



HAL
open science

Graph Convolution Networks for Unsupervised Learning

Maria Al Jreidy, Joseph Constantin, Fadi Dornaika, Denis Hamad

► **To cite this version:**

Maria Al Jreidy, Joseph Constantin, Fadi Dornaika, Denis Hamad. Graph Convolution Networks for Unsupervised Learning. Hamlich, Mohamed; Bellatreche, Ladjel; Siadat, Ali; Ventura, Sebastian. Smart Applications and Data Analysis: 4th International Conference, SADASC 2022, Marrakesh, Morocco, September 22-24, 2022, Proceedings, 1677, Springer International Publishing, pp.24-33, 2022, Communications in Computer and Information Science (CCIS, volume 1677), 978-3-031-20489-0. 10.1007/978-3-031-20490-6_3 . hal-04460546

HAL Id: hal-04460546

<https://hal.science/hal-04460546v1>

Submitted on 15 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Smart Applications and Data Analysis	
Series Title		
Chapter Title	Graph Convolution Networks for Unsupervised Learning	
Copyright Year	2022	
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Switzerland AG	
Corresponding Author	Family Name	Jreidy
	Particle	
	Given Name	Maria Al
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	LISIC -ULCO
	Address	50 rue Ferdinand Buisson, BP 699, 62228, Calais Cedex, France
	Division	LaRRIS, Faculty of Sciences
	Organization	Lebanese University
	Address	Fanar, Jdeidet BP, 90656, Lebanon
	Email	mariajreidy@hotmail.com
Author	Family Name	Constantin
	Particle	
	Given Name	Joseph
	Prefix	
	Suffix	
	Role	
	Division	LaRRIS, Faculty of Sciences
	Organization	Lebanese University
	Address	Fanar, Jdeidet BP, 90656, Lebanon
	Email	
Author	Family Name	Dornaika
	Particle	
	Given Name	Fadi
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	University of the Basque Country UPV/EHU
	Address	San Sebastian, Spain
	Division	
	Organization	IKERBASQUE, Basque Foundation for Science
	Address	Bilbao, Spain

	Email
Author	<p>Family Name Hamad</p> <p>Particle</p> <p>Given Name Denis</p> <p>Prefix</p> <p>Suffix</p> <p>Role</p> <p>Division</p> <p>Organization LISIC -ULCO</p> <p>Address 50 rue Ferdinand Buisson, BP 699, 62228, Calais Cedex, France</p> <p>Email</p>
Abstract	<p>In recent years, graph convolution networks (GCN) have been proposed as semi-supervised learning approaches. In this paper, we introduce a new objective function to train a GCN in order to adapt it in unsupervised learning context. More precisely, we propose a loss function composed only of unsupervised terms, the first term is the kernel k-means objective function used to capture the shared features information between nodes, and the second is a regularization term that imposes the smoothness of the predicted clusters of the whole data. Thanks to the proposed objective function, we are able to keep the advantages of the classical semi-supervised GCN, while using it in an unsupervised way. Experiments on benchmark datasets show that our proposed unsupervised GCN achieves superior performance compared to state-of-the-art clustering algorithms.</p>
Keywords (separated by '-')	<p>Graph convolution network - Unsupervised learning - Kernel k-means - Spectral clustering - Regularization term</p>



Graph Convolution Networks for Unsupervised Learning

Maria Al Jreidy^{1,2(✉)}, Joseph Constantin², Fadi Dornaika^{3,4},
and Denis Hamad¹

¹ LISIC -ULCO, 50 rue Ferdinand Buisson, BP 699, 62228 Calais Cedex, France

² LaRRIS, Faculty of Sciences, Lebanese University,
Fanar, Jdeidet BP 90656, Lebanon

mariajreidy@hotmail.com

³ University of the Basque Country UPV/EHU, San Sebastian, Spain

⁴ IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract. In recent years, graph convolution networks (GCN) have been proposed as semi-supervised learning approaches. In this paper, we introduce a new objective function to train a GCN in order to adapt it in unsupervised learning context. More precisely, we propose a loss function composed only of unsupervised terms, the first term is the kernel k-means objective function used to capture the shared features information between nodes, and the second is a regularization term that imposes the smoothness of the predicted clusters of the whole data. Thanks to the proposed objective function, we are able to keep the advantages of the classical semi-supervised GCN, while using it in an unsupervised way. Experiments on benchmark datasets show that our proposed unsupervised GCN achieves superior performance compared to state-of-the-art clustering algorithms.

