Applied Intelligence (2021) 51:5007–5015
https://doi.org/10.1007/s10489-020-02093-5

E-GCN: graph convolution with estimated labels

Jisheng Qin[1] · Xiaoqin Zeng[1] · Shengli Wu[2] · E. Tang[1]

Accepted: 24 November 2020 / Published online: 6 January 2021

Abstract

Graph ***C***onvolutional ***N***etwork (GCN) has been commonly applied for semi-supervised learning tasks. However, the established GCN frequently only considers the given labels in the topology optimization, which may not deliver the best performance for semi-supervised learning tasks. In this paper, we propose a novel ***G***raph ***C***onvolutional ***N***etwork with ***E***stimated labels (E-GCN) for semi-supervised learning. The core design of E-GCN is to learn a suitable network topology for semi-supervised learning by linking both estimated labels and given labels in a centralized network framework. The major enhancement is that both given labels and estimated labels are utilized for the topology optimization in E-GCN, which assists the graph convolution implementation for unknown labels evaluation. Experimental results demonstrate that E-GCN is significantly better than ***s***tate-***o***f-***t***he-***a***rt (SOTA) baselines without estimated labels.

Keywords Semi-supervised · Learning graph convolution · Topology optimization · Estimated labels

1 Introduction

Graphs are powerful instruments and the connection between real-world objects is often the graph structure, such as human social networks, paper networks, protein networks, etc. There are many significant graph applications such as predicting whether they have an edge between pair nodes [1, 2], what are the most probable label of nodes [3], what is the personalization needs of users [4, 5]. Node classification can be described as analyzing the features to obtain hidden information in the graph for nodes with unknown labels. It is one of the widest topics in current research [6–10].

✉ Xiaoqin Zeng
xzeng@hhu.edu.cn

Jisheng Qin
qinjisheng2018@gmail.com

Shengli Wu
S.Wu1@ulster.ac.uk

E. Tang
tange@hyit.edu.cn

[1] Institute of Intelligence Science and Technology, Hohai University, Nanjing, China

[2] School of Computing, Ulster University, Belfast, BT15 1ED, UK

Node classification develops swiftly after taking advantage of network topology. Although the traditional classification method can already complete the task significantly, there is still a lot of room for improvement in classification performance due to the inherent sparseness and noise of usual networks.

At present, to improve classification performance, the most frequently operated measure is to make full employ of label information, node characteristics, and network topology. Recently, with the improvement of graph convolutional networks, researchers have extended Convolutional Neural Networks (CNN) to non-Euclidean graph data with great success. Among the node classification research, Graph Convolutional Network (GCN) [3], which simplifies ChebNet, has drawn wide attention as a result of its simplicity and high performance. As illustrated in Fig. 1a, the purpose of GCN is to perform specific filtering operations on features and employ a fully connected network process. GCN employs network topology and labels information to train, but they are fixed. As illustrated in Fig. 1b, Yang et al. [11] utilized the given labels to learn the graph network simultaneously, which provides more flexibility compared to GCN.

Unfortunately, current GCN research has not fully explored the information of labels for classified nodes. Specifically, although Yang et al. [11] assumed some sparse and noise and dynamically learn the given network topology and make full employ of the given labels information, the

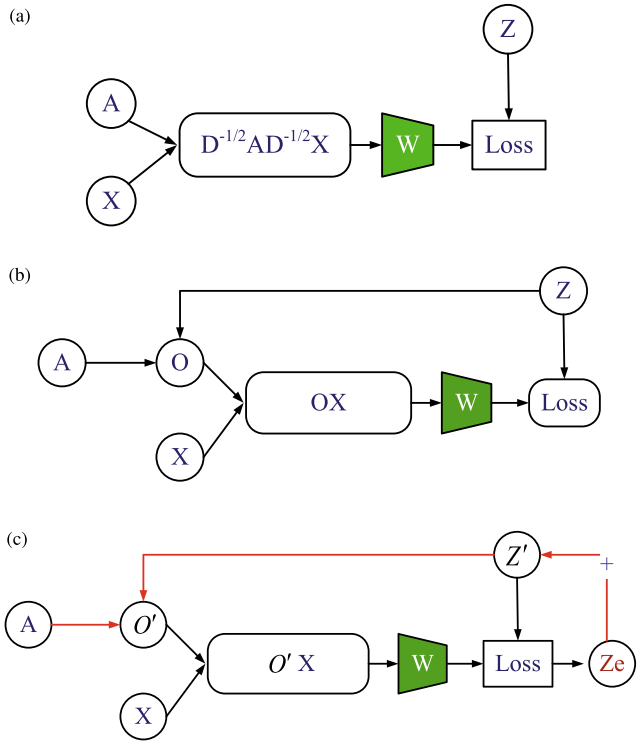

Fig. 1 The comparison between GCN, TO-GCN, and our proposed E-GCN. GCN adopts the designated topology and feature X as input to learn the parameters W. TO-GCN exploits the given labels to participate in learning. E-GCN concurrently learns the W to comprehensively explore the given labels and estimated labels

corresponding information of the classified nodes has not been considered.

