Information Sciences 584 (2022) 467–478

Contents lists available at ScienceDirect

Information Sciences

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

On the use of high-order feature propagation in Graph Convolution Networks with Manifold Regularization

F. Dornaika *

Henan University, Kaifeng, China
University of the Basque Country UPV/EHU, San Sebastian, Spain
IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

ARTICLE INFO

Article history:
Received 19 November 2020
Received in revised form 18 October 2021
Accepted 19 October 2021
Available online 5 November 2021

Keywords:
Graph-based semi-supervised learning
Graph Convolution Networks (GCN)
Graph Convolution Networks with Manifold Regularization (GCNMR)
Feature propagation
Label prediction
Manifold regularization
Semi-supervised image classification

ABSTRACT

Graph Convolutional Networks (GCNs) have received a lot of attention in pattern recognition and machine learning. In this paper, we present a revisited scheme for the new method called "GCNs with Manifold Regularization" (GCNMR). While manifold regularization can add additional information, the GCN-based semi-supervised classification process cannot consider the full layer-wise structured information. Inspired by graph-based label propagation approaches, we will integrate high-order feature propagation into each GCN layer. High-order feature propagation over the graph can fully exploit the structured information provided by the latter at all the GCN's layers. It fully exploits the clustering assumption, which is valid for structured data but not well exploited in GCNs.

Our proposed scheme would lead to more informative GCNs. Using the revisited model, we will conduct several semi-supervised classification experiments on public image datasets containing objects, faces and digits: Extended Yale, PF01, Caltech101 and MNIST. We will also consider three citation networks. The proposed scheme performs well compared to several semi-supervised methods. With respect to the recent GCNMR approach, the average improvements were 2.2%, 4.5%, 1.0% and 10.6% on Extended Yale, PF01, Caltech101 and MNIST, respectively.

1. Introduction

Representation learning has recently become a hot research topic. This paper aims to find solutions for Manifold Learning (e.g., graph-based dimensionality reduction), as well as Deep Learning (e.g., graph convolutional networks). Given the widespread use of graphs, graph representation learning plays an important role in machine learning, with applications in classification, clustering, knowledge discovery and information retrieval. Graph-based data representation thus becomes a powerful tool for learning representation and dimensionality reduction [48,47]. For example, in [41], the authors generated the low-dimensional representation using Extreme Learning Machines. They reconstructed all samples according to the graph weights containing the supervised information. In [45], the authors proposed a subspace learning approach that employs a regularized low-rank and sparse representation for multi-shot person re-identification problems.

Nowadays, semi-supervised learning (SSL) is considered a hot topic in machine learning [34,18]. Thanks to the SSL, some remarkable progress has been made in image classification [3,13,11,44]. Semi-supervised methods use both labeled and

* Corresponding author at: University of the Basque Country, San Sebastian, Spain.
E-mail address: fadi.dornaika@ehu.eus

https://doi.org/10.1016/j.ins.2021.10.041
0020-0255/

unlabeled samples [9]. The labeled data provide the final model's discriminative power, while the unlabeled ones preserve the data's geometric structure [30,40].

Graph-based data projection methods and their variants have attracted much attention in the field of dimensionality reduction [47]. Semi-supervised methods such as the Local and Global Consistency (LGC) [46] and Gaussian Random Field (GRF) [49] methods assume that the nearby data samples (i.e., the connected nodes) should have close labels. These methods therefore classify the unlabeled data samples by propagating the available labels over the graph. In [5], the authors proposed the Semi-supervised Discriminant Analysis (SDA) method as an extension of the supervised LDA method. Huang et al. [6] extended the supervised Local Discriminant Embedding (LDE) method to semi-supervised Discriminant Embedding (SDE). A joint embedding learning and sparse regression (JELSR) framework was proposed in [21]. This scheme jointly estimates the nonlinear embedding and linear transformation. In [27], the authors introduced a model based on multiple views which simultaneously recovers the clustering/semi-supervised classification and graph similarity matrix. The work of [29] dealt with label propagation over a graph. It introduced Flexible Manifold Embedding (FME) which consists of a label embedding's joint recovery and regression model. In [15,12], the authors proposed a kernel version of FME.

In [28], the authors presented an approach called semi-supervised projection with graph optimization (SPGO) for both semi-supervised classification and dimensionality reduction, which computes the projection and graph. It adaptively estimates the graph matrix based on the representations obtained in the low-dimensional space.

Graph Neural Networks (GNNs) can discover complicated structures in high-dimensional spaces [8,7,25]. They are also excellent for all types of learning: unsupervised, semi-supervised and supervised learning [19,16,38]. They are used in many applications [10,31,39,33].

Graph Convolutional Networks (GCNs) were introduced in [24]. They can be considered a special case of GNNs and are used as a semi-supervised learning method that exploits graph-structured data.

This method learns an architecture that outputs the soft labels associated with the training data. The work of [2] learns a network of GCNs using the neighboring nodes at different distances in the random walk. Each GCN block takes a different power of the adjacency matrix. They each learn a combination of the instance outputs that optimizes the classification objective. Although this method can outperform the basic GCN, it requires learning a large number of parameters since it is based on a network of GCNs. In [43], the authors proposed an unsupervised embedding method called Manifold Regularized Deep Learning (MRDL). It is based on a sequence of layers, where each layer estimates its own kernelized sparse graph and nonlinear projection. In [19], the authors presented a general unsupervised learning method (GraphSAGE) where features are sampled and aggregated based on the nodes' neighborhoods. This paradigm can be employed for inductive classification and representation. In [36], the authors presented Deep Graph Infomax (DGI). This unsupervised method can provide nodes' representation.

In this correspondence, we revisit our recent method called "Graph Convolutional Network with Manifold Regularization" (GCNMR) [23]. More precisely, we propose a high order feature propagation in each GCN block. The resulting scheme is called HO-GCNMR. This model's main goal is to retain two key strengths of data-driven graphs. First, unlike the classical GCN that performs one-hop feature propagation, we use high-order feature propagation so that distant neighbors can improve the data's resulting representation before applying the linear and nonlinear transformations associated with a given GCN block. Second, labels' inconsistency is reduced by integrating the concept of manifold smoothness. Thus, the presented approach takes into account additional information and is able to estimate the labels in a more consistent and accurate way.

In our proposed model, the cluster structure also intervenes in the feature propagation scheme, which can improve the representations' class discrimination. This is achieved by mainly focusing the feature propagation on same cluster nodes.

