Information Sciences 514 (2020) 484–498

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

HesGCN: Hessian graph convolutional networks for semi-supervised classification

Sichao Fu[a], Weifeng Liu[a,*], Dapeng Tao[b], Yicong Zhou[c], Liqiang Nie[d]

[a] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China
[b] School of Information Science and Engineering, Yunnan University, Kunming 650091, China
[c] Faculty of Science and Technology, University of Macau, Macau 999078, China
[d] School of Computer Science and Technology, Shandong University, Qingdao 266237, China

ARTICLE INFO

Article history:
Received 7 August 2019
Revised 10 October 2019
Accepted 13 November 2019
Available online 13 November 2019

Keywords:
Graph convolutional networks
Hessian
Semi-supervised learning
Manifold assumption

ABSTRACT

Manifold or local geometry of samples have been recognized as a powerful tool in machine learning areas, especially in the graph-based semi-supervised learning (GSSL) problems. Over recent decades, plenty of manifold assumption-based SSL algorithms (MSSL) have been proposed including graph embedding and graph regularization models, where the objective is to utilize the local geometry of data distributions. One of most representative MSSL approaches is graph convolutional networks (GCN), which effectively generalizes the convolutional neural networks to deal with the graphs with the arbitrary structures by constructing and fusing the Laplacian-based structure information. However, the null space of the Laplacian remains unchanged along the underlying manifold, it causes the poor extrapolating ability of the model. In this paper, we introduce a variant of GCN, i.e. Hessian graph convolutional networks (HesGCN). In particularly, we get a more efficient convolution layer rule by optimizing the one-order spectral graph Hessian convolutions. In addition, the spectral graph Hessian convolutions is a combination of the Hessian matrix and the spectral graph convolutions. Hessian gets a richer null space by the existence of its two-order derivatives, which can describe the intrinsic local geometry structure of data accurately. Thus, HesGCN can learn more efficient data features by fusing the original feature information with its structure information based on Hessian. We conduct abundant experiments on four public datasets. Extensive experiment results validate the superiority of our proposed HesGCN compared with many state-of-the-art methods.

1. Introduction

With the rapid development of the computer network technology and the diversification of obtaining data methods, the information that people can obtain is growing rapidly in the form of an index. How to effectively utilize these massive data to accelerate the development of social productive force is one of the common challenges for global researchers and technical experts at present. Semi-supervised learning (SSL) [1–4] is an important sub-field in machine learning and plays a key role in modern intelligent technology.

* Corresponding author.
E-mail address: liuwf@upc.edu.cn (W. Liu).

https://doi.org/10.1016/j.ins.2019.11.019
0020-0255/

Currently, the graph-based semi-supervised learning (GSSL) [5–8] is the mainstream method in the SSL. In this field, GSSL mainly uses the graph to describe the space structure of data. Under the framework of the graph theory [9–12], it can construct a model with excellent performance and generalization ability on the data with missing a lot of label information. In GSSL, how to better preserve the manifold or local geometry structure of samples is extremely important. The manifold assumption reflects the local smoothness of the discriminant function, i.e. the samples in the same local neighborhood should be similar. Under this assumption, the role of the massive unlabeled samples is to more accurate describe the distribution characteristics of the local neighborhood, so that the discriminant function fits better. Manifold assumption can be easily applied to the GSSL algorithms. In the past few decades, many manifold assumption-based SSL methods have been successfully applied to the computer vision [13–16] and machine learning areas [17–20].

A large number of manifold assumption-based SSL (MSSL) algorithms have been proposed and have been divided into two parts, i.e. methods based on the graph regularization or graph embedding.

In the graph regularization methods of MSSL, it utilizes the local geometry structure of all unlabeled and labeled samples to increase the generalization ability of the training model. Zhu et al. [4] proposed a label propagation (LP) algorithm, i.e. it first used the relationships between the samples to establish a complete graph model, and then predicted the label information of the unlabeled nodes from the label information of the labeled nodes. Belkin et al. [21] exploited the manifold structure of the data probability distributions and took it as an additional regularization term of the loss function, which is used to control the complexity of data distributions. Weston et al. [22] generalized the non-linear embedding algorithms that applied for the shallow SSL to the deep-layer networks, which can be used for the regularization term or the network architectures.

Graph embedding is a process of mapping the graph structured data (high-dimensional dense matrix) into the low-dimensional dense vector, which aims to solve the problem that graph structured data is difficult to input into the machine learning algorithms efficiently. Perozzi et al. [23] took the sequence of the nodes obtained by random walk as a sentence, and then obtained the local information of the network from the truncated random walk sequence, which can learn the potential representation of the nodes through the local information. Wang et al. [24] compared the edge generated by the generator (the negative sample of the generator) with the observable edge of the networks (the positive sample of the discriminator), i.e. the trained generator approximated the first-order information of the networks. Hamilton et al. [25] used an inductive method to compute the node representations, i.e. the model first extracted a fixed number of nodes from the neighbor nodes of each node, and then used a specific way to fuse the information of these neighbor nodes.

Because the traditional discrete convolution cannot maintain the translation invariance of the non-Euclidean data, traditional convolutional neural networks is unable to process these data. However, there exists a large number of graph structured data in real life. In the past year, a novel graph structured data representation learning method has obtained more and more attentions of the researchers, i.e. GCN [26], which aimed to extract the spatial features of graph structured data in the spectral domain. Kipf and Welling [26] obtained the sample structure information based on the graph Laplacian by the one-order approximation of spectral graph filter. And then, it made a fusion for original feature information and the structure information of the samples by the convolution operation. Finally, it can get the richer sample features. (GCN only utilized the structure information between each node and its direct neighbours). Fu et al. [27] utilized the spectral graph filter with two-order polynomials to capture the direct and indirect relationships between nodes. Yadati et al. [28] used hypergraph Laplacian to describe the complex space structure relationships between different nodes, not the single pairwise connection relationships. However, due to the null space of the Laplacian is a constant function about geodesic distance, it causes the poor extrapolation ability for unseen data [29]. In other words, GCN cannot learn more representative data features because of the lack of richer structure information.

To overcome the above-mentioned issue, we propose the Hessian graph convolutional networks (HesGCN) for the graph structured data representation and classification problem. We use the Hessian [30–32] matrix to describe the manifold distribution of the nodes. And then, we can get the spectral graph Hessian convolutions by applying it to the spectral convolutions on graphs. Finally, we build a deep-layer HesGCN though the optimized convolution layer rule for semi-supervised classification. Because the linear relationships between the null space of the Hessian and geodesic distance function, Hessian can more accurate preserve the local manifold distribution information of the data [33]. HesGCN can improve the semi-supervised classification results effectively in the case of fewer labeled samples by the convolution of input sample features and structure information based on Hessian. We compare the HesGCN with the GCN to demonstrate the effectiveness of the HesGCN. Extensive experiment results on the Citeseer, Cora, Pubmed and NELL datasets verify the performance of the HesGCN.

The contribution of this paper can be summarized as the following some folds:

(1) By the effective combination of the graph Hessian matrix and spectral graph convolutions, this paper proposes a spectral graph Hessian convolutions to address the issue that graph convolutional networks (GCN) fails to better preserve and utilize the local geometry structure relationships between samples.
(2) This paper further acquires a novel form of the convolution layer rule via the optimization of one-order approximation of spectral graph Hessian convolutions.
(3) To evaluate the semi-supervised classification performance of the proposed Hessian graph convolutional networks (HesGCN), this paper conducts extensive experiments on the citation networks datasets.

Table 1
Some important mathematic notations.

