:

$$Z^{(l)} = \hat{D}^{-1}(A_k \odot H)Z^{(l-1)} \tag{17}$$

When ignoring the non-linear transformation and initialize $Z^{(0)} = XW$, we have:

$$\begin{aligned} Z^{(l)} =& \hat{D}^{-1}(A_k \odot H)Z^{(l-1)} \\ =& [\hat{D}^{-1}(A_k \odot H)]^2 Z^{(l-2)} \\ =& [\hat{D}^{-1}(A_k \odot H)]^l Z^{(0)} \\ =& [\hat{D}^{-1}(A_k \odot H)]^l XW \end{aligned} \tag{18}$$

which corresponds to the propagation process of HOG-GCN.

Experiments

We first give the experimental setup, then compare our model HOG-GCN with the state-of-the-art methods on transductive node classification and visualization. Last, we give the parameter analysis and homophily degree matrix analysis.

Experimental setup

Datasets. We evaluate the performance of the proposed HOG-GCN and existing methods on seven real-world datasets. To demonstrate that HOG-GCN can adaptively learn propagation mechanism for both homophily and heterophily, we use four heterophilic networks and three homophilic networks for evaluation. The heterophilic networks include three web page datasets, Cornell, Texas and Wisconsin (Pei et al. 2020), and a film industry dataset Film (Tang et al. 2009). The homophilic networks include Cora, Citeseer and Pubmed which are all citation networks (Sen et al. 2008; Namata et al. 2012). The detailed statistics of these datasets is summarized in Table 1, where H.R. represents the homophily ratio of networks.

Datasets	Texa.	Wisc.	Corn.	Film	Cora	Cite.	Pubm.
Nodes	183	251	183	7600	2708	3327	19717
Edges	309	499	295	33544	5429	4732	44338
Features	1703	1703	1703	931	1433	3703	500
Classes	5	5	5	5	7	6	3
H.R.	0.09	0.19	0.30	0.22	0.81	0.74	0.80

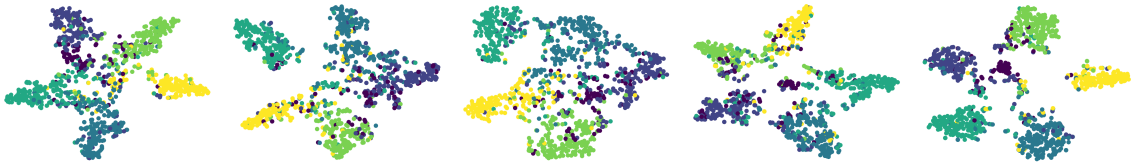
Table 1: Statistics of datasets

Baselines. We compare our proposed approach HOG-GCN with the following baselines: (1) MLP, which only uses attribute information; (2) DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), which uses network topology information alone through random walks; (3) traditional GNN models : GCN (Kipf and Welling 2017) and GAT (Velickovic et al. 2018), which work under the assumption of homophily; and (4) GNN models tackling heterophily: Geom-GCN (Pei et al. 2020), H2GCN (Zhu et al. 2020), CPGNN (Zhu et al. 2021), GPR-GNN (Chien et al. 2021) and AM-GCN (Wang et al. 2020). Particularly, as CPGNN has four different variants, we choose the best two for comparison.

Parameter Setup. For all datasets, we generate 10 random splits for training, validation and test. For each split we select 48% of nodes in each class to form the training set, 32% of nodes for the validation set and the remaining as the test set. For a fair comparison, all methods share the same 10 random splits. All the parameters of the baseline methods were set as what were used by their authors. In our approach HOG-GCN, for the homophily degree matrix estimation, we use a two-layer MLP with 512 units in the hidden layer. For the homophily-guided propagation, we use two-layer graph convolution operation with 256 units in the hidden layer. We set α to 1 and β to 0.1, and set both γ and μ to 1. We adopt Adam optimizer (Kingma and Ba 2015) and the default initialization in pytorch.

