



Community Discovery Algorithm Based on Improved Deep Sparse Autoencoder

Dianying Chen¹, Xuesong Jiang¹(✉), Jun Chen¹, and Xiumei Wei²(✉)

¹ Qilu University of Technology (Shandong Academy of Sciences),
Jinan, Shandong, China
jxs@qlu.edu.cn

² Qilu University of Technology, Jinan, Shandong, China
wxm@qlu.edu.cn

Abstract. Community structures are everywhere, from simple networks to real-world complex networks. Community structure is an important feature in complex networks, and community discovery has important application value for the study of social network structure. When dealing with high-dimensional matrices using classical clustering algorithms, the resulting communities are often inaccurate. In this paper, a community discovery algorithm based on an improved deep sparse autoencoder is proposed, which attempts to apply to the community discovery problem through two different network similarity representations. This can make up for the deficiency that a single network similarity matrix cannot fully describe the similarity relationship between nodes. These similarity representations can fully describe and consider local information between nodes in the network topology. Then, a weight-bound deep sparse autoencoder is constructed based on an unsupervised deep learning method to improve the efficiency of feature extraction. Finally, feature extraction is performed on the similarity matrix to obtain a low-dimensional feature matrix, and the k-means clustering method is used to cluster the low-dimensional feature matrix to obtain reliable clustering results. In various extensive experiments conducted on multiple real networks, the proposed method is more accurate than other community discovery algorithms using a single similarity matrix clustering algorithm, and the efficiency of the community discovery algorithm is much more improved.

Keywords: complex network · node similarity · community discovery · deep learning · deep sparse autoencoders

1 Introduction

The wide application of complex networks in real life and their superior performance have made many researchers have a strong interest in the study of complex networks. In recent years, community detection has become a hot topic

This work was supported in part by National Key R&D Program of China (No. 2019YFB1707000).

in complex network research, aiming to discover the underlying network structure and important information [1], so community discovery can provide important research value and scientific significance for the analysis of network structure. Real-world networks have complex network structures, and it is difficult to obtain accurate community structures using traditional clustering methods. Fully mining the complex information in the network and constructing the similarity relationship between network nodes is the key to improve the accuracy of community discovery algorithm. Usually, we use the similarity matrix to describe the similarity relationship between nodes. The similarity matrix is obtained by transforming the adjacency matrix of the network by a single function.

In recent years, the community discovery of using deep learning to solve large-scale complex networks has become a hot topic. Although researchers have proposed many deep learning-based models, there are problems of high model complexity and low training efficiency of models with too many parameters. Wang et al. [2] proposed to learn the similarity matrix representation using a multi-layer autoencoder and an extreme learning machine, which improved the accuracy and stability, but the training time was too high. The algorithm proposed by Jia et al. [3] uses an adversarial network to optimize the strength of the node membership community, so that the generator and the discriminator compete with each other. The alternate iteration of the two improves the accuracy, but the many parameters cause poor universality. Li et al. [4] proposed an algorithm that learns the characterization of the edges of the network by using an edge clustering algorithm to transform into the overlapping community division of nodes, which improves the accuracy, but the stability is not high. The algorithm proposed by Shang et al. [5] uses a multi-layer sparse auto-encoder to reduce the dimensionality of similar matrices and perform representation learning, and uses k-means clustering to improve the accuracy, but the parameters of the algorithm are not easy to choose, and the universality is relatively low. Zhang et al. [6] used multi-layer spectral clustering to divide the community of the network. The accuracy of this algorithm is higher than that of a single layer, but the number of layers is an unstable parameter.

The above algorithms all use a single function-based similarity matrix as the input matrix. However, this similarity matrix based on a single function cannot reflect the local information of each node. In addition to the directly connected nodes in the network, there are also indirectly connected nodes, and there are also different similarity relationships between these nodes. Second, the training of deep learning models is inefficient due to the large amount of experimental data and excessive parameters.

Therefore, a community discovery algorithm CoIDSA (Community Discovery Algorithm Based on Improved Deep Sparse Autoencoder) based on an improved deep sparse autoencoder is proposed to address the above problems. There are three main contributions of this paper.