Based on Yang et al. [11], to completely explore the topology of the graph network and enhance the effective utilization of the estimated labels of the classified nodes, we propose a novel Graph Convolutional Network with Estimated Labels (E-GCN). As illustrated in Fig. 1c, We initially estimate the maximum possible label of an unknown label node based on the existing GCN model and then learn the network topology simultaneously with the given labels Z and the estimated labels Z_e. Compared with GCN and TO-GCN, our proposed E-GCN has greater flexibility and accuracy.

Specifically, we first utilize the given labels to optimize the network topology and estimate the labels of new vertexes that are adopted to optimize the network topology. Optimize the network topology by making full explore of the given labels and estimated labels, and finally learn the optimal parameters of the model.

Our contributions are two-fold as follows:

- We indicate that the optimization of the model with estimated labels and the given labels can enhance the node classification performance in the semi-supervised learning tasks.
- We propose a novel Graph Convolution with Estimated Labels for Semi-supervised Node Classification (E-GCN), which simultaneously learns the topology of the graph network and the parameters of the fully connected network with fully explored estimated labels and the given labels.

2 Related works

2.1 Semi-supervised node classification

Semi-supervised node classification is performed to learn the potential distribution of the entire data with little labeled data and a large amount of unlabeled data. Various methods are available for semi-supervised node classification, and we focus on two major methods, graph representation-based, and spectral representation-based graph convolution.

The graph representation projects a high latitude vector or sparse vector to low latitude, primarily by capturing the neighbor similarity and community membership of the node in space. As proposed in [12], walks are constructed using random wandering over the graph to catch local spatial information, and finally to extract all spatial information to attain label-independent embedding.

The spectral-based method of graph convolution is to complete node classification employing a spectral representation of a graph. Consider $G = (V, E)$ being an undirected graph, where V is the set of all vertices $v_i \in V, (i = 1, 2, \cdots, N)$, and E represents the set of all edges in the graph. $A \in R^{N \times N}$ denotes the neighborhood matrix of the graph, which describes the presence or absence of connections between nodes in the graph, and the value of element A_{ij} is the binary case of 0 or 1. If the value of attribute A_{ij} is a real value, the adjacency matrix reflects the similarity between the nodes of the graph in the encoding for the similarity measure. As well, we can generate the degree matrix D from the adjacency matrix, which defines the degree matrix as a diagonal matrix with a specific element whose value is $D_{ii} = \sum_j A_{ij}$.

For M-classification problems the M-dimensional vector is typically derived after a 1-of-M coding solution, denoting the probability that a node corresponds to each classification. Regularization is normally implemented in schemes with the generic form as follows:

$$\mathcal{L} = \mathcal{L}_L\left(f\left(\mathbf{X}_L\right), \mathbf{T}_L\right) + \lambda f(\mathbf{X})^T \mathbf{L} f(\mathbf{X}) \quad (1)$$

where $f(\bullet)$ depicts a vector of classifications mapping node representations from dimension D to dimension M, T_L is a vector of classification markers of actual nodes, and $L = D - A$ shows a non-standardized Laplace matrix. The first

part of the equation is the categorical loss of training the model on the labeled nodes, and the second part represents the regularization based on Laplace, $\lambda > 0$ signifies the weight of two items. The labels of the nodes on the graph are propagated throughout the graph by this approach.

With the increasing requirement for semi-supervised node classification tasks on large graph datasets, researchers have proposed various graph convolutional networks [13–15]. For example, Kipf and Welling [3] proposed a scalable graph GCN model that can linearly scale the number of graph edges and learn the graph representation by encoding the local graph structures and nodes attributes. Yu et al. [16] proposed a double-convolution graph neural network to handle the classification of semi-supervised nodes.

2.2 Graph convolutional network

Graph Neural Networks are deep learning-based methods that function on graphs. Briefly offering an introduction to deep learning at first, deep learning is a representational learning method that exploits multilayer perceptron structures to process nonlinear information, transforming raw data into high-level, more abstract expressions through some simple, nonlinear multilayer representational models. The motivation is to build neural networks that can simulate the human brain for analytic learning and mimic the human brain to give interpretations of the data. Based on graph convolution and graph embedding in deep learning, graph neural networks are proposed to aggregate information about graph structures. This paper focuses on graph convolutional neural networks.