Notation	Description
M	The sub-manifold of the M-dimensional data in R^d
$C^\infty(M)$	The set of smooth functions about M
$S_\Delta(f)$	Laplacian regularizer
$S_{Hess}(f)$	Hessian regularizer
X	The set of N samples: $X = \{x_i\}_{i=1}^n$
$dV(x)$	The natural volume element
N_s	The null space of $S_\Delta(f)$: $N_s = \{f \in C^\infty(M) \mid S(f) = 0\}$
f	A function f: $M \to R$ with $f \in C^\infty(M)$
g_θ	Spectral graph filter
θ or $W^{(L)}$	Filter parameters or weight matrix of each layer
L	Graph Laplacian: $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ with $D_{ii} = \sum_j A_{ij}$
Hes	Graph Hessian
A	Local geometry relationships between samples
$\lambda_{\max}$	The largest eigenvalue of matrix
$H^{(L+1)}$	The extracted sample features of each layer
$\widetilde{L}$	$\widetilde{L} = \frac{2}{\lambda_{\max}}L - I_N$
$\widetilde{A}$	$\widetilde{A} = A + I_N$
$T_k(x)$	$T_0(x) = I_N, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$

The rest of this paper is organized as follows. Section 2 reviews the related works. Sections 3 and 4 describe the proposed HesGCN algorithm in detail. The experimental results are discussed and analyzed in Section 5, followed by the conclusion in Section 6.

2. Realted works

In this section, we first give a brief introduction of the null space description of the Hessian matrix. And then, we briefly summarize the basic terms and concepts of GCN [26]. To better understand this paper for readers, we give some important mathematic notations in Table 1.

2.1. The null space description of the Hessian matrix

Let M denotes the sub-manifold of the M-dimensional data in R^d, which also is known as the Euclidean space. In addition, we denote the $C^\infty(M)$ as the set of smooth functions about M. Suppose X is a set of N samples $X = \{x_i\}_{i=1}^n$. The definition of the Laplacian regularizer $S_\Delta(f)$ is as follows [29,34]:

$$S_\Delta(f) = \int_M \| \bigtriangledown f\|^2 dV(x) \tag{1}$$

Here, $dV(x)$ represents the natural volume element [35]. f is a real-valued function, i.e. f: $M \to R$ with $f \in C^\infty(M)$. However, these are only the constant functions about M for the null space N_s of the Laplacian regularizer $S_\Delta(f)$, i.e. $N_s = \{f \in C^\infty(M) \mid S(f) = 0\}$. Because of all the functions meet the condition $S(f) = 0$, thus the functions are not penalized. In addition, the S: $C^\infty(M) \to R$ is the regularization function.

To solve the above-mentioned limitation of the Laplacian regularizer $S_\Delta(f)$, a new method has been proposed, i.e. Hessian regularizer $S_{Hess}(f)$. It can be defined as the following expression:

$$S_{Hess}(f) = \int_M \sum_{r,s=1}^{m} \left(\frac{\partial^2 f}{\partial x_\gamma \partial x_s} \right)^2 dV(x) \tag{2}$$

Obviously, from the definition, we can see that, the Hessian regularizer $S_{Hess}(f)$ does the second covariant derivative on f. In addition, Eells and Lemaire [33] has been given the proof process of the following proposition. From the proposition, we can know that the null space of the $S_{Hess}(f)$ is richer compared to the $S_\Delta(f)$.

Proposition 1 (Eells and Lemaire [33]). *A function f: $M \to R$ with $f \in C^\infty(M)$ has zero second derivative, $\bigtriangledown_a \bigtriangledown_b f|_x = 0$, $\forall x \in M$, if and only if for any geodesic $\gamma : (-\varepsilon, \varepsilon) \to M$ parameterized by arc length s, there exists a constant c_γ depending on γ such that*

$$\frac{\partial}{\partial s} f(\gamma(s)) = c_\gamma \quad \forall -\varepsilon < s < \varepsilon \tag{3}$$

Here, the functions f, which is called the geodesic functions, satisfy the condition $\frac{\partial}{\partial s} f(\gamma(s)) = const$. It corresponds to the linear map relationships in the Euclidean space.

Owing to the null space of the Laplacian regularizer $S_\Delta(f)$ are constant functions with regard to the geodesic distance [29], thus it does not extrapolate exactly for unseen data or poor extrapolation capability. However, the relationship between

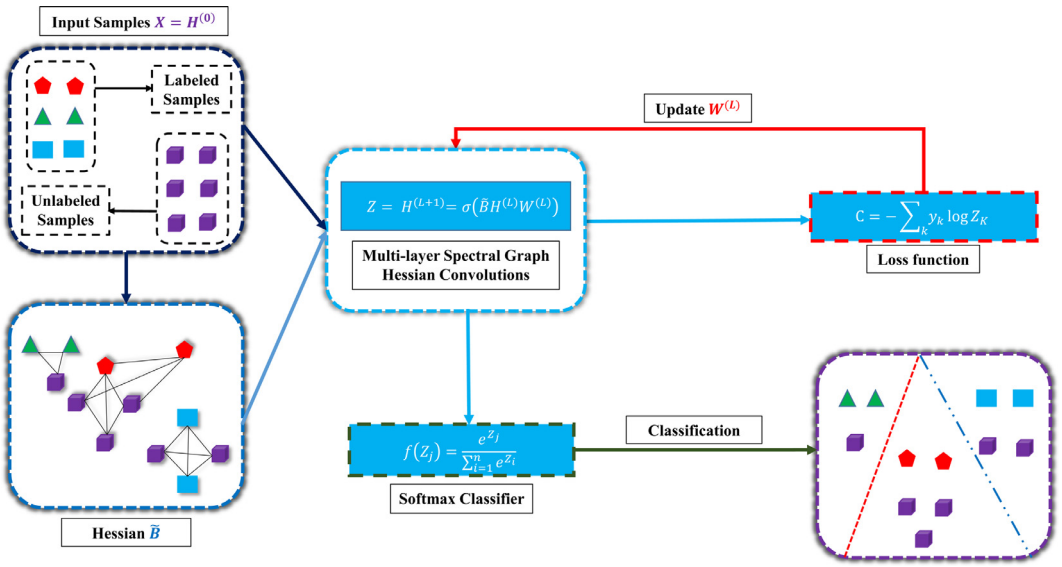

Fig. 1. The framework of the multi-layer HesGCN model for semi-supervised classification.

the geodesic distance and the null space of the $S_{Hess}(f)$ is linear, i.e. geodesic functions f exist, so the $S_{Hess}(f)$ can fit the data that beyond data points perfectly. In other words, due to the richer null space of the $S_{Hess}(f)$, Hessian can better describe the local geometry of data. Fig. 2 shows the differences of the graph Laplacian and graph Hessian in the local geometry preserving or the null space description of data. Thus, it can get the richer structure information of the data. GCN can extract the richer sample features by fusing the richer structure information and feature information of the data.

So far, Liu et al. [36] exploited the Hessian to better describe the intrinsic manifold structure of the data. Liu and Tao [37] proposed multiview Hessian regularization to exploit the complementary information of the different view. Feng et al. [38] explored graph Hessian matrix to encode the data manifold and proposed Hessian regularized multitask dictionary learning for remote sensing image recognition.

In the following, we briefly describe the computational process of the Hessian. First, we construct the k nearest neighbor matrix A^i for each sample. Then, we get the tangent coordinates U of the sample through the singular value decomposition on A^i. Next, we obtain a Hessian matrix Hes^i through the least-squares estimation about U. Finally, we get a symmetric Hessian matrix Hes_{ij} by the accumulation of the Hes^i. The detailed process can be found in [36,37].

2.2. Spectral graph Laplacian convolutions

Recently, a new model, i.e. GCN [26], has caused wide public concern in the graph structured data representation and classification. GCN generalized the convolutional neural network from the regular spatial structure data to the irregular graph structured data. Now, we briefly describe the related works of the GCN model.

