Deep Attributed Network Embedding with Community Information

Li Xue[1,2], Wenbin Yao[1,2(✉)], Yamei Xia[1], and Xiaoyong Li[3]

[1] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
ymxia@bupt.edu.cn

[2] Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China

[3] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract. Attributed Network Embedding (ANE) aims to learn low-dimensional representation for each node while preserving topological information and node attributes. ANE has attracted increasing attention due to its great value in network analysis such as node classification, link prediction, and node clustering. However, most existing ANE methods only focus on preserving attribute information and local structure, while ignoring the community information. Community information reveals an implicit relationship between vertices from a global view, which can be a supplement to local information and help improve the quality of embedding. So, those methods just produce sub-optimal results for failing to preserve community information. To address this issue, we propose a novel method named DNEC to exploit local structural information, node attributes, and community information simultaneously. A novel deep neural network is designed to preserve both local structure and node attributes. At the same time, we propose a community random walk method and incorporate triplet-loss to preserve the community information. We conduct extensive experiments on multiple real-world networks. The experimental results show the effectiveness of our proposed method.

Keywords: Graph structured data · Network embedding · Deep learning · Node classification

1 Introduction

Networks are important and ubiquitous structures in the real world, including social networks, citation networks, and communication networks. Network Embedding (NE) aims to map vertex into low-dimensional space and is valuable

National Nature Science Foundation of China (61672111) and the Joint Fund of NSFC-General Technology Fundamental Research (U1836215,U1536111).

J. Lokoč et al. (Eds.): MMM 2021, LNCS 12572, pp. 653–665, 2021.
https://doi.org/10.1007/978-3-030-67832-6_53

for many data-mining applications such as node classification, link prediction [1], visualization [2], and anomaly detection [3].

Inspired by the success of Word2vec [4], early works based on skip-gram [5] mainly focus on exploring network structure. Node2Vec [6] explores network structure by biased random walks. Line [7] designs two loss functions to preserve network structure. Those methods only concentrate on preserving the local structure, ignoring community information. CARE [8] adopts a community-aware random walk to preserve community information. COME [9] designs a novel community embedding framework. M-NMF [10] designs a loss function based on modularity to preserve community structure. However, those NE methods just concentrate on network structure and pay less attention to node attributes, which play an important role in many applications. So, those NE methods just consider plain network and are not suitable for attributed networks.

Thus, another line of works is proposed for attributed network embedding, such as TADW [11] and DANE [12]. TADW incorporates node attributes into the matrix factorization framework. DANE designs two autoencoders to preserve node attributes and network structure together. However, those ANE methods don't take community information into account. When the network is sparse and the attribute is noisy, utilizing community structure can greatly improve the quality of node representations.

In order to obtain node representation of high quality, we try to incorporate network structure, node attributes, and community structure into the ANE framework. We propose a novel framework, called DNEC, which preserves community structure. DNEC employs two embedding layers to compress network structure and attribute separately and generate structure representation and attribute representation. Structure representation and attribute representation are connected as the input of the shared encoder. The shared encoder will compress two different representations into the unified representations spaces. Dual decoder employs two traditional fully connected neural networks to reconstruct node attributes and structure of the network. To preserve community structure, we propose a biased random walk to construct community triplets to calculate triplet-loss. In summary, our main contributions can be summarized as follows:

(1) We design a novel ANE framework, which seamlessly integrates network structure, node attributes, and community structure into low-dimensional representation space.
(2) A biased random walk is proposed to construct community triplets and then calculate community triplet-loss.
(3) We evaluate and validate our method through three tasks: node classification, node clustering, and visualization. Experimental result demonstrate the effectiveness of our method.