By applying this deep model, we can see that semi-supervised classification can be better than most competing methods on image datasets and citation network data. Moreover, it is worth noting that classical GCNs can use high-order node blending across a cascade of layers. However, as shown in [23], increasing the number of layers did not improve the final semi-supervised learning model. This can be explained by the fact that the blending is achieved on the nodes represented in different subspaces (GCN blocks) as each block has its own linear and nonlinear transformations. However, in our proposed model, feature propagation occurs in the same subspace of features associated with a block.

The paper is structured as follows: Section 2 contains some notations and definitions. It gives a brief overview of the GCN and GCNMR methods. Section 3 presents the proposed high-order GCNMR model. Section 4 presents a performance evaluation on four image databases and three citation networks. It also provides some qualitative results. Section 5 provides an analysis of the parameters' sensitivity. Section 6 contains some discussions. Finally, Section 7 provides some concluding remarks.

2. Background

In this section we will provide some notation. Then we will briefly discuss the GCN and GCNMR methods.

2.1. Definitions and notations

In this section, we present the main definitions and notations. The big bold letters in this paper denote matrices while small bold letters denote vectors. We define the data matrix by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d\times(l+u)}$, where $\mathbf{x}_i|_{i=1}^{l}$ and

$\mathbf{x}_i|_{i=l+1}^{l+u}$ are the labeled data samples and unlabeled data samples, respectively. The number of labeled (unlabeled) samples is denoted by l (u). The samples' dimension is denoted by d. The total number of samples is $N = l + u$. The number of classes is denoted by C. The labeled samples are denoted by the matrix $\mathbf{X}_l = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l] \in \mathbb{R}^{d\times l}$. Let the matrix $\mathbf{X}_u = [\mathbf{x}_{1+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d\times u}$ denote the unlabeled data samples. Let $\mathbf{S} \in \mathbb{R}^{(l+u)\times(l+u)}$ be the corresponding graph matrix, where S_{ij} is the similarity value between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. The Laplacian matrix, denoted $\mathbf{L}$, of $\mathbf{S}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$. Here $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the row or column sums (since the graph matrix is symmetric) of $\mathbf{S}$.

2.2. Graph Convolutional Networks (GCN): A brief review

The Graph Convolutional Networks (GCN) [24] method is a special deep neural network used for semi-supervised label inference on structured data. Specifically, used for a semi-supervised context based on label inference, a GCN model adopting two layers has the following structure:

$$\mathbf{F} = \mathrm{GCN}_{2-layer}\left(\mathbf{X}, \mathbf{S}; \mathbf{W}^{(0)}, \mathbf{W}^{(1)}\right) = \mathrm{softmax}\left(\widehat{\mathbf{S}}\,\sigma\left(\widehat{\mathbf{S}}\mathbf{X}^T\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{1}$$

where $\mathbf{X}^T \in \mathbb{R}^{N\times d}$ denotes the data matrix associated with N data samples. $\widehat{\mathbf{S}}$ is the renormalized graph matrix. It is given by $\widehat{\mathbf{S}} = \widehat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{S}+\mathbf{I})\widehat{\mathbf{D}}^{-\frac{1}{2}}$ and $\widetilde{D}_{ii} = \sum_j(\mathbf{S}+\mathbf{I})_{ij}$. $\mathbf{W}^{(0)} \in \mathbb{R}^{d\times H}$ is a linear transformation mapping the input data to a hidden representation whose dimension is H, σ is the rectified linear activation function *ReLU ()*, and $\mathbf{W}^{(1)} \in \mathbb{R}^{H\times C}$ is a linear transformation mapping the hidden representations to a feature vector whose dimension is C, where C is the number of classes.

Softmax is the normalization function that maps each output to class probabilities. This neural network's output is the matrix $\mathbf{F} \in \mathbb{R}^{N\times C}$, which constitutes the soft label matrix associated with all samples. The work in [24] estimated the model $\mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \ldots$. by minimizing the cross-entropy loss between the model's output and the known labels.

In the original GCN, the feature propagation performed in the first layer produces the following data matrix:

$$\mathbf{Y} = \hat{\mathbf{S}}\mathbf{X}^T \tag{2}$$

2.3. Review of Graph Convolutional Networks with manifold Regularization (GCNMR)

The method presented in [23] improved the GCN model by adding a manifold regularization term to the original loss function of the GCNs. Thus, the GCN layers are trained to produce labels that minimize the label fitting term (associated with labeled samples) and the smoothness term associated with all samples. The global GCNMR model's loss function is given by:

$$\begin{aligned}\mathscr{L}\left(\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(L)}\right) &= -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\,log(F_{ik}) + \tfrac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}||\mathbf{F}_{i*} - \mathbf{F}_{j*}||^2 S_{ij} \\ &= -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\,log(F_{ik}) + \lambda\,Trace\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right)\end{aligned} \tag{3}$$

where λ is a parameter that is used to balance the two criteria (label fitness and label smoothness). $Trace()$ denotes the trace of a matrix. $\mathbf{F}i*$ denotes the ith row of the soft label matrix $\mathbf{F}$. The latter is the deep neural network's output. y_{ik} is the real label distribution of the ith sample. l is the number of labeled samples, and C is the number of classes.

3. High-order GCNMR

Since the GCN layers are trained on the propagated features, this entails that the propagation strategy chosen have a significant impact on the final performance. Indeed, the feature propagation used in the original GCN models employs first-order propagation. This means that the features of a given node (sample) are replaced by those of the node and its immediate neighbors. This type of propagation hence restricts the interaction between samples to the direct neighbors only. Therefore, other relationships or similarities are not considered. The vanilla GCN model is not able to directly learn the similarity matrix's high powers and might have difficulties with modeling information about distant nodes.

We propose to use all the structured information in the graph to perform feature propagation in each layer. Inspired by label propagation, we propose a feature propagation that directly enables the network to better utilize information across distant nodes. This is referred to as high-order feature propagation.

3.1. High-order feature propagation

Let $\mathbf{S} \in \mathbb{R}^{N\times N}$ be the affinity matrix of a given data graph. Let $\mathbf{A} = \mathbf{D}^{-1}\mathbf{S}$ be the row normalized version of $\mathbf{S}$. Here $\mathbf{D}$ is a diagonal matrix whose diagonal entries are given by $D_{ii} = \sum_j S_{ij}$.