Hammond et al. [39] proposed the initial definition of the spectral convolutions on graph domain, which is equal to the convolution of the signal X and spectral graph filter g_θ. However, this method is only applicable for the small graph structured data because of the eigendecomposition of the graph Laplacian. To solve the above problem, Defferrard et al. [39] optimized this method by introducing the Chebyshev polynomials. Finally, the definition used the following form:

$$g_\theta(L) \star X = \sum_{K=0}^{K} \theta_K T_K(\overrightarrow{L})X \tag{4}$$

Here, Defferrard et al. [39] used the normalized graph Laplacian to describe the manifold structure of the data, i.e. $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, $D_{ii} = \sum_j A_{ij}$. The A expresses the adjacent relationships matrix and the I_N represents the identity matrix. $\overrightarrow{L}$ is calculated by the $\overrightarrow{L} = \frac{2}{\lambda_{\max}}L - I_N$. $\lambda_{\max}$ is the graph Laplacian's largest eigenvalue. θ is the parameters matrix of the spectral graph filter. This definition reduced the computational complexity of the model for the large graph structured data. In addition, it considered the K-hop area of each sample. Defferrard et al. [39] constructed the model by the spectral graph Laplacian convolutions with K-order polynomials and provided the proof process of the above mentioned definition.

To further reduce the model's computational complexity and the radius of the receptive field, Kipf and Welling [26] only used the direct neighborhood of each node in each convolution layer, i.e. $K = 1$. Finally, it proposed the linear layer-wise aggregate rule according to the one-order spectral graph convolutions, i.e.

$$H^{(L+1)} = \sigma(\overrightarrow{D}^{-\frac{1}{2}}\overrightarrow{A}\overrightarrow{D}^{-\frac{1}{2}}H^{(L)}W^{(L)}) \tag{5}$$

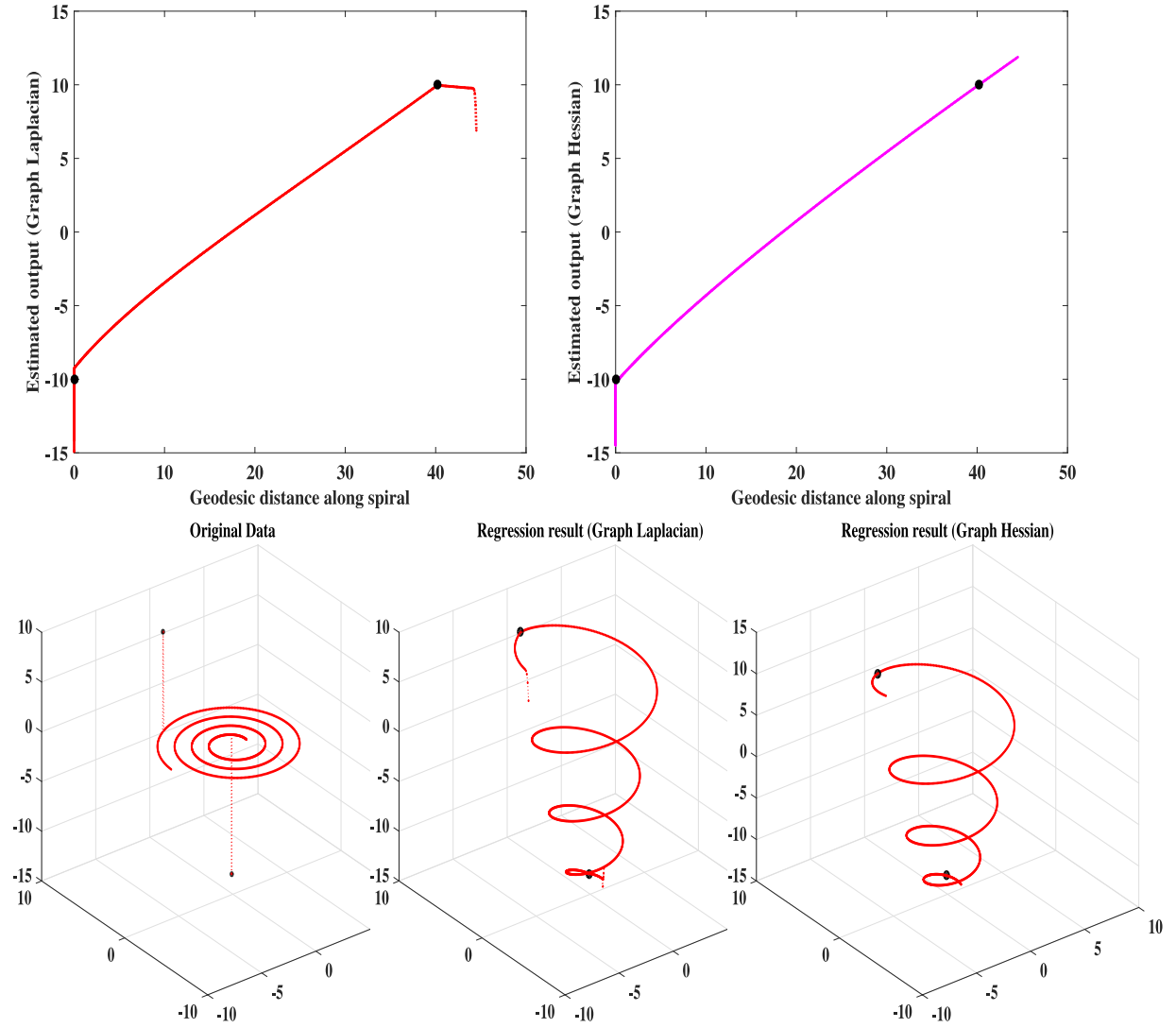

Fig. 2. Differences of the semi-supervised regression for fitting two points on the one-dimensional spiral by respectively utilizing graph Laplacian and graph Hessian matrix to preserve the manifold structure information of data distributions. In the local geometry preserving of data, graph Laplacian always biases the solution towards a constant function which cannot extrapolate exactly to unseen data. Different from graph Laplacian, graph Hessian varies linearly with the geodesic distance which can fit the data perfectly.

Here, σ is an activation function. At present, the commonly-used σ is rectified linear unit function (RELU), i.e. $f(x) = \max(0, x)$. $\overrightarrow{D}^{-\frac{1}{2}}\overrightarrow{A}\overrightarrow{D}^{-\frac{1}{2}}$ expresses the structure information based on the graph Laplacian matrix, $\overrightarrow{D}_{ii} = \sum_j \overrightarrow{A}_{ij}$. $\overrightarrow{A} = A + I_N$ are the spatial adjacency relationships matrix including self-connections. $H^{(L)}$ are the input sample features of each convolution layer. $H^{(L+1)}$ are the output sample features that fuse the underlying graph structure information with the input sample features. A multi-layer spectral graph convolutions-based model can be built by stacking the layer-wise aggregate rule, which is named GCN [26].

3. Spectral graph Hessian convolutions

In this section, first of all we describe the motivation of our proposed convolution layer rule, i.e. we generalize the spectral graph Laplacian convolutions to the spectral graph Hessian convolutions by utilizing the Hessian matrix to preserve the manifold structure of the samples. And then, we introduce a novel convolution layer rule by optimizing the one-order approximation of the spectral graph Hessian convolutions. In the following, we introduce the works of the proposed spectral graph Hessian convolutions in detail.

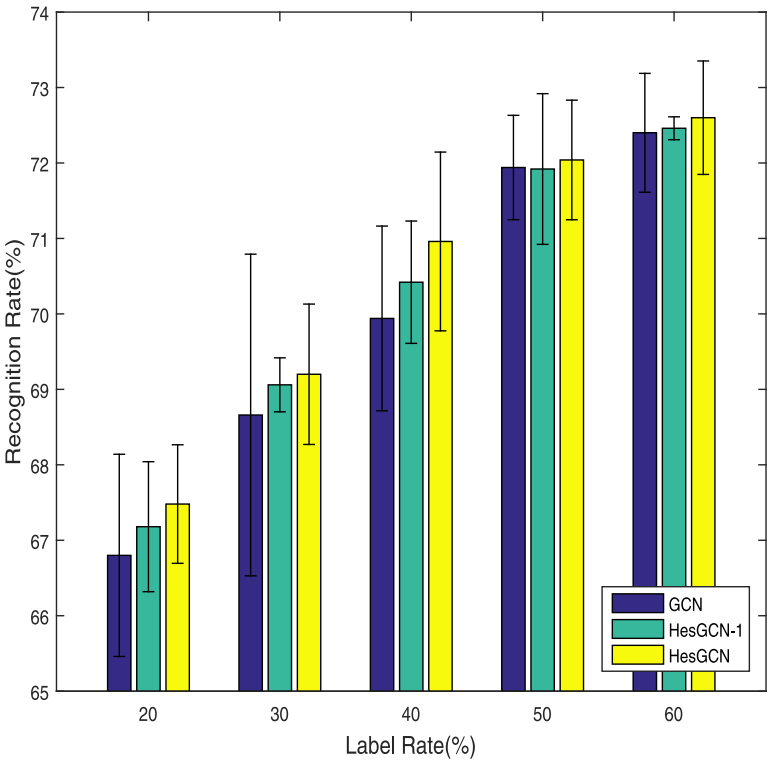

Fig. 3. Recognition accuracy of all classes in the Citeseer database.

