Masked Graph Convolutional Network

Liang Yang[1,2], **Fan Wu**[1,2], **Yingkui Wang**[3], **Junhua Gu**[1,2] and **Yuanfang Guo**[4,*]
[1]School of Artificial Intelligence, Hebei University of Technology, China
[2]Hebei Province Key Laboratory of Big Data Calculation, Hebei University of Technology, China
[3]College of Intelligence and Computing, Tianjin University, China
[4]School of Computer Science and Engineering, Beihang University, China
yangliang@nextseason.cc, wufanslient@outlook.com, ykwang@tju.edu.cn, jhgu@hebut.edu.cn, andyguo@buaa.edu.cn

Abstract

Semi-supervised classification is a fundamental technology to process the structured and unstructured data in machine learning field. The traditional attribute-graph based semi-supervised classification methods propagate labels over the graph which is usually constructed from the data features, while the graph convolutional neural networks smooth the node attributes, i.e., propagate the attributes, over the real graph topology. In this paper, they are interpreted from the perspective of propagation, and accordingly categorized into symmetric and asymmetric propagation based methods. From the perspective of propagation, both the traditional and network based methods are propagating certain objects over the graph. However, different from the label propagation, the intuition "the connected data samples tend to be similar in terms of the attributes", in attribute propagation is only partially valid. Therefore, a masked graph convolution network (Masked GCN) is proposed by only propagating a certain portion of the attributes to the neighbours according to a masking indicator, which is learned for each node by jointly considering the attribute distributions in local neighbourhoods and the impact on the classification results. Extensive experiments on transductive and inductive node classification tasks have demonstrated the superiority of the proposed method.

1 Introduction

Semi-supervised classification, which leverages both the labelled and unlabelled data for prediction, is a traditional yet popular topic in machine learning for both the unstructured and structured data [Chapelle *et al.*, 2009; Zhu, 2006]. Traditional attribute-graph based semi-supervised classification (AGSS) methods, such as label propagation (LP [Zhu *et al.*, 2003]) and label spreading (LS [Zhou *et al.*, 2003]), can effectively classify the unstructured data, i.e., there is no structural correlations among these data samples. On the other hand, the graph convolutional neural networks (GCNNs) [Niepert *et al.*, 2016; Duvenaud *et al.*, 2015; Gao *et al.*, 2018], such as graph convolutional network (GCN [Kipf and Welling, 2017]) and graph attention network (GAT [Veličković *et al.*, 2018]), are recently proposed for semi-supervised node classification of the structured data, i.e., there exists structural correlations among these data samples.

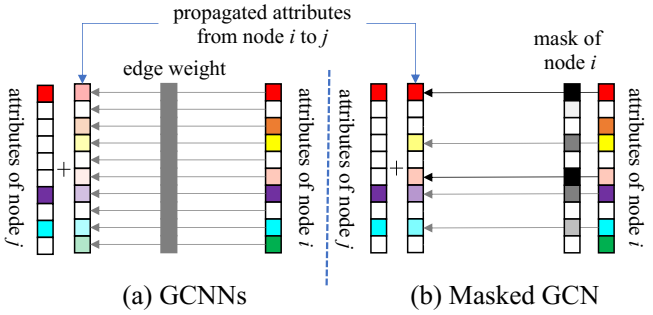

Figure 1: Comparison between traditional GCNNs and our proposed Masked GCN. GCNNs propagates the node attributes entirely. The propagated attributes in GCNNs is computed by the product of the attributes and edge weight. Masked GCN only propagates certain part of the node attributes via a mask vector learned for each node. The propagated attributes in Masked GCN is obtained by the element-wise product of the attributes and the learned mask. Note that a darker colour in the mask represents a higher value.

The traditional AGSS methods usually take all the features and labels of the data samples as input. Note that only a certain portion of the data samples possesses labels and other samples are unlabelled. Then, they propagate the given labels over the graph constructed from the input features to predict the unlabelled data samples. On the contrary, GCNNs take the real graph topologies as well as the given features and labels as input. Then, they smooth the node attributes with respect to the graph topology and predict the target nodes based on the smoothed attributes [Li *et al.*, 2018]. Many approaches with different smoothing operators have been proposed from either the spectral or spatial perspectives [Monti *et al.*, 2017; Defferrard *et al.*, 2016; Hamilton *et al.*, 2017].

In this paper, we interpret GCNNs from the perspective of propagation by analyzing three latest methods, i.e., GCN [Kipf and Welling, 2017], GAT [Veličković *et al.*, 2018] and PageRank GCN (PR-GCN [Klicpera *et al.*, 2019]). Instead of propagating the labels over the attribute-based graph in

*Corresponding author.