Let $\mathbf{H} = (\mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_N)$ be the data's initial features, i.e., $\mathbf{H} = \mathbf{X}^T$. Each row in $\mathbf{H}$ represents a particular node (sample). Our neighborhood graph-based feature propagation aims to map the data representation $\mathbf{H}$ to a new data representation $\mathbf{Y} = (\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_N) \in \mathbb{R}^{N \times d}$ for all graph nodes by integrating features of other nodes. Here we employ a trick similar to that used in label propagation [37]. Accordingly, the features $\mathbf{H}$ are iteratively propagated using the graph $\mathbf{A}$'s edges and weights.

For each propagation step and for each node $\mathbf{h}_i$ (a row vector), the new row vector, $\mathbf{y}_i$, is thus the sum of the feature information for its neighbors and the initial features $\mathbf{h}_i$.

$$\mathbf{y}_i^{(t+1)} = (1-\alpha)\sum_{j=1}^{N} A_{ij}\mathbf{y}_j^{(t)} + \alpha\mathbf{h}_i$$

where $t = 0, 1, 2, \ldots$ and α is a positive number less than one. In a matrix form, the above equation becomes:

$$\mathbf{Y}^{(t+1)} = (1-\alpha)\mathbf{A}\mathbf{Y}^{(t)} + \alpha\mathbf{H} \tag{4}$$

with $\mathbf{Y}^{(0)} = \mathbf{H}$. Since $\mathbf{A}$ is row-normalized, it is known that the above recursive propagation converges to a stable solution given by:

$$\mathbf{Y} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{A})^{-1}\mathbf{H} \tag{5}$$

The feature propagation provided by Eq. (5) is different from that proposed by GCN. Specifically, in Eq. (2), the feature propagation uses equal weights for the initial and neighbors' features. Furthermore, the propagation performed by GCNs is based on one step on the graph.

In short, high-order feature propagation is based on Eq. (5). In each layer, the input features $\mathbf{H}$ are transformed into the features $\mathbf{Y}$ given by:

$$\mathbf{Y} = \mathbf{B}\mathbf{H}$$

where $\mathbf{B} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{A})^{-1}$ is a transformation matrix computed once and for all. Fig. 1 illustrates the difference between one-step feature propagation and high-order feature propagation. The figure's left part illustrates how the new features of the node $\mathbf{x}_2$ are obtained when one-step propagation is used. It shows that this node's new features depend only on the node itself and its direct neighbors (these are shown in red). The figure's right part illustrates the propagation done using high order feature propagation. In this case, the node's new features depend on a large set of nodes belonging to the same connected component within the graph. This way, high-order propagation integrates the concept of data clusters into GCN learning by taking into account the long-range relationships between samples.

3.2. High-Order Graph Convolution Networks with Manifold Regularization

Our proposed model adopts the same loss used in the GCMR-MR model. However, the feature propagation (i.e., data convolution) will be based on high-order feature propagation.

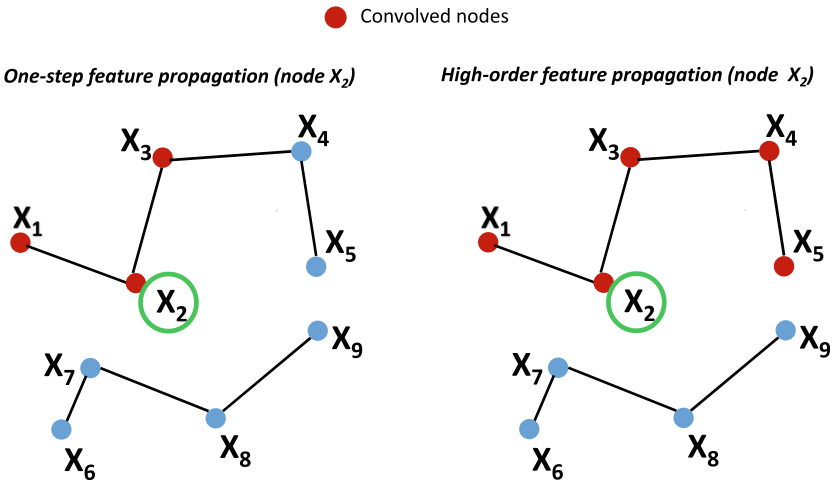

Fig. 1. Illustration of the difference between one-step feature propagation and high-order feature propagation. The figure's left part illustrates how the new features of node $\mathbf{x}_2$ are obtained when one-step propagation is used. It shows that this node's new features depend only on the node itself and its direct neighbors (these are shown in red). The figure's right part illustrates the high-order feature propagation. In this case, the new features of $\mathbf{x}_2$ are derived from a cluster of nodes that share some properties.

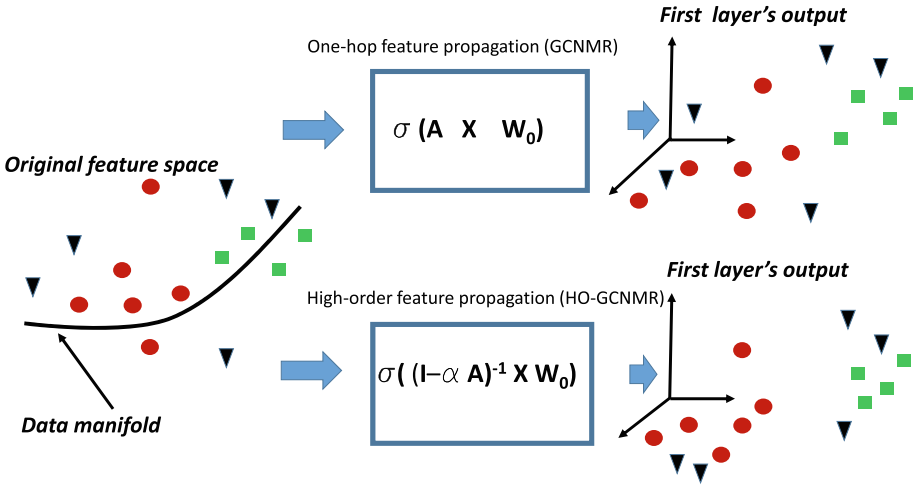

Fig. 2. Schematic representation of the first layer of GCNMR and HO-GCNMR. The top image corresponds to GCNMR using one-step feature propagation. The bottom image corresponds to the proposed high-order feature propagation (HO-GCNMR). The data representation obtained from the HO-GCNMR layer is expected to have better properties in terms of data smoothness and class discrimination than that obtained from GCNMR. The colored dots correspond to the labeled data while the black triangles correspond to the unlabeled data.