- The new similarity matrix is constructed from two different functions, which can fully exploit the similarity relationship between each node in the network topology.

- Develop a learning method of deep sparse autoencoder based on weight binding, extract the feature representation in the similarity matrix, and obtain a low-dimensional feature matrix. This learning method halved the model parameters, speeding up training and reducing the risk of overfitting.
- Extensive experiments are conducted on multiple real datasets. The experimental results show that the CoIDSA algorithm proposed in this paper can obtain a more accurate network community structure.

2 Related Work

2.1 Community Discovery

Given a network, the subgraph corresponding to a subset of closely connected nodes in the network is called a community, and the process of finding out its community structure is called community discovery.

The current mainstream community discovery algorithms are deep learning-based community discovery algorithms. Autoencoders are very common in community discovery because they can efficiently represent nonlinear real-world networks. In [7], Cao et al. proposed a new method to combine a modular model and a normalized cut model via an autoencoder. In [8], the DeCom model is proposed, which exploits the idea of autoencoders. For extracting overlapping communities in large-scale networks. Different from other autoencoder-based schemes, Cavallari et al. [9] proposed the ComE community embedding framework, where the community embedding problem is treated as a multivariate Gaussian distribution to enhance community discovery. Xu et al. [10] proposed the community discovery approach of CDMEC. This framework combines migration learning and stacked autoencoders to generate feature representations of low-dimensional complex networks.

2.2 Deep Sparse Autoencoder

An autoencoder is a neural network that uses a backpropagation algorithm to make the output value equal to the input value. It first compresses the input into a latent space representation, and then utilizes the representation to reconstruct the output. Autoencoders are divided into two parts: encoder and decoder. Autoencoders can learn efficient representations of input data through unsupervised learning. This efficient representation of the input data is called an encoding, and its dimensions are generally much smaller than the input data, making autoencoders useful for dimensionality reduction.

Sparse autoencoder [11] is a kind of autoencoder, which is a derivative autoencoder generated by adding sparsity constraints to the hidden layer neurons of autoencoder, and is able to learn the features that best express the sample in a harsh environment and effectively dimensionalize the data sample. The model is finally trained by calculating the error between the output of the autoencoder and the original input, and continuously adjusting the parameters of the autoencoder.

3 Method

In this paper, we propose a community discovery algorithm CoIDSA based on improved deep sparse autoencoder, which mainly consists of three steps: Firstly, two similarity matrices are obtained by preprocessing the adjacency matrix according to two different functions to enhance the similarity of nodes; Secondly, a weight-bound deep sparse autoencoder is constructed to extract features from the similarity matrix to obtain a low-dimensional feature matrix with obvious features and improve the efficiency of model training; Finally, the two low-dimensional feature matrices are combined into a new matrix and clustered using the k-means method to obtain the community structure. The algorithm model diagram is shown in Fig. 1.

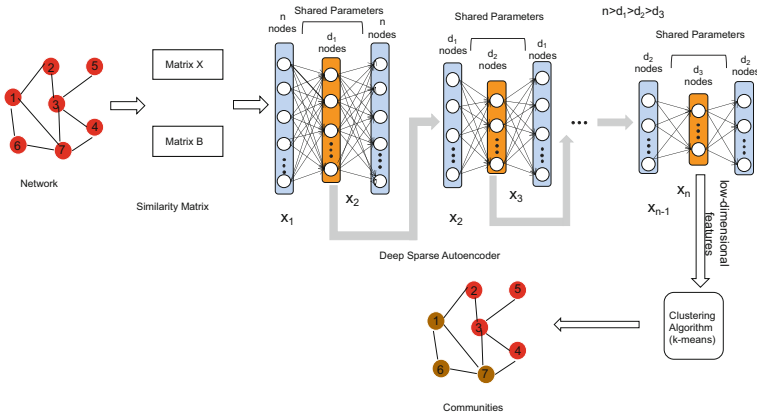


Fig. 1. Algorithm model diagram

3.1 Enhance Node Similarity

Use the s -hop function to preprocess the obtained matrix as the first similarity matrix. It can solve the problem of losing the similarity relationship information between many nodes and not reflecting the complete local information of each node because the similarity relationship between the nodes that are not directly connected cannot be represented.