GCN is mainly utilized to deal with non-Euclidean graph data and usually adopts two strategies: spatial convolution and spectral convolution. Space-based methods update features by aggregating the spatially nearest neighbor vectors of each node, and in these methods the convolution operation is defined by specifying the neighborhood size, allowing information to be propagated locally. Recently, Jiang et al. [17] proposed utilizing a Gaussian mixture model to encode adjacent regions to solve the aggregation process.

Spectral convolution employs a convolution operator and then performs spectral filters by decomposing the graph Laplacian. According to spectral convolution, the attributes of the nodes are regarded as the signals in the graph and the convolution operations are directly applied to the spectrum of the graph.

$$g_\theta * x = U g_\theta(\Lambda) U^T x \tag{2}$$

Where $x \in \mathbb{R}^N$ is a signal and a filter $g(\theta) = \mathrm{diag}(\theta)$ parameterized by $\theta \in \mathbb{R}^N$ in the Fourier domain, the vectors U and Λ are generated from $L = D^{-1/2}(D - A)D^{-1/2}$ which is the graph Laplacian.

Unfortunately, the spectral method often requires singular value decomposition. For a large graph, it often leads to higher computational complexity. To reduce computational complexity, Defferrard et al. [18] truncated Chebyshev polynomials as:

$$g_\theta * x = \sum_{p=0}^{P} \theta'_p T_p(L) x \tag{3}$$

Where T_p and θ'_p represent Chebyshev polynomials and coefficients. Equation (3) was further simplified by Kipf and Welling [3] as:

$$g_\theta * x = \theta \left(I + D^{-1/2} A D^{-1/2} \right) x \tag{4}$$

Where I represents the identity matrix. Let $\tilde{A} = A + I$ and $\tilde{D}_m = \sum_j \tilde{A}_{nj}$, (4) can be rephrased as:

$$H = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} x \Theta \right) \tag{5}$$

where $\sigma(\bullet)$ represents the nonlinear activation function, such as $Softmax$ or $ReLU$. As illustrated in Fig. 1a, Kipf and Welling [3] proved that stacking two GCNs may produce outstanding performance, (5) becomes:

$$Z = f(X, A) = \mathrm{soft\,max} \left(\hat{A} \mathrm{Relu} \left(\hat{A} X \Theta^{(0)} \right) \Theta^{(1)} \right) \tag{6}$$

where $\hat{A}$ represents $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$. $\Theta^{(0)}$ and $\Theta^{(1)}$ will be obtained by minimizing the cross-entropy error of label nodes.

$$\ell = - \sum_{n \in V_l} \sum_{c=1}^{C} Y_{nc} \log (Z_{nc}) \tag{7}$$

where V_l represents the set of labeled nodes.

2.3 Topology optimization

Label propagation is to estimate the label of the new node through network topology, features, and the given labels. Li et al. [19] provided that GCN with the Laplacian smoothing term performs significantly improve than without. Yang et al. [11] declared a more obvious network structure could raise the performance of classification. Network topology plays a leading role in classification tasks. As illustrated in Table 1, a *F*ully *C*onnected *N*etwork (FCN) just employs node features, without graph convolution.

Table 1 Classification results with different topology

Dateset	FCN	GCN	TO-GCN	GCN-GT
Citeseer	57.1%	72.0%	72.7%	100%
Cora	56.2%	81.3%	83.1%	100%
PubMed	70.7%	79.2%	79.5%	100%

*G*raph *C*onvolutional *N*etworks with *G*round *T*ruth (GCN-GT) adopts the ground truth membership matrix, where a connection exists between two vertices if and only if they belong to the same classification and the attribute is equal to 1. The accuracy of classification is raised when the network structure becomes clearer. Therefore, topology optimization is an attractive measure to improve classification performance.

3 Optimization

The shortcoming of existing work motivates is to improve the topology of the network. Although existing work has utilized given labels to redefine network topology, it has not yet exploited estimated labels of classified nodes. Therefore, in this paper, we consider utilizing the given labels and the estimated labels to simultaneously and jointly refine the network topology. The flow chart of our proposed design is illustrated in Fig. 1c. In this section, we first introduce the existing network topology optimization methods, then Graph Convolution with Estimated Labels for Semi-supervised Node Classification (E-GCN) is presented.