2 Related Work

Some earlier works can be traced back to the graph embedding problem, such as Laplacian Eigenmaps and LPP [13], which utilizes manifold learning to capture structure proximity. Inspired by word2vec [4], Deepwalk [14] generates node sequences by truncating random walks and train skip-gram model to preserve structural proximity. Node2vec [8] introduces a biased random walk to explore network structure flexibly. Line [7] proposes an explicit objective function to preserve first-order proximity and second-order proximity. Deepwalk [14], Node2vec [6], and Line [7] are all based on shallow neural network that cannot preserve the non-linear structure of the network. SDNE [15] employs autoencoder to preserve first-order proximity, second-order proximity, and non-linear structure of the network simultaneously. DNGR [16] introduces a random-surfing model and directly construct positive pointwise mutual information matrix (PPMI) and employs stacked denoising autoencoder to extract feature. GraRep [17] calculates similarity matrices of the different order, factorize these matrices to retain representations of the different order. The above-mentioned works ignore community structure. CARE designs a community-aware random walk to generate node sequence and feed into skip-gram to preserve the local structure and community information. The M-NMF [10] adopts modular non-negative matrix factorization to retain the node's representation which preserves both the community structure and node's local structure simultaneously.

The above NE methods just explore the structure of networks. Thus, they are not suitable for attributed network containing rich semantic information that should be preserved to improve the quality of representations. State-of-the-art ANE models considering both node attributes and network structure have a better performance. TADW proves that deepwalk is equivalent to matrix factorization and incorporates text information into the matrix factorization framework to preserve node attributes. DANE utilizes two autoencoders to extract node attributes and network structure respectively. Attribute representation and structure representation are connected as the final representation. Tri-Dnr [18] incorporates network structure, node content, and node label into a unified framework to learn the representation of the node.

3 Problem Definition

We consider an attributed information network $G = (V, X, A)$,where $V = \{v_1, , , v_n\}$, $X = \{x_1, , , x_n\}$ and $A = \{a_1, , , a_n\}$ represent the node set, set of attribute vectors and set of adjacent vectors respectively. In detail, attribute vector x_i and adjacent vector a_i is associated with the node v_i. In case of unweighted networks, if v_i is connected to v_j, $a_{ij} = 1$, otherwise, $a_{ij} = 0$. In case of weighted networks, if v_i is connected to v_j, a_{ij} reflects how strongly two individual nodes are connected to each other, otherwise, $a_{ij} = 0$. x_i that holds l different attributes and each element x_{ij} represents whether node v_i contains the j-th attribute. We define a function $com(v_i)$. When $com(v_i) = c$, the vertex v_i belongs to community c. Attributed network consist of network structure and node attributes of vital significance. The aim of ANE is to learn the low-dimensional representation of each node, while preserving node attributes and network structure.

4 The Model

In Sect. 4.1, we firstly perform community detection on the whole graph. After community detection, each node in network is assigned to corresponding community. Then, we perform community random walk to construct community triplets. In Sect. 4.2, Deep Attribute Network Embedding (DNE) framework is designed to integrate network structure and attributes and map two information into the unified representations spaces. In Sect. 4.3, we use community triplets to calculate community triplet-loss and form the final model DNEC.

4.1 Construct Community Triplets

Firstly, we adopts infomap [19] to obtain community information. According to community information, we perform community random walk on the whole graph.

Community Detection: Adjacent matric only reflects the local relationship between nodes. Community structure can reveal the hidden relationship between nodes. To obtain Community structure, we use Infomap [19] to get every node's community. Infomap encodes the shortest vertex sequence based on information theory, and detects communities through a deterministic greedy search strategy. In order to obtain the vertex sequence, a random walk strategy is used to collect high-order information. The greedy search strategy integrates information on a global view and integrates communities. Because random walk and greedy search are common strategies for obtaining community information, Infomap is employed as community detection module.