Table 2
The experiment datasets.

Datasets	Samples	Classes	Citation-Links	Dimensions
Pubmed	19,717	3	44,338	500
Citeseer	3327	6	4732	3703
Cora	2708	7	5429	1433
NELL	65,755	210	266,144	5414

GCN fused the manifold structure relationships based on the graph Laplacian in the convolution process of each layer. Because the poor null space of the graph Laplacian, it is unable to accurately describe the manifold structure of the samples, i.e. GCN cannot learn the most representative sample features in the imbedded process of the structure relationships and feature information. To overcome this problem, we describe the space manifold structure information by substituting the Hessian matrix for the graph Laplacian. And then, we make an extension for the spectral convolutions on graph domain and get a novel definition, i.e. spectral graph Hessian convolutions.

$$g_\theta(Hes) \star X = \sum_{K=0}^{K} \theta_K T_K(\overrightarrow{Hes})X \tag{6}$$

Here, Hes denotes the graph Hessian matrix. In addition, in the computational process of the Hessian, we exploit the Euclidean distance-based k-Nearest Neighbor method to calculate A. $\overrightarrow{Hes} = \frac{2}{\lambda_{\max}} Hes - I_N$. $\lambda_{\max}$ expresses the graph Hessian Hes matrix's the biggest eigenvalue. $T_K(\overrightarrow{Hes})$ can be shown through the following form: $T_0(\overrightarrow{Hes}) = I_N, T_1(\overrightarrow{Hes}) = \overrightarrow{Hes}, T_k(\overrightarrow{Hes}) = 2\overrightarrow{Hes}T_{k-1}(\overrightarrow{Hes}) - T_{k-2}(\overrightarrow{Hes})$. In addition, because the Hessian and Laplacian matrix have been calculated in pre-processing, Eqs. (6) and (4) has the same computation complexity $O(K|\varepsilon|)$.

To increase the computational efficiency of the HesGCN model and inspired by the experience of the deep learning, we also use the one-order localized approximation of the spectral graph Hessian convolutions, i.e. $K = 1$. (In other words, spectral graph Hessian convolutions with one-order polynomial can well capture the structure information between nodes). In addition, with the diversification of adjacent relationships matrix A computing methods and the differences of different

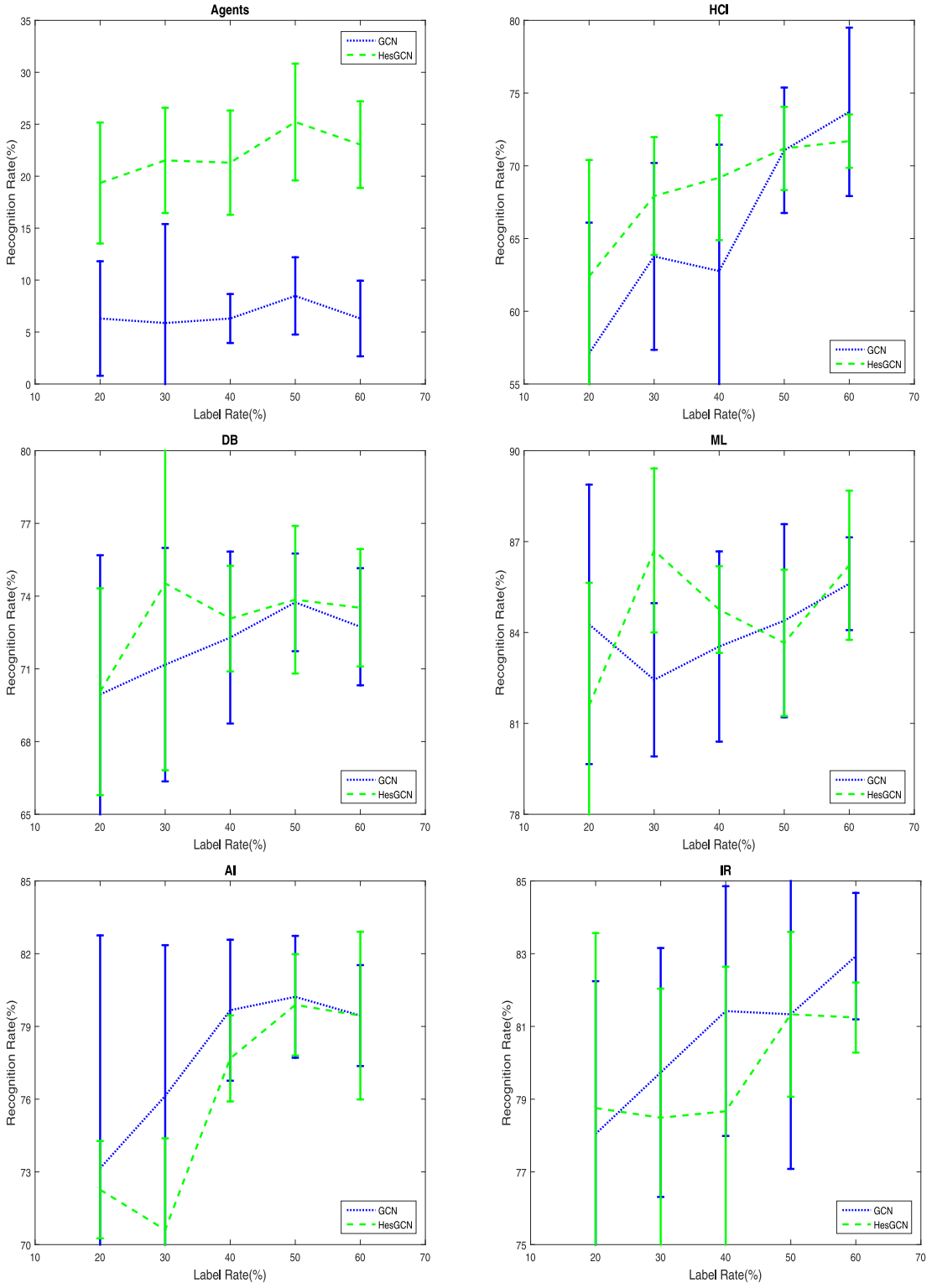

Fig. 4. Recognition accuracy of each class in the Citeseer database, including Agents, AI, DB, IR, ML, HCI. Each subfigure corresponds on single class.