Keywords: Graph convolution network · Unsupervised learning · Kernel k-means · Spectral clustering · Regularization term

1 Introduction

Graphs are powerful forms of describing and modeling complex systems. In recent years, the problem of graph clustering has been well studied [1–3]. Several methods have been proposed to achieve this goal, some of them rely only on the structure of the graph in the learning process, [4, 5], while others just rely on data features.

One of the well-known methods in the clustering process is k-means clustering [6, 7] where the number of clusters is initially needed to know. k-means clusters graph by considering node features only. After that, the spectral clustering algorithm [8] was introduced where the graph clustering is done based on the structure of the graph built from data features. Then, other methods combining auto-encoders with the clustering algorithm [9, 10] were proposed to

give better solutions with high-dimensional databases. For example, adversarially regularized variational graph autoencoder (ARVGE) [11], that learns node embedding by graph autoencoder and graph variational autoencoder. The spectral embedding network (SENet) [12] where the graph structure and node feature information via higher-order graph convolution are used to train the model with a spectral clustering loss function.

All these methods can be considered as unsupervised methods as they do not impose prior knowledge of the fundamental truth label in their learning phase.

However, there are other graph-based methods which require a small part of their inputs to be labelled, we can mention the Graph Convolution Network (GCN) [13] which has a very important advantage where the structure of the graph and the features of the data are convolved together in order to obtain the final outputs, namely the soft labels. In this sense, a GCN processes nodes in a graph based on their structure as well as their feature similarities. Yet, as said before a GCN usually relies on prior knowledge of some labels that are often not available for multiple databases, as data labeling is a tedious task to perform.

To keep the advantage of the GCN, and to be able to use it with unlabeled data, unsupervised graph clustering using GCN model where proposed. In [14] an unsupervised model for clustering graphs via a joint GCN has been introduced. In [15] a label sampling method was introduced and used before the GCN model. However, these methods are considered complex as they require labelling generation techniques followed by the GCN to use it in an unsupervised way.

In this paper, we propose a new unsupervised learning framework based on GCN only. The main contribution of our work is as follows: Without any prior knowledge of node labels, we propose a GCN model trained with the kernel k-means objective function where the kernel matrix K represents the data features, integrated with a regularization term reflecting the graph structure. In other words, our proposed loss has two clustering-friendly terms. In this case, if two nodes share a larger proportion of neighbors and features, they have a higher probability of belonging to the same cluster. Therefore, both network topology and data features are considered during model training to obtain consistent clustering.

The rest of this paper is organized as following. In Sect. 2, we introduce some related work. In Sect. 3, we present some preliminary notations and formulations related to this work. Section 4 gives a detailed description of the proposed method. We present experimental results on various real-world networks to validate the performance of the method proposed in Sect. 5, we also compare our approach with other state-of-the-art algorithms. Finally, in Sect. 6, we give the conclusion.

2 Related Work

Early graph clustering techniques merely grouped nodes based on the graph topology. Such as, DeepWalk [4] and node2vec [16], which produce node sequences using reduced random walks before obtaining node embeddings using

the Skip-gram model [17]. Different k-order proximities between nodes are captured by GraRep [18]. Then, models based on autoencoders were created in order to capture the highly nonlinear graph structures [19]. In order to increase the model strength and generalizability, ANE [20], a clustering technique that uses adversarial learning to learn node representations, attempts to regularize the embedding learning process.