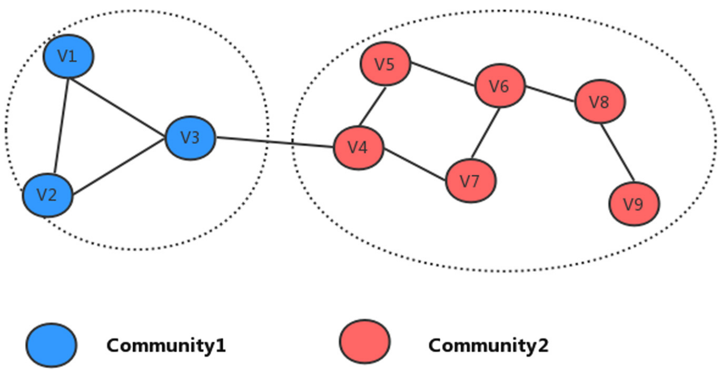

Fig. 1. Simple network with two communities

Community Triplets Construction: On the basis of community detection, we know the community distribution of each node. We perform biased random walk to construct community triplets. For each node a, which belong to community $com(a)$, we randomly select node p belonging to the community $com(a)$. Then, we chose another community which is not equal to the community $com(a)$ and randomly select a node f from this community. we get triplet $< a, p, f >$, where $com(a) = com(p)$ and $com(a) \neq com(f)$. Repeat the above process to construct t triplets for each node in network. For v_1 in Fig. 1, we randomly select a node v_3 from the first community and randomly select a node v_7 from the second community. Then, we obtain a community triple $< v_1, v_3, v_7 >$. Then, we construct t triplets for v_1. We construct community triplets set $ComSet$ for the whole graph. There are $n \times t$ triplets in $ComSet$.

4.2 Framework of DNE

The framework of DNE is as shown in the Fig. 2. DNE is consist of embedding layer, shared encoder and dual decoder. Embedding layer is designed to extract two different representations. Shared encoder is designed to map two representations into unified representations spaces. Finally, dual decoder is designed to reconstruct adjacent and node attributes respectively.

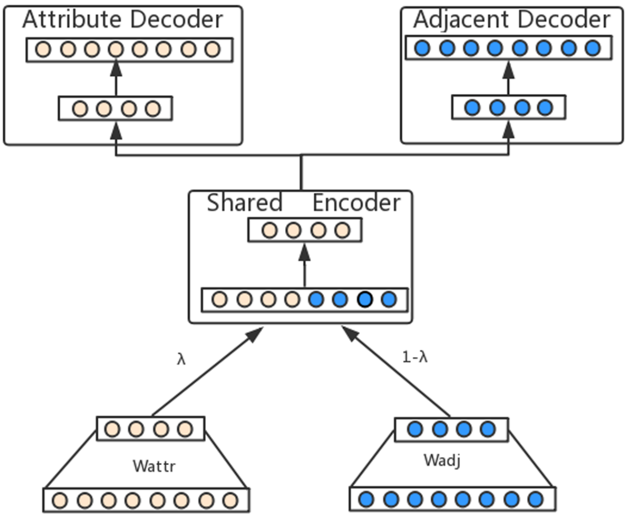

Fig. 2. The framework of the deep model of DNE

Embedding Layer: We design two fully connected layers to extract two different representations and use weight λ and $1-\lambda$ to connect two representations. As shown in Fig. 2, Attribute embedding layer and structure embedding layer compress structural vector and attribute vector into two representations respectively. The structure vector of v_i is a_i and the attribute vector of v_i is a_i. The weights of two layer are W_{attr} and W_{stru} separately. The final output of embedding layer of node v_i is denoted as follows:

$$h_i^{(0)} = [\lambda\sigma(W_{attr} \cdot x_i), (1 - \lambda)\sigma(W_{stru} \cdot a_i)] \tag{1}$$

where W_{attr} and W_{struc} are the weight parameters to be learned. σ is the activation function.