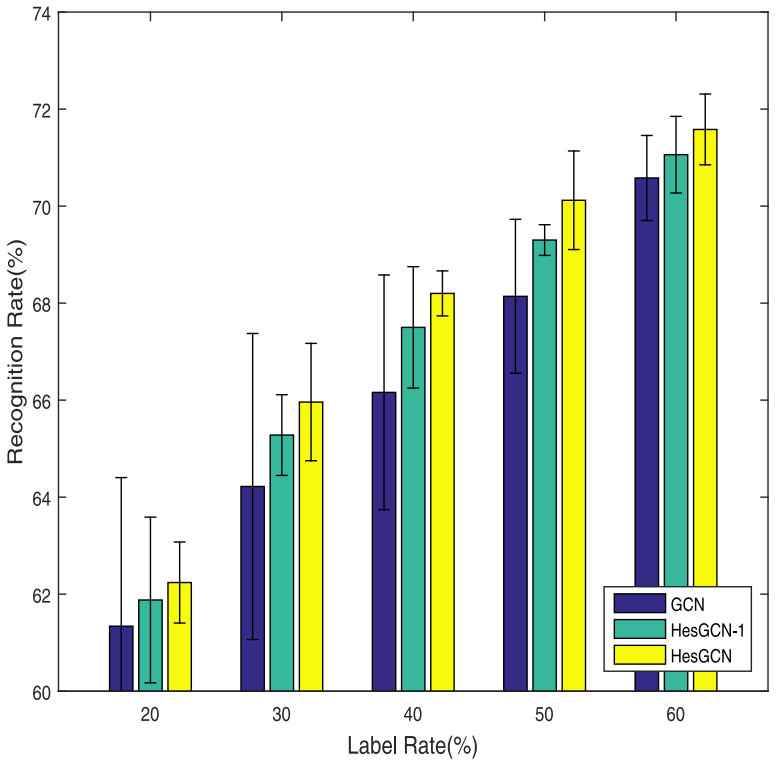

Fig. 5. Recognition accuracy of all classes in the Cora database.

datasets, $\lambda_{\max} = 2$ is not optimal. Thus, the definition can be written the following form and it is named HesGCN-1.

$$g_\theta(Hes) \star X = \theta_0 X + \theta_1 \left(\frac{2}{\lambda_{max}} Hes - I_N \right) X \tag{7}$$

Here, the above definition need two filter parameters θ_0 and θ_1 in each convolution layer. However, in the practice, the successive convolution of this filter will cause the overfitting and increase the operation numbers of the model. To surmount the above problem, we optimize the method as the following expression. It is called HesGCN, which is known as the convolution layer rule.

$$Z = g_\theta(Hes) \star X = \theta \left(\frac{2}{\lambda_{max}} Hes - I_N \right) X \tag{8}$$

In this definition, the $\frac{2}{\lambda_{\max}} Hes - I_N$ denotes the Hessian-based samples structure information matrix, which is an $N \times N$ symmetric matrix. X is the sample input feature matrix in each layer. Z is the extracted sample feature information.

4. Multi-layer Hessian GCN

Based on Eq. (8), our proposed convolution layer rule *f(X, W, Hessian)* is also defined as the following expression, i.e.

$$H^{(L+1)} = \sigma \left(\left(\frac{2}{\lambda_{max}} Hes - I_N \right) H^{(L)} W^{(L)} \right) \tag{9}$$

Fig. 1 describes the basis model framework of multi-layer HesGCN for semi-supervised classification. In this paper, to validate the performance of our proposed method, we use the convolution layer rule to design a two-layer HesGCN networks model. First, we need to do some preparatory work, i.e. the structure information matrix $\overrightarrow{B} = \frac{2}{\lambda_{\max}} Hes - I_N$. And then, we use the following expression for the convolution of the first layer.

$$H^{(1)} = RELU(\overrightarrow{B} H^{(0)} W^{(0)}) \tag{10}$$

In the first layer, it makes a learning for the feature information $H^{(0)}$ and structure information $\overrightarrow{B}$ simultaneously by the convolution. Specially, $H^{(0)}$ is the initial sample feature matrix. $W^{(0)}$ is the networks model weight parameters of the first layer. We also use the most powerful RELU function. After the first layer, HesGCN can extract the sample features of the first layer, i.e. $H^{(1)}$.

We use the output $H^{(1)}$ of the first convolution layer to constitute the sample input matrix of the second layer. The convolution operation of the second layer is same as the last layer, which also need a preparatory work $\overrightarrow{B}$. And then, we get the network form of this layer, i.e.

$$H^{(2)} = \overrightarrow{B} H^{(1)} W^{(1)} \tag{11}$$

Here, $W^{(1)}$ expresses the weight parameter matrix in the second layer. By the convolution operation of this layer, we can get the final sample features, i.e. $H^{(2)}$. It makes a fusion for the input sample $H^{(1)}$ and local geometry structure information $\overrightarrow{B}$.

After the convolution operation of the two layer, the final part is to make a prediction for the sample, i.e. the classifier uses the training data with given category to learn classification rules, and then predicts the unknown data. In this paper, we use the Softmax classifier, which is a generation of the logistic function. It compresses a k-dimensional vector Z containing any real number into another k-dimensional real vector $\sigma(Z)$, so that the range of each element in the vector is between zero and one that the sum of all elements is 1. The Softmax function is given through the following form, i.e.

$$f(Z_j) = \frac{e^{Z_j}}{\sum_{i=1}^{n} e^{Z_i}} \tag{12}$$

Here, we take the final sample features as the input of the Softmax function, i.e. $H^{(2)}$. To get the best classification performance of the model, we use the cross entropy loss function to measure the error of the model, i.e.

$$C = -\sum_{K} y_K \log Z_K \tag{13}$$

Here, the Z_K denotes probability vector matrix, i.e. the output of the Softmax function. y_K is the true label information of the sample. In addition, the networks weight $W^{(0)}$ and $W^{(1)}$ are updated through the gradient descent method. We summarize the experiment process of the two-layer HesGCN in Algorithm 1.

Algorithm 1 Two-layer Hessian GCN model.

Input: Data X
Parameter: Dropout rate, learning rate etc.
Output: Mean classification accuracy

1: Construct adjacency matrix A between data X.
2: Compute structure information $\overrightarrow{B}$.
3: Initialize the hyperparameters.
4: **for** $j = 0 \rightarrow k - 1$
5: $H^{(1)} = RELU(\overrightarrow{B} H^{(0)} W^{(0)})$
6: $H^{(2)} = \overrightarrow{B} H^{(1)} W^{(1)}$
7: **until convergence**
8: Get the optimal $W^{(0)}$ and $W^{(1)}$.
9: Return the mean classification accuracy of data.

5. Experiments

In this section, to demonstrate the effectiveness of the proposed HesGCN, we utilize the HesGCN model for semi-supervised classification on four real-life datasets including Citeseer [40], Cora [41], Pubmed [42] and NELL [43]. And then we make the comparison between it and many state-of-the-art algorithms, such as GCN [26], HyperGCN [28], graph attention network (GAT) [42] and so on. In the following, we give a detailed description of the experiment datasets, the experiment parameters setting and the experiment results of HesGCN in turn.

5.1. Datasets

The Citeseer [40] dataset consists of 3327 scientific publications from six classes, such as Agents, ML, HCI, AI, DB and IR. Each scientific publication contains 3703 different words, which is described by the number zero and one. This dataset contains 4732 citation relationships between the different scientific publications.

The Cora [41] dataset contains 2708 scientific books collected from seven categories. The category attributes of the Cora contain case-based, rule-learning, theory, genetic-algorithms, neural-networks, probabilistic-methods and reinforcement-learning. Each book is described by 1433 distinct words. There consists of 5429 citation links between 2708 publications.

The Pubmed [42] dataset is composed of 19,717 diabetes publications with three class attributes including diabetes mellitus type 2, diabetes mellitus experimental and diabetes mellitus type 1. The size of each publication is 500-dimensional vectors. The citation relationships of all diabetes books are 44,338.

The NELL [43] dataset consists of totally 65,755 nodes collected from the knowledge graph [44]. It is divided into 210 categories. Each node has the 5414 dimensions features. The briefly description of all datasets are shown in Table 2.