Suppose network graph $G = (V, E)$, for nodes $v, u \in V$, if the shortest path from node v to u is called s , then node v can reach node u through s hops.

Node Similarity: Given a network graph $G = (V, E)$, for nodes $v, u \in V$, the similarity $\text{sim}(v, u)$ between nodes v and u is defined as:

$$\text{sim}(v, u) = e^{\sigma(1-s)} \tag{1}$$

Among them, $s \geq 1$, with the increase of the number of hops s , the similarity between nodes decreases continuously, σ is called the attenuation factor,

$\sigma \in (0, 1)$, which controls the attenuation degree of the similarity of nodes, the greater the σ , the similarity between nodes decay faster.

Network similarity matrix: Given a network graph $G = (V, E)$, $X = [x_{ij}]_{n \times n}$ is a matrix corresponding to the network graph G . Use formula (1) to calculate the similarity $x_{ij} = \text{sim}(v_i, v_j)$ between the corresponding two nodes v_i and v_j in X , $v_i, v_j \in V$, then X is called the similarity matrix of G .

When the number of hops is greater than a certain threshold, two nodes that are not in the same community will also get a certain similarity value, which will make the boundary of the community structure more blurred. Therefore, the hop threshold s is set, and only the similarity between nodes that can reach each other within s hops is calculated to ensure that the topology information of the graph is enhanced without affecting the division of community boundaries.

Inspired by the modular function Q , which is defined as the difference between the number of edges within communities and the expected number of such edges among all pairs of nodes, the adjacency matrix is preprocessed by the modular function Q , and the modular matrix B is used as the second similarity matrix.

$$Q = \frac{1}{4m} \sum_{i,j} \left(A(i,j) - \frac{k_i k_j}{2m} \right) (h_i h_j) = \frac{1}{4m} h^T B h \quad (2)$$

where $\frac{k_i k_j}{2m}$ is the number of edges between nodes i and j , h is the community membership vector, k_i is the degree of node i , A is the adjacency matrix, and m is the total number of edges in the network.

3.2 Feature Extraction

This section describes the detailed process of feature extraction by CoIDSA algorithm. The process of constructing a deep sparse autoencoder based on weight binding is described, and the preprocessed similarity matrices X and B are subjected to feature extraction, and then the two low-dimensional feature matrices after feature extraction are merged, and finally the community structure is obtained by clustering.

We give the similarity matrix $X = [x_{ij}]_{n \times n}$ of the network graph G as an example, and take the similarity matrix X as the input matrix of the autoencoder, and the encoder maps the input data X to the hidden layer features. The hidden layer feature $\xi_i \in R^d$ is obtained by formula (3), which is expressed as:

$$\text{Encoding: } \xi_i = s_f(Wx_i + b) \quad (3)$$

Among them, S_f is the nonlinear activation function, such as the sigmoid function $\text{sigmoid}(x) = \frac{1}{(1+\exp(-x))}$; $W \in R^{d \times n}$ is the weight matrix, and $b \in R^{d \times 1}$ is the bias vector of the coding layer. The decoder reconstructs the hidden layer feature ξ_i , and the decoding result $x'_i \in R^{n \times 1}$ can be obtained by formula (4) as the output information:

$$\text{Decoding : } x'_i = s_g \left(\hat{W} \xi_i + \hat{b} \right) \tag{4}$$

where S_g is an activation function of the decoder. $\hat{W} = W^T \in R^{n \times d}$ is the weight matrix and $\hat{b} \in R^{n \times 1}$ is the bias vector of the decoding layer.