Category	Symmetric Propagation			Asymmetric Propagation	
Method	GCN	PR-GCN	Label Spreding	GAT	Label Propagation
Objective Function	$\mathcal{L}=\sum_{i,j} w_{ij}\left\|\frac{h_i}{\sqrt{d_i}}-\frac{h_j}{\sqrt{d_j}}\right\|_2^2$	$\mathcal{L}+\mu\sum_i \|\|h_i-y_i\|\|_2^2$		$\sum_{i,j} w_{ij}\|\|h_i-h_j\|\|_2^2$	
Graph Laplacian	$\hat{L}=D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$			$L=D-W$	
Parameters w_{ij}	a_{ij}		$\exp\left(-\sum_d \frac{(x_{id}-x_{jd})^2}{\sigma_d^2}\right)$	$\exp\left(b^T[h_i\|\|h_j]\right)$ for connected nodes i and j	$\exp\left(-\sum_d \frac{(x_{id}-x_{jd})^2}{\sigma_d^2}\right)$
Propagation Medium	Network topology		Attribute-based network	Attribute-enhanced network topology	Attribute-based network
Propagation Object	Attributes x_i		Labels y_i	Attributes x_i	Labels y_i
Analytical Solution	All nodes with same h_i	$(1-\alpha)(I-\alpha\hat{L})^{-1}Y$		Unknown	$(I-P_{uu})^{-1}P_{ul}Y_l$
Iteration $h_i^{(k+1)}$	$\sum_j \frac{w_{ij}}{\sqrt{d_i d_j}}h_j^{(k)}$	$(1-\alpha)\sum_j \frac{w_{ij}}{\sqrt{d_i d_j}}h_j^{(k)}+\alpha y_i$		$\sum_j \frac{w_{ij}}{d_i}h_j^{(k)}$	
Propagation Symmetry	Symmetric $(w_{ij}=w_{ji})$			Asymmetric $w_{ij}=b_1^Th_i+b_2^Th_j$ $w_{ji}=b_1^Th_j+b_2^Th_i$	Asymmetric $w_{ij}/d_i\neq w_{ji}/d_j$ although $w_{ij}=w_{ji}$
Constraints	$h_i^{(0)}=x_i$	With regularization $\sum_i \|\|h_i-y_i\|\|_2^2$		$h_i^{(0)}=x_i$	$h_i^{(k)}=y_i$ for the labelled nodes
Parameter Learning	No hyper-parameters		Tune $\sigma_d=\sigma$ manually	Minimize the cross entropy of the labelled data	Minimize the entropy of the unlabelled data

Table 1: Comparisons of the traditional attribute-graph based label propagation algorithms and GCNNs.

traditional attribute-graph based semi-supervised classification techniques, attributes are propagated base on real graph topologies in GCNNs.

By deep analogy, both the AGSS approaches and GCNNs can be interpreted from the perspective of propagation and classfied into two categories (as shown in Table 1):

- GCN, PR-GCN and label spreading (LS) minimize the objective functions based on symmetric normalized graph Laplacian, which results in symmetric propagations.
- GAT and label propagation (LP) minimize the objective functions based on original graph Laplacian, which results in asymmetric propagations.

This unified interpretation motivates us to consider that *"Should the attributes be propagated as the labels?"*. In general, propagation is based on the intuition that "the connected data samples tend to be similar". Typically, this intuition is valid for the labels, i.e., the connected nodes usually possess similar labels. Therefore, labels are often propagated entirely. Unfortunately, this intuition is not necessarily correct for the attributes. Different from the labels, certain attributes should not be propagated over the network. For example, the interests of a superstar tend be propagated to his fans, i.e., the fans tend to have the same interests as their idol. On the other hand, the gender of the superstar should not be propagated to his fans, because the gender of the fans cannot be changed according to the gender of the superstar. In fact, the connected nodes only possess a certain portion of similar attributes.

According to the observation above, a masked graph convolutional network (Masked GCN) is proposed in this paper, as described in Fig. 1. Instead of directly propagating all the attributes in each node, Masked GCN only propagates a portion of its attributes to the neighbours. The selection of the to-be-propagated attributes is achieved by assigning a mask to each node. The masks are learned by jointly considering the local and global information, i.e., the distributions of the attributes in local neighbourhoods and their impacts on the classification result. Although the backbone fully-connected network is similar to GCN, Masked GCN possesses another two interactive learnable components, the edge weights and masks for the nodes, which enhance the flexibility of our proposed Masked GCN. These three learnable components can be jointly trained by minimizing the cross-entropy between the predicted and given labels.

The contributions are summarized as follows:

- We analyze the traditional attribute-graph based semi-supervised classification and graph convolutional neural networks from the perspective of propagation, and classify them into two categories, symmetric and asymmetric propagations.
- We conclude that the common assumption in label propagation cannot satisfy the practical demands of the attribute propagation and only *part of* the attributes should be propagated.
- We propose a masked graph convolution network (Masked GCN), which satisfies the demands of attribute propagation, by learning a mask vector for each node.