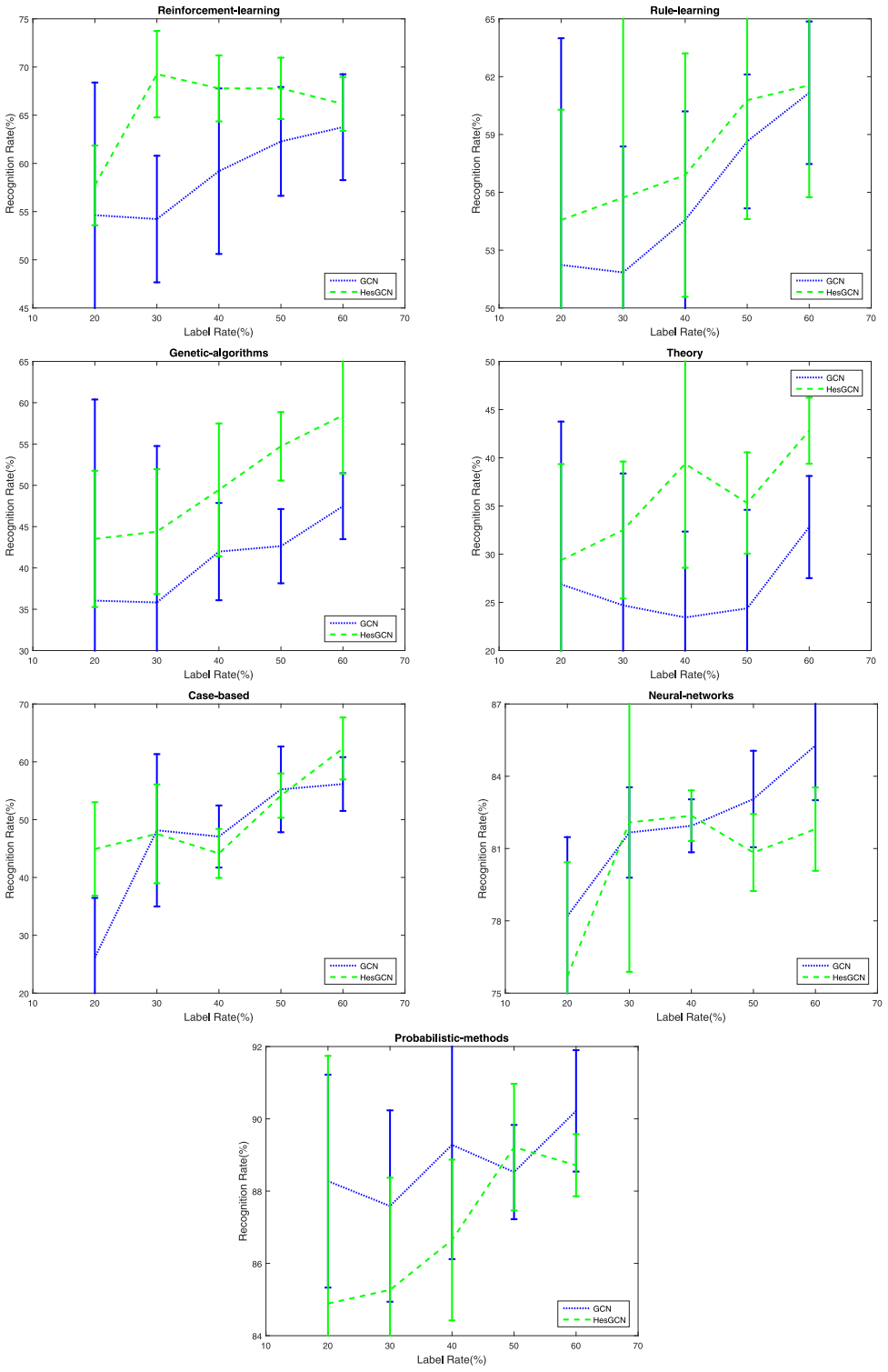

Fig. 6. Recognition accuracy of each class in the Cora database, including Theory, Case-based, Genetic-algorithms, Neural-networks, Probabilistic-methods, Reinforcement-learning, Rule-learning. Each subfigure corresponds on single class.

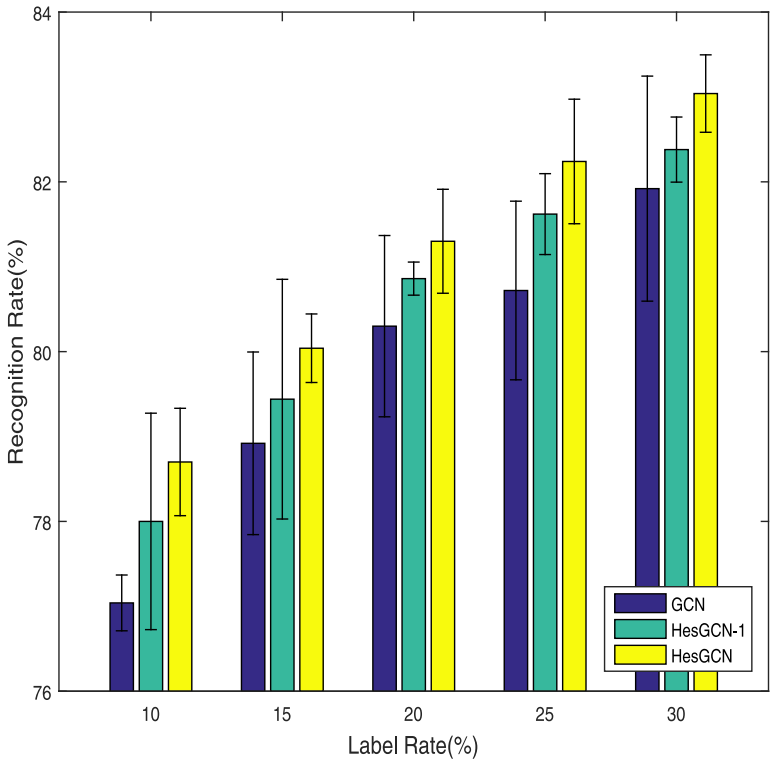

Fig. 7. Recognition accuracy of all classes in the Pumbed database.

Table 3
The comparison of HesGCN and different semi-supervised learning methods on the Citeseer and Cora dataset.

Method	Citeseer (120)	Cora(140)
Multi-layer Perception	46.5	55.1
Manifold Regularization	60.1	59.5
Semi-supervised Embedding	59.6	59
Chebyshev($K = 2$)	53.6	49.8
Chebyshev($K = 3$)	53.7	50.5
GAT	59.8	57
HyperGCN	55	59.4
HesGCN	60.6	59.7

5.2. Experiment parameters

In the experiment, owing to the reason of the hardware, we use the all samples of the Citeseer and Cora datasets, 5000 samples of the Pubmed and NELL databases. The four datasets are divided into three parts, i.e. validation set, testing set and training set. In addition, the 1000 labeled samples are used for test, the 500 labeled samples are selected for validation and the remaining are training samples. In the process of the training, we randomly select a certain percentage label rate samples (20%, 30%, 40%, 50% and 60%) for semi-supervised learning in the Citeseer and Cora datasets. For Pubmed and NELL datasets, the 10 %, 15%, 20%, 25% and 30% samples are selected as labeled data.

In the training process of the model, the maximum iteration numbers are 200 epochs. We use the adaptive moment estimation [45] (Adam) method with a learning rate of 0.01 and 0.1 (NELL) (other three default parameters: $\beta_1 = 0.9$ and $\beta_2 = 0.999$—exponential decay rates, $\epsilon = 10^{-8}$—for numerical stability) to optimize the loss function. If the loss values of the cross entropy loss function in the validation set remain unchanged with ten consecutive times, the HesGCN will stop training automatically. To increase the rate of convergence, we use the method that proposed in [46] to initialize networks weight. The L2 regularization parameter with value of 5×10^{-4} and 5×10^{-5} (NELL) is used to avoid the overfitting problem. We also use the following parameters in three datasets: Citeseer: 0.5 (dropout rate) and 32 (hidden units), Cora: 0.5 (dropout rate) and 32 (hidden units), Pubmed: 0.5 (dropout rate) and 16 (hidden units), and NELL: 0.6 (dropout rate) and 128 (hidden units).