During the training process, the autoencoder adjusts the four parameters $\delta = \{W, \hat{W}, b, \hat{b}\}$ of the weight matrix and the bias vector, but in order to improve the training efficiency of the autoencoder, the method of binding weights is adopted here. The encoder and the decoder share the weights, that is, the weights of the decoder and the encoder are transposed $W = \hat{W}$ to each other, and will be updated during backpropagation public weight matrix. Then minimize the reconstruction error of x_i and x'_i :

$$\min_{W, \hat{W}, b, \hat{b}} \text{mize} \sum_{i=1}^n \left\| s_g \left(\hat{W} s_f (W x_i + b) + \hat{b} \right) - x_i \right\|_2^2 \tag{5}$$

We use KL divergence, adding a sparsity constraint to the autoencoder:

$$\sum_{j=1}^d KL \left(\rho \parallel \frac{1}{n} \sum_{i=1}^n \xi_i \right) \tag{6}$$

Then the reconstruction error of building a sparse autoencoder is:

$$L(\delta) = \sum_{i=1}^n \left\| s_g \left(\hat{W} s_f (W x_i + b) + \hat{b} \right) - x_i \right\|_2^2 + \alpha \sum_{j=1}^d KL \left(\rho \parallel \frac{1}{n} \sum_{i=1}^n s_f (W x_i + b) \right) \tag{7}$$

where α is the weight coefficient that controls the sparse penalty term, and ρ is a sparse parameter, usually a small value close to 0 ($\rho = 0.01$).

4 Experimental Results and Analysis

In order to test the performance of the proposed algorithm in this paper, real datasets are used for performance evaluation. The existing algorithms are compared with the CoIDSA algorithm proposed in this paper. This section is divided into three parts: experimental preparation, evaluation criteria and experimental results.

4.1 Experiment Preparation

DataSet. In this paper, we use four real datasets Football [12], Polblogs [13], Polbooks [14], and Dolphins [15] to validate our algorithm.

The specific dataset description is shown in Table 1.

Table 1. Dataset Information

Dataset	Number of nodes	Number of sides	communities
Football	115	613	12
Polblogs	1490	16718	2
Polbooks	105	441	3
Dolphins	62	159	2

For better experiments, we set up neural networks with different depths in different layers of the real-world network. For small datasets, choosing different depths of neural network layers can achieve better results in the feature extraction process. For large datasets, we use powers of 2 to reduce the depth of the neural network in the number of nodes in each layer. In the experiments, we set the learning rate of the deep sparse autoencoder to 0.1. Considering the sparsity limitation, we set the sparsity parameter to 0.01. For the number of layers of the deep sparse autoencoder, we choose a suitable 3-layer training layer in each network to provide more accurate feature extraction results. Such parameter settings can effectively utilize the characteristics of deep sparse autoencoders to ensure the accuracy of the feature extraction process in datasets of different sizes through layer-by-layer greedy training. The details are shown in Table 2.

Table 2. Structure of deep neural network

Dataset	Number of nodes
Dolphins	62-32-16
Football	64-32-16
Polbooks	64-32-16
Polblogs	256-128-64

4.2 Evaluation Criteria and Comparison Algorithms

Evaluation Criteria. The accuracy of the resulting community is judged by the community $RC = \{C'_1, C'_2, \dots, C'_k\}$ obtained by the real community $GT = \{C_1, C_2, \dots, C_n\}$ through our algorithm. where n is the number of real communities and k is the number of resulting communities. This paper uses two general community evaluation criteria, NMI and modularity Q , to analyze the accuracy of community discovery.

Contrast Algorithm. In order to fully demonstrate the effectiveness of the proposed algorithm in this paper, the proposed algorithm is compared with existing algorithms, namely CoDDA [5], SSCF [16], DNR [17] and DSACD [18]. The

CoDDA algorithm is a community discovery algorithm based on sparse autoencoder, which performs feature extraction on the similarity matrix of a single function, and then obtains the community structure by clustering; The SSCF algorithm is a sparse subspace community detection method based on sparse linear coding; The DNR algorithm is to use the nonlinear model in the deep neural network for community detection; The DSACD algorithm is to construct similarity matrices to reveal the relationships between nodes and to design deep sparse autoencoders based on unsupervised learning to reduce the dimensionality and extract the feature structure of complex networks.

4.3 Experimental Results

In this section, the experiments consist of three parts, which are ablation experiment, contrast experiment and parameter experiment. We compare the algorithm of this paper with other algorithms based on two performance evaluation criteria, NMI and modularity Q . To explain the results of parameter selection, we give the parameter experiments.