3.1 Network topology refinement

Semi-supervised classification is mainly to classify nodes without labels by making full utilization of a limited number of node labels. The principle behind it is neighboring vertices in the graph are inclined to enjoy the same labels. Equation (8) is minimizing the objective function:

$$L(Y) = \frac{1}{2}\sum_{i,j} w_{ij}\left(y_i - y_j\right)^2 \tag{8}$$

where w_{ij} defines the similarity between the vertices v_i and v_j. Then consider the topology information and utilize the given labels to participate in the graph topology reconstruction. With the assumption that the adjacent vertices tend to share the same labels and guaranteed topology value to be non-negative, (2) can be transformed to:

$$\begin{aligned} &\min_Y \frac{1}{2}\sum_{i,j} a_{ij}\left\|y_i - y_j\right\|_2^2 = \min_Y \frac{1}{2}\mathrm{Tr}\left(Y^T L Y\right) \\ &\text{s.t. } \mathrm{y_n} = \mathrm{z_n} \forall v_n \in V_l \end{aligned} \tag{9}$$

where $z_n \in \{0,1\}^{1\times K}$ represents the true labels and $y_n \in \{0,1\}^{1\times K}$ is the predicted labels of the vertex v_n. $L = D - A$ is the Laplacian of the graph. Yang et al. [11] relaxed the Y to be $\mathbb{R}^{N\times K}$. Equation (9) can be relaxed to:

$$\min_Y \frac{1}{2}\mathrm{Tr}\left(Y^T L Y\right) + \frac{\lambda}{2}\|Y - Z\|_F^2 \tag{10}$$

According to Wang et al. [20], (10) equals to:

$$\min_Y \frac{1}{2}\mathrm{Tr}\left(LYY^T\right) + \frac{\lambda}{2}\left\|YY^T - ZZ^T\right\|_F^2 \tag{11}$$

To reduce the weight between nodes that don't belong to the same class, Yang et al. [11] proposed that if nodes come from different classes and have high similarity will be penalized. simultaneously, let $O = \left[O_{ij}\right] = YY^T \in \mathbb{R}^{N\times N}$ and $Q = \left[Q_{ij}\right] = ZZ^T \in \{0,1\}^{N\times N}$, (11) can be rewritten as:

$$\min_Y \frac{1}{2}\mathrm{Tr}(LO) + \frac{\lambda}{2}\|O - Q\|_F^2 + \frac{\alpha}{2}\sum_{i,j} O_{ij}\left\|z_i - z_j\right\|_2^2 \tag{12}$$

where $G = \left[G_{ij}\right] \in \{0,1\}^{N\times N}$, when the vertices v_i and v_j are of different classes, let $G_{ij} = 1$, (12) becomes:

$$\min_Y \frac{1}{2}\mathrm{Tr}((L + \alpha G)O) + \lambda\|O - Q\|_F^2 \tag{13}$$

Finally, the additional loss objective function with the given labels is:

$$\min_Y \varphi_{\text{refine}} = \min_Y \frac{1}{2}\mathrm{Tr}((L + \alpha G)O) + \lambda\|O - Q\|_F^2 \tag{14}$$

To maximize the exploration of network topology optimization in GCN, the latest research is to promote the classification of semi-supervised nodes through highly modular networks. However, the information contained in the classification nodes is ignored. Therefore, we propose a topologically optimized graph convolutional network simultaneously and jointly utilize the given labels and estimated labels together. The flow chart for our design is illustrated in Fig. 1c. The final objective function is:

$$\min_{\min} \varphi_{all} = \min_{\min} \varphi_{\text{classify}} + \varphi_{\text{refine}} + \varphi_{\text{e-classify}} + \varphi_{\text{e-refine}} \tag{15}$$

where the primary item represents classification loss and the next item represents topology refinement loss. The third term is the classification loss including the estimated labels and the fourth is the topology refinement loss with the estimated labels and the given labels.

3.2 Topology optimization process analysis

Compared with supervised learning, the biggest drawback of semi-supervised learning is that too little label information is accessible. In this paper, we utilize the validation set as a complement to the training set and apply estimated nodes labels to supplement the participating training label set. For the final objective function, we still employ the gradient descent method to minimize the loss function. The first term utilizes cross-entropy loss:

$$\min_{\min} \varphi_{\text{classify}} = \min -\sum_{n\in V_l}\sum_{c=1}^{c} Z_{nc}\log\left(H_{nc}\right) \tag{16}$$

The second term is an additional objective loss like (14), the gradient of L_{refine} relative to O is:

$$\frac{\partial \varphi_{\text{refine}}}{\partial O} = (L + \alpha G - \lambda Q) + \lambda O \tag{17}$$

where $L + \alpha G - \lambda Q$ can be computed in advance and remain fixed. However, the gradients $\varphi_{\text{classify}}$ and $\varphi_{\text{e-refine}}$ are dynamic. With the assistance of the gradient descent optimization of the cross-entropy loss of the first term, the prediction accuracy of the model for the validation set is constantly improved. When the probability of a certain category exceeds a set threshold, It considers this estimated label equal to the real label. we employ the classification results of the model on the validation set to construct the label matrix Z' of the validation set. It will approach the actual label value of the validation set with gradient optimization $\varphi_{\text{classify}}$.