The developed node-clustering techniques that use both node properties and graph structure are referred to as the attributed methods. Some techniques employ spectral clustering [8], random walks [21], matrix factorization, and Bayesian models [22] on them to make use of both structure and feature information. Other approaches [23] develop a trade-off distance metric between them.

Graph convolutional network (GCN) is a semi-supervised approach that was recently presented for graph classification problems [13]. GCN-based algorithms help to merge network topology and attribute information, but they depend on a large number of node labels to classify unlabeled nodes, in contrast to most semi-supervised methods that concentrate on maintaining network structure. Sun et al. [24] suggested a network embedding framework for node clustering based on a graph convolutional autoencoder. Additionally, few unsupervised techniques have been recently developed. An unsupervised model for community detection using GCN embedding was proposed by Jin et al. [25]. A spectral embedding network for attributed graph clustering (SENet) was suggested in [12] and uses a spectral clustering loss with GCN to learn node embeddings while also enhancing graph structure. In this work, the GCN model formed with two loss functions commonly used in clustering problems, which manages both data structure and data attributes.

3 Preliminary

The section briefly presents the preliminary knowledge of this work, and the architecture of convolution networks of graphs.

3.1 Weighted Kernel K-Means and Spectral Clustering

Weighted kernel k-means is a k-means clustering algorithm enhanced by the use of a kernel function [26]. The kernel weighted k-means objective function is defined as follows:

$$D(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^k \sum_{x_i \in \pi_j} w(x_i) \|\phi(x_i) - m_j\|^2 \quad (1)$$

with $w(x_i)$ is the weight of data x_i , π_j denote the clusters, k is the number of clusters, ϕ a non-linear function and $m_j = \frac{\sum_{x_i \in \pi_j} w(x_i) \phi(x_i)}{\sum_{x_i \in \pi_j} w(x_i)}$ the i-th cluster center.

According to [26], there is a direct relationship between the trace maximization of the normalized cut in spectral clustering and the kernel k-means problems. Therefore, the objective function represented in (1) can be written as follows:

$$D(\{\pi_j\}_{j=1}^k) = \text{trace}(W^{\frac{1}{2}}\phi^T\phi W^{\frac{1}{2}}) - \text{trace}(Y^T W^{\frac{1}{2}}\phi^T\phi W^{\frac{1}{2}}Y) \quad (2)$$

where, W is the diagonal matrix of all the weights. Y is the $N * k$ orthonormal cluster assignment matrix, i.e., $Y^T Y = I$.

Considering that $\phi^T\phi$ is simply the kernel matrix K of the data, and $\text{trace}(W^{\frac{1}{2}}\phi^T\phi W^{\frac{1}{2}})$ is a constant term, the minimization of the objective function in (2) is equivalent to the minimization of $-\text{trace}(Y^T W^{\frac{1}{2}}K W^{\frac{1}{2}}Y)$.

In situations where computing the spectral clustering algorithm is difficult, the weighted-kernel k-means algorithm is particularly useful as an alternative algorithm.

3.2 Graph Convolution Network Overview

A graph is represented by $G = (V, E, X)$ where V is a set of N nodes, and E a set of edges such as an edge (v_i, v_j) is a link connecting the nodes v_i and v_j . $X = [x_1; x_2; \dots; x_N]$, $\in R^{N*d}$ is a node feature matrix, where $x_i \in R^d$ denotes a feature vector of node v_i [12]. The structure of a graph G can be represented by two principal matrices: the adjacency matrix $A \in R^{N*N}$ where $a_{ij} = 1$ if there is an edge between nodes v_i, v_j , and $a_{ij} = 0$ otherwise, and the similarity matrix S which is a square matrix of $N * N$ and symmetric ($s_{ij} = s_{ji}$), s_{ij} represents the weight of the edge (v_i, v_j) .

The GCN presented in [13] uses a semi-supervised learning approach on data structured in graphs. It handles the challenge of labeling nodes in a network where labels are only known for a limited portion of nodes by estimating labels for unlabeled data, for that it is referred to as semi-supervised learning technique. A GCN model learns hidden layer representations that encode both local graph structure and node characteristics.