Shared Encoder: To compress attribute and structure into common representation space, we then use a fully connected neural network of multiple layers to map each node into a non-linear latent representation space. The input data of shared encoder is the output of embedding layer $h_i^{(0)}$ and the representation of hidden layers can be denoted as follows:

$$h_i^{(t)} = f(W^t(h_i^{(t-1)}) + b^t) \tag{2}$$

where W^t is the $t-$th hidden layer weight matrix, b^t is the biases and $f(.)$ is the activation function. The final representation of node v_i is represented as $emb(v_i)$.

Dual Decoder: The representations obtained by the shared encoder layer contains both attribute and structure and is the input of dual decoder. Attributed decoder and structure decoder are designed to reconstruct the node attribute and structure separately. Attribute decoder consists of multiple layers. The loss of reconstructing attributes is denoted by the mean square error (MSE) given by

$$Loss_{attr} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})^2 \tag{3}$$

where $\hat{x_i}$ is the output of attribute decoder. Structure decoder consist of multiple layers and directly reconstructs structure vector of node v_i. The structure reconstruct loss is as follows:

$$Loss_{stru} = \frac{1}{n}\sum_{i=1}^{n}(a_i - \hat{a_i})^2 \tag{4}$$

where $\hat{a_i}$ is the output of the structure decoder.

4.3 DNEC

The homogeneity theory indicates that nodes with same community should be closer to each other in low-dimensional space, while nodes belong to different communities should stay away from each other in the representation space. Therefore, we calculate the community triplet-loss on the basis of DNE and form

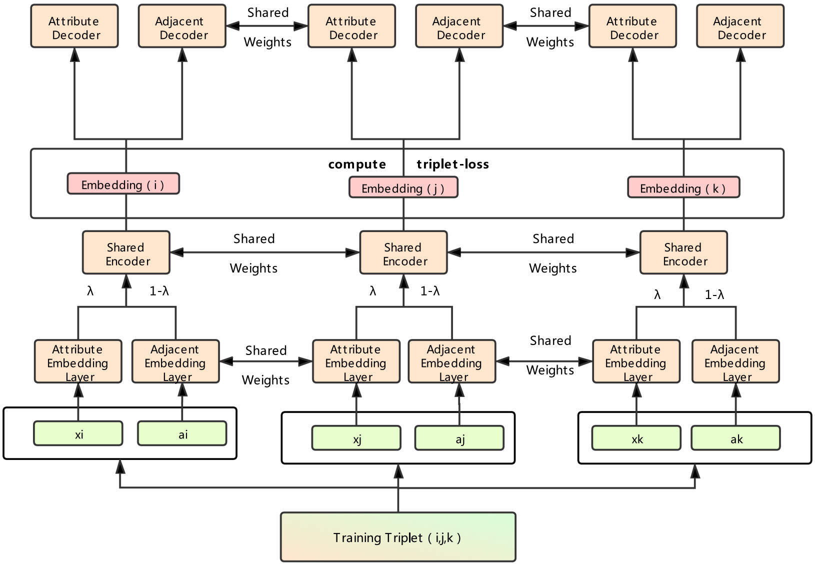

Fig. 3. DNEC: The framework of the deep model of DNE with Triplet-loss

the final model DNEC. Overall, we define the following community triplet-loss function as follows:

$$Loss_{com} = \sum_{<a,p,f>\in ComSet} max(dis(a,p) - dis(a,f) + margin, 0) \tag{5}$$

where $dis(a,p) = (emb(a) - emb(p))^2$, $com(a) = com(p)$, $com(a) \neq com(f)$. In Fig. 3, all triplets in $ComSet$ are used as training sets to train the model. Given a triplet $< a, p, f >$, we get the node's representation $< emb(a), emb(p), emb(f) >$ through embedding layer and shared encoder. Then, we calculate the $Loss_{com}$ of all triplets in $ComSet$. Minimize L_{com}, $dis(a,p)$ becomes smaller and $dis(a,f)$ becomes bigger. Nodes in the same community will be closer to each other and nodes in the different communities will be away from each other in the representation space.