Ablation Experiment. The experimental results are shown in Fig. 2. It can be seen from the figure that the similarity matrix obtained based on the number of hops is 5.5% higher than the clustering result obtained based on the similarity matrix of the modular function, which is due to the preprocessing method based on the number of hops, which recalculates the similarity matrix of the network nodes and improves the local information of the nodes more. The clustering results of the similarity matrix obtained based on two functions are on average 5.2% higher than the similarity matrix obtained after the hop-count based preprocessing, so it can be clearly seen that the experimental results of two similarity matrices outperform the experimental results of one similarity matrix. Thus we can conclude that using two different similarity matrices is effective for the accuracy improvement of the community discovery algorithm.

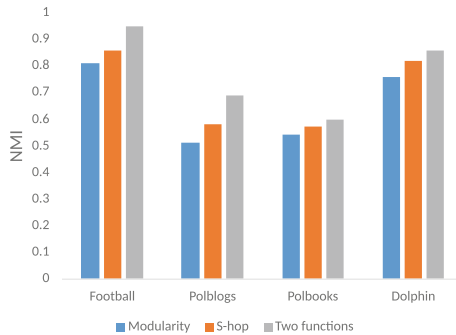


Fig. 2. Experimental results of different functions as input.

Comparative Experiment. In order to verify the accuracy of this algorithm in community discovery, the CoIDSA algorithm is used to compare with the existing algorithms, and the results of the community evaluation indicators NMI and Q are shown in Fig. 3.

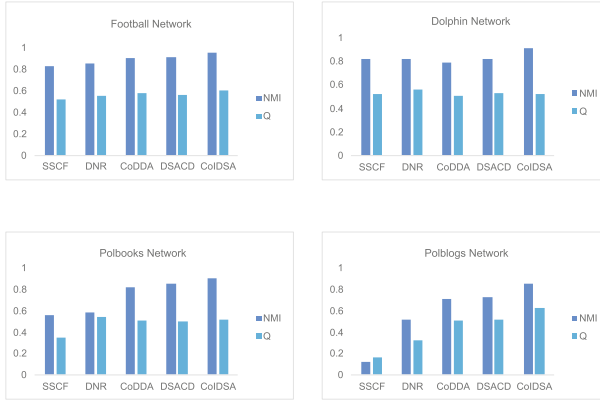


Fig. 3. Comparison of NMI and Q performance of CoIDSA algorithm with 4 existing community detection algorithms on 4 real datasets.

Table 3. Analysis of community discovery results

Algorithm	Dolphins		Football		Polbooks		Polblogs	
	NMI	Q	NMI	Q	NMI	Q	NMI	Q
SSCF	0.82	0.52	0.83	0.52	0.56	0.35	0.12	0.16
DNR	0.81	0.56	0.85	0.55	0.58	0.54	0.51	0.32
CoDDA	0.79	0.51	0.90	0.58	0.82	0.51	0.71	0.51
DSACD	0.82	0.53	0.91	0.56	0.85	0.50	0.73	0.52
CoIDSA	0.91	0.52	0.95	0.60	0.90	0.52	0.86	0.63

The metrics of community detection on different algorithms for the test dataset are presented in Table 3 and Fig. 3. The NMI metrics of the CoIDSA algorithm as a whole are higher compared to all other algorithms. This is because the algorithm adopts the preprocessing method based on the number of hops and the modularization function before clustering, and recalculates the similarity matrix of the network nodes, so that the complexity of the network structure is fully considered, and different methods are used to construct multiple similarity matrix. Then, a deep sparse autoencoder bound by weights is used to perform feature extraction on the similarity matrix to obtain a low-dimensional feature matrix with more obvious features, and then cluster to obtain a more accurate community. Community results using the CoIDSA algorithm are on

average 5% higher than other algorithms. This is because the CoIDSA algorithm uses multiple similarity matrix inputs before feature extraction, which improves the local information of nodes, and the obtained low-dimensional feature matrix can better express the structure of the network, which proves the effectiveness of the CoIDSA algorithm. The DSACD algorithm is higher than the CoDDA algorithm in all indicators, because the improvement of the DSACD algorithm in the CoDDA algorithm is consistent with the description of the paper. The clustering results on the Polblogs dataset are lower than those of the other three datasets, because the Polblogs dataset has many times more data than the other three datasets, but the indicators of the CoIDSA algorithm also reach a higher level, which shows that the CoIDSA algorithm is also effective on larger datasets.

