A Dual Fusion Model for Attributed Network Embedding

Kunjie Dong[1], Lihua Zhou[1(✉)], Bing Kong[1], and Junhua Zhou[2]

[1] School of Information, Yunnan University, Kunming 650091, China
kunjiedong@qq.com, {lhzhou,Bingkong}@ynu.edu.cn
[2] School of Public Administration, Yunnan University, Kunming 650504, China
ynuzjh@163.com

Abstract. Attributed network embedding (ANE) maps nodes in network into the low-dimensional space while preserving proximities of both node attributes and network topology. Existing methods for ANE integrated node attributes and network topology by three fusion strategies: the early fusion (EF), the synchronous fusion (SF) and the late fusion (LF). In fact, different fusion strategies have their own advantages and disadvantages. In this paper, we develop a dual fusion model named as DFANE. DFANE integrated the EF and the LF into a united framework, where the EF captures the latent complementarity and the LF extracts the distinctive information from node attributes and network topology. Extensive experiments on eight real-world networks have demonstrated the effectiveness and rationality of the DFANE.

Keywords: Network analysis · Attributed network embedding · Fusion strategy · Auto-encoder

1 Introduction

With more and more information becoming available, nodes in real-world networks are often associated with attributed features, for example, papers in academic citation networks generally have a published conference, author, research topic and keywords, which are known as attributed networks [4]. Recently, ANE, aiming to learn the low-dimensional latent representations of nodes which can well preserve the proximities based node attributes and network topology at the same time, has attracted lots of researchers' interests.

It plays an important role in integrating the node attributes and network topology for ANE, because ANE focuses on capturing latent relationships in

Supported by organization of the National Natural Science Foundation of China (61762090, 61262069, 61966036 and 61662086), The Natural Science Foundation of Yunnan Province (2016FA026), the Project of Innovative Research Team of Yunnan Province (2018HC019), Program for Innovation Research Team (in Science and Technology) in University of Yunnan Province (IRTSTYN), and the National Social Science Foundation of China under Grant No. 18XZZ005.

G. Li et al. (Eds.): KSEM 2020, LNAI 12274, pp. 86–94, 2020.
https://doi.org/10.1007/978-3-030-55130-8_8

terms of node attributes and network topology. Existing ANE approaches, based on the different fusion styles, can be divided into three categories: the EF embedding models [2,3], the SF embedding models [1,8,9], and the LF embedding models [6]. The EF and SF allow attributes modeling and topology modeling closely interact each other, but they cannot guarantee the individual distinctive characteristic; the LF trains individual models separately without knowing each other and results are simply combined after training, thus it can guarantee individual characteristic, but it may lose the consistency information of nodes [1].

To fully excavate the relationships and intrinsic essences between node attributes and network topology, we propose a dual fusion model named DFANE for ANE in this paper. DFANE consists of two components, the EF component and the LF component, where the former first concatenates the node attributes and network topology into a united vector on the input layer, and then conducts collaborative training to capture the relationships between node attributes and network topology; and the latter, including the node attributes Auto-Encoder and the network topology Auto-Encoder, first to capture the individual inherent essences from node attributes and network topology without interacting each other, and then achieves information integration via concatenating two types of individual inherent essences obtained by the two independent Auto-Encoders.

To summarize, our main contributions are as follows:

(i) A dual fusion model with the EF and LF components is proposed. The EF component extracts the latent interrelationship, the LF component captures the peculiarity of attributes and topology, and then the unity of two components preserve the consistency and complementarity information.
(ii) We conduct abundant experiments on eight datasets by the tasks of node classification and node clustering. The experimental results demonstrate the effectiveness and rationality of the DFANE.

2 Related Work

In this section, we briefly summarize the development of ANE methods in items of the strategies of fusing node attributes and network topology.

NANE [3] designed a self-feedforward layer to capture weight features in node attributes and network topology, which employed the pairwise constraint based the input data and weights to preserve the local information and the global information respectively. NetVAE [2] adopted a shared encoder to perform co-training and introduced the attributed decoder and networked decoder to reconstruct node attributes and network topology.

DANE [1] captured the latent highly non-linearity in both node attributes and network topology by two symmetrical Auto-Encoders, in which considered the consistency and complementary. ANRL [8] fed the node attributes into the encoder and the decoder reconstructed node's target neighbors instead of node itself under the guidance of network topology. The partial correlation, i.e. nodes with similar attributes may be dissimilar in topology and vice versa, was proposed in PRRE [9], and EM algorithm was utilized to tune two thresholds to define node relations: positive, ambiguous, negative.

