Robust Semi-Supervised Representation Learning for Graph-Structured Data

Lan-Zhe Guo, Tao Han, and Yu-Feng Li

National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{guolz,hant,liyf}@lamda.nju.edu.cn

Abstract. The success of machine learning algorithms generally depends on data representation and recently many representation learning methods have been proposed. However, learning a good representation may not always benefit the classification tasks. It sometimes even hurt the performance as the learned representation maybe not related to the ultimate tasks, especially when the labeled examples are few to afford a reliable model selection. In this paper, we propose a novel robust semi-supervised graph representation learning method based on graph convolutional network. To make the learned representation more related to the ultimate classification task, we propose to extend label information based on the smooth assumption and obtain pseudo-labels for unlabeled nodes. Moreover, to make the model robust with noise in the pseudo-label, we propose to apply a large margin classifier to the learned representation. Influenced by the pseudo-label and the large-margin principle, the learned representation can not only exploit the label information encoded in the graph-structure sufficiently but also can produce a more rigorous decision boundary. Experiments demonstrate the superior performance of the proposal over many related methods.

Keywords: Robust · Representation Learning · Semi-Supervised Learning · Graph Convolutional Network

1 Introduction

The performance of machine learning methods is heavily dependent on the choice of data representation (or features). Representation learning, i.e., learning representations of the data that needed for the learning classifiers, has already become an important field in machine learning [2].

One challenge of representation learning is that it faces a paradox between preserving as much information about the input as possible, and attaining nice properties for the output learning task [5]. Recently, there are many researches pointed out that, the representation learning may fail to improve the performance of classification task [2, 5]. The main reason is that the learned representation may be far from the ultimate learning task. For example, representation learning typically pursuits the whole factors of the raw data, while the ultimate

task may only be related to a small subset of these factors. For this reason, it is essential to learn a robust representation, especially when the labeled examples are few to afford a reliable model selection.

In this paper, we focus on learning a robust representation for semi-supervised graph-structured data. It is widely accepted that graph-structured data occurs in numerous application domains, such as social networks [14], citation networks [9] and many others [7]. Learning an appropriate vector representation of nodes in graphs has proved extremely useful for a wide variety of predictive and graph analysis tasks [6, 14, 16]. Figure 1 illustrates a visualization of a classical graph-structured data and the corresponding node embeddings. A number of graph representation learning methods such as DeepWalk [14], LINE [16], have been proposed recently. However, these methods require a multi-step pipeline where the representation learning model and the classifier are trained separately. In other words, the learned representation may be far from the classification task, and thus hurt the performance.

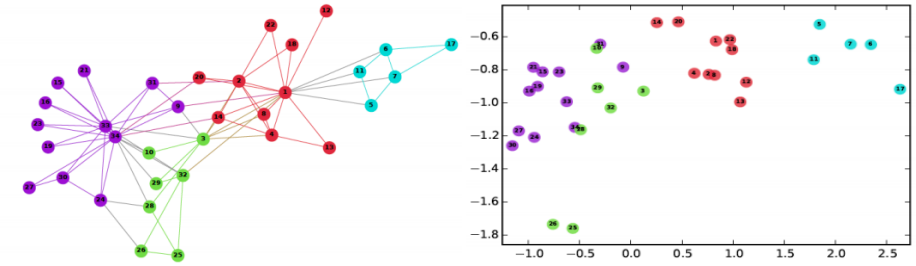

Fig. 1. Graph structure of the Zachary Karate Club social network (left) and the two dimensional visualization of node embeddings (right).

Most recently, Graph Convolutional Network (GCN) [9] is proposed to fill the gap. Unlike previous studies where the learned representation and the trained classifier are conducted separately, GCN jointly optimizes the representation learning model and the ultimate classifier. Nevertheless, most ultimate classifiers in GCN work under the labeled data, which is insufficient to learn a robust representation in semi-supervised learning.

In this paper we propose to obtain high-confidence pseudo-labels for unlabeled nodes from the well-known label propagation strategy to enhance the label capacity. Our basic idea is that given graph-structured data, many label information are encoded in the graph structure based on the smooth assumption [22], i.e., connected nodes are likely to share similar labels. We further propose a large-margin classifier to overcome the noise pseudo-labels induced from label propagation. Figure 2 shows the pipeline of the proposal.