Datasets / Accuracy (%)	Texas	Wisconsin	Cornell	Film	Cora	Citeseer	Pubmed
GCN	54.05±4.36	50.39±7.55	53.78±8.59	28.78±1.48	86.48±1.43	73.58±1.37	87.34±0.65
MLP	81.08±4.83	85.49±3.53	83.24±7.03	36.58±1.44	71.29±1.60	66.96±2.61	86.48±0.63
GAT	57.30±3.38	54.31±5.62	54.59±7.33	28.99±1.44	87.16±1.17	75.64±1.95	85.25±0.60
DeepWalk	49.19±3.38	53.51±5.10	44.12±9.12	23.74±0.56	80.08±1.84	53.59±2.63	81.14±0.54
H2GCN	79.73±7.27	82.55±4.33	78.38±4.35	36.71±1.41	86.48±1.63	75.56±2.18	88.77±0.65
CPGNN-MLP	79.73±6.54	84.53±6.48	73.51±6.02	36.73±1.03	84.57±1.59	72.10±2.70	87.67±0.72
CPGNN-Cheby	74.32±7.38	81.76±6.74	63.51±5.83	35.51±1.85	**87.18±1.13**	75.52±1.84	**89.08±0.67**
GPR-GNN	84.59±4.37	83.92±3.14	82.97±5.68	36.47±1.38	86.70±1.03	75.12±1.98	87.38±0.63
AM-GCN	78.38±7.25	81.76±4.96	78.38±4.98	33.60±1.17	86.66±1.36	76.01±1.90	86.78±0.60
Geom-GCN	66.22±6.65	62.55±5.22	55.68±8.04	32.39±1.49	84.91±1.12	73.16±1.92	88.41±0.63
HOG-GCN	**85.17±4.40**	**86.67±3.36**	**84.32±4.32**	**36.82±0.84**	87.04±1.10	**76.15±1.88**	88.79±0.40

Table 2: Classification results with mean value and standard deviation. The best result is bold and the second best is underlined.



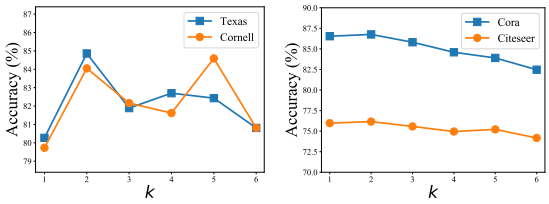
(a) GCN (b) H2GCN (c) CPGNN (d) GPRGNN (e) HOG-GCN

Figure 2: Visualization results on Citeseer dataset

Node Classification

The node classification results are reported in Table 2. We use mean accuracy as the evaluation metric along with the standard deviation of 10 splits. We have the following observations.

- Our approach HOG-GCN outperforms all the other methods compared on all of the four heterophilic networks, i.e., Texas, Cornell, Wisconsin and Film. It demonstrates the importance of incorporating and learning the homophily degree matrix into graph convolution framework for automatically changing the propagation process. To be specific, HOG-GCN significantly outperforms the traditional GNN models, i.e., GCN and GAT, by 26.49% and 24.47% on average, since they cannot generalize to the scenario of heterophily (and are even defeated by MLP which only uses attributes). Compared with methods that focus on tackling heterophily, such as H2GCN, Geom-GCN, CPGNN and GPR-GNN, our HOG-GCN also achieves an improvement between 3.9% and 19.03% in terms of mean accuracy. These results demonstrate the effectiveness of HOG-GCN in the scenario of heterophily.
- On homophilic networks (i.e., Cora, Citeseer, Pubmed), the proposed HOG-GCN performs better or comparable to the baselines. To be specific, HOG-GCN performs best on Citeseer, and achieves the second best on Pubmed. Notably, HOG-GCN outperforms GCN and GAT which have an implicit assumption of strong homophily on these three homophilic networks by 1.52% and 0.98% on average. These results demonstrate that our method has the best performance in the scenario of heterophily while maintaining comparable or better performance in the scenario of homophily, and further validate the effectiveness and robustness of the proposed approach.