2 Notations

For a set of data samples $V=\{v_i|i=1,...,N\}$, where $|V|=N$, each data sample is associated with a feature $x_n\in\mathbb{R}^T$. $X\in\mathbb{R}^{N\times T}$ is the collection of these features, each row of which corresponds to a sample. If there exists a network among these data samples, it can be represented by an attributed graph $G=(V,E,X)$, where E stands for a set of edges and each edge connects two vertices in V. The sparse adjacency matrix $A=[a_{ij}]\in\{0,1\}^{N\times N}$ is employed to represent the network topology, where $a_{ij}=1$ if an edge exists between the vertices v_i and v_j, and vice versa. If the network is allowed to possess self-edges, then $a_{nn}=1$. Otherwise, $a_{nn}=0$. If a network is constructed based on the

node features instead of observed links, its adjacency matrix is represented as $W = [w_{ij}] \in \mathbb{R}^{N\times N}$, which is the generalized A. $d_n = \sum_j w_{nj}$ is the degree of the vertex v_n, and $D = \mathrm{diag}(d_1, d_2, ..., d_N)$ is the degree matrix of the adjacency matrix W. Then, the normalized adjacency matrix is $P = D^{-1}A$. The graph Laplacian and its normalized form are defined as $L = D - A$ and $\hat{L} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$, respectively.

In semi-supervised classification, the labels $Y_l = \{y_i\} \in \mathbb{R}^{|V_l|\times F}$ of a set of vertices $V_l \subset V$, where F is the number of classes, are given. For simplicity, nodes $\{v_n\}_{n=1}^{l}$ is assumed be labelled, while nodes $\{v_n\}_{n=l+1}^{|V|}$ are unlabelled. Then, the label matrix Y_l can be reformed as $Y = [Y_l; Y_u] \in \mathbb{R}^{|V|\times F}$, where Y_u is a zero matrix. The normalized adjacency matrix can then be expressed as

$$P = \left[\begin{array}{c|c} P_{ll} & P_{lu} \\ \hline P_{ul} & P_{uu} \end{array}\right],$$

where $P_{ll} \in \mathbb{R}^{|V_l|\times|V_l|}$ and $P_{uu} \in \mathbb{R}^{|V-V_l|\times|V-V_l|}$ are the sub-matrices of P which correspond to the labelled and unlabelled nodes, respectively.

3 Comparisons between AGSS and GCNNs

In this section, two traditional AGSS methods and three latest GCNNs are firstly reviewed. Then, the similarities between these two types of techniques are analyzed and summarized accordingly. Note that the symbol h_i is employed to denote the object, which is propagated over the graph. Typically, the labels are the objects propagated in traditional AGSS approaches, while the attributes are the objects propagated in GCNNs. Therefore, we simply employ h_i to represent both the propagated labels in AGSS classification (in Sec. 3.1), and the propagated attributes in GCNNs (in Sec. 3.2).

3.1 Traditional Attribute-graph based Semi-supervised Classification

In transductive semi-supervised learning, the unlabeled data samples given in the training stage is also the to-be-predicted data in the testing stage. Traditional AGSS approaches are well-studied transductive semi-supervised learning strategies.

A graph, which reveals the similarities between data samples, is usually constructed with edges. Let w_{ij} denote the edge between the data samples v_i and v_j, and it is defined as

$$w_{ij} = \exp\left(-\sum_d \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right), \tag{1}$$

where σ_d's are learnable parameters. Then, the semi-supervised learning process is conducted by propagating the given labels over the constructed graph. With different propagation strategies and constraints, numerous algorithms have been proposed, such as label propagation [Zhu *et al.*, 2003], label spreading [Zhou *et al.*, 2003] and etc.

Label Propagation

According to Gaussian random field on the graph, where the mean of the field is characterized by the harmonic functions, Label Propagation (LP) directly minimizes the energy function

$$E(H) = \sum\nolimits_{i,j} w_{ij}||h_i - h_j||_2^2 = \mathrm{tr}(H^T LH), \tag{2}$$

with $h_i = y_i$ which is fixed as the label y_i of the i^{th} data sample in V_l. Note that $H = \{h_i\}_{i=1}^N$ is the predicted labels, $L = D - A$ represents the graph Laplacian matrix of W and $\mathrm{tr}(.)$ stands for the trace operator. By minimizing Eq. (2) with respect to h_i, its iterative updating formula for the unlabelled data is computed as

$$h_i^{(k+1)} = \sum_j \frac{w_{ij}}{\sum_j w_{ij}} h_j^{(k)} = \sum_j \frac{w_{ij}}{d_i} h_j^{(k)}. \tag{3}$$

Then, the analytical solution can be expressed as

$$Y_u = (I - P_{uu})^{-1} P_{ul} Y_l, \tag{4}$$

where I is an identity matrix with the size of $|V - V_l|$. The parameters σ_d's are learned by minimizing the average entropy of the labels for the unlabelled data, which is formulated as

$$-\sum\nolimits_{i=l+1}^{N} h_i \log h_i + (1 - h_i)\log(1 - h_i), \tag{5}$$

with respect to σ_d. This strategy is based on the intuition that a well constructed graph will generate confident predictions, i.e., h_i is close to either 0 or 1.