In conclusion, we make several noteworthy contributions as follows:

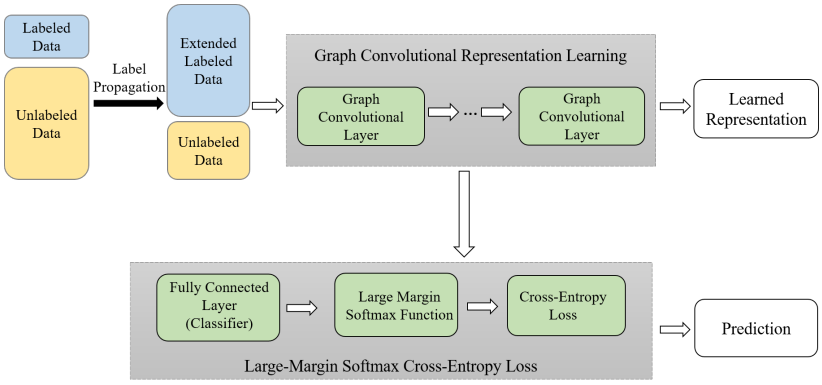

Fig. 2. The structure of the proposed RoGraph.

- We propose a robust representation learning method RoGraph for semi-supervised graph-structured data, with the idea of the classical label propagation and large margin principle, which is very easy to implement.
- Experiments on real-world network datasets are conducted. The experimental results demonstrate that RoGraph achieves clearly better results than many related methods.

The rest of this paper is organized as follows. We first introduce related works and present preliminaries with an introduction to GCN. Next, we present the proposed RoGraph, and then show the experimental results and discuss why the large-margin principle can benefit graph representation learning. Finally, we conclude this paper.

2 Related Work

The proposed algorithm is conceptually related to semi-supervised graph representation learning and large-margin learning methods.

Semi-supervised graph representation learning Inspired by the Skip-Gram [13], many semi-supervised learning methods for graph-structured data have been proposed in recent years. DeepWalk [14] learns embeddings via the prediction of the local neighborhood of nodes, sampled from random walks on the graph. There are also many works based on DeepWalk, such as LINE [16] extends DeepWalk with more sophisticated random walk and node2vec [6] extends DeepWalk with breadth-first search schemes. For all these methods, however, a multi-step pipeline including random walk generation and semi-supervised training is required where each step has to be optimized separately, so the learned representation may not the best representation for the classification task. Planetoid [20] alleviated this by injecting label information in the process of learning embeddings. GCN [9] generalize traditional convolutional networks to graph-structured data and can learn representations end-to-end.

Large-margin learning methods There have been many large-margin learning methods in many fields. Max-margin markov network [17] firstly introduced the large-margin principle into markov networks. MedLDA [21] proposed a maximum entropy discrimination LDA to learn a discriminative topic model (e.g., latent Dirichlet allocation [3]). LEAD [10] adopted the large-margin principle to judge the quality of graph. MMDW [18] combined large-margin loss function with DeepWalk to learn discriminative network representations.

It is notable that to our best knowledge, the large-margin principle has rarely been applied to graph-based methods. We show that large-margin principle is indeed helpful for robust graph representation.

3 Preliminaries

In this section, we introduce some notations used in our method and give a brief introduction to the idea of graph convolutional networks.

3.1 Notations

We consider the problem of representation learning on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The given information includes a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ which x_i is a feature description for every node i, N is the number of nodes and M is the dimension of input features; an adjacency matrix $\mathbf{A} = A_{ij} \in \mathbb{R}^{N \times N}$, where $A_{ij} = A_{ji} = 1$ if node i and node j has a link, otherwise $A_{ij} = A_{ji} = 0$; a labeling matrix $\mathbf{Y} \in \mathbb{R}^{N \times K}$ with K being the number of classes. In the setting of semi-supervised learning, we have set $\mathcal{Y}_L$ which includes all labeled nodes and set $\mathcal{Y}_U$ which includes all unlabeled nodes. The size of $\mathcal{Y}_L$ is much smaller than the size of $\mathcal{Y}_U$. The learned representation is matrix $\bar{\mathbf{X}} \in \mathbb{R}^{N \times F}$, where F is the dimension of output feature per node. The prediction is matrix $\mathbf{Z} \in \mathbb{R}^{N \times K}$, where Z_{ij} indicates the probability that node i belongs to class j.