The layer-wise propagation rule of the GCN for semi-supervised learning is:

$$H^i = \sigma(\hat{A}H^{i-1}W^i) \quad (3)$$

H^i : output matrix of the i -th inner layer of the GCN, $H^i \in R^{N*h_i}$ with h_i the number of features of the inner-layer i . H^{i-1} : the input of the i -th inner layer of the GCN, $H^{i-1} \in R^{N*h_{i-1}}$. For the first layer, H^{i-1} is the features matrix X of the graph. A : indicates the adjacency matrix of the graph. \hat{A} its normalized matrix calculated as follows: $\hat{A} = \hat{D}^{-\frac{1}{2}}(A + I)\hat{D}^{-\frac{1}{2}}$ where, $D = \sum_i (A + I)_{ij}$, and I the identity matrix. W^i : the weight of the layer i , $W^i \in R^{d_{i-1}*d_i}$. σ is an activation function of internal layers such as rectified linear unit (ReLU) [13].

Then, the Softmax activation function, defined in [13] is applied in order to obtain the predicted label matrix $F \in R^{N*k}$ where k is the number of classes.

The GCN network is usually trained with the cross-entropy loss function which is typically used in classification problems. In Kejani, Dornaika & Talebi

(2020) [27], multiple regularization loss was added to the cross-entropy loss function to give better clustering results. The loss equation then becomes:

$$Loss = - \sum_{i=1}^l \sum_{c=1}^k y_{ic} \log(F_{ic}) + \lambda \text{trace}(F^T L F) \quad (4)$$

where y_{ic} represents the true label of the dataset, F_{ic} is the estimated probability of the i -th sample to be in class c . l denotes the total number of labels samples, λ is a hyper-parameter, and L denotes the Laplacian matrix of the graph. The model parameters W^i of GCN can be trained by minimizing Eq. (4) with gradient descent.

4 The Proposed Method

Given a graph $G(V, E, X)$, and its adjacency matrix A , our goal is to partition the nodes of this graph into k different clusters using GCN in an unsupervised way.

4.1 Model Architecture

To achieve our goal, we design a three-layer GCN as shown in Fig. 1. We first feed the node feature matrix $X \in R^{N*d}$ into the network, then learn the first propagation layer output $H^1 \in R^{N*h_1}$, and $H^2 \in R^{N*h_2}$ for the second layer by:

$$H^1 = \text{ReLu}(\widehat{A}XW^1) \quad (5)$$

$$H^2 = \text{ReLu}(\widehat{A}H^1W^2) \quad (6)$$

where $\text{ReLu}()$ is an activation function, \widehat{A} the normalized adjacency matrix of the graph, $W^1 \in R^{d*h_1}$, and $W^2 \in R^{h_1*h_2}$ are the trainable weight matrices of the inner layers and h_1, h_2 the number of features in the hidden layers.

In the output layer H^3 , we map H^2 to a k -dimensional space, where k indicates the number of clusters:

$$H^3 = H^2W^3 \quad (7)$$

$W^3 \in R^{h_2*k}$ is a learnable model parameter matrix. Then, to be able to use the output of the GCN in an unsupervised learning strategy, we perform a cholesky decomposition [12] on $(H^3)^T H^3$, i.e. $(H^3)^T H^3 = Q^T Q$, where $Q \in R^{k*k}$ is a lower triangular matrix. Finally, we obtain the orthogonal form of the GCN output which also represents the soft clustering assignment matrix output of the model:

$$H = H^3(Q^{-1})^T \quad (8)$$

After getting the orthogonalized output, our clustering loss function will be applied to it in order to train the model parameters.