In order to retain node attributes, local structure, and community structure, we designed the DNEC model. Overall, we minimize the following loss function:

$$L_{total} = Loss_{attr} + Loss_{stru} + Loss_{com} \tag{6}$$

5 Experiment

In this part, we conduct experiments on three public datasets such: cora, citeseer, and pubmed. We compare our method with the state-of-art methods. The experimental results prove that our method has significant improvements over baselines. Firstly, we introduce the datasets we used in the experiments, and then simply list the comparison methods, finally, we present the experimental results and discuss the advantages of our method.

5.1 Experimental Settings

Datasets : An overview of the network datasets we consider in our experiments is given in Table 1.

Table 1. Statistics of the datasets

Datasets	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3707	6
PubMed	19717	44338	500	3

The datasets are paper citation networks. The nodes in Table 1 represent papers. The edge of each network is the citation relationship between two papers. The attribute of each node is the bag-of-words representation of the corresponding paper.

Baselines: We use the following five state-of-the-art NE methods as our baselines. All baselines are published recently and all have good performance on NE. The descriptions of the baselines are as follows:

Deepwalk [14]: uses random walk to generate node sequences and feed node sequences into skip-gram to learn node representation vector of the nodes using only structure.

Line [7]: exploits the network structure's first-order proximity and second-order proximity.

Node2vec [6]: uses two parameters to simulate BFS and DFS search strategies to generate node sequences and then preserve global and local proximity by a flexible random-walk way.

TADW [11]: incorporates text into Matrix Factorization and preserve node content and network struct simultaneously.

DANE [12]: adopts two deep neural networks to extract node structure and node attribute separately and connect two different representations as the final representation.

Parameter Settings. For a fair comparison, we set the embedding dimension to 100 for all methods. For Deepwalk and Node2vec, we set the window size t to 10 and walk length l to 80. For Node2vec, we set the BFS parameter q to 2.0 and DFS to 0.5. For LINE, the starting value of learning rate is 0.025. The number of negative samples is set as 5 and the number of training samples are set as 10,000. For TADW, we set the parameters λ to 0.5. For DANE, we set the parameters as shown in the paper. For our method, we set λ to 0.5, margin to 0.5 and t to 5. We summarize the parameter settings of the three datasets in Table 2. For PubMed, the network structure contains more useful information than node attribute. So, we set the dimension of Attribute Embedding layer to 200 and set the dimension of Structure Embedding layer to 800.

Table 2. Parameter settings of the three datasets

Datasets	Attr-Emb	Stru-Emd	Shar-Encoder	Attr-Decoder	Stru-Decoder
Cora	512	512	1024-512-100	100-512-1433	100-512-2708
Citeseer	512	512	1024-512-100	100-512-3707	100-512-3327
PubMed	200	800	1000-500-100	100-200-500	100-800-19717

5.2 Results and Analysis

Node Classification. We conduct node classification on learned node's representation to demonstrate the great performance of on semi-supervised classification task. We applied the Lib-SVM(SVM) software packages as the classifier for all baselines. For a comprehensive assessment, we randomly select$\{10\%, 30\%, 50\%\}$ nodes from the dataset as the training set, and the remaining nodes as the testing set. We adopt the method of five-fold cross-validation to train the SVM classifier with training set and use Micro-F1 (Mi-F1) and Macro-F1 (Ma-F1) as Metrics on the testing set to measure the classification result. The average accuracy of node classification of all methods are

Table 3. Average of Micro-F1 and Macro-F1 scores in Cora dataset

Training percent	10%	30%	50%
Method	Mi-F1 Ma-F1	Mi-F1 Ma-F1	Mi-F1 Ma-F1
Deepwalk	0.7568 0.7498	0.8064 0.7943	0.8287 0.8177
Node2vec	0.7477 0.7256	0.8201 0.8121	0.8235 0.8162
Line	0.7338 0.7191	0.8122 0.8105	0.8353 0.8254
TADW	0.7510 0.7234	0.8006 0.7801	0.8354 0.8187
DANE	0.7867 0.7748	0.8281 0.8127	0.8502 0.8377
DNEC	**0.7979 0.7832**	**0.8384 0.8213**	**0.8697 0.8456**

shown in Tables 3, 4, and 5, where the best results are bold. We find that our method performs better than baselines. Deepwalk, Node2Vec and Line just consider structure. So, the performance of those methods are worsen than TADW.