Label Spreading

By considering the smoothness property of the graph, Label Spreading (LS) formulates the propagation as $H^{(k+1)} = (1 - \alpha)\hat{L}H^{(k)} + \alpha Y$, where $\hat{L}$ is the normalized Laplacian matrix of W. This formulation is equivalent to the updating formula

$$h_i^{(k+1)} = (1 - \alpha)\sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} h_j^{(k)} + \alpha y_i, \tag{6}$$

which possesses the analytical solution as Eq. (7) shows.

$$H = (1 - \alpha)(I - \alpha\hat{L})^{-1} Y \tag{7}$$

Eq. (7) is proved to be the solution of the following objective function [Zhou *et al.*, 2003]

$$\begin{aligned} E(H) &= \sum_{i,j} w_{ij} \left\| \frac{h_i}{\sqrt{d_i}} - \frac{h_j}{\sqrt{d_j}} \right\|_2^2 + \mu \sum_i ||h_i - y_i||_2^2 \\ &= \mathrm{tr}(H^T \hat{L} H) + \mu||H - Y||_F^2, \end{aligned} \tag{8}$$

with $\alpha = \frac{1}{1+\mu}$. As can be observed, the hard constraint of the given labels in LP is replaced by a soft constraint, i.e., the regularization term $\sum_i ||h_i - y_i||_2^2$. Here, all the hyper-parameters σ_d's are set to be σ and tuned manually.

3.2 Graph Convolutional Neural Networks

Graph convolutional neural networks are extended from the convolutional neural network (CNN [Krizhevsky *et al.*, 2012]) which processes structured data samples, such as images or speeches, to process the general graph data samples. Different from the traditional attribute-graph based semi-supervised learning, where only the attributes and labels are given and networks are constructed from the attributes as described in Sec. 3.1, GCNNs exploits the network topologies as well as the attributes and labels.

Graph Convolutional Network

Graph Convolutional Network (GCN) [Kipf and Welling, 2017] simplifies many previous models which possess high complexities, and defines the graph convolution operation as

$$H_{GCN}^{(k+1)} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H_{GCN}^{(k)}, \tag{9}$$

with $H_{GCN}^{(0)} = X$, where $\tilde{A} = A + I_N$ and $\tilde{D}_{nn} = \sum_j \tilde{A}_{nj} = d_n + 1$. The unknown labels can be predicted by feeding $H_{GCN}^{(k+1)}$ into a fully-connected layer as $Q_{GCN} = H_{GCN}W$. The parameter W is obtained by minimizing the cross-entropy between the given labels and the predictions as

$$\mathcal{L} = -\sum_{n\in V_l}\sum_{f=1}^{F} Y_{nf}\log(Q_{nf}). \tag{10}$$

PageRank GCN

Since GCN is equivalent to Laplacian smoothing the attributes in the graph [Li *et al.*, 2018], all the nodes tend to obtain identical attributes as the number of iterations (i.e. the number of graph convolutional layers) increases. To alleviate this issue, PageRank GCN (PR-GCN [Klicpera *et al.*, 2019]) improves the original GCN by considering its relationship with PageRank. The convolution operator in PageRank GCN is defined as

$$H_{PR\_GCN} = (1-\alpha)(I - \alpha\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}})^{-1}X. \tag{11}$$

Unfortunately, directly calculating the fully personalized PageRank matrix $(1-\alpha)(I-\alpha\hat{L})^{-1}$ is computationally inefficient, because it will generate a dense matrix, which will induce extra computations in the multiplication with the attribute matrix X. PR-GCN resolves this issue by approximating H_{PR_GCN} in an iterative manner as

$$H_{PR\_GCN}^{(k+1)} = (1-\alpha)\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H_{PR\_GCN}^{(k+1)} + \alpha X. \tag{12}$$

Its prediction and training procedures are identical to GCN.

Graph Attention Network

To give consistent performances with variable sized inputs and effectively exploit the most important parts of the inputs, the attention mechanism is introduced by GAT to handle neighbourhoods with various sizes. GAT replaces the graph convolutional layer in GCN with a graph attention layer by leveraging attention mechanism [Bahdanau *et al.*, 2015]. To obtain the nodes which should be focused on, the attention coefficients

$$o_{nk} = \frac{\exp\left(c(x_n^T W, x_k^T W)\right)}{\sum_{k\in N(n)}\exp\left(c(x_n^T W, x_k^T W)\right)} \tag{13}$$

is computed to reveal the amount of attentions received at node v_k from node v_n. GAT adopts the Leaky-ReLU nonlinearity mapping, $\mathrm{LeakyReLU}([x_n^T W||x_k^T W]b)$, as $c(.,.)$ in Eq. (13), where $[x_n^T W||x_k^T W]$ represents the concatenation of $x_n^T W$ and $x_k^T W$, and $b \in \mathbb{R}^{2F}$ stands for the shared parameters. $O = [o_{nk}] \in \mathbb{R}^{N\times N}$ can be regarded as the re-assigned weights to the adjacency matrix $A = [a_{nk}] \in \mathbb{R}^{N\times N}$. $o_{nk} \neq 0$ when $a_{nk} = 1$, i.e., the nodes v_n and v_k are connected. Then, GAT can be represented as $Q_{GAT} = OXW$ and considered as a re-weighted adjacency matrix based smoothing as

$$H_{GAT}^{(k+1)} = OH_{GAT}^{(k)}, \tag{14}$$

with $H_{GAT}^{(0)} = X$. After the smoothing operation, a projection $Q_{GAT} = H_{GAT}W$ is performed. The parameter W and attention parameter b can be computed by minimizing the cross-entropy between the given labels and predictions according to Eq. (10).