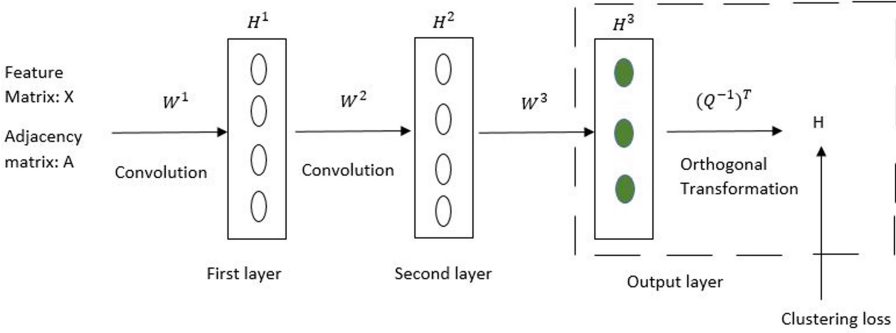


Fig. 1. The model architecture of the graph convolution neural network.

4.2 Model Loss Function and Clustering

As mentioned before, our main goal is to train the GCN in an unsupervised manner, i.e. the training process does not include any prior information about the ground truth labels. To achieve it, we have merged the rectified version of the kernel k-means objective function given by $-\text{trace}(Y^T W^{\frac{1}{2}} K W^{\frac{1}{2}} Y)$ with the classic spectral loss that use the graph. IN the rectified version of the kernel k-means loss we replace W by I (Identity matrix) and Y by H the model output. the model output. In this case, our proposed model is trained by minimizing the following loss function:

$$Loss = -\text{trace}(H^T K H) + \lambda \text{trace}(H^T L H) \quad (9)$$

where H is the model orthogonalized output, L the Laplacian graph matrix, λ is a hyper-parameter used to balance between the two terms, usely $\lambda \in [0, 1]$. K is the kernel matrix, in our case, we choose K the Gaussian kernel matrix given by: $K = \exp(-\frac{\|x_i - x_j\|^2}{s^2})$, $x_i, x_j \in X$ and s^2 is equal to the average value of the distance between the data points. Using this loss function, the main advantage is that no prior knowledge is needed to train the parameters of our model. Therefore, our GCN model is an unsupervised model. Moreover, the use of kernel k-means loss function allows us also to acquire the advantages of spectral clustering as already mentioned in paragraph Sect. 3.1. The objective function of spectral clustering and kernel k-means are similar. Furthermore, during the feature propagation phase and the learning phase, the model adapts both the structure and the features of the graph.

The parameters of the model W^1, W^2, W^3 are trained by minimizing equation (9) with gradient descent. After finishing the model training process, we perform k-means on its output H to obtain the partitions of nodes.

With the proposed loss function (9), two interesting and friendly clustering types of constraints are imposed on the unknown representation matrix H : (i) the kernel K-means loss enforces compactness in the space induced by the kernel matrix (information derived from features) and (ii) the spectral clustering term

enforces smoothness of the representation over the graph (information derived from data structures).

5 Experiments

We evaluate our model on three datasets widely used for the analysis of attributed graphs: Cora, Citeseer, Pubmed [28].

These datasets are citation networks where the nodes correspond to publications and are connected by an edge if one cites the other. Each node is associated with a feature vector, which represents the content of the document. Informations of these datasets are summarized in Table 1.

Table 1. Datasets Information

Datasets	Nodes	Edges	Features	Clusters
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
Pubmed	19717	44338	500	3

5.1 Baselines

We compare our methods with two different types of baselines:

The first type is methods that use only node features or graph structure: k-means partition nodes using the graph features matrix [7], spectral clustering [8] which takes the adjacency matrix of nodes hence the structure of graph.

The second type is methods adopting both node characteristics and graph structure: such as variational graph autoencoder (ARVGE) [11], and SENet [12] method which improves graph structure by leveraging the information of shared neighbors, and learns node embedding with the help of a spectral clustering loss.

5.2 Evaluation Metrics and Experimental Setups

To assess the clustering performance of our method, we use three performance measures [29]: clustering accuracy (Acc), normalized mutual information (NMI), and adjusted rand index (ARI). Acc is the ratio of correctly predicted data. NMI assesses cluster quality by measuring the match between true and predicted labels. ARI measures the separation ability between clusters and the recognition ability for each cluster.