Because TADW considers attributed information on three datasets. DANE perform better than TADW. Because DANE employs deep neural network to persevere structure information and attributed information. It can be seen from Tables 3, 4, and 5, our method uses a more reasonable method to map structure and node attributes into the unified representation spaces. In the network, nodes with same community tent to have same category. Community triplet-loss will pull nodes with same category cluster in representation spaces. So, DNEC performs better than all baselines.

Table 4. Average of Micro-F1 and Macro-F1 scores in Citeseer dataset

Training percent	10%	30%	50%
Method	Mi-F1 Ma-F1	Mi-F1 Ma-F1	Mi-F1 Ma-F1
Deepwalk	0.5052 0.4645	0.5783 0.5329	0.5900 0.5486
Node2vec	0.5233 0.4832	0.6110 0.5651	0.6335 0.5972
Line	0.5139 0.4726	0.5761 0.5384	0.6075 0.5700
TADW	0.6048 0.5344	0.6481 0.5769	0.6578 0.5897
DANE	0.6444 0.6043	0.7137 0.6718	0.7393 0.6965
DNEC	**0.6534 0.6219**	**0.7248 0.6956**	**0.7524 0.7126**

Table 5. Average of Micro-F1 and Macro-F1 scores in PubMed dataset

Training percent	10%	30%	50%
Method	Mi-F1 Ma-F1	Mi-F1 Ma-F1	Mi-F1 Ma-F1
Deepwalk	0.8047 0.7873	0.8168 0.8034	0.8176 0.8034
Node2vec	0.8027 0.7849	0.8110 0.7965	0.8103 0.7981
Line	0.8037 0.7892	0.8129 0.8007	0.8110 0.7994
TADW	0.8258 0.8143	0.8286 0.8214	0.8343 0.8294
DANE	0.8298 0.8179	0.8311 0.8205	0.8475 0.8349
DNEC	**0.8395 0.8279**	**0.8473 0.8341**	**0.8582 0.8421**

Node Clustering. To prove the performance of our method on node clustering task, we apply K-means on cora dataset. We use the label information as the true community information and use the clustering accuracy to measure the clustering result. The result is shown in Table 6, the clustering accuracy of our method is higher than all baselines. The accuracy of DANE is higher than Deepwalk, Node2vec. Because DANE preserves non-linear structure and attributed

Table 6. Clustering Accuracy

Method	Cora	Citeseer	Pubmed
Deepwalk	0.6813	0.4145	0.6660
Node2vec	0.6473	0.4504	0.6754
Line	0.4789	0.3913	0.6614
TADW	0.5993	0.6642	0.6257
DANE	0.7027	0.4797	0.6942
DNEC	**0.7213**	**0.6942**	**0.7181**

information. DNEC has the best performance for the reason that DNEC incorporates local structure, node attributes and community information. Equipped with triplet-loss, nodes with same community will cluster in low-dimensional space.

Visualization. To further show the embedding result obtained by our method, we apply t-sne to visualize the node's representation in lwo-dimensional space.we conduct t-sne task on citeseer dataset. The result is shown in Fig. 4. The boundary of TADW is not explicit. Because DANE consider node attribute and network structure together and uses non-linear neural network, the boundary of different class is more explicit than TADW. We can see from Fig. 4 that the visualization of our method have clear boundaries and compact cluster. Because triplet-loss makes nodes in same community cluster in representation spaces and make nodes in different communities away from each other.