To make a full explore of estimated labels, we've mainly worked on two aspects. On the one hand, the prediction value of the classification result of the validation set is compared with the threshold to obtain the estimated labels set and then calculate the cross-entropy $\varphi_{\text{e-refine}}$ of the estimated labels and prediction labels.

On the other hand, let $G' = \left[G'_{ij}\right] \in \{0, 1\}^{N\times N}$, when the vertices V_i and V_j are of different classes, let $G'_{ij} = 1$. We construct the additional objective loss of the validation set as:

$$\varphi'_{\text{e-refine}} = \min_{Y'} \frac{1}{2}\text{Tr}\left(\left(L + \alpha G'\right) Y' Y'^{T}\right) + \lambda \left\| Y' Y'^{T} - Z' Z'^{T} \right\|_F^2 \tag{18}$$

where Y' is to combine the prediction labels of the training set and the prediction labels of the validation set and then scale the elements to $R^{N\times K}$. Z' is to merge the given labels and the estimated labels Z_e and then scale the elements to $R^{N\times K}$.

Then, let $O' = \left[O'_{ij}\right] = Y' Y'^{T} \in \mathbb{R}^{N\times N}$ and $Q' = \left[Q'_{ij}\right] = Z' Z'^{T} \in \{0, 1\}^{N\times N}$, (18) can be rewritten to:

$$\varphi'_{\text{e-efine}} = \min_{Y'} \frac{1}{2}\text{Tr}\left(\left(L + \alpha G'\right) O'\right) + \lambda \left\| O' - Q' \right\|_F^2 \tag{19}$$

Finally, cumulate the four parts to generate the objective function we provide.

Compared with the existing semi-supervised classification methods, we make full explore of the node labels that have been classified and alleviate the problem of fewer labels in semi-supervised classification. What E-GCN has the advantages that related methods haven't is fully exploring estimated labels and the given labels. Analogous to GCN and TO-GCN, we can utilize the given labels Y and the estimated labels Y' to participate in the construction of the network topology simultaneously. Although the E-GCN objective function appears to be much more complex than the GCN objective function, the time complexity of both E-GCN and GCN is O(|E|CHF), where E is the number of edges and C denotes the dimension of X and H refers to the dimension of the hidden layer and F is the dimension of the output layer. E-GCN is more flexible and can simultaneously add multiple validation sets, which can effectively improve the accuracy of vertex classification.

It is worth noting that the effectiveness of our E-GCN depends to some extent on the accuracy of GCN classification. The more accuracy the GCN classification, the better the E-GCN performance.

4 Experiments

In this section, we will evaluate the performance of E-GCN via a large number of real experiments and provide some visualizations to help illustrate.

4.1 Datasets

For Cora, Citeseer, and Pubmed datasets, the vertices represent documents and edges represent undirected citations between documents. We follow the settings of prior experiments of Kipf and Welling [3]. We utilized the network topology, the features of all nodes, and the labels of 20 nodes of each class. We also adopted the Adam optimizer with a learning rate equal to 0. 1 and the regularization factor limited to 5e-4. All performance comparisons were performed on 1000 test sets and 500 nodes were used for verification. The statistics of the three databases are presented in (Table 2).

The papers in the CiteSeer dataset are divided into six broad categories, containing a total of 3312 papers, and information on the references and citations between the papers is recorded, while 3703 unique words are collated and used as a feature word dictionary for the papers after removing unused words and words that have a probability of occurring less than 10 times in the document.

Cora is a dataset of 2708 machine learning papers, which are grouped into seven different topics, each citing or cited by at least one other paper, if the paper is viewed as a vertex in the graph and the citation relationship is viewed as an edge, then the graph has a total of 2708 vertices and 5429

Table 2 Datasets statistics

Dateset	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

edges. For each paper, the papers are represented by a 1433-dimensional word vector, i.e., each paper has 1433 features, words Each element of the vector corresponds to a word and has only two values, 0 and 1, indicating whether the word present in the paper.

The Pubmed dataset consists of 19,717 diabetes-related papers from the PubMed database, divided into three categories. There are a total of 44,338 sets of citation relationships in the dataset. Each paper in the dataset is described by a word vector from a dictionary of 500 unique words.