The loss function adopted by HO-GCNMR is given by:

$$\mathscr{L}\left(\mathbf{W}^{(0)},\ldots,\mathbf{W}^{(L)}\right) = -\sum_{i=1}^{l}\sum_{k=1}^{C} y_{ik}\, log(F_{ik}) + \lambda\, Trace\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right) \tag{6}$$

In HO-GCNMR, the soft label matrix $\mathbf{F}$ is provided by the output of the following neural net:

$$\mathbf{F} = f\left(\mathbf{X},\mathbf{B},\mathbf{W}^{(0)},\mathbf{W}^{(1)},\ldots\right) = \text{softmax}\left(\cdots\left(\mathbf{B}\,\sigma\left(\mathbf{B}\mathbf{X}^T\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right)\cdots\right) \tag{7}$$

Since the transformation matrix $\mathbf{B}$ is used by each GCN layer, it entails that each GCN layer performs high-order feature propagation.

If all intermediate layers yield a node whose size is H, the neural network then has N_p parameters, where $N_p = d.H + H.H + \ldots + H.C$. The neural network's output is the label matrix $\mathbf{F} \in \mathbb{R}^{N\times C}$.

Fig. 2 illustrates the difference between the GCNMR model, which assumes one-step feature propagation, and the proposed HO-GCNMR model, which assumes high-order feature propagation. The colored dots correspond to the labeled data while the black triangles correspond to the unlabeled data. For the sake of simplicity, only the first layer's input and output are shown. The figure's upper part illustrates the one-step convolution (GCNMR). The lower part illustrates the high-order convolution (HO-GCNMR). It is expected that the data representation obtained by the output of the HO-GCNMR layer have better properties in terms of data smoothness and class discrimination than that obtained by GCNMR.

4. Performance evaluation

We will evaluate the performance of the proposed HO-GCNMR model on different types of data. To compare the performance of different competing methods, we will use four public image databases: Extended Yale, PF01, Caltech101 and MNIST, and three citation databases: Cora, Citeseer and Pubmed. The image databases are face, scene and handwriting databases.

4.1. Datasets

1. **Extended Yale Face Dataset**[1]: The face database used in this work, Extended Yale B Face, corresponds to 38 people (groups) with different viewing conditions (9 poses and 64 lighting conditions).
   Three supervision levels are used in the semi-supervised learning. Thus, 9, 14 and 20 images per class are randomly selected as labeled images while the remaining ones are employed as unlabeled images.

[1] http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html.

2. **PF01**[2]: This dataset consits of the face images of 103 people: 53 men and 50 women. Each person has 17 different images. All individuals in the dataset are Asian.
   The reported results correspond to three supervision levels. Thus, 5, 8 and 12 images in each class are used as labeled images.
3. **Caltech101 Dataset**[3]: Caltech101 is a large dataset consisting of 9144 images grouped into 101 classes [17]. We used a subset that contains 2020 images, each class of the 101 classes containing 20 images. In the context of semi-supervised learning, three supervision levels are used. Thus, 3, 6 and 9 images per class are randomly selected as labeled images. The image descriptor was set to the deep features extracted by the pre-trained ResNet-50 [20] convolutional neural network.
4. **MNIST**[4]: The MNIST database of handwritten digits includes 70,000 images. It is divided into a training set of 60,000 images and a test set of 10,000 images. It has 10 classes corresponding to digits from 0∼9. In our experiments, we used a subset of 5,000 images. These images were size-normalized and centered in an image of 28× 28 pixels. We used the deep features extracted by the VGG16 convolutional neural network as the image descriptor for this database. 10, 20 and 30 images per class are randomly selected as labeled images.
5. **Cora.** This dataset contains 2708 references. The graph already exists as an adjacency matrix. The corresponding network has 2708 nodes and 5429 edges. Each node has a 1433-dimensional feature descriptor. The number of classes is six.
6. **Citeseer.** This dataset contains 3327 references. The corresponding network dataset contains 3327 nodes and 4732 edges. Each node has a 3703-dimensional feature descriptor. The number of classes is six.
7. **Pubmed.** This dataset contains 19717 references. The corresponding network has 19717 nodes and 44338 edges. Each node has a 500-dimensional dimension feature descriptor. The number of classes is three.

4.2. Experimental setup

We compared the proposed scheme HO-GCNMR with the semi-supervised method GCN [24] and other semi-supervised methods, which include the Semi-Supervised Discriminant Embedding (SDE) [42], the Exponential Semi-Supervised Discriminant Embedding (ESDE) [14], the Gaussian Random Field (GRF) [49], the Kernel Flexible Model Embedding (KFME) [15], the Multi-view learning with adaptive neighbors (MLAN) [27] and the GCN with manifold regularization (GCNMR) method [23]. This comparison also includes the HO-GCN model obtained from a GCN implementing high-order feature propagation.

To obtain a fair comparison, a single descriptor is used by the MLAN method. This descriptor was used by all other competing methods. In the used four image databases, the images represent the nodes, and the edges represent the pairwise similarities between the images. For each image database, we created a K- Nearest Neighbor graph that adopts ten neighbors. This setting is similar to many semi-supervised works that build graphs on image data. The pair similarity (i.e., edge weight) is given by a Gaussian function. This function's variance is set using the mean of all square pair distances in the image database.

It is worth noting that the graphs associated with the Cora, Citeseer and Pubmed citation networks are provided by the datasets themselves and correspond to adjacency matrices.

The solution of GCN, GCNMR and the introduced HO-GCNMR method is based on the use of a deep neural network. Therefore, similar setups were used in these methods' training phase [24]. These models were implemented in detail in TensorFlow [1]. The training of the deep networks was performed in a similar way for GCN, GCNMR and the introduced HO-GCNMR. More precisely, the learning rate was set to 0.01. We used two layers and set the nodes' dimension in the hidden layers to 256 neurons.

The proposed scheme has two parameters: λ and α. λ is chosen from $\left\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\right.$. α is fixed at 0.6. Note that α varies when we explicitly study its impact on the final performance. We quantified the classification accuracy (best average recognition rate) of all compared methods. We then splitted the dataset into a labeled part and an unlabeled one. We repeated this process five times. We reported the average performance on the five random splits. For the labeled sets, we considered three different sets of labeled images for each image dataset.