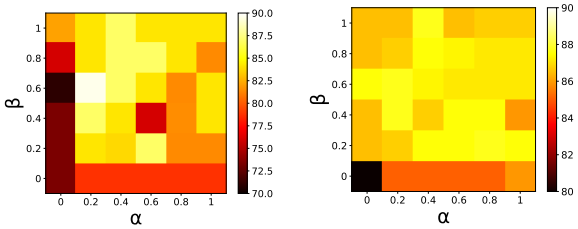


(a) Heterophilic networks (b) Homophilic networks

Figure 3: Analysis results of order k

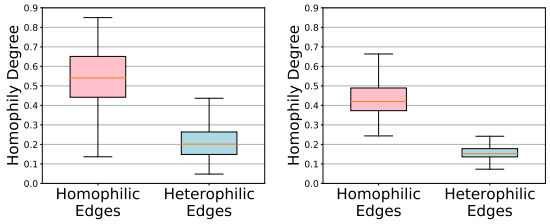
Visualization

For intuitively illustrating that our approach can gain better results, we use t-SNE (Hinton 2008), which can project the learned node representations onto a two-dimensional space, to visualize the derived representations. Fig. 2 shows the visualization results of GCN, H2GCN, CPGNN, GPR-GNN and our proposed HOG-GCN on Citeseer dataset as an illustrative example, where nodes with same color have same class label. As shown, the visualization results of GCN and CPGNN are less satisfactory in this case, since points with same color are dispersed and some points with different color are mixed with each other. The results of GPR-GNN and H2GCN are relatively better but the borders between different classes are still not so clear. Obviously, the visualization of our HOG-GCN performs much better, where the representations have the highest intra-class similarity and form more discernible clusters. The results of visualization validate that our proposed method can gain better result, and demonstrate the effectiveness of our theoretical analysis.



(a) Texas (b) Cora

Figure 4: Analysis results of weight α and β



(a) Texas (b) Cora

Figure 5: Analysis results of homophily degree distribution

Parameter Analysis

We also investigate the sensitivity of parameters in HOG-GCN. We take Texas, Cornell, Cora, Citeseer as examples, where Texas and Cornell exhibit heterophily while Cora and Citeseer exhibit homophily.

Analysis of order k. In our proposed method, we conduct generalized label propagation and homophily-guided feature propagation over the network topology within k-order neighborhoods, since we may not know the distribution of heterophily of network. Here, we vary the order k from 1 to 6 and report the classification results. The results are shown in Fig. 3. As shown, results on Texas and Cornell have a significant improvement when $k = 2$ than $k = 1$. While on Cora and Citeseer, the method only has a slight improvement when $k = 2$ than $k = 1$, and the accuracy drops as increasing k from 2 to 6. Based on these results, we conclude that on heterophilic networks where most directly connected nodes are from different classes, we often need a proper receptive field to capture enough intra-class information; while on homophilic networks where most connected nodes are from the same class, direct neighborhoods may be enough to provide intra-class information. For both homophilic and heterophilic networks, aggregating much information from higher-order neighborhoods will mix much noise and irrelated information that will negatively influence the prediction of nodes.

Analysis of weights α and β. In our model, α and β represent the weight of homophily degree matrix estimated from attribute space and topology, respectively. Here we investigate the influence of α and β on node classification by varying them from 0 to 1. The results are shown in Fig. 4. On Texas dataset, the performance is relatively poor when ignoring attribute information or topology information (i.e., α is 0 or β is 0). The method performs best when α is 0.4 and β is 0.6, demonstrating that combining attribute information and topology information can learn better homophily degree matrix and further obtain better performance. On Cora dataset, the performance is relatively stable, demonstrating that homophilic networks may not need fine-grained guidance during the propagation process. The results demonstrate the importance of combining node attributes and network topology in the estimation of homophily degree matrix, and further validate the effectiveness and robustness of the proposed approach.

Homophily Degree Matrix Analysis

In the proposed graph convolution framework, we incorporate a learnable homophily degree matrix, which can reflect the homophily or heterophily of networks, to adaptively change the propagation process. Therefore, in order to investigate whether the homophily degree matrix learned by our proposed model are meaningful, here we analyze the homophily degree distribution on homophilic edges (i.e., two nodes with same class) and heterophilic edges (i.e., two nodes with different classes) within 2-order neighborhoods, respectively. Fig. 5 shows the analysis results on a heterophilic network Texas and a homophilc network Cora.

As shown, for both homophilc and heterophilic networks, our approach can adaptively learn higher homophily degree between node with the same class, and lower homophily degree between nodes with different classes. During the feature propagation process, the homophilic edges are given greater weights while the heterophilic edges are given smaller weights according to homophily degree matrix. The results demonstrate that our approach can adaptively learn different propagation process for both homophily and heterophily, enabling it to go beyond the strong assumption of homophily.