However, due to the characteristics of deep learning, more time is needed during the training process, and the running time of each algorithm on the four datasets is shown in Table 4.

Table 4. Running time on different datasets

Dataset	Dolphins(s)	Football(s)	Polbooks(s)	Polblogs(min)
SSCF	1.91	2.89	3.65	5.89
DNR	2.23	3.78	4.67	6.35
CoDDA	1.85	2.56	2.43	4.36
DSACD	1.83	2.43	2.25	4.12
CoIDSA	1.36	1.19	1.07	3.54

From the experimental results in Table 4, it can be seen that the CoIDSA algorithm proposed in this paper is better than the current mainstream community discovery algorithm in time. The reason is that we choose weight binding as the optimization method, and bind the weight of the decoder layer to the encoder layer. This halved the model parameters, speeding up training and reducing the risk of overfitting. Where the weights are shared among the layers and the common weight matrix will be updated during backpropagation for the purpose of improving training efficiency. A weight-bound deep sparse autoencoder is used for training, and the experimental results show that this algorithm is effective in improving the efficiency of community discovery.

Parametric Experiment. Deep sparse autoencoders for community discovery contain two important parameters: jump threshold (s) and decay factor (σ) in the first similarity matrix. These two parameters have a direct impact on the clustering results. This section sets up experiments to find the optimal parameters.

- The number of jump thresholds s

For the Football dataset, the number of nodes in each layer of a deep sparse autoencoder is [64–32], and the decay factor $\sigma = 0.5$ is used to analyze the impact of different jump thresholds on NMI . As shown in Fig. 4, under different values of the jump threshold s , compared to directly using the k-means algorithm to cluster the similarity matrix, the result community obtained by the CoIDSA algorithm is more accurate. The CoIDSA algorithm can greatly improve the accuracy of community results.

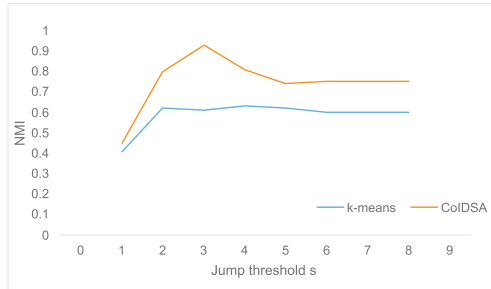


Fig. 4. NMI values of CoIDSA algorithm and k-means algorithm under different jump threshold s in Football dataset.

It can be seen from Fig. 4 that with the increase of s , the NMI value shows a trend of increasing first and then decreasing, which is also in line with the actual situation. Because in the real network, there is a certain degree of similarity between nodes that are not directly connected but can be reached after a certain number of hops. If the number of hops is too large, there is a certain degree of similarity between nodes far away, but the ambiguity of the community identification boundary is increased. For the smaller datasets Football, Polbooks, and Dolphins, the jump threshold was chosen to be 3 hops, and for the larger dataset Polblogs, 5 hops was chosen to achieve optimal results.

– Decay factor σ

For the Football dataset, the number of nodes in each layer of a deep sparse autoencoder is [64–32], and the jump threshold $s=3$, and the influence of different attenuation factors on NMI is analyzed. As shown in Fig. 5, under different values of the attenuation factor, compared to directly using the k-means algorithm to cluster the similarity matrix, the result community obtained by the CoIDSA algorithm is more accurate. The CoIDSA algorithm can greatly improve the accuracy of community results.

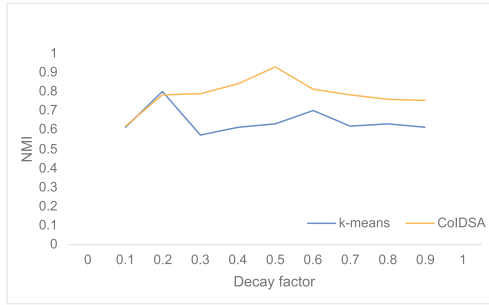


Fig. 5. *NMI* values of CoIDSA algorithm and k-means algorithm under different attenuation factors in Football dataset.