4.3. Method comparison

The citation networks Cora, Citeseer and Pubmed have their own evaluation protocol, which is explained in [35]. Therefore, we present the obtained results for the image and citation datasets in two separate tables. Bold figures correspond to the best results.

Table 1 depicts the results obtained by several semi-supervised methods on the Extended Yale, PF01, Caltech101 and MNIST databases. This table summarizes the mean classification rate in % and the associated standard deviation obtained with five random splits.

For every dataset, three numbers of labeled samples per class were used (these numbers are shown in the second column). In our experiments, the number of epochs used by the deep methods is fixed at 200.

[2] https://sites.google.com/site/postechimlab2012/databases/face-database-2001.
[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101/.
[4] http://yann.lecun.com/exdb/mnist/.

Table 1
Mean classification accuracies and standard deviations (%) of different methods on the image datasets.

Dataset	Lab.	SDE	ESDE	GRF	KFME	MLAN	GCN	GCNMR	HO-GCN	HO-GCNMR
Ext. Yale	9	76.8 ± 5.2	76.7 ± 3.8	72.4 ± 2.9	72.9 ± 7.8	77.4 ± 3.4	78.4 ± 8.5	81.9 ± 7.3	80.4 ± 8.5	**84.3 ± 4.5**
	14	78.9 ± 4.0	84.4 ± 7.0	75.7 ± 3.0	80.4 ± 5.1	79.7 ± 3.0	84.6 ± 3.6	85.6 ± 2.4	87.0 ± 2.1	**87.8 ± 2.1**
	20	83.5 ± 3.7	86.9 ± 3.8	78.0 ± 2.7	86.1 ± 3.2	81.5 ± 1.7	87.6 ± 2.7	88.4 ± 2.3	**90.7 ± 2.6**	90.6 ± 2.2
PF01	5	49.3 ± 3.8	50.9 ± 4.6	45.1 ± 3.9	50.0 ± 4.3	47.5 ± 4.3	50.3 ± 4.7	54.1 ± 5.1	54.2 ± 4.7	**58.1 ± 4.8**
	8	60.1 ± 4.8	61.3 ± 6.3	52.4 ± 5.7	57.6 ± 6.8	55.4 ± 6.9	58.8 ± 7.9	62.6 ± 6.5	64.0 ± 7.4	**68.0 ± 6.5**
	12	56.5 ± 12.0	63.5 ± 12.4	50.1 ± 10.8	55.7 ± 11.3	53.1 ± 11.6	60.4 ± 10.0	64.7 ± 9.7	64.9 ± 6.5	**68.9 ± 6.9**
Caltech101	3	61.6 ± 2.7	72.2 ± 0.9	77.0 ± 1.7	77.8 ± 1.4	76.6 ± 1.3	77.9 ± 0.6	78.7 ± 0.7	79.2 ± 0.3	**79.2 ± 0.2**
	6	77.9 ± 1.5	79.2 ± 0.4	80.0 ± 0.6	80.9 ± 0.7	79.8 ± 0.9	82.1 ± 0.5	82.4 ± 0.6	80.4 ± 1.4	**84.3 ± 1.3**
	9	83.2 ± 0.9	82.0 ± 0.7	81.9 ± 0.7	83.4 ± 0.6	81.3 ± 1.0	83.9 ± 0.5	84.1 ± 0.6	84.3 ± 0.2	**84.6 ± 0.3**
MNIST	10	76.0 ± 1.2	81.3 ± 1.1	47.6 ± 0.2	79.4 ± 0.7	83.8 ± 1.9	75.7 ± 1.2	76.1 ± 1.1	84.9 ± 1.6	**85.3 ± 1.2**
	20	84.6 ± 0.2	85.7 ± 1.6	47.3 ± 0.4	83.9 ± 1.1	86.0 ± 1.4	78.0 ± 1.2	78.0 ± 1.2	88.9 ± 1.4	**89.2 ± 1.0**
	30	86.4 ± 1.1	86.9 ± 1.7	47.5 ± 0.4	85.3 ± 0.7	89.9 ± 0.9	79.3 ± 1.4	78.6 ± 1.4	89.9 ± 1.9	**90.1 ± 1.8**

Table 2
Classification accuracies (%) of different methods on the citation datasets.

Dataset	ManiReg [4]	GRF [49]	DeepWalk [32]	GAT [35]	GCN [24]	GLCN [22]	GCNMR [23]	HO-GCNMR
Cora	59.5	68.0	67.2	83.2	82.9	85.5	84.3	**85.7**
Citeseer	60.1	45.3	43.2	71.0	70.9	72.0	72.0	**72.8**
Pubmed	70.7	63.0	65.3	78.0	77.9	78.3	79.0	**80.1**

Table 2 summarizes the classification rate in % obtained with different semi-supervised methods on Cora, Citeseer and Pubmed. The compared semi-supervised methods are: Manifold Regularization (ManiReg) [4], GRF [49], DeepWalk [32], Graph Attention Networks (GAT) [35], GLCN [22], GCNMR [23] and the proposed HO-GCNMR. In these experiments, the number of epochs used by the deep methods is set to 200.

4.4. Qualitative evaluation

To better understand the label matrix obtained by the proposed HO-GCNMR method, we provide two ways of visualizing the obtained results. The first visualizes the entire predicted label matrix **F**, while the second presents the pairwise correlation associated with the predicted labels..

For the first type, we considered the 2,414 images of the Extended Yale dataset with 38 classes (subjects). We considered the case where each class has 14 labeled images. The proposed method's output is the predicted label matrix **F** with 2,414 rows and 38 columns. Each row is the predicted label probability distribution (a 38-vector) for a given image.

To visualize these 2414 vectors, we used the well-known t-SNE [26] technique. Fig. 3 shows the 2D t-SNE visualization of the predicted label matrix **F**'s rows associated with the Extended Yale dataset. As can be seen, the 38 classes within the predicted label space were globally well separated.

For the second type, we considered the Caltech101 dataset used above. We considered the case of 6 labeled images per class. After applying the HO-GNCMR to the Caltech101 data, we obtained the predicted label matrix **F**. For each pair of unlabeled images, we calculated the correlation between their predicted labels. We used the Pearson correlation coefficient (normalized zero-mean correlation). When this coefficient is one, we have a perfect correlation between the two images' label probabilities. However, when this coefficient is zero, there is no correlation at all between the two images. Fig. 4 shows the obtained correlation matrix associated with 144 unlabeled images' predicted labels. Since these images' ground truth labels are known, we were able to arrange them in blocks of 12 images for each class (12 classes). The diagonal blocks in the correlation matrix thus correspond well to the correlation of samples belonging to the same class. As can be seen in the figure, most pairs belonging to the same class have a high correlation, indicating that the estimated labels bu HO-GCNMR were consistent between classes.