4.2 Comparison with SOTA methods

Baselines We compare our E-GCN model with several SOTA approaches over three graph datasets, including GCN (Graph Convolutional Network) [3], TO-GCN (Topology Optimization based Graph Convolutional Network) [11], AGNN (Attention-based Graph Neural Network) [14], GAT (Graph Attention Network) [15], Chebyshev (Graph Convolution with Chebyshev Filters) [18], LP (Label Propagation) [21], SemiEmb (Semi-supervised embedding) [22], DeepWalk (Graph Embedding) [12], ICA (Iterative Classification Algorithm) [23], Planetoid (Graph-based Semi-supervised Learning Framework) [24], MoNet (Mixture Model Networks) [25], TAGCN (Topology Adaptive Graph Convolutional Networks) [26], and DGCN (Dual Graph Convolutional Networks) [27]. The codes of GCN have been published by authors and we utilized them in the experiment and made some modifications according to the requirements.

Results Table 3 summarizes the comparison results of three citation network datasets. E-GCN* represents that we only include the validation set in the training, while E-GCN indicates that we include both the validation set and the test volumes in the training, and the experimental results demonstrate that the combination of more data is beneficial for the performance of the model. Table 3 demonstrates higher prediction accuracy on the semi-supervised nodes classification of E-GCN by utilizing the estimated labels and the given labels to participate in. E-GCN consistently outperformed them on a node classification task for each of the three benchmark datasets over the best-performing approach achieves gains of up to 1.1%, 2.1%, and 0.7%.

4.3 Sensitivity analysis

We divide the data of the validation set into 10 groups and add them to the training set in batches, and then record the classification accuracy of the test set after each training is completed. As illustrated in Fig. 2, after adding the validation set data in batches, the accuracy of the Cora classification has increased significantly, while the classification accuracy of the Pubmed has increased at a limited rate. The performance improvement of the Pubmed is limited, because the structure is messy and has fewer features. The adaptability to the optimization of the network topology is very poor, and it is difficult to obtain the optimal learnable weights.

The same approach was followed to evaluate whether a large dataset would improve the efficiency of the model, using a similar method of joining all the data from the test set in batches for training. The experimental results are illustrated in the previous part of Fig. 3, and we could

Table 3 Node classification result

Methods	Cora	Citeseer	Pubmed
SemiEmb [22]	59.0%	59.6%	71.7%
LP [21]	68.0%	45.3%	63.0%
DeepWalk [12]	67.2%	43.2%	65.3%
ICA [23]	75.1%	69.1%	73.9%
Planetoid [24]	75.7%	64.7%	77.2%
Chebyshev [18]	81.2%	69.8%	74.4%
GCN [3]	81.5%	70.3%	79.0%
MoNet [25]	81.7%	69.9%	78.8%
TO-GCN [11]	83.1%	72.7%	79.5%
AGNN [14]	83.1%	71.7%	79.9%
TAGCN [26]	83.3%	72.5%	79.0%
DGCN [27]	83.5%	72.6%	80.0%
GAT [15]	83.0%	72.5%	79.0%
E-GCN*	84.0%	74.1%	80.2%
E-GCN	84.6%	74.8%	80.7%
GAIN	1.1%	2.1%	0.7%

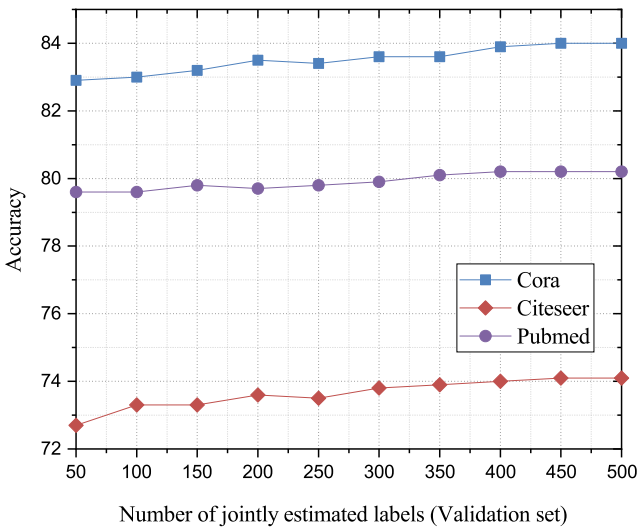

Fig. 2 The classification accuracy adding various numbers of jointly validation estimated labels