3.3 Comparisons

Here, the traditional AGSS techniques (in Sec. 3.1) and GCNNs (in Sec. 3.2) are analyzed from the perspective of propagation, and they are classified into two categories: symmetric and asymmetric propagations as summarized in Table 1.

Symmetric Propagation (GCN, PR-GCN and LS)

Comparing Eq. (11) and Eq. (12) with Eq. (7) and Eq. (6), respectively, the differences between PR-GCN and LS are summarized as follows: 1) the labels are propagated in LS while the attributes are propagated in PR-GCN; 2) the normalization is $\frac{1}{\sqrt{d_i d_j}}$ in LS and $\frac{1}{\sqrt{(d_i+1)(d_j+1)}}$ in PR-GCN, because $A_{ii} = 1$ in PR-GCN while $W_{ii} = 0$ in LS. If self-loop is allowed in LS, its normalization will also be $\frac{1}{\sqrt{(d_i+1)(d_j+1)}}$ which is identical to PR-GCN. Therefore, PR-GCN [Klicpera *et al.*, 2019] is equivalent to LS [Zhou *et al.*, 2003], despite the different propagated objects. The objective function of LS in Eq. (8) can also serve for PR-GCN by replacing the label y_i with the attribute x_i.

Comparing Eq. (12) from PR-GCN with Eq. (9) from GCN, PR-GCN averages the propagated attributes $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H$ and the original attributes X with weights $1-\alpha$ and α, respectively, while GCN does not perform these averages. The objective function of GCN is the first part of Eq. (8) (which is the cost function of LS), i.e., $\mathcal{L} = \sum_{i,j} w_{ij}||\frac{h_i}{\sqrt{d_i}} - \frac{h_j}{\sqrt{d_j}}||_2^2$.

The propagation weights in GCN, PR-GCN and LS are $\frac{w_{ij}}{\sqrt{d_i d_j}}$. Since $w_{ij} = w_{ji}$ holds in these algorithms, their propagations are considered to be symmetric.

Asymmetric Propagation (GAT and LP)

Comparing Eq. (3) from LP with Eq. (14) (where o_{nk} defined in Eq. (13)) from GAT, the difference between them is the edge weight function. LP adopts Eq. (1), while GAT utilizes

$$w_{ij} = \exp\left(b^T[h_i||h_j]\right) = \exp\left(b_1^T h_i + b_2^T h_j\right), \tag{15}$$

if v_i and v_j are connected. Note that $[h_i||h_j]$ is the concatenated vector of h_i and h_j, and h_i possesses the identical size compared to h_j. Therefore, their objective functions and updating formulas are the same as shown in Eqs. (2) and (3), respectively. Since $w_{ij} = w_{ji}$ holds in LP, it performs symmetric propagations if the normalization is disabled. Since $b_i = b_j$ does not exist in most of the situations, $w_{ij} \neq w_{ji}$ and thus LP and GAT perform asymmetric propagations.

Dataset	#Nodes	#Edges	#Classes	#Features
CiteSeer	3,327	4,732	6	3,703
Cora	2,708	5,429	7	1,433
PubMed	19,717	44,338	3	500
NELL	65,755	266,144	210	5,414
PPI	56,944	818,716	121	50

Table 2: Datasets.

4 Masked Graph Convolutional Network

In this section, we propose our Masked Graph Convolutional Network (Masked GCN) by providing the motivations, formulations and solutions accordingly.

4.1 Motivations

With the comparisons and summaries in Sec. 3, the inherent difference between the traditional AGSS techniques and GCNNs can be considered as the object to be propagated before the predictions. Labels are propagated in traditional AGSS methods, while attributes are propagated in GCNNs. The common assumption in these methods is "the connected data samples tend to be similar". For traditional label propagations, this assumption usually holds because the connected data samples tend to possess similar labels. However, this assumption is debatable for attribute propagations. For example, each people usually has many characters in online social networks. Two connected people are often similar for some characters instead of all the characters. Therefore, the connected nodes are usually similar for a certain portion of their attributes.