3.2 Graph Convolutional Networks

Graph convolutional network [9] generalizes the convolutional network into graph-structured data and proposes an efficient layer-wise propagation rule. The traditional GCN model contains the following components:

(1) Renormalization: Adding an self-loop to each node, which results in a new adjacency matrix $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ where $\mathbf{I}$ is the identity matrix and the new degree matrix $\tilde{\mathbf{D}}$ with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. After that, symmetrically normalize $\tilde{\mathbf{A}}$ and obtain $\tilde{\mathbf{A}}_s = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$.

(2) Graph Convolutional layer: The graph convolutional layer uses the propagation rule:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}_s \mathbf{H}^{(l)} \mathbf{W}^{(l)}) \tag{1}$$

where $\mathbf{H}^{(l)}$ is the matrix of activations in the l-th layer and $\mathbf{H}^{(0)} = \mathbf{X}$, $\mathbf{H}^{(L)} = \bar{\mathbf{X}}$ with L is the number of layers in the network, $\mathbf{W}^{(l)}$ is a layer specific trainable

weight matrix in layer l, and $\sigma(\cdot)$ denotes an activation function, such as the $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$.

(3) Softmax cross-entropy loss: Applying a fully connected layer as the classifier to the learned representation $\bar{\mathbf{X}}$: $\mathbf{Z} = \bar{\mathbf{X}}\mathbf{W}^L$, where $\mathbf{W}^L \in \mathbb{R}^{F\times K}$ is a trainable weight matrix of the fully connected layer. Then evaluates the softmax cross-entropy loss over labeled nodes. The loss can be written as:

$$\mathcal{L} = \sum_{i\in\mathcal{Y}_L} -\ln(\frac{e^{Z_{iy_i}}}{\sum_j e^{Z_{ij}}}) \tag{2}$$

where $\mathcal{Y}_L$ is the set of labeled nodes and y_i is the label of the i-th node.

4 Our Proposed Method

In semi-supervised learning, the number of labeled nodes is usually limit to provide a reliable model selection, thus, the learned representation using only the labeled data may be not robust for the ultimate classification task.

Observed that in graph-structured data connected nodes are likely to share the same label, we propose to assign a high-confidence pseudo label for some unlabeled nodes according to this property so that we can exploit more label information encoded in the graph structure and enhance labeled data. Moreover, the large-margin principle is often used to train a robust classifier, thus, we propose to adopt the large-margin softmax cross-entropy loss function [12] instead of the original softmax function to help decrease the impact of noise in the pseudo-label and produce even more robust representation.

4.1 Enhance Labeled Data

The underlying assumption in graph-based semi-supervised learning is the smooth assumption, i.e., connected nodes likely to share the same label. With this assumption, we can exploit more label information using the graph structure information and produce a classification task related representation.

A simple method to mine the label information encoded in the graph structure for unlabeled nodes is label propagation [22]. The label propagation method only takes the graph matrix $\mathbf{A}$ and the labeling matrix $\mathbf{Y}$ as input and the objective is to find a prediction matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times K}$ of the same size as the labeling matrix $\mathbf{Y}$ by minimizing both fitting error and smooth regularization:

$$\begin{aligned}\hat{\mathbf{Y}} &= \arg\min_{\hat{\mathbf{Y}}} \mathcal{C}(\hat{\mathbf{Y}}) \\ &= \arg\min_{\hat{\mathbf{Y}}} \{\underbrace{||\hat{\mathbf{Y}} - \mathbf{Y}||_2^2}_{\text{fitting error}} + \alpha \underbrace{\mathrm{tr}(\hat{\mathbf{Y}}^\top \mathbf{L}\hat{\mathbf{Y}})}_{\text{regularization}}\} \\ &= \arg\min_{\hat{\mathbf{Y}}} \{\sum_{i\in\mathcal{Y}_L} (\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2 + \alpha \sum_{i,j}^{n} A_{ij}(\hat{\mathbf{Y}}_i - \hat{\mathbf{Y}}_j)^2\}\end{aligned} \tag{3}$$

where $\mathbf{L}$ is the graph laplacian matrix.

In Eq.(3), the fitting error term enforces the prediction matrix $\hat{\mathbf{Y}}$ to agree with the label matrix $\mathbf{Y}$, while the smooth regularization term enforces each column of $\hat{\mathbf{Y}}$ to be smooth along the edges. The scalar α is a balancing parameter.