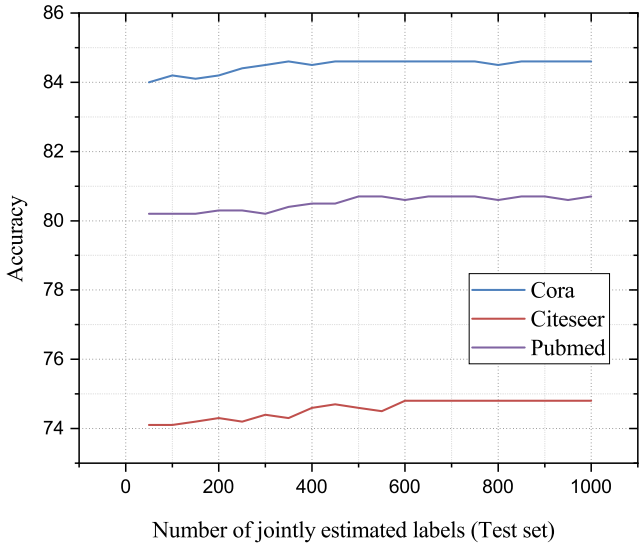

Fig. 3 The classification accuracy after adding various numbers of jointly test estimated labels

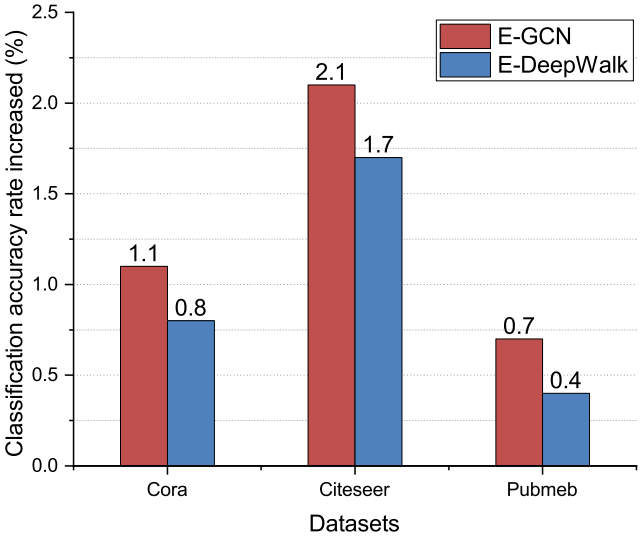

Fig. 4 Classification accuracy growth of different methods with estimated labels

observe a significant improvement in our classification performance when a large amount of data is included in the training. The experimental results demonstrate that the larger the amount of data to be included in the training of the E-GCN model, the better the efficiency of the model.

Meanwhile, we attempt to figure out the law of growth inaccuracy. The second half of Fig. 3 reveals that as we increase the number of datasets we join, the tendency for classification accuracy to rise becomes slower and slower, and eventually stays the same, demonstrating that classification accuracy does not rise significantly when we continue to gather more data.

4.4 Additional experiments

Here we provide a brief additional experiment to prove the validity of the E-GCN idea. Specifically, we deploy the idea of E-GCN on DeepWalk to add the validation set and test set of Cora and Citeseer to the training set in batches, with their estimated labels also adopting the same settings as E-GCN, and the experimental test results are illustrated in Table 4.

The results in Table 4, demonstrate that it is effective to employ the E-GCN idea to incorporate training with data to be tested in the future to boost the capabilities of the model. Where E-DeepWalk* means that we only participate in the training with data from the validation set, while E-DeepWalk means that all data from both the validation set and test set are joined to the training. The model's ability to effectively and consistently improve the model is improved when validation set data is appended until the final model's accuracy stabilizes with the addition of a partial test set.

We compare the classification accuracy growth of nodes with evaluation labels on GCN and DeepWork. As indicated in Fig. 4, it is evident that the effect of our proposed method is more satisfactory on GCN. It can be derived that the higher the classification accuracy of the original method, the more pronounced the improvement of the abilities of the model after incorporating training with evaluated labels.

Table 4 The classification accuracy of the deep walk with estimated labels

Dataset	E-DeepWalk* (%)				E-DeepWalk (%)				% GAIN
	25%	50%	75%	100%	25%	50%	75%	100%	
Cora	67.4%	67.6%	67.8%	67.7%	68.0%	68.0%	67.9%	68.0%	0.8%
Citeseer	43.5%	44.0%	44.3%	44.7%	44.9%	44.9%	44.9%	44.9%	1.7%
Pubmed	65.5%	65.4%	65.7%	65.7%	65.7%	65.7%	65.7%	65.7%	0.4%

5 Conclusion

In this paper, we propose a novel Graph Convolution with Estimated Labels for Semi-supervised Node Classification (E-GCN). E-GCN integrates both the given labels and the estimated labels into the topology learning of the graph. This method can learn the topology of the graph that is most suitable for GCN semi-supervised classification. Experimental results demonstrate that E-GCN is significantly better than existing GCN without using estimated labels.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61170089, No. 60971088).