4.2 Formulations and Solutions

Here, Masked Graph Convolutional Network (Masked GCN) is formulated based on asymmetric propagation and attribute-enhanced network topology. According to the above intuition, the objective function is defined as follows.

$$\mathcal{L}_{Masked\_GCN\_Asym} = \sum\nolimits_{i,j} w_{ij}||h_i - M^{(j)}h_j||_2^2 \quad (16)$$

where h_i stands for the attributes of the node v_i, and w_{ij}, which is defined in Eq. (15), represents the edge weight between the nodes v_i and v_j. Note that $M^{(j)} = \mathrm{diag}(m_1^{(j)}, m_2^{(j)}, ..., m_T^{(j)})$ is the diagonal mask matrix of the node v_i, where T is the number of attributes, i.e., the length of h_i (h_j). This mask matrix $M^{(j)}$ relaxes the hard constraint that h_i and h_j tend to be completely similar and only constrain a certain portion of h_i and h_j to be similar. By minimizing Eq. (16) with respect to h_i, the updating formula can be obtained as

$$h_i^{(k+1)} = \sum_j \frac{w_{ij}}{\sum_j w_{ij}} M^{(j)} h_j^{(k)}. \quad (17)$$

As can be observed, the propagated attributes is masked by $M^{(j)}$ as illustrated in Fig. 1. The mask $m_t^{(j)}$ of attribute t in node j is exploited to determine whether the attribute t in node j should be propagated to its neighbours. If $m_t^{(j)}$ is large, most of the attribute t in node j will be propagated (e.g. the red and pink attributes in Fig. 1(b)), and vice versa (e.g. the orange and green attributes).

Then, the formulation of $M^{(j)}$ can be introduced. Intuitively, the mask $m_t^{(j)}$ of the attribute t in node j should be determined by both the local and global behaviors of attribute t. Typically, $m_t^{(j)}$ should be affected by the confidence that node i possesses attribute t. Therefore, the consistencies of attribute t between node v_i and its neighbours, i.e., $-\sum_{p\in N(j)} w_{jp}(h_{pt} - h_{jt})^2$, are taken in to account in the proposed method. If the consistency is low, the mask tends to prevent the attribute t from being propagated, i.e., $m_t^{(j)}$ should be small. On the other hand, $m_t^{(j)}$ should globally benefit the classification result. Therefore, a learnable parameter σ_t, which is independent of the nodes, is assigned. By jointly considering the local and global behaviors, the mask is defined as

$$m_t^{(j)} = \exp\left(-\frac{1}{d_j}\sum_{p\in N(j)} \frac{w_{jp}(h_{pt} - h_{jt})^2}{\sigma_t^2}\right), \quad (18)$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_T)$ is the matrix which contains the learnable attribute weights. Similar to GCN and GAT, two backbone convolutional layers are utilized in Masked GCN, i.e., the attributes are propagated twice.

By setting $H^{(0)}_{Masked\_GCN} = X$ which is the original attributes, $H^{(1)}_{Masked\_GCN} = \{h_i^{(1)}\}_{i=1}^N$ and $H^{(2)}_{Masked\_GCN} = \{h_i^{(2)}\}_{i=1}^N$ are calculated via Eq. (17). The unknown labels are predicted by feeding the obtained attributes $H^{(2)}_{Masked\_GCN}$ into a fully-connected layer as follow.

$$T_{Masked\_GCN} = \mathrm{softmax}(H^{(2)}_{Masked\_GCN} W) \quad (19)$$

where W is the weights in the fully-connected layer.

There exists three learnable components in the proposed Masked GCN, W in the fully-connected layer (Eq. (19)), b in edge re-weighting (Eq. (15)) and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_T)$ in the learned mask (Eq. (18)). They can be jointly trained via gradient back-propagation.

Remark 1: In GCN, all the attributes are propagated, which tend to amend all the nodes to possess the identical attributes after convergence. Since Masked GCN only propagates part of the attributes, this issue can be alleviated.

Remark 2: The formulation of Masked GCN are constructed for the transductive semi-supervised node classification task. Since GCNNs propagate attributes based on the network topology instead of propagating labels, the proposed Masked GCN can be directly applied to inductive semi-supervised node classification task where the target graphs in the testing stage are not provided in the training stage.

Remark 3: Although only GAT in asymmetric propagation is improved to the masked version, other GCNNs, such as GCN and PR-GCN in symmetric propagation, can also be improved with the proposed mask. For example, GCN can be

Methods	Cora	Citeseer	Pubmed	NELL
MLP	55.1%	46.5%	71.4%	22.9%
ManiReg [Belkin *et al.*, 2006]	59.5%	60.1%	70.7%	21.8%
SemiEmb [Weston *et al.*, 2012]	59.0%	59.6%	71.7%	26.7%
LP [Zhu *et al.*, 2003]	68.0%	45.3%	63.0%	26.5%
DeepWalk [Perozzi *et al.*, 2014]	67.2%	43.2%	65.3%	58.1%
ICA [Lu and Getoor, 2003]	75.1%	69.1%	73.9%	23.2%
Planetoid [Yang *et al.*, 2016]	75.7%	64.7%	77.2%	61.9%
Chebyshev [Defferrard *et al.*, 2016]	81.2%	69.8%	74.4%	-
MoNet [Monti *et al.*, 2017]	81.7%	69.9%	78.8%	64.2%
Random-scheme Mask GCN	14.8%	17.2%	35.1%	4.8%
GCN [Kipf and Welling, 2017]	81.5%	70.3%	79.0%	66.0%
Masked GCN (Sym)	**82.7%**	**72.0%**	**79.3%**	**68.2%**
GAT [Veličković *et al.*, 2018]	83.0%	72.5%	79.0%	-
Masked GCN (Asym)	**84.4%**	**73.8%**	**80.2%**	**68.9%**