Fig. 5 shows some qualitative results obtained with the Extended Yale dataset when the number of labelled images per class was set to 14. Fig. 5(a) shows some sample images correctly classified by the proposed method HO-GCNMR. Fig. 5(b) shows some example images that were misclassified by the proposed method HO-GCNMR. The misclassified images obviously suffered a significant illumination anomaly.

5. Parameter sensitivity

In this section, we present a study of the proposed semi-supervised scheme's performance as a function of the main parameters. The proposed scheme has three main parameters. These are as follows: (1) λ, which controls the relative

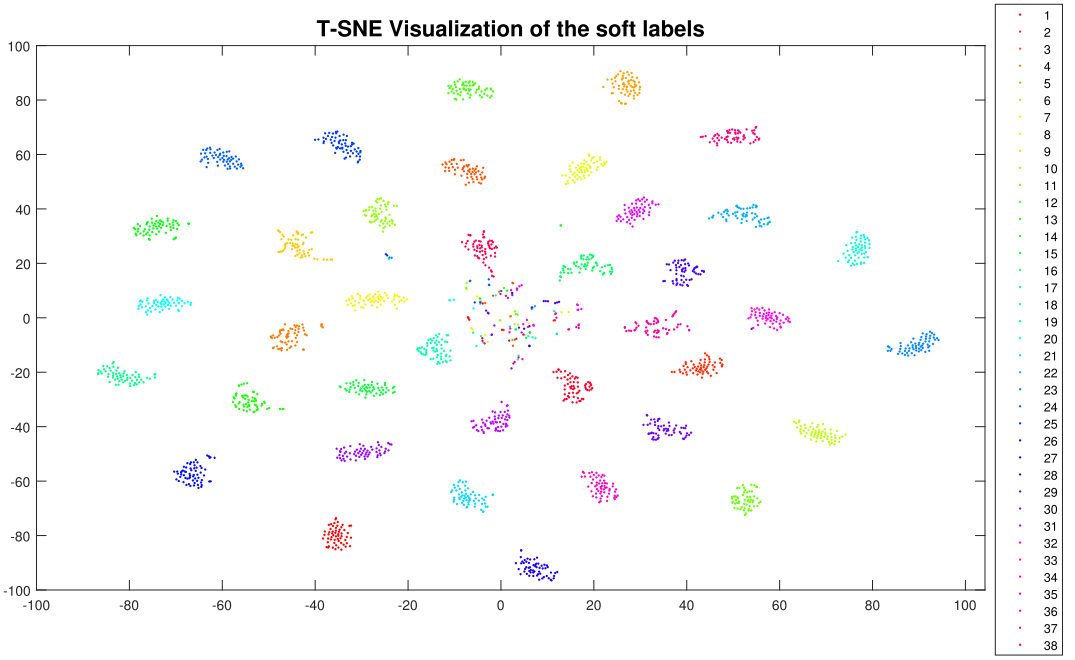

Fig. 3. Visualization of predicted labels using the t-SNR technique. The 2414 predicted labels were obtained using the HO-GCNMR method. Each colored dot corresponds to the 38-vector representing the predicted label distribution of a particular image. Each color is associated with a particular class (for illustration).

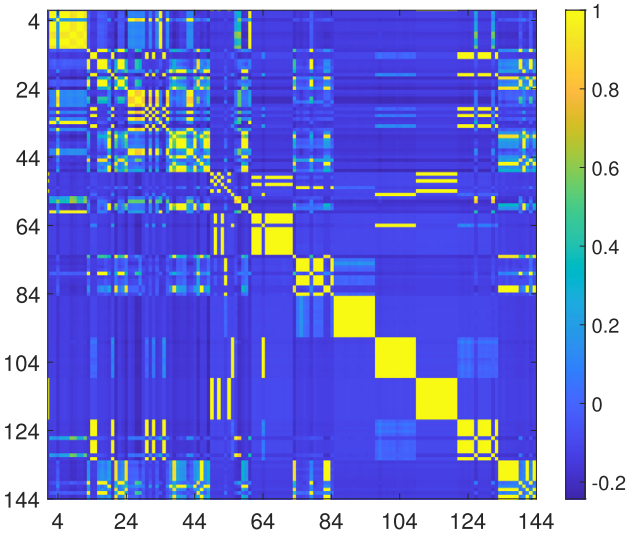

Fig. 4. Visualization of pairwise correlation between predicted labels. The correlation matrix corresponds to 144 unlabeled images in the pruned Caltech101 dataset belonging to 12 classes.

influence of the cross entropy loss and the manifold regularization term in the global loss, (2) α, which controls the feature propagation in a given layer,and (3) the number of layers.

5.1. Effect of the parameter λ

The proposed learning model HO-GCNMR has the balance parameter λ which controls the importance of the supervised and unsupervised loss within the total loss (6). Fig. 6 shows the classification accuracy versus the values of λ for the four datasets: Extended Yale, PF01, Caltech101 and MNIST. λ varies in the set $\left\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\right\}$. The number of

Fig. 5. Qualitative results obtained on the Extended Yale dataset: (a) Some example images correctly classified by the proposed method HO-GCNMR. (b) Some example images that are misclassified by the proposed method HO-GCNMR.

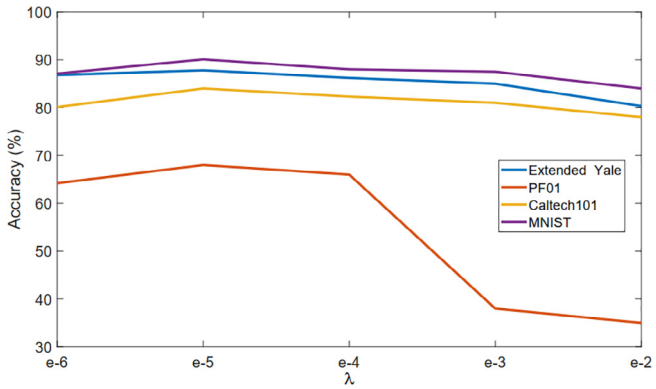

Fig. 6. Classification accuracy of the proposed method HO-GCNMR versus parameter λ. Experiments were performed on the following datasets: Extended Yale, PF01, Caltech101 and MNIST.

labeled images per class was 14, 8, 6 and 20 for Extended Yale, PF01, Caltech101 and MNIST, respectively. According to this figure, it is obvious that the value of 10^{-5} for λ can be considered as a good choice for all four datasets.