LINE [6] considered different roles of each node in network, i.e. node itself and "context" of the other nodes in network, the node's embedding representations were learned by two separate models for preserving the first-order and second-order proximity of network topology respectively. The final representation for each node was obtained by concatenating the distinctive representations learned from two separate models. However, LINE only taken the network topology into account.

3 The Proposed Model

In this section, we first to present the definition of ANE and then develop a dual fusion ANE model.

3.1 Problem Definition

Let $G = (V, E, \boldsymbol{X})$ be an attributed network with n nodes, where V represents the set of nodes, E represents the set of edges, and $\boldsymbol{X} \in R^{n\times l}$ represents the attributes matrix in which the row-vector $\boldsymbol{x}_i \in R^l$ corresponds to the attributes vector of the node v_i. Besides, the adjacency matrix $\boldsymbol{M} \in R^{n\times n}$ represents the link relationship between nodes, in which the element $m_{ij} > 0$ represents there existing the edge between the nodes v_i and v_j, while $m_{ij} = 0$ represents the edge is absent.

It is necessary to preserve the proximities of both node attributes and network topology in ANE. Let $\boldsymbol{A} \in R^{n\times n}$ be the attributes similarity matrix, and the element $a_{ij} \in \boldsymbol{A}^{n\times n}$ can be measured by the functions of distance similarity of attributes vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of nodes v_i and v_j, such as Cosine similarity, Euclidean distance. The Cosine similarity can be calculated by $a_{ij} = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{||\boldsymbol{x}_i|| \times ||\boldsymbol{x}_j||}$, where the operator "$\cdot$" represents the dot product, "$\times$" represents the scalar multiplication, "$||\boldsymbol{x}_*||$" represents the L2-norm of the vector.

Based on attributes similarity matrix $\boldsymbol{A}$, the semantic proximity, reflecting the attributes homogeneity effect amongst nodes, can be computed. For example, the semantic proximity $b_{ij} \in \boldsymbol{B}^{n\times n}$ between the nodes v_i and v_j can be calculated by $b_{ij} = \frac{\boldsymbol{a}_i \cdot \boldsymbol{a}_j}{||\boldsymbol{a}_i|| \times ||\boldsymbol{a}_j||}$, where $\boldsymbol{a}_i = (a_{i1}, \cdots, a_{in}) \in \boldsymbol{A}$ and $\boldsymbol{a}_j = (a_{j1}, \cdots, a_{jn}) \in \boldsymbol{A}$.

The proximities related to the topology of network include the first-order proximity which corresponds to the direct neighbor relationships and the high-order proximity which corresponds to multi-hops neighbor relationships connected by the shared neighbor. The first-order proximity between nodes v_i and v_j can be measured by the value of $m_{ij} \in \boldsymbol{M}$. Specifically, the larger value of m_{ij} indicates the stronger proximity between two nodes. Let $\hat{\boldsymbol{M}}^1$ is the 1-step probability transition matrix amongst nodes, which can be obtained by the row-wise normalization of $\boldsymbol{M}$; $\hat{\boldsymbol{M}}^t$ be the t-step probability transition matrix, which can be computed by $\hat{\boldsymbol{M}}^1$: $\hat{\boldsymbol{M}}^t = \underbrace{\hat{\boldsymbol{M}}^1 \cdots \hat{\boldsymbol{M}}^1}_{t}$; the neighborhood proximity matrix $\boldsymbol{S} = \hat{\boldsymbol{M}}^1 + \hat{\boldsymbol{M}}^2 + \cdots + \hat{\boldsymbol{M}}^t$, then the high-order proximity between nodes v_i and v_j can be measured by the similarity of vectors $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

ANE aims to find a map function $f(\boldsymbol{a}_i, \boldsymbol{s}_i) \rightarrow \boldsymbol{h}_i$ that maps node vectors $\boldsymbol{a}_i$ and $\boldsymbol{s}_i$ into a unified embedding representation $\boldsymbol{h}_i \in R^d$, such that the node semantic proximities and network topological proximities can be captured and preserved, d is the dimension of the embedding representation and $d \ll l$.

3.2 The Architecture of DFANE

The proposed DFANE model consists of the EF component and the LF component. They fuse node attributes and network topology at different stages to implement ANE. The architecture of DFANE is displayed in Fig. 1.