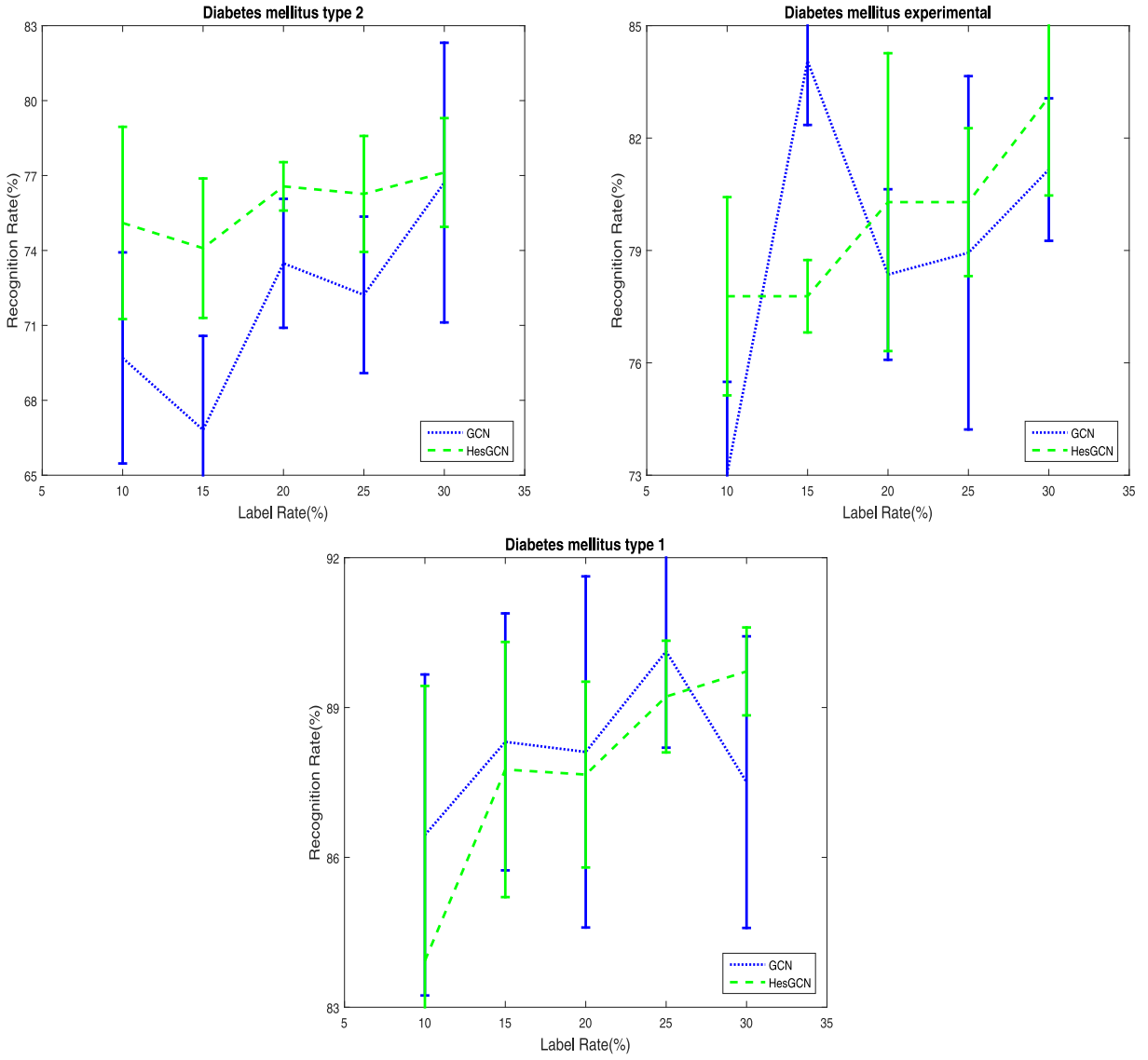

Fig. 8. Recognition accuracy of each class in the Pubmed database, including Diabetes mellitus experimental, Diabetes mellitus type 1, Diabetes mellitus type 2. Each subfigure corresponds on single class.

5.3. Experiment results

In this part, we compare the different variants of the HesGCN model with the GCN model. Figs. 3, 5, 7 and 9 give the mean recognition rate of all classes in the Citeseer, Cora, Pubmed and NELL dataset. The x-axis is the different label rate of the training set and the y-axis is the mean recognition rate of the different database on all classes. In addition, to better demonstrate the mean recognition performance of the single class for the HesGCN, Figs. 4, 6 and 8 show the mean recognition accuracy of the different model on the different class. The y-axis denotes the mean recognition accuracy of each class.

Figs. 3 and 4 show the experiment results of the Citeseer database. As we can see in the Fig. 3 that HesGCN outperforms the GCN model especially when the label rate is low. In addition, we can find that the mean recognition accuracy of the three methods are on the increase, when the label rate of the samples increases. We also observe that, under most conditions, the HesGCN performs better than other methods in some classes, such as Agents, HCI, DB and ML, as shown in Fig. 4.

Fig. 5 illustrates the mean recognition rate with the standard deviation in the Cora database. It shows that HesGCN gets the best performance in contrast to the GCN and HesGCN-1. In particular, it also reveals the effectiveness of the optimized

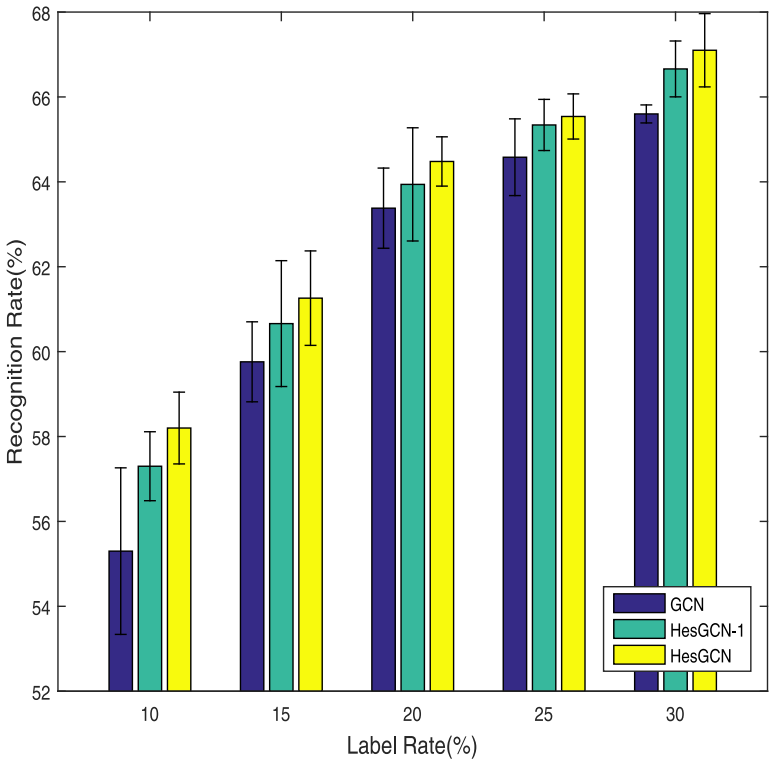

Fig. 9. Recognition accuracy of all classes in the NELL database.

method that our proposed. The each subfigure of the Fig. 6 corresponds the mean recognition accuracy of the single class. And then, in the majority of cases, the HesGCN model obtains an excellent performance.

From the experimental results of the Figs. 7 and 9 over all the categories, both methods achieve a higher recognition accuracy. The best algorithm of the above-mentioned methods is HesGCN model. We can find that the lower the label rate of training samples, our proposed model performs far better compared to other methods. From the line chart of the single class with the standard deviation, i.e. Fig. 8, we can see that in the Diabetes mellitus experimental, Diabetes mellitus type 1 and Diabetes mellitus type 2, our proposed HesGCN outperforms HesGCN-1 and GCN in most cases.