5.2. Effect of the parameter α

The proposed HO-GCNMR model has the parameter α that controls the high-order feature propagation. We set the number of layers to two and λ to a small value, since this configuration gave high performance in the GCNMR method [23]. We then varied α in the interval [0, 1] in a step of 0.1. Fig. 7 shows the classification accuracy of the HO-GCN and HO-GCNMR models as a function of α on the PF01 unlabeled data. This experiment corresponds to a split of the PF01 dataset where the number of labeled samples per class was set to 5. We can observe that the value of 0.9 can be considered as the best value for both models: HO-GCN and the proposed HO-GCNMR.

5.3. Number of layers' impact

We have also studied the effect of the number of layers on the performance of the proposed HO-GCNMR. Fig. 8 shows the classification rate versus the number of layers for two databases: Extended Yale and PF01.

6. Discussion

From the experimental results obtained we may draw the following observations:

- In almost all cases, the proposed HO-GCNMR model outperformed all other compared graph-based semi-supervised approaches (Tables 1 and 2). This applies to both types of data: Images and Citation Networks. On average the improvements with respect to the GCNMR model were 2.2%, 4.5%, 1.0% and 10.6% for Extended Yale, PF01, Caltech101 and MNIST, respectively. With regards to GCN's new variant (i.e. HO-GCN) we got some improvements with respect to the GCN model.

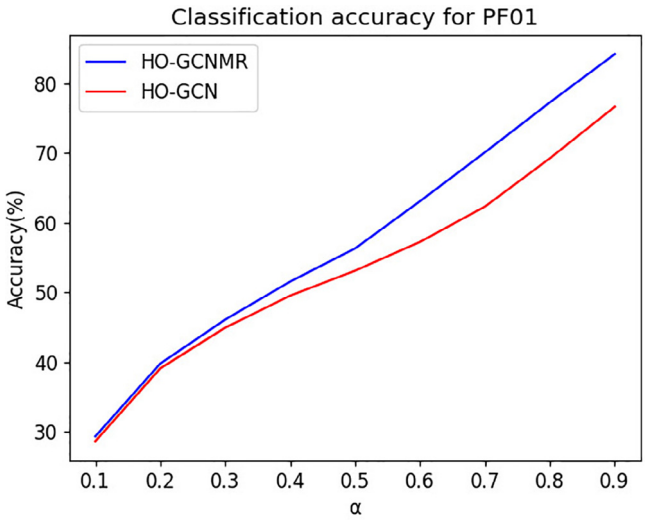

Fig. 7. Classification accuracy of the HO-GCN and HO-GCNMR models versus parameter α. These results correspond to the PF01 dataset.

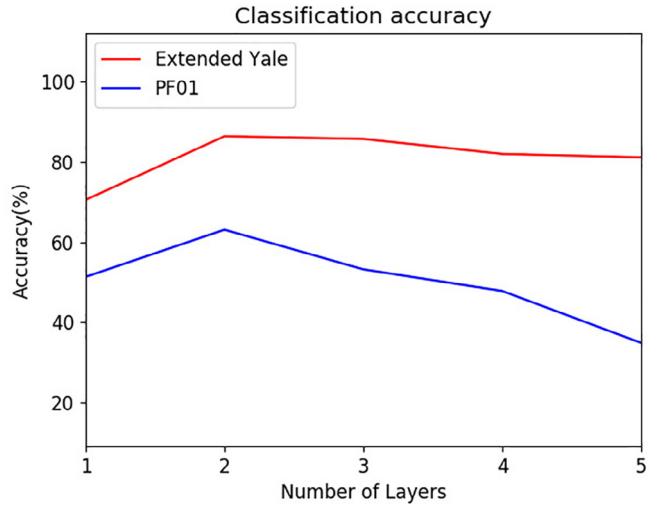

Fig. 8. Classification accuracy of the HO-GCNMR method versus the number of layers. Two databases are used: Extended Yale and PF01.

- Regarding the specific parameters of the proposed HO-GCNMR model, we can conclude that a small value for λ could give good results. On the other hand, a value close to one for α seemed to be a good choice. It is worth mentioning that the proposed HO-GCNMR model provided a good classification rate in all the used image datasets, even without using the best parameter α: Extended Yale, PF01, Caltech101 and MNIST.
- The performance of HO-GCNMR was significantly better than that of the original GCN model.
- It has been shown that in the experiments conducted with the image datasets and citation networks, the use of two layers was sufficient to achieve good performance. Using more than two layers did not necessarily improve performance. This result suggests that the obtained data representation was suitable for semi-supervised label inference. As future work, we could investigate progressive topology estimation, where the deep neural system is extended layer by layer and stops adding layers when no further improvement is obtained.

7. Conclusion

In this paper, we revisited Graph Convolution Network with manifold regularization. In addition to using a loss function that incorporates unsupervised and supervised information, we used high-order feature propagation adopted by each layer.

The GCN, GCNMR and HO-GCMR (proposed) models used the same deep neural network structure. Therefore, the training and testing phase's computational cost is the same for all these models.

Despite the fact that the deep neural network associated with the proposed HO-GCNMR model has the same structure as the GCN and GCNMR models, the adoption of high-order feature propagation resulted in an improvement in the final performance without any additional computational cost. Although the presented idea was simple, we found that the resulting model achieved better classification performance on unlabeled data than previous GCN models.

The proposed method's main limitation is that it is a transductive model. This means that both labeled and unlabeled data must be available in the training phase. This limitation is also shared by many competing methods, including GCN and GCNMR. Another limitation of these approaches is the assumption that the data graph is provided in advance. Future work will explore strategies to extend the proposed model to an inductive model. An inductive model can naturally make a prediction for unseen out-of-sample data. In addition, future work could include a joint estimation of the graph and the classification model.

CRediT authorship contribution statement

F. Dornaika: Software, Conceptualization, Investigation, Resources, Writing - original draft, Writing - review & editing.