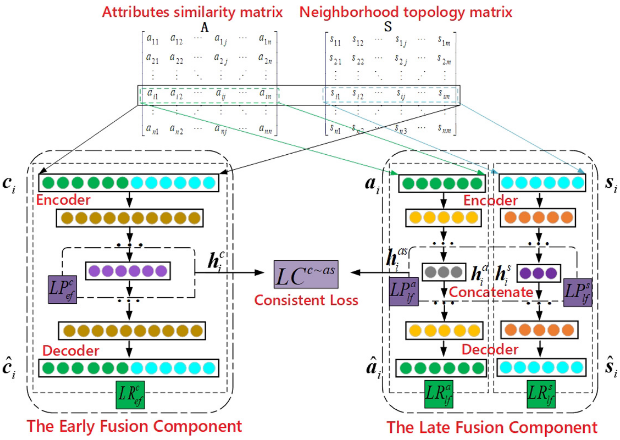

Fig. 1. The architecture of DFANE.

The Early Fusion Component. The EF component is implemented by a deep Auto-Encoder. Let $\boldsymbol{C} = [\boldsymbol{A}, \boldsymbol{S}] \in R^{n \times 2n}$ be a matrix formed by concatenating the attributes similarity matrix $\boldsymbol{A} \in R^{n \times n}$ and the neighborhood topology matrix $\boldsymbol{S} \in R^{n \times n}$, i.e. $\boldsymbol{c}_i = (\boldsymbol{a}_i, \boldsymbol{s}_i) = (a_{i1}, \cdots, a_{in}, s_{i1}, \cdots, s_{in})$, where $\boldsymbol{c}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{s}_i$ are the i-th row vector of $\boldsymbol{C}$, $\boldsymbol{A}$ and $\boldsymbol{S}$ respectively. Let the deep Auto-Encoder have $2T - 1$ layers, where the layers $1, 2, \cdots, T$ for the encoder, the layers $T, T+1, \cdots, 2T-1$ for the decoder, the layer T be shared for the encoder and decoder; the vectors $\boldsymbol{h}^c_{i,t}(t = 1, \cdots, T)$ and $\boldsymbol{y}^c_{i,t}(t = 1, \cdots, T)$ be the hidden representations of the node v_i at t-th layer of the encoder and decoder respectively, $\boldsymbol{h}^c_i = \boldsymbol{h}^c_{i,T} \in R^{2d}$ be the desired underlying compact representation of the node v_i, $\hat{\boldsymbol{c}}_i = \boldsymbol{y}^c_{i,T} \in R^{2n}$ be the reconstructed data point from the decoder.

Proximity Loss. To preserve the first-order proximity in the concatenate matrix $\boldsymbol{C}$, the negative log-likelihood $LP_{ef}^{c} = -\sum_{m_{ij}>0} \log p_{ij}^{c}$ should be minimized, where $p_{ij}^{c} = \frac{1}{(1+\exp(\boldsymbol{h}_i^c \cdot (\boldsymbol{h}_j^c)^T))}$ is the joint probability between the $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$.

Reconstruction Loss. To preserve the highly non-linear relationship existed in the concatenate matrix $\boldsymbol{C}$, the reconstruction loss between the input $\boldsymbol{c}_i$ and output $\hat{\boldsymbol{c}}_i$, i.e. $LR_{ef}^{c} = \sum_{i=0}^{n} ||\hat{\boldsymbol{c}}_i - \boldsymbol{c}_i||_2^2$, should be minimized to train the Auto-encoder.

The Late Fusion Component. The LF complement consists of two symmetrical Auto-Encoders: the node attributes Auto-Encoder and the network topology Auto-Encoder. Both of them have the same layers architecture with the Auto-Encoder of the EF complement. Let $\boldsymbol{h}_i^a = \boldsymbol{h}_{i,T}^a \in R^d$ and $\boldsymbol{h}_i^s = \boldsymbol{h}_{i,T}^s \in R^d$ be the desired underlying compact representations, $\hat{\boldsymbol{a}}_i = \boldsymbol{y}_{i,T}^a \in R^d$ and $\hat{\boldsymbol{s}}_i = \boldsymbol{y}_{i,T}^s \in R^d$ be the reconstructed representations of the node v_i with respect to the node attributes Auto-Encoder and the network topology Auto-Encoder. The two Auto-Encoders are trained independently, and the desired representation of a node is the concatenation of two representations obtained from the node attributes Auto-Encoder and the network topology Auto-Encoder, i.e. $\boldsymbol{h}_i^{as} = \boldsymbol{h}_i^a \oplus \boldsymbol{h}_i^s = (h_{i1}^a, \cdots, h_{id}^a, h_{i1}^s, \cdots, h_{id}^s) \in R^{2d}$, $\hat{c}_i^{as} = \hat{\boldsymbol{a}}_i \oplus \hat{\boldsymbol{s}}_i = (\hat{a}_{i1}, \cdots, \hat{a}_{in}, \hat{s}_{i1}, \cdots, \hat{s}_{in}) \in R^{2n}$.