A closed-form solution of the unconstrained quadratic optimization problem can be obtained by setting the derivative of the objective function to zero:

$$\hat{\mathbf{Y}} = (\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{Y} \tag{4}$$

For small-scale data, we can simply use Eq.(4) to get the prediction of unlabeled nodes. However, for large-scale data, Eq.(4) is time consuming because it needs to compute the inverse of the matrix $\mathbf{I} + \alpha\mathbf{L}$. To address this problem, we use Stochastic Gradient Descent (SGD) to solve Eq.(3).

Let

$$\mathcal{C}_i(\hat{\mathbf{Y}}) = \alpha \sum_{j=1}^{n} A_{ij}(\hat{\mathbf{Y}}_i - \hat{\mathbf{Y}}_j)^2 + \mathbb{I}(i) \cdot (\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2 \tag{5}$$

where $\mathbb{I}(i) = 1$ if $i \in \mathcal{Y}_L$, otherwise, $\mathbb{I}(i) = 0$, $i \in \{1, 2, \cdots, N\}$ represents the index of the chosen instance. It is easy to verify that $\mathbb{E}[\nabla\mathcal{C}_i(\hat{\mathbf{Y}})] = \frac{1}{n}\nabla\mathcal{C}(\hat{\mathbf{Y}})$.

Therefore $\nabla\mathcal{C}_i(\hat{\mathbf{Y}})$ is an unbiased estimator of $\frac{1}{n}\nabla\mathcal{C}(\hat{\mathbf{Y}})$ where $\frac{1}{n}$ is a constant given a graph. Hence, we can adopt SGD strategy to solve $\hat{\mathbf{Y}}$ by updating:

$$\hat{\mathbf{Y}}^{(t+1)} = \hat{\mathbf{Y}}^{(t)} - \eta\nabla\mathcal{C}_i(\hat{\mathbf{Y}})$$

where the gradient $\nabla\mathcal{C}_i(\hat{\mathbf{Y}}) = \alpha\mathbf{A}_i^{\top}(\hat{\mathbf{Y}}_i - \hat{\mathbf{Y}}_j) + \mathbb{I}(i) \cdot (\hat{\mathbf{Y}}_i - \mathbf{Y}_i)$. We adopt the interesting stochastic label propagation method [11] to derive the gradient efficiently. The element $\hat{Y}_{ij}$ in the learned prediction matrix $\hat{\mathbf{Y}}$ indicates the probability of node i belongs to class j. For an unlabeled node i, if the maximum Y_{ij} in $\hat{\mathbf{Y}}_i$ is greater than a threshold, we think the node i has a high confidence in class j and add node i to the labeled node set $\mathcal{Y}_L$. After this process, we derive a larger labeled node set $\tilde{\mathcal{Y}}_L$ to learn a better representation.

4.2 Large-Margin Cross-Entropy Loss

Obviously the pseudo-label derived by the label propagation may consist of noise. To make the learned representation robust, it is necessary to overcome the affect caused by noise. Intuitively, if the decision boundary has a large margin to the nearest training data point, the model turns out to be a robust classifier according to margin theory. An additional benefit is that the large margin principle works as a regularizer and thus help avoid overfitting issues, which is particularly useful when labeled data is limited. Thus, to learn a robust representation [12], we use a generalization of the original softmax cross-entropy loss function termed Large-Margin Cross-Entropy Loss.

Observed in GCN that $Z_{ij} = \bar{\mathbf{X}}_i\mathbf{W}_j^L$. This can be reformulated as: $Z_{ij} = \|\mathbf{W}_j^L\|\|\bar{\mathbf{X}}_i\|\cos(\theta_j)$ where θ_j is the angle between the vector $\mathbf{W}_j^L$ and $\bar{\mathbf{X}}_i$. Thus

the original softmax cross-entropy loss function can be rewritten as:

$$L_i = -\ln(\frac{e^{\|\mathbf{W}_{y_i}^L\|\|\bar{\mathbf{X}}_i\|\cos(\theta_{y_i})}}{\sum_j e^{\|\mathbf{W}_j^L\|\|\bar{\mathbf{X}}_i\|\cos(\theta_j)}}) \tag{6}$$

In RoGraph, we propose to use large-margin softmax instead of the original softmax. For example, once an instance x with the label $+1$, the original softmax is to force $\mathbf{W}_1^\top \mathbf{x} > \mathbf{W}_2^\top \mathbf{x}$, i.e., $\|\mathbf{W}_1\|\|\mathbf{x}\|\cos(\theta_1) > \|\mathbf{W}_2\|\|\mathbf{x}\|\cos(\theta_2)$, in order to classify x correctly. In contrast, the large-margin softmax want to make the classification more rigorous in order to produce a large-margin decision boundary. Thus, the large-margin softmax requires $\|\mathbf{W}_1\|\|\mathbf{x}\|\cos(m\theta_1) > \|\mathbf{W}_2\|\|\mathbf{x}\|\cos(\theta_2)(0 \le \theta_1 \le \frac{\pi}{m})$ where m is a positive integer.