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 is supported in part by the University of the Basque Country UPV/EHU grant GIU19/027.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint, 2016..
[2] S. Abu-El-Haija, A. Kapoor, B. Perozzi, J. Lee, N-GCN: Multi-scale graph convolution for semi-supervised node classification, vol. 115 of Proceedings of Machine Learning Research, 2020, pp. 841–851..
[3] R.A.R. Ashfaq, X.-Z. Wang, J.Z. Huang, H. Abbas, Y.-L. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Information Sciences 378 (2017) 484–497.
[4] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.
[5] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1–7.
[6] H.-T. Chen, H.-W. Chang, T.-L. Liu, Local discriminant embedding and its variants, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 2, IEEE, 2005, pp. 846–853..
[7] L. Chen, H. Huang, D. Chen, Join cardinality estimation by combining operator-level deep neural networks, Information Sciences 546 (2021) 1047–1062.
[8] S. Chen, X. Tian, C. Ding, B. Luo, Y. Liu, H. Huang, Q. Li. Graph convolutional network based on manifold similarity learning, Cognitive Computation 12 (2020) 1144–1153..
[9] Y. Chong, Y. Ding, Q. Yan, S. Pan, Graph-based semi-supervised learning: A review, Neurocomputing 408 (2020) 216–230.
[10] M. Coskun, M. Koyuturk, Node Similarity Based Graph Convolution for Link Prediction in Biological Networks, Bioinformatics (2020)..
[11] F. Dornaika, Y. El Traboulsi, Learning flexible graph-based semi-supervised embedding, IEEE Transactions on Cybernetics 46 (1) (2016) 206–218.
[12] F. Dornaika, Y. El Traboulsi, Margin based semi-supervised elastic embedding for face image analysis, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1313–1320.
[13] F. Dornaika, Y. El Traboulsi, Matrix exponential based semi-supervised discriminant embedding for image classification, Pattern Recognition 61 (2017) 92–103.
[14] F. Dornaika, Y.E. Traboulsi, Matrix exponential based semi-supervised discriminant embedding, Pattern Recognition 61 (2017) 92–103.
[15] Y. El Traboulsi, F. Dornaika, A. Assoum, Kernel flexible manifold embedding for pattern classification, Neurocomputing 167 (2015) 517–527.
[16] S. Fu, W. Liu, K. Zhang, Y. Zhou, D. Tao, Semi-supervised classification by graph p-Laplacian convolutional networks, Information Sciences 560 (2021) 92–106.
[17] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, 2007..
[18] H. Guo, H. Zou, J. Tan, Semi-supervised dimensionality reduction via sparse locality preserving projection, Applied Intelligence (2020).
[19] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034..
[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[21] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Transactions on Cybernetics 44 (6) (2014) 793–804.
[22] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.
[23] M.T. Kejani, F. Dornaika, H. Talebi, Graph convolution networks with manifold regularization for semi-supervised learning, Neural Networks (2020).
[24] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017..
[25] Z. Li, H. Liu, Z. Zhang, T. Liu, N.N. Xiong, Learning Knowledge Graph Embedding With Heterogeneous Relation Attention Networks, in: IEEE Transactions on Neural Networks and Learning Systems, 2021, pp. 1–13.
[26] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9(Nov) (2008) 2579–2605..

[27] F. Nie, G. Cai, J. Li, X. Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, IEEE Transactions on Image Processing 27 (3) (2018) 1501–1511.
[28] F. Nie, X. Dong, X. Li, Unsupervised and semisupervised projection with graph optimization, IEEE Transactions on Neural Networks and Learning Systems 2020 (2020).
[29] F. Nie, D. Xu, I.W.-H. Tsang, C. Zhang, Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction, IEEE Transactions on Image Processing 19 (7) (2010) 1921–1932.
[30] D.C.G. Pedronette, Y. Weng, A. Baldassin, C. Hou, Semi-supervised and active learning through manifold reciprocal knn graph for image retrieval, Neurocomputing 340 (2019) 19–31.
[31] W. Peng, X. Hong, G. Zhao, Tripool: Graph triplet pooling for 3D skeleton-based action recognition, Pattern Recognition 115 (2021) 107921.
[32] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.
[33] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, JKT: A joint graph convolutional network based Deep Knowledge Tracing, Information Sciences 580 (2021) 510–523.
[34] Y.E. Traboulsi, F. Dornaika, Y. Ruichek, Semi-supervised two phase test sample sparse representation classifier, Knowledge-Based Systems 160 (2018) 16–27.
[35] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[36] P. Velickovic, W. Fedus, W.L. Hamilton, P. Lio, Y. Bengio, R.D. Hjelm, Deep graph infomax, in: International Conference on Learning Representations, 2019.
[37] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Transactions on Knowledge and Data Engineering 20 (1) (2008) 55–67.
[38] J. Wang, J. Liang, J. Cui, J. Liang, Semi-supervised learning with mixed-order graph convolutional networks, Information Sciences 573 (2021) 171–181.
[39] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A Comprehensive Survey on Graph Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2021) 4–24.
[40] J. Xie, S. Liu, H. Dai, Manifold regularization based distributed semi-supervised learning algorithm using extreme learning machine over time-varying network, Neurocomputing 355 (2019) 24–34.
[41] L. Yang, S. Song, S. Li, Y. Chen, G. Huang, Graph embedding-based dimension reduction with extreme learning machine, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020) 1–12.
[42] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, Pattern Recognition 45 (3) (2012) 1119–1135.
[43] Y. Yuan, L. Mou, X. Lu, Scene recognition by manifold regularized deep learning architecture, IEEE Transactions on Neural Networks and Learning Systems 26 (10) (2015) 2222–2233.
[44] J. Zhang, P. Zhang, B. Li, L. Jing, T. Lv, Semisupervised feature extraction based on collaborative label propagation for hyperspectral images, IEEE Geoscience and Remote Sensing Letters (2019) 1–5.
[45] A. Zheng, X. Zhang, B. Jiang, B. Luo, C. Li, A subspace learning approach to multishot person reidentification, IEEE Transactions on Systems, Man, and Cybernetics: Systems 50 (1) (2020) 149–158.
[46] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, 2004, pp. 321–328..
[47] R. Zhu, F. Dornaika, Y. Ruichek, Joint graph based embedding and feature weighting for image classification, Pattern Recognition 93 (2019) 458–469.
[48] R. Zhu, F. Dornaika, Y. Ruichek, Learning a discriminant graph-based embedding with feature selection for image categorization, Neural Networks 111 (2019) 35–46.
[49] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in: International Conference on Machine Learning, 2003, pp. 912–919.