For our model, we build the network with randomly initialized weights with 16 hidden units and two internal layers as in [12]. We adopt the Adam optimizer [30] with a 0.001 learning rate and run experiments on Pytorch. The λ hyperparameter in the proposed loss function is fixed to 0.5.

5.3 Result Analysis

Tables 2, 3, and 4 show the clustering and standard deviation results obtained after running each method 10 times on each dataset. The best results are in bold. The interpretations are as follows: U-GCN, the algorithm proposed in this paper, outperforms clustering approaches that just use node attributes or graph structure such as k-means and spectral clustering, as it effectively incorporates and captures both kinds of information.

Furthermore, U-GNC improves the results of methods that use the attributes and structure of the graph at the same time, since U-GCN adapts the structure and attributes of the graph in the propagation of features in the model and in the training process not as ARVGE as it only uses them both during the propagation phase. In addition, the use of the weighted kernel k-means objective loss function which is similar to the spectral clustering objective loss function facilitates the transmission of information about the cluster structure at each layer.

Table 2. Clustering results on Cora

Methods	ACC	NMI	ARI
k-means	0.3465 \pm 0.0136	0.1673 \pm 0.0154	0.0958 \pm 0.0107
Spectral	0.3419 \pm 0.0203	0.1949 \pm 0.0194	0.0181 \pm 0.0186
ARVGE [11]	0.6380 \pm 0.0096	0.4500 \pm 0.0083	0.3740 \pm 0.0071
SENet [12]	0.7192 \pm 0.0066	0.5508 \pm 0.0065	0.4896 \pm 0.0109
U-GCN(this paper)	0.73 \pm 0.00872	0.56 \pm 0.00569	0.495 \pm 0.00859

Table 3. Clustering results on Citeseer

Methods	ACC	NMI	ARI
k-means	0.3849 \pm 0.0237	0.1702 \pm 0.0206	0.1243 \pm 0.0192
Spectral	0.2591 \pm 0.0109	0.1184 \pm 0.0168	0.0012 \pm 0.0137
ARVGE [11]	0.5440 \pm 0.0139	0.2610 \pm 0.0172	0.2450 \pm 0.0164
SENet [12]	0.6752 \pm 0.0075	0.417 \pm 0.0082	0.4237 \pm 0.0097
U-GCN(this paper)	0.68 \pm 0.00617	0.417 \pm 0.00632	0.43 \pm 0.00859

Table 4. Clustering results on Pubmed

Methods	ACC	NMI	ARI
k-means	0.5732 \pm 0.0381	0.2912 \pm 0.0352	0.2505 \pm 0.0346
Spectral	0.3974 \pm 0.0267	0.0346 \pm 0.0309	0.0018 \pm 0.0291
ARVGE [11]	0.5822 \pm 0.0100	0.2062 \pm 0.0057	0.2045 \pm 0.0065
SENet [12]	0.6759 \pm 0.0062	0.3061 \pm 0.0145	0.2966 \pm 0.0113
U-GCN(this paper)	0.68 \pm 0.00527	0.3105 \pm 0.0102	0.31 \pm 0.0010

6 Conclusion

In this work, we proposed an unsupervised method based on GCN for the partitioning problem in graphs without prior knowledge. Inspired by the mechanism of GCN and the relationship between kernel weighted k-means and spectral clustering, we integrated two objective functions to achieve clustering of nodes by merging structural and feature information. The proposed loss function is suitable for training the GCN model when there is no information available on the labels. This approach is also a new solution for clustering graphs in an unsupervised way. The experimental results demonstrate that the proposed method achieves better performance on three citation networks, compared to state-of-the-art algorithms. In future work, we aim to study the effect of changing the trade-off hyper-parameter λ .