Observed that the following inequality always holds:

$$\|\mathbf{W}_1\|\|\mathbf{x}\|\cos(\theta_1) \ge \|\mathbf{W}_1\|\|\mathbf{x}\|\cos(m\theta_1) > \|\mathbf{W}_2\|\|\mathbf{x}\|\cos(\theta_2) \tag{7}$$

Therefore, we have $\|\mathbf{W}_1\|\|\mathbf{x}\|\cos(\theta_1) > \|\mathbf{W}_2\|\|\mathbf{x}\|\cos(\theta_2)$. Therefore, the new classification criteria correctly classifies x, and produces a more rigorous decision boundary. Figure 3 illustrates a geometric interpretation for the advantage of the large-margin softmax function [12].

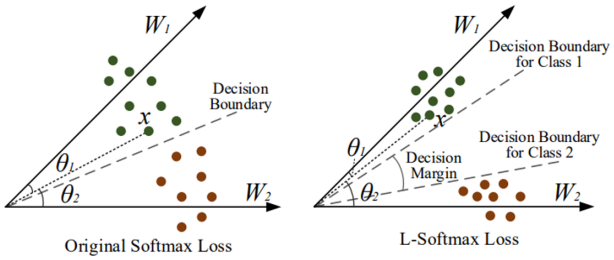

Fig. 3. Illustrative geometric interpretation. The left (right) presents the original softmax (large-margin softmax).

Formally, the large-margin softmax cross-entropy loss is defined as:

$$L_i = -\ln(\frac{e^{\|\mathbf{W}_{y_i}^L\|\|\bar{\mathbf{X}}_i\|\phi(\theta_{y_i})}}{e^{\|\mathbf{W}_{y_i}^L\|\|\bar{\mathbf{X}}_i\|\phi(\theta_{y_i})} + \sum_{j\neq y_i} e^{\|\mathbf{W}_j^L\|\|\bar{\mathbf{X}}_i\|\phi(\theta_j)}}) \tag{8}$$

where

$$\phi(\theta) = \begin{cases} \cos(m\theta), & 0 \le \theta \le \frac{\pi}{m} \\ \mathcal{D}(\theta), & \frac{\pi}{m} \le \theta \le \pi \end{cases}$$

and $\mathcal{D}(\theta)$ is a monotonically decreasing function, $\mathcal{D}(\frac{\pi}{m}) = \cos(\frac{\pi}{m})$.

According to the work in [12], we let $\phi(\theta) = (-1)^k \cos(m\theta) - 2k, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ where $k \in [0, m-1]$. The m is related to the classification margin. The larger

the value m is, the larger the classification margin becomes. Then, we have the final loss function

$$\text{L-Softmax}(\bar{\mathbf{X}}, \mathbf{W}^L) = \sum\nolimits_{i \in \tilde{\mathcal{Y}}_L} L_i \tag{9}$$

where $\tilde{\mathcal{Y}}_L$ is the set of extended labeled nodes after label propagation.

In short summary, the proposed RoGraph includes these steps: we first propagate the label information from the labeled nodes to unlabeled nodes using the graph structure. Then, we renormalize the adjacency matrix and use graph convolutional layers to produce the representation of each node. Finally, we add a fully connected layer to the learned representation as the classifier and adopt the large-margin cross-entropy loss to derive the final representation.

5 Experiments

In this section, we evaluate the proposed RoGraph in benchmark network datasets and show the effectiveness of our proposal. Besides, we give the loss vs. epoch in both training set and validation set.

5.1 Experimental Setup

Cora, CiteSeer and PubMed [15] are three benchmark network datasets. The statistics of datasets are summarized in Table 1. In these networks, nodes are documents and edges are citation links. Each document is represented by a sparse 0/1 feature vector. Citation links between documents constitute a 0/1 undirected graph. If v_i cites v_j or vice versa, then $A_{ij} = A_{ji} = 1$, otherwise $A_{ij} = A_{ji} = 0$. Each document has a class label. For training, we only use 20 labels per class and all feature vectors for each dataset.