Proximity Loss. To preserve the first-order proximity of vectors $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ in the attributes similarity matrix $\boldsymbol{A}$, the negative log-likelihood $LP_{lf}^{a} = -\sum_{m_{ij}>0} \log p_{ij}^{a}$ should be minimized, where $p_{ij}^{a} = \frac{1}{(1+\exp(\boldsymbol{h}_i^a \cdot (\boldsymbol{h}_j^a)^T))}$ is the joint probability between the $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. Similarly, the negative log-likelihood $LP_{lf}^{s} = -\sum_{m_{ij}>0} \log p_{ij}^{s}$ should be minimized to preserve the first-order proximity of vectors $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, where $p_{ij}^{s} = \frac{1}{(1+\exp(\boldsymbol{h}_i^s \cdot (\boldsymbol{h}_j^s)^T))}$ is the joint probability between the $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

Reconstruction Loss. To preserve the semantic proximity associated with node attributes, the reconstruction loss between node attributes encoder's input $\boldsymbol{a}_i$ and decoder's output $\hat{\boldsymbol{a}}_i$, i.e. $LR_{lf}^{a} = \sum_{i=1}^{n} ||\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i||_2^2$, should be minimized. Similarly, the reconstruction loss between network topology encoder's input $\boldsymbol{s}_i$ and decoder's output $\hat{\boldsymbol{s}}_i$, i.e. $LR_{lf}^{s} = \sum_{i=1}^{n} ||\hat{\boldsymbol{s}}_i - \boldsymbol{s}_i||_2^2$, should be minimized.

In DFANE, we propose a consistency loss function to measure the consistency of the two components, shown as $LC^{c\tilde{\ }as} = \sum_{i=1}^{n} ||\boldsymbol{h}_i^c - \boldsymbol{h}_i^{as}||_2^2 + \sum_{i=1}^{n} ||\hat{\boldsymbol{c}}_i - \hat{\boldsymbol{c}}_i^{as}||_2^2$, where the first item indicates the embedding consistency loss and the second item means the reconstruction consistency loss of two components.

3.3 The Total Loss of DFANE

To sum up, the total loss of the DFANE is defined as the $LT = \alpha LP + \beta LR + \gamma LC^{c\tilde{\ }as}$, where the proximity loss $LP = LP^{c}_{ef} + LP^{a}_{lf} + LP^{s}_{lf}$ and the reconstruction loss $LR = LR^{c}_{ef} + LR^{a}_{lf} + LR^{s}_{lf}$ are the sum of the proximity loss and the reconstruction loss in the concatenate matrix $\boldsymbol{C}$, attributes similarity matrix $\boldsymbol{A}$ and neighborhood topology matrix $\boldsymbol{S}$, respectively. α, β and γ are hyper-parameters used to balance the weights among different losses.

4 Experiment

In this section, we conduct experiments on eight publicly network datasets to evaluate the performance of the DFANE, compare with the several state-of-the-art methods, by the tasks of node classification and node clustering.

4.1 Datasets and Baselines Method

In experiments, three types of real-world networks are used, i.e. WebKB[1] network, Social network and Bibliographic network, where WebKB network consists of Texas, Cornell (corn), Washington (Wash), Wisconsin (Wisc) datasets; Social network contains Hamilton[2] (Hami) and Wiki datasets; Bibliographic network includes Cora and Pubmed[3] (Pubm) datasets.

We select NANE [3], DANE [1], ANRL [8], PRRE [9] as the baselines. NANE adopted the EF strategy to implement ANE; other baselines all employed the SF strategy to implement ANE. Besides, we develop two variants of DFANE, i.e. DFANE-E and DFANE-L, that only including the single complement.

4.2 Parameter Setting

The number of neurons in each layer is arranged in Table 1, which is a template for all baselines in experiments. The first layer of input of encoder and the last layer of output of decoder for node attributes and network topology correspond to the dimension of node attributes and the number of nodes in network.

DFANE contains three hyper-parameters α, β and γ for balancing the proximity loss, reconstruction loss and consistency loss. The value of hyper-parameters tuned through the algorithm of grid search and listed in Table 1. In experiments, both node classification and node clustering use the same parameters. The parameters of the baselines were set the same as the original papers.

1 https://linqs-data.soe.ucsc.edu/public/lbc/.
2 https://escience.rpi.edu/data/DA/fb100/.
3 https://linqs.soe.ucsc.edu/data.

Table 1. The architecture of neural networks and the hyper-parameters of DFANE.