In this section, we compare many state-of-the-art semi-supervised learning methods, such as Multi-layer Perception [47], Manifold Regularization [21], Semi-supervised Embedding [22], Chebyshev($K = 2$) [48], Chebyshev($K = 3$) [48], HyperGCN [28] and graph attention network (GAT) [42]. Under the circumstances of 120 (Citeseer) and 140 (Cora) labeled samples of the training set, we report the average recognition rate with 100 random runs in Table 3. From the results of the Figs. 3, 5, 7, 9 and Table 3, our proposed HesGCN performs better than the above mentioned methods. Compared with GCN and HesGCN-1, these data prove the effectiveness of the optimization method that our proposed, i.e. from Eqs. (5) and (7) to Eq. (8). Compared with other models, we can know that HesGCN can learn richer sample features to improve its classification performance by the effective convolution fusion of the Hessian-based structure information and the original data information. In addition, it also indicates the richer null space of the Hessian, i.e. Hessian can better describe the local geometry of data compared with graph Laplacian (from Eqs. (1) to (2)).

6. Conclusion

In the past few years, many MSSL algorithms have achieved great success in the problems of the graph structured data representation and classification. However, how to choose the appropriate expression methods of data manifold structure and better exploit these manifold structure information is still a significant problem. In this paper, we introduce a new form of the spectral graph theory, i.e. spectral graph Hessian convolutions, by replacing the graph Laplacian with the Hessian matrix. And then, we get a novel linear convolution layer rule by simplifying and deducing the one-order polynomial of the spectral graph Hessian convolutions. Finally, we propose the Hessian graph convolutional networks for semi-supervised classification by stacking the proposed convolution layer rule. Due to the richer null space of the Hessian in contrast to Laplacian, HesGCN can get the most representative sample features and increase the classification performance of the model. Extensive experiments results show that HesGCN outperforms and more stable than the popular GCN.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant 61671480, in part by the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) under grant 18CX07011A and YCX2019080, in part by the Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by the Research Committee at University of Macau under grants MYRG2016-00123-FST and MYRG2018-00136-FST.

References

[1] P. Ji, N. Zhao, S. Hao, J. Jiang, Automatic image annotation by semi-supervised manifold kernel density estimation, Inf. Sci. 281 (2014) 648–660.
[2] J. Yu, X.-J. Wu, J. Kittler, Semi-supervised hashing for semi-paired cross-view retrieval, in: Proceedings of International Conference on Pattern Recognition, 2018, pp. 958–963.
[3] J. Zhang, J. Yu, J. You, D. Tao, N. Li, J. Cheng, Data-driven facial animation via semi-supervised local patch alignment, Pattern Recognit. 57 (2016) 1–20.
[4] X. Xu, T. Hospedales, S. Gong, Transductive zero-shot action recognition by word-vector embedding, Int. J. Comput. Vis. 123 (3) (2017) 309–333.
[5] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (7) (2012) 3262–3272.
[6] Z. Kang, H. Pan, S.C. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. (2019).
[7] J. Ma, T.W. Chow, Robust non-negative sparse graph for semi-supervised multi-label learning with missing labels, Inf. Sci. 422 (2018) 336–351.
[8] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (2017) 210–219.
[9] Z. Kang, H. Xu, B. Wang, H. Zhu, Z. Xu, Clustering with similarity preserving, Neurocomputing (2019).
[10] R. Sheikhpour, M.A. Sarram, E. Sheikhpour, Semi-supervised sparse feature selection via graph Laplacian based scatter matrix for regression problems, Inf. Sci. 468 (2018) 14–28.
[11] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl. Based Syst. 163 (2019) 510–517.
[12] E. Tu, Y. Zhang, L. Zhu, J. Yang, N. Kasabov, A graph-based semi-supervised k nearest-neighbor method for nonlinear manifold distributed data classification, Inf. Sci. 367 (2016) 673–688.
[13] C. Hong, J. Yu, J. You, X. Chen, D. Tao, Multi-view ensemble manifold regularization for 3d object recognition, Inf. Sci. 320 (2015) 395–405.
[14] S. Liu, J. Wu, L. Feng, H. Qiao, Y. Liu, W. Luo, W. Wang, Perceptual uniform descriptor and ranking on manifold for image retrieval, Inf. Sci. 424 (2018) 235–249.
[15] L. Zhang, L. Zhang, B. Du, J. You, D. Tao, Hyperspectral image unsupervised classification by robust manifold matrix factorization, Inf. Sci. 485 (2019) 154–169.
[16] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multi-modal face pose estimation with multi-task manifold deep learning, IEEE Trans. Ind. Inf. (2018).
[17] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, B. Du, Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding, Pattern Recognit. 48 (10) (2015) 3102–3112.
[18] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, Y. Wen, Multiview vector-valued manifold regularization for multilabel image classification, IEEE Trans. Neural Netw. Learn. Syst. 24 (5) (2013) 709–722.
[19] T. Ni, F.-L. Chung, S. Wang, Support vector machine with manifold regularization and partially labeling privacy protection, Inf. Sci. 294 (2015) 390–407.
[20] C. Turchetti, L. Falaschetti, A manifold learning approach to dimensionality reduction for modeling data, Inf. Sci. 491 (2019) 16–29.
[21] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (Nov) (2006) 2399–2434.
[22] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep Learning via Semi-supervised Embedding, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 639–655.
[23] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[24] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, M. Guo, Graphgan: graph representation learning with generative adversarial nets, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
[25] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[26] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the International Conference on Learning Representations, 2017.
[27] S. Fu, W. Liu, S. Li, Y. Zhou, Two-order graph convolutional networks for semi-supervised classification, IET Image Proc. (2019).
[28] N. Yadati, M. Nimishakavi, P. Yadav, A. Louis, P. Talukdar, Hypergcn: hypergraph convolutional networks for semi-supervised classification, in: Proceedings of the International Conference on Multimedia and Expo, 2019.
[29] M. Belkin, P. Niyogi, Semi-supervised learning on Riemannian manifolds, Mach. Learn. 56 (1–3) (2004) 209–239.
[30] S. Si, D. Tao, K.-P. Chan, Discriminative hessian eigenmaps for face recognition, in: Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 5586–5589.
[31] S. Si, D. Tao, K.-P. Chan, Evolutionary cross-domain discriminative hessian eigenmaps, IEEE Trans. Image Process. 19 (4) (2009) 1075–1086.
[32] J. Zhang, Y. Wan, Z. Chen, X. Meng, Non-negative and local sparse coding based on l2-norm and hessian regularization, Inf. Sci. 486 (2019) 88–100.
[33] J. Eells, L. Lemaire, Selected Topics in Harmonic Maps, 50, American Mathematical Soc., 1983.
[34] J.J. Verbeek, N. Vlassis, Gaussian fields for semi-supervised regression and correspondence learning, Pattern Recognit. 39 (10) (2006) 1864–1875.
[35] J.M. Lee, Riemannian Manifolds: An Introduction to Curvature, Springer New York, 2007.
[36] W. Liu, X. Yang, D. Tao, J. Cheng, Y. Tang, Multiview dimension reduction via hessian multiset canonical correlations, Inf. Fusion 41 (2018) 119–128.
[37] W. Liu, D. Tao, Multiview hessian regularization for image annotation, IEEE Trans. Image Process. 22 (7) (2013) 2676–2687.
[38] G. Feng, W. Liu, S. Li, D. Tao, Y. Zhou, Hessian-regularized multitask dictionary learning for remote sensing image recognition, IEEE Geosci. Remote Sens. Lett. 16 (5) (2018) 821–825.
[39] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.
[40] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29 (3) (2008). 93–93
[41] C. Cabanes, A. Grouazel, K.v. Schuckmann, M. Hamon, V. Turpin, C. Coatanoan, F. Paris, S. Guinehut, C. Boone, N. Ferry, et al., The cora dataset: validation and diagnostics of in-situ ocean temperature and salinity measurements, Ocean Sci. 9 (1) (2013) 1–18.
[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proceedings of the International Conference on Learning Representations, 2018.
[43] Z. Yang, W.W. Cohen, R. Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, in: Proceedings of the International Conference on Machine Learning, 2016.

[44] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka, T.M. Mitchell, Toward an architecture for never-ending language learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2010.
[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2014.
[46] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[47] C. Bai, J. Guo, L. Guo, J. Song, Deep multi-layer perception based terrain classification for planetary exploration rovers, Sensors 19 (14) (2019) 3102.
[48] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.