Table 1. The statistics of experimental network datasets.

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

For the used datasets, we trained RoGraph with two graph convolutional layers and a fully connected layer and evaluate the prediction accuracy on a test set of $1,000$ labeled examples. We train our models on all three datasets for a maximum of 200 epochs using Adam [8] with a learning rate 0.01, 0.5 dropout rate, 5×10^{-4} weight decay rate and early stopping with a window size of 10. We use a hidden layer of 16 units and we initialize weights using the Xavier initialization [4]. The threshold of assigning a pseudo-label to unlabeled nodes is set to 0.8 for all the experiments. The parameter m is fixed to 2 for the large-margin softmax loss on all the experiments.

5.2 Compared Results

We compared the proposed RoGraph with many state-of-the-art methods [9], including label propagation (LP) [22], semi-supervised embedding (SemiEmb) [19], manifold regularization (ManiReg) [1], skip-gram based graph embeddings (DeepWalk) [14], iterative classification algorithm (ICA) [15] and Planetoid [20].

We further compare against with conventional GCN. For GCN, the hyper-parameters are same with RoGraph. To validate the effectiveness of the two proposed technologies separately, we also compare with two variants of RoGraph, i.e., RoGraph-P (RoGraph with pseudo-label only) and RoGraph-M (RoGraph with L-Softmax only).

The experimental results are summarized in Table2. For GCN, RoGraph-P, RoGraph-M, RoGraph, we reported the mean accuracy over five random splits. Results for all other methods are taken from [9].

From Table 2, we can see that, on all the datasets, RoGraph achieves a clear performance gain over the GCN method. It demonstrates the effectiveness of the proposed RoGraph. In addition, RoGraph-P and RoGraph-M also achieve better results than GCN but are not as good as RoGraph. It indicates that the two introduced technologies (label propagation and large-margin principle) are both useful to robust representation.

Table 2. Summary of results in terms of classification accuracy.

Method	Citeseer	Cora	Pubmed
MainReg	60.1	59.5	70.7
SemiEmb	59.6	59.0	71.1
LP	45.3	68.0	63.0
DeepWalk	43.2	67.2	65.3
ICA	69.1	75.1	73.9
Planetoid	64.7	75.7	77.2
GCN	69.6	80.8	77.8
RoGraph-P	72.4	82.4	77.8
RoGraph-M	71.4	83.2	78.1
RoGraph	**73.1**	**83.8**	**79.1**

5.3 The Confidence of Assigning Pseudo-Label

In this section, we show the confidence of assigning a pseudo-label to unlabeled nodes after label propagation. For each unlabeled node i, The confidence of assigning a pseudo-label is the maximum number in $\hat{Y}_i$ after row-normalization. The results for all three datasets are shown in Figure 4. From Figure 4, we can see that even we set the threshold as 0.8, we can still give a pseudo-label to about half of the unlabeled nodes. This demonstrates that the label-propagation process does help us exploit more label information encoded in the graph structure.

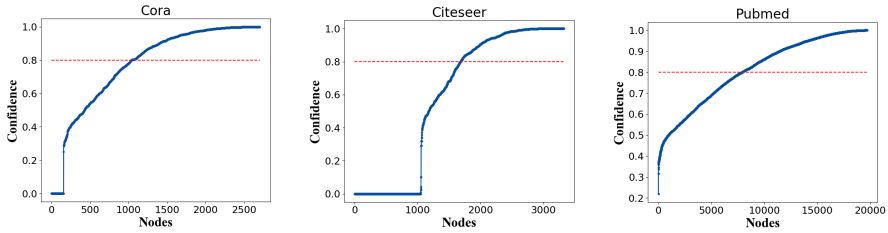

Fig. 4. The confidence of assigning a pseudo-label to unlabeled nodes.

5.4 Loss vs. Epoch

Figure 5 illustrates the relationship between the loss and the epoch on Cora dataset (On the other two datasets, we achieve similar results). One can see that the proposed RoGraph not only achieves the lowest loss in both training set and validation set but also needs fewer epochs to converge than traditional GCN, though we adopt a harder loss function. This is consistent with the numerical results in Table 2 and also verifies the effectiveness of the proposal.