It can be seen from Fig. 5 that with the increase of the attenuation factor, the *NMI* value shows a trend of first increasing and then decreasing. When the attenuation factor is set to 0.5, the local characteristics of the node can be enhanced to achieve the optimal result.

5 Conclusions and Future Work

In this paper, in order to more effectively detect complex network structures with high-dimensional feature representations, we propose a community discovery algorithm based on an improved deep sparse autoencoder. Through experiments on real data sets, the community discovery algorithm based on the improved deep sparse autoencoder proposed in this paper has higher accuracy, stronger stability and faster training efficiency for community discovery. In addition, the main research object of this paper is static networks. Therefore, community discovery on dynamic networks will be the direction of future research.

Acknowledgment. This work was supported in part by National Key R&D Program of China (No. 2019YFB1707000).

References

1. Ding, Z., Chen, X., Dong, Y., Herrera, F.: Consensus reaching in social network degroot model: the roles of the self-confidence and node degree. *Inf. Sci.* **486**, 62–72 (2019)
2. Wang, F., Zhang, B., Chai, S., Xia, Y.: Community detection in complex networks using deep auto-encoded extreme learning machine. *Mod. Phys. Lett. B* **32**(16), 1850180 (2018)
3. Jia, Y., Zhang, Q., Zhang, W., Wang, X.: Communitygan: Community detection with generative adversarial nets. In: *The World Wide Web Conference*, pp. 784–794 (2019)

4. Li, S., Zhang, H., Wu, D., Zhang, C., Yuan, D.: Edge representation learning for community detection in large scale information networks. In: Doukeridis, C., Vouros, G.A., Qu, Q., Wang, S. (eds.) *MATES 2017*. LNCS, vol. 10731, pp. 54–72. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73521-4_4
5. Shang, J., Wang, C., Xin, X., Ying, X.: Community detection algorithm based on deep sparse autoencoder. *J. Softw.* **28**(3), 648–662 (2017)
6. Zhang, X., Newman, M.E.: Multiway spectral community detection in networks. *Phys. Rev. E* **92**(5), 052808 (2015)
7. Cao, J., Jin, D., Yang, L., Dang, J.: Incorporating network structure with node contents for community detection on large networks using deep learning. *Neurocomputing* **297**, 71–81 (2018)
8. Bhatia, V., Rani, R.: A distributed overlapping community detection model for large graphs using autoencoder. *Future Gener. Comput. Syst.* **94**, 16–26 (2019)
9. Cavallari, S., Zheng, V.W., Cai, H., Chang, K.C.-C., Cambria, E.: Learning community embedding with community detection and node embedding on graphs. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 377–386 (2017)
10. Xu, R., Che, Y., Wang, X., Hu, J., Xie, Y.: Stacked autoencoder-based community detection method via an ensemble clustering framework. *Inf. Sci.* **526**, 151–165 (2020)
11. Qiao, S., et al.: A fast parallel community discovery model on complex networks through approximate optimization. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1638–1651 (2018)
12. Khorasgani, R.R., Chen, J., Zaiane, O.R.: Top leaders community detection approach in information networks. In: *4th SNA-KDD Workshop on Social Network Mining and Analysis*, Citeseer (2010)
13. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 us election: divided they blog. In: *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43 (2005)
14. Newman, M.E.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
15. Lusseau, D., Newman, M.E.: Identifying the role that animals play in their social networks. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 271, no. suppl.6, pp. S477–S481 (2004)
16. Mahmood, A., Small, M.: Subspace based network community detection using sparse linear coding. *IEEE Trans. Knowl. Data Eng.* **28**(3), 801–812 (2015)
17. Yang, L., Cao, X., He, D., Wang, C., Wang, X., Zhang, W.: Modularity based community detection with deep learning. In: *IJCAI*, vol. 16, pp. 2252–2258 (2016)
18. Fei, R., Sha, J., Xu, Q., Hu, B., Wang, K., Li, S.: A new deep sparse autoencoder for community detection in complex networks. *EURASIP J. Wireless Commun. Networking* **2020**(1), 1–25 (2020). <https://doi.org/10.1186/s13638-020-01706-4>