References

1. Shah, S.A., Koltun, V.: Deep continuous clustering. [arXiv:1803.01449](https://arxiv.org/abs/1803.01449) (2018)
2. Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel M., Cremers D.: Clustering with deep learning: taxonomy and new methods. [arXiv:1801.07648](https://arxiv.org/abs/1801.07648) (2018)
3. Wang, X., Li, J., Yang, L., Mi, H.: Unsupervised learning for community detection in attributed networks based on graph convolutional network, *Neurocomputing* **456**, 147–155 (2021)
4. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2014)
5. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI) (2016)
6. Moradi Fard, M., Thonet, T.: Gaussier E. Jointly clustering with k-means and learning representations, *pattern recognition letters, deep k-means* (2020)
7. Gao, B., Yang, Y., Gouk, H., Hospedales, T.M.: Deep Clustering with Concrete k-means. University of Edinburgh, United Kingdom Samsung AI Centre, Cambridge, United Kingdom (2020)
8. von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416 (2007). <https://doi.org/10.1007/s11222-007-9033-z>
9. Yang, X., Deng, C., Zheng, F., Yan, J., Liu, W.: Deep spectral clustering using dual autoencoder network. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2019)
10. Affeldt, S., Labiod, L., Nadif, M.: Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recogn.* **108**, 107522 (2020)
11. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI) (2018)
12. Zhang, X., Liu, H., Xiao-Ming, W., Zhang, X., Liu, X.: Spectral embedding network for attributed graph clustering. *Neural Netw.* **142**, 388–396 (2021)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with convolutional networks. In: 5th International Conference on Learning Representations (2017)

14. Jin, D., Li, B., Jiao, P., He, D., Shan, H.: Community detection via joint graph convolutional network embedding in attribute network. In: Proceedings of ICANN, pp. 594–606 (2019)
15. Wang, X., Li, J., Yang, L., Mi, H.: Unsupervised learning for community detection in attributed networks based on graph convolutional network. *Neurocomputing* **456**, 147–155 (2021)
16. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 855–864 (2016)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of 27th Annual Conference on Neural Information Processing Systems (NeurIPS), pp. 3111–3119 (2013)
18. Cao, S., Lu, W., Xu, Q.: Grarep: learning graph representations with global structural information. In: Proceedings of the 2015 ACM on Conference on Information and Knowledge Management (CIKM), pp. 891–900 (2015)
19. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 1225–1234 (2016)
20. Dai, Q., Li, Q., Tang, J., Wang, D.: Adversarial network embedding. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), pp. 2167–2174 (2018)
21. Zhou, Y., Cheng, H., Yu, J.X.: Clustering large attributed graphs: an efficient incremental approach. In: Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia (ICDM), pp. 689–698 (2010)
22. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: A model-based approach to attributed graph clustering. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 505–516 (2012)
23. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. *VLDB Endowment* **2**(1), 718–729 (2009)
24. Sun, H., He, F., Huang, J., Sun, Y., Li, Y., Wang, C., et al.: Network embedding for community detection in attributed networks. *ACM Trans. Knowl. Discov. Data* **14**, 1–25 (2020)
25. Jin, D., Li, B., Jiao, P., He, D., Shan, H.: Community detection via joint graph convolutional network embedding in attribute network. In: Proceedings of ICANN, pp. 594–606 (2019)
26. Dhillon, I. S., Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
27. Kejani, M.T., Dornaika, F., Talebi, H.: Graph Convolution Networks with manifold regularization for semi-supervised learning. *Neural Network* **127**, 160–167 (2020)
28. Kipf, T.N., Welling, M.: Variational graph auto-encoders (2016)
29. Aggarwal, C.C., Reddy, C.K.: Data clustering: algorithms and applications. The CRC Press, Boca Raton (2014)
30. Kingma, D. P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)

Author Queries

Chapter 3

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author and email address have been identified as per the information available in the Copyright form.	
AQ2	This is to inform you that as the Institutional email address of the corresponding author is not available in the manuscript, we are displaying the private email address in the PDF and SpringerLink. Do you agree with the inclusion of your private e-mail address in the final publication?	