Table 3: Transductive learning results.

improved to

$$\mathcal{L}_{Masked\_GCN\_Sym} = \sum_{i,j} w_{ij} \left\| \frac{h_i}{\sqrt{d_i}} - M^{(j)} \frac{h_j}{\sqrt{d_j}} \right\|_2^2, \tag{20}$$

where $w_{ij} = a_{ij}$ is obtained from graph topology and it is not learnable as in GCN. Similar to Eq. (16), $M^{(j)}$ also serves as masks in Eq. (20).

Remark 4: Although T additional learnable parameters σ_t's are introduced in Masked GCN, the number of additional parameters is relatively small compared to the numbers of parameters in GCN and GAT. The parameters in the one-layered GCN are the weights of the fully connected layer, whose number is $T \times C$ (where T and C are the numbers of attributes and classes, respectively). In addition to the weights of the fully connected layer, GAT also possesses another $2T$ parameters, which is employed to determine the weights of the edges. Therefore, the T additional learnable parameters in Masked GCN tend not to cause overfitting.

5 Evaluations

In this section, we validate the proposed Masked GCN by empirically evaluating its performances in the semi-supervised node classification task with both the transductive and inductive learning settings. Masked GCN (Asym) and Masked GCN (Sym) denotes the proposed method with asymmetric and symmetric propagations, respectively.

5.1 Datasets

For the transductive learning task, the experiments are conducted on three commonly utilized citation networks [Sen *et al.*, 2008], Cora, CiteSeer and PubMed, as shown in Table 2. In each network, nodes and edges are research papers and undirected citations, respectively. The node content is constructed by extracting the words from the documents. Papers are categorized into various classes according to the disciplines. In each citation network, 20 nodes per class, 500 nodes and 1000 nodes are employed for training, validation and performance assessment, respectively. Besides, another bipartite large network [Carlson *et al.*, 2010], NELL, is constructed from a knowledge graph as shown in Table 2. Except for the entity nodes in the original knowledge graph, separate relation nodes (e_i, r) and (e_j, r) are extracted from each entity pair (e_i, r, e_j). The edges are constructed between each entity e_i and its all relation nodes (e_i, r).

For the inductive learning task, the protein-protein interaction (PPI) dataset [Zitnik and Leskovec, 2017] is employed. PPI dataset consists of 24 attributed graphs, each of which corresponds to a different human tissue and contains 2,373 nodes in average. Each node possesses 50 features including the positional gene sets, motif gene sets and immunological signatures. 121 cellular functions are employed from the gene ontology sets, which are collected from the Molecular Signatures Database [Subramanian *et al.*, 2005], as labels. Algorithms are trained on 20 graphs, validated on 2 graphs and tested on 2 graphs, accordingly. The training and validation graphs are fully labelled, while the graphs for testing are not given during training and validation processes.

5.2 Baselines

For the transductive learning task, 11 baseline semi-supervised node classification algorithms are employed, including multilayer perceptron (MLP), label propagation (LP) , semi-supervised embedding (SemiEmb), manifold regularization (ManiReg), graph embedding (DeepWalk), iterative classification algorithm (ICA), attribute-graph based semi-supervised learning framework (Planetoid), graph convolution with Chebyshev filters (Chebyshev), graph convolutional network (GCN), mixture model networks (MoNet), and graph attention networks (GAT), for comparisons.

For the inductive learning task, 7 state-of-the-art algorithms are employed, including random classifier (Random), logistic regression based on node feature without network structure (Logistic Regression), inductive variant of GCN (Inductive GCN) [Kipf and Welling, 2017], three variants of

Methods	PPI
Random	0.396
Logistic Regression	0.422
GraphSAGE-mean	0.598
GraphSAGE-LSTM	0.612
GraphSAGE-pool	0.600
Inductive GCN	0.500
Masked GCN (Sym)	**0.892**
GAT	0.934
Masked GCN (Asym)	**0.952**

Table 4: Inductive learning results.

GraphSAGE [Hamilton *et al.*, 2017] with different aggregator functions, and graph attention network (GAT) [Veličković *et al.*, 2018]. GraphSAGE aggregates the representations (i.e. features) in local neighbourhoods and concatenates the aggregations with the corresponding node representations. GraphSAGE-mean calculates the element-wise means in local neighbourhoods as the representations. GraphSAGE-LSTM feeds the representations from local neighbourhoods into an LSTM by considering its superior expressive capability. GraphSAGE-pool, which is symmetric and trainable, inputs the representations from local neighbourhoods into a fully-connected neural network and then processes the outcomes by performing an element-wise max-pooling operation.