Conclusion

In this paper, we propose a novel homophily-guided graph convolution network which can be universally suitable for both homophilic and heterophilic networks. In specific, we incorporate a learnable homophily degree matrix into graph convolution framework for modeling the homophily and heterophily of networks and further adaptively changing the propagation process according to homophily degree between node pairs. The homophily degree matrix is learned from attribute and topology information via extracting class-aware information, which can conduct the propagation process. In return, the result of propagation process can further help learn homophily degree matrix through downstream semi-supervised tasks. These two process can be enhanced by each other and trained jointly. Experiments on seven real-world datasets demonstrate that the proposed new approach outperforms existing methods under heterophily, while gains competitive performance under homophily.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61876128, 61772361) and the George Washington University Facilitating Fund (UFF) FY21.

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; and Galstyan, A. 2019. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *Proceedings of the 36th International Conference on Machine Learning*, 21–29.

Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2021. Dual quaternion knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6894–6902.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive universal generalized pagerank graph neural network. In *Proceeding of the 9th International Conference on Learning Representations*.

Gao, S.; Denoyer, L.; and Gallinari, P. 2011. Temporal link prediction by integrating content and structure information. In *International Conference on Information and Knowledge Management*, 1169–1174.

Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and van den Hengel, A. 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the 2019 International Conference on Computer Vision*, 1705–1714.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, 1024–1034.

He, D.; Wang, T.; Zhai, L.; Jin, D.; Yang, L.; Huang, Y.; Feng, Z.; and Yu, P. S. 2021. Adversarial representation mechanism learning for network embedding. *IEEE Transactions on Knowledge and Data Engineering, https://doi.org/10.1109/TKDE.2021.3103193*.

Hinton, G. E. 2008. Visualizing high-dimensional data using t-sne. *Vigiliae Christianae*, 9: 2579–2605.

Jin, D.; Huo, C.; Liang, C.; and Yang, L. 2021a. Heterogeneous graph neural network via attribute completion. In *Proceedings of the Web Conference*, 391–400.

Jin, D.; Yu, Z.; Jiao, P.; Pan, S.; Yu, P. S.; and Zhang, W. 2021b. A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering, https://doi.org/10.1109/TKDE.2021.3104155*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proceeding of the 5th International Conference on Learning Representations*.

Li, C.; and Goldwasser, D. 2019. Encoding social information with graph convolutional networks for political perspective detection in news media. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2594–2604.

Namata, G. M.; London, B.; Getoor, L.; and Huang, B. 2012. Query-driven active surveying for collective classification. In *Workshop on Mining and Learning with Graphs (MLG)*.

Pei, H.; Wei, B.; Chang, K. C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric graph convolutional networks. In *Proceeding of the 8th International Conference on Learning Representations*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 701–710.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI Magazine.*, 29(3): 93–106.

Tang, J.; Aggarwal, C. C.; and Liu, H. 2016. Node classification in signed social networks. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 54–62.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 807–816.

Tian, F.; Gao, B.; Cui, Q.; Chen, E.; and Liu, T. 2014. Learning deep representations for graph clustering. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1293–1299.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *Proceeding of the 6th International Conference on Learning Representations*.

Wang, C.; Wang, H.; Zhou, C.; Li, J.; and Gao, H. 2019. ECOQUG: An effective ensemble community scoring function. In *Proceedings of the 35th International Conference on Data Engineering*, 1702–1705.

Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; and Pei, J. 2020. AM-GCN: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1243–1253.

Wang, Z.; Wang, C.; Pei, J.; Ye, X.; and Yu, P. S. 2016. Causality based propagation history ranking in social networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3917–3923.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations*.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2021. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*.

Yan, Y.; Zhu, J.; Duda, M.; Solarz, E.; Sripada, C. S.; and Koutra, D. 2019. GroupINN: Grouping-based interpretable

neural network for classification of limited, noisy brain data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 772–782.

Yu, X.; and Han, J. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 283–292.

Zhu, J.; Rossi, R. A.; Rao, A. B.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021. Graph neural networks with heterophily. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems 33*.