References

1. Wei X, Xu L, Cao B, et al. (2017) Cross view link prediction by learning noise-resilient representation consensus. In: Proceedings of the 26th international conference on World Wide Web, pp 1611–1619
2. Gao M, Chen L, Li B, et al. (2018) A link prediction algorithm based on low-rank matrix completion. Applied Intelligence 48(12):4531–4550
3. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. arXiv:1609.02907
4. Ying R, He R, Chen K, et al. (2018) Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 974–983
5. Zhang C, Li T, Ren Z, et al. (2019) Taxonomy-aware collaborative denoising autoencoder for personalized recommendation. Applied Intelligence 49(6):2101–2118
6. Fortunato S, Hric D (2016) Community detection in networks: a user guide. Physics Reports 659:1–44
7. Tao Z, Liu H, Li S, et al. (2017) From ensemble clustering to multi-view clustering. In: IJCAI
8. Wang T, Liu L, Liu N, Zhang H, et al. (2020) A multi-label text classification method via dynamic semantic representation model and deep neural network. Applied Intelligence 50:2339–2351
9. Li M, Ma Z, Wang YG, et al. (2020) Fast Haar transforms for graph neural networks. Neural Networks
10. Wang YG, Li M, Ma Z, et al. (2020) Haar graph pooling. In: ICML
11. Yang L, Kang Z, Cao X, et al. (2019) Topology optimization based graph convolutional network. In: IJCAI, pp 4054–4061
12. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 701–710
13. Abu-El-Haija S, Kapoor A, Perozzi B, et al. (2018) N-gcn: multi-scale graph convolution for semi-supervised node classification. arXiv:1802.08888
14. Thekumparampil KK, Wang C, Oh S, et al. (2018) Attention-based graph neural network for semi-supervised learning. arXiv:1803.03735
15. Veličković P, Cucurull G, Casanova A, et al. (2017) Graph attention networks. arXiv:1710.10903
16. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv:1709.04875
17. Jiang J, Cui Z, Xu C, et al. (2019) Gaussian-induced convolution for graphs. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 4007–4014
18. Defferrard M, Bresson X (2016) Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems, pp 3844–3852
19. Li Q, Han Z, Wu XM (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI conference on artificial intelligence
20. Wang X, Qian B, Davidson I (2012) Labels vs. pairwise constraints: a unified view of label propagation and constrained spectral clustering. In: 2012 IEEE 12th international conference on data mining, pp 1146–1151
21. Zhu X, Ghahramani Z, Lafferty JD (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 912–919
22. Weston J, Ratle F, Mobahi H, et al. (2012) Deep learning via semi-supervised embedding. Neural networks: tricks of the trade. Springer, Berlin, pp 639–655
23. Lu Q, Getoor L (2003) Link-based classification. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 496–503
24. Yang Z, Cohen W (2016) Revisiting semi-supervised learning with graph embeddings. arXiv:1603.08861
25. Monti F, Boscaini D, Masci J, et al. (2017) Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5115–5124
26. Du J, Zhang S, Wu G, et al. (2017) Topology adaptive graph convolutional networks. arXiv:1710.10370
27. Zhuang C, Ma Q (2017) Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 World Wide Web conference, pp 499–508

Jisheng Qin received the M.S. degree in School of Computer Science and Technology from Anhui University of Technology. He is currently pursuing his PH.D. degree in the Institute of Intelligence Science and Technology, Hohai University, China. His main research interests include Machine Learning, Deep Learning and Neural Network.

Xiaoqin Zeng received the Ph.D. degree from Hong Kong Polytechnic University, the M.S. degree from Southeast University, and the B.S. degree from Nanjing University, all in Computer Science. He currently is a professor and Ph.D. student supervisor in Hohai University, China. His research interests include Computational Intelligence, machine learning, pattern recognition, and graph grammars.

Shengli Wu received the Ph.D. degree in computer science from Southeast University, China. He is a lecturer in the School of Computing, Ulster University, UK. His current research interests include database and information retrieval, scient metrics and citation analysis, and machine learning.

E Tang received the M.S. degree in Computer Science and Technology from Nanjing University of Science and Technology. She is currently pursuing her Ph.D. degree in the Institute of Intelligence Science and Technology, Hohai University, China. She is a lecturer with the College of Computer and Software Engineering, Huaiyin Institute of Technology, China. Her current research focuses on machine learning and image processing.