All the results of the baseline methods are either from their original papers or produced by running the codes from the authors with their default settings.

5.3 Results

Before we present the results for both the transductive and inductive learning tasks, the performance of our method with a random-scheme mask matrix is introduced to show the necessity of learning the mask as in Eq. (18). The classification accuracies are shown in Table 3. As can be observed, the performance with a random mask matrix is similar to that of random classification and the accuracies are much lower than that of our mask scheme, because adding a random-scheme mask matrix to each node is equivalent to assigning random attributes to each node.

The results for transductive learning task in terms of classification accuracies are shown in Table 3. To respectively highlight the improvements of Masked GCN in symmetric and asymmetric propagations, GCN and its improved version Masked GCN (Sym) are placed together while GAT and Masked GCN (Asym) are placed together. We can find that Masked GCN significantly improves the performances compared to GCN and GAT. Besides, Masked GCN (Asym) outperforms other methods including Masked GCN (Sym).

The results for inductive learning task in terms of micro F-1 scores are shown in Table 4. Similar conclusion can be obtained as the transductive learning results. However, the gain achieved by our proposed method for the symmetric propagation (GCN) is much significant than that for the asymmetric propagation (GAT), because the propagation weights in GCN are only determined by the degrees of two connected nodes. Therefore, the performance of GCN is relatively low and the gain induced by Masked GCN is much significant. Besides, although GAT has achieved a high performance on PPI, its performance can be further improved by our Masked GCN.

Methods	Cora	Citeseer	Pubmed
GAT	44.8	72.6	270.6
Masked GCN (Asym)	58.9	92.1	312.3

Table 5: Running time comparison (in seconds).

According to the results in both the transductive and inductive tasks, the proposed Masked GCN can obviously improve the performances compared to the baseline methods, which also verifies the effectiveness of our principle, i.e., propagating partial attributes instead of the entire ones.

The running time of our Masked GCN is compared to that of the state-of-the-art method, GAT, on the four networks with the transductive learning setting as shown in Table 5. The results have shown that the running time of Masked GCN is 1.24 times compared to that of GAT in average. In fact, the extra time is mainly utilized to learn the parameters of masks.

Case Study. To verify our motivations and contributions, a case study is conducted on the Pubmed network to specifically show which propagations will be masked. The Pubmed network consists of publications about diabetes. Each publication is described by a word vector from a dictionary which consists of 500 unique words. As can be observed from our results, most of the common words (such as "increase", "measure", etc.) are masked, because their propagations do not significantly impact the classification performance. On the contrary, the medical terms (such as "kinase", "metabolic", etc.), which tend to give large contributions in the classification, possess high propagation rates.

6 Conclusions

In this paper, we observe that the connected nodes are usually similar for a certain portion of their attributes. According to this observation, we propose a masked graph convolutional network (Masked GCN) by masking the attributes to be propagated. The mask is learned by jointly considering the attribute distributions in local neighbourhoods and the impact on the classification results. The experimental results in both the transductive and inductive tasks verify the correctness and superiority of propagating partial attributes instead of the entire ones. In the future, our Masked GCN will be extended to other heterogeneous networks [Serafino *et al.*, 2018].

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No.2017YFC0820106), in part by the National Natural Science Foundation of China under Grant 61503281 and Grant 61802391, in part by the Foundation for Innovative Research Groups through the National Natural Science Foundation of China under Grant 61421003, and in part by the Fundamental Research Funds for the Central Universities.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[Carlson *et al.*, 2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, pages 1306–1313, 2010.

[Chapelle *et al.*, 2009] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE TNN*, 20(3):542–542, 2009.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3837–3845, 2016.

[Duvenaud *et al.*, 2015] David K. Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, pages 2224–2232, 2015.

[Gao *et al.*, 2018] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *ACM SIGKDD*, pages 1416–1424, 2018.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1025–1035, 2017.

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

[Klicpera *et al.*, 2019] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *ICLR*, 2019.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, pages 3538–3545, 2018.

[Lu and Getoor, 2003] Qing Lu and Lise Getoor. Link-based classification. In *ICML*, pages 496–503, 2003.

[Monti *et al.*, 2017] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE CVPR*, pages 5425–5434, 2017.

[Niepert *et al.*, 2016] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *ACM SIGKDD*, pages 701–710, 2014.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[Serafino *et al.*, 2018] Francesco Serafino, Gianvito Pio, and Michelangelo Ceci. Ensemble learning for multi-type classification in heterogeneous networks. *IEEE TKDE*, 30(12):2326–2339, 2018.

[Subramanian *et al.*, 2005] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Weston *et al.*, 2012] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 639–655. 2012.

[Yang *et al.*, 2016] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, pages 40–48, 2016.

[Zhou *et al.*, 2003] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2003.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

[Zhu, 2006] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.

[Zitnik and Leskovec, 2017] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.