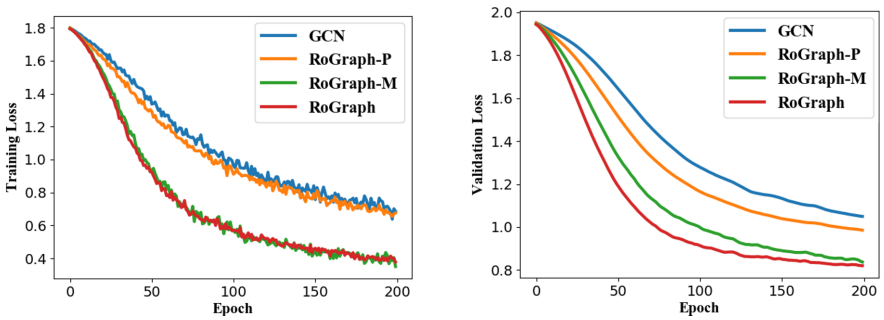

Fig. 5. Loss vs. Epoch on Cora. The left (right) presents training (validation) loss.

6 Discussion

Graph representation learning has attracted significant attention in recent years. Meantime, the large-margin principle is also widely used to train a robust classifier and help avoid overfitting issues. However, to our best knowledge, the large-margin principle has rarely been applied to graph representation learning. In this paper, we successfully fill this blank by applying the large-margin principle to GCN and show the effectiveness through empirical results. It demonstrates that the large-margin principle can work well with graph-based methods.

We think that one key reason for why the large-margin principle can work well with graph-based methods is that, the underlying assumption for the large-margin principle (large-margin assumption) and graph-based methods (manifold assumption or smooth assumption) are kind of complementary. Specifically, the manifold assumption requires the learned representation of nodes in the same classes are similar to each other. It emphasizes that the data closeness within the same classes, whereas ignores the data separability between different classes. By contrast, the large-margin assumption requires the learned representation of nodes between different classes have a large margin. It emphasizes the data separability between different classes but ignores the data closeness. Therefore, by taking the two assumptions into account simultaneously, one can encourage both the inter-class separability and intra-class compactness between learned representations for graphs and leads to a better decision boundary. It is innovative for the algorithm design of graph representation learning.

7 Conclusion

We have introduced a novel and easy-to-implement approach RoGraph for robust semi-supervised representation learning on graph-structured data. RoGraph leverages label propagation to obtain high-confidence pseudo-label for unlabeled nodes, which can exploit label information encoded in the graph structure sufficiently. Besides, we adopt the large-margin softmax cross-entropy loss function instead of the traditional softmax function to produce a more rigorous decision boundary. Both of these technologies can help produce a robust representation. Experiments on a number of benchmark network datasets suggest that the proposed RoGraph achieves better results than many state-of-the-art methods. In future, we will extend the proposed strategy to edge representation rather than only node representation in this work.

Acknowledgments This research was supported by the National Key R&D Program of China (2017YFB1001903) and the National Natural Science Foundation of China (61772262).

References

1. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research pp. 2399–2434 (2006)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1798–1828 (2013)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research pp. 993–1022 (2003)
4. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010)

5. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning. MIT press Cambridge (2016)
6. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864 (2016)
7. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. IEEE Data Engineering Bulletin pp. 52–74 (2017)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
10. Li, Y.F., Wang, S.B., Zhou, Z.H.: Graph quality judgement: a large margin expedition. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. pp. 1725–1731 (2016)
11. Liang, D.M., Li, Y.F.: Lightweight label propagation for large-scale network data. In: Proceedings of 27th International Joint Conference on Artificial Intelligence. pp. 3421–3427 (2018)
12. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: Proceedings of the 33rd International Conference on Machine Learning. pp. 507–516 (2016)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 701–710 (2014)
15. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine p. 93 (2008)
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077 (2015)
17. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Advances in Neural Information Processing Systems. pp. 25–32 (2004)
18. Tu, C., Zhang, W., Liu, Z., Sun, M.: Max-margin deepwalk: Discriminative learning of network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. pp. 3889–3895 (2016)
19. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade, pp. 639–655 (2012)
20. Yang, Z., Cohen, W.W., Salakhutdinov, R.: Revisiting semi-supervised learning with graph embeddings. In: Proceedings of the 33rd International Conference on Machine Learning. pp. 40–48 (2016)
21. Zhu, J., Ahmed, A., Xing, E.P.: Medlda: maximum margin supervised topic models. Journal of Machine Learning Research pp. 2237–2278 (2012)
22. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning. pp. 912–919 (2003)