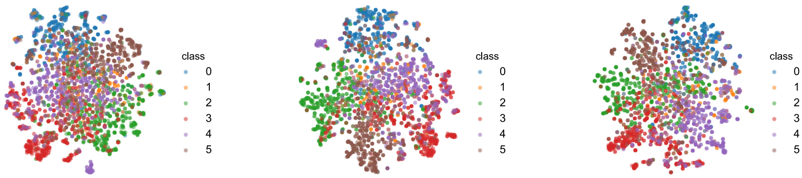

Fig. 4. t-SNE visualization the dataset Citeseer by using TADW, DANE, and our proposed method. The left is the visualization using TADW, and the right is the visualization using the method we proposed, and the median is the visualization using DANE.

6 Conclusion

In this paper, we propose ANE framework, using a more reasonable way to preserve both the network topology and the node attribute. We also take community information into account to improve the quality of representations. The experimental results prove that our method has a great performance on node classification, node clustering, and visualization tasks. Compared to the previous works, we incorporate community information into ANE and obtain a better performance. In future, we will consider the scalability of our method in heterogeneous networks.

References

1. Dong, Y., Zhang, J., Tang, J., Chawla, N.V., Wang, B.: CoupledLP: link prediction in coupled networks. In: 21th International Conference on Knowledge Discovery and Data Mining(SIGKDD 2015), pp. 199–208 (2015)
2. Tang, J., Liu, J., Zhang, M., Mei, Q.: Visualizing large-scale and high-dimensional data. In: 25th International Conference on World Wide Web(WWW 2016), pp. 287–297 (2016)
3. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. Commun. Surv. Tutorials **2014**, 303–336 (2014)
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In:1st International Conference on Learning Representations (ICLR 2013), pp. 1–12 (2013)
5. Tomas, M., Ilya, S., Kai, C., Greg, S., Je, D.: Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111–3119 (2013)
6. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In:22nd International Conference on Knowledge Discovery and Data Mining (KDD 2016), pp. 855–864. ACM (2016)
7. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: 24th International Conference on World Wide Web (WWW 2015), pp. 1067–1077 (2015)
8. Keikha, M.M., Rahgozar, M., Asadpour, M.: Community aware random walk for network embedding. Knowl. Based Syst. **148**, 47–54 (2018)
9. Yang, C., Lu, H., Chen, K.: CONE: Community Oriented Network. http://arxiv.org/abs/1709.01554
10. Yang, L., Cao, X., Wang, C., Zhang, W.: Modularity based community detection with deep learning. In: 25th International Joint Conference on Artificial Intelligence (AAAI 2016), pp. 2252–2258 (2016)
11. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: 24th International Conference on Artifificial Intelligence (IJCAI 2015), pp. 2111–2117 (2015)
12. Hongchang, G., Heng, H.: Deep Attributed network embedding. In: 27th International Joint Conference on Artificial Intelligence (IJCAI 2018), pp. 3364–3370 (2018)
13. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and spectral techniques for embedding and clustering. In: Advances Neural Information Processing Systems, pp. 585–591 (2001)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: 20th International Conference on Knowledge Discovery and Data Mining (KDD 2014), pp. 701–710. ACM (2014)
15. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: 22nd SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2016), pp. 1225–1234. ACM (2016)

16. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: 30th AAAI Conference on Artificial Intelligence (AAAI 2016), pp. 1145–1152 (2016)
17. Cao S., Lu W., Xu Q.: GraRep: learning graph representations with global structural. In 24th International Conference on Information and Knowledge Management (CIKM 2015), pp. 891–900. ACM (2015)
18. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), pp. 1895–1901 (2016)
19. Rosvall, M., Bergstrom, C.: Maps of random walks on complex networks reveal community structure. In: Proceedings of the National Academy of Sciences of the United States of America, vol. 105(4), pp. 1118–1123 (2018)