Datasets	Methods number of neurons in each layer		Hyper-parameters		
	Node attributes	Network topology	α	β	γ
Texas	1703-200-100-200-1703	187-200-100-200-187	0.1	1	0.001
Corn	1703-200-100-200-1703	195-200-100-200-195	0.001	100	0.01
Wash	1703-200-100-200-1703	230-200-100-200-230	1	1000	0.001
Wisc	1703-200-100-200-1703	265-200-100-200-230	10	1000	200
Hami	144-200-100-200-144	2314-200-100-200-2314	100	100	200
Wiki	4973-256-128-256-4973	2405-256-128-256-2405	1	1000	10
Cora	1433-256-128-256-1433	2708-256-128-256-2708	10	100	500
Pubm	500-256-128-256-500	19717-256-128-256-19717	0.01	50	0.01

4.3 Results and Analysis

Node classification is carried out on the learned node representations, and L2-regularized Logistic Regression [5] is used as the classifier. Then {10%, 20%, 30%, 40%, 50%} labeled nodes are randomly selected as the training set and the remained nodes as the testing set. For node clustering, we adopt K-means algorithm as the clustering method. In experiments, these processes are repeated ten times, and the average performances with respect to the Micro-F1, Macro-F1, AC and NMI [7] are reported for each dataset. Here due to the space constrains we only list the node classification results selected 40% labeled nodes as training set and node clustering results for eight datasets in Table 2, where bold numbers represent the best results; the trend of node classification results with respect to the training rates of {10%, 20%, 30%, 50%} are similar to the 40%.

Table 2. Node classification performance of different methods on eight datasets with training rate is 40% in Micro-F1 and Macro-F1, and node clustering performance of different methods on eight datasets in AC and NMI.

Metrics	Methods	Texas	Corn	Wash	Hami	Wisc	Wiki	Cora	Pubm
Micro-F1	NANE	53.98	44.44	46.38	86.68	48.43	49.41	50.46	57.51
	ANRL	65.49	41.03	65.94	79.99	57.23	72.14	40.00	81.31
	PRRE	72.66	64.27	77.03	93.94	79.56	73.20	82.61	84.3
	DANE	79.38	56.41	73.98	94.09	76.04	78.62	83.93	87.86
	DFANE-L	79.12	57.18	73.98	93.78	77.23	79.07	83.39	87.6
	DFANE-E	82.3	61.54	79.71	**94.26**	76.54	79.51	81.75	85.44
	DFANE	**83.54**	**70.34**	**81.6**	93.97	**81.76**	**79.85**	**84.18**	**88.44**

(*continued*)

Table 2. (*continued*)

Metrics	Methods	Texas	Corn	Wash	Hami	Wisc	Wiki	Cora	Pubm
Macro-F1	NANE	14.02	12.31	12.67	13.05	25.83	32.03	46.71	52.38
	ANRL	28.73	17.56	30.24	26.2	14.81	55.03	16.8	81.88
	PRRE	52.09	43.22	47.57	58.11	**36.92**	56.30	81.56	83.99
	DANE	53.73	41.35	50.48	54.75	30.58	70.23	82.46	87.6
	DFANE-L	53.46	44.26	48.93	50.57	30.44	70.02	81.84	87.42
	DFANE-E	57.34	46.59	58.56	52.41	30.7	70.15	80.14	85.11
	DFANE	**59.53**	**53.71**	**59.1**	**59.08**	30.52	**70.86**	**82.72**	**88.17**
AC	NANE	5.92	3.24	3.06	10.27	2.54	9.43	6.59	2.68
	ANRL	52.14	40.2	56.09	31.49	39.47	44.56	30.45	63.58
	PRRE	**53.53**	38.41	**60.32**	34.43	50.49	46.06	68.86	64.42
	DANE	38.5	39.08	39.09	**36.85**	32.3	46.32	64.8	64.14
	DFANE-L	35.56	31.62	39.35	36.83	32.19	44.97	66.11	66.20
	DFANE-E	50.91	47.03	48.48	35.69	49.24	34.47	54.22	43.23
	DFANE	50.37	**51.28**	54.22	36.24	**60.86**	**48.07**	**72.66**	**70.8**
NMI	NANE	5.2	4.32	3.26	16.12	4.25	16.13	10.23	3.7
	ANRL	15.19	19.32	20.97	12.08	12.16	44.14	16.71	25.39
	PRRE	24.78	19.87	**34.32**	34.7	27.93	43.52	48.99	28.02
	DANE	8.56	8.94	10.48	37.02	5.98	47.48	50.49	29.09
	DFANE-L	7.58	4.59	11.06	37.39	5.27	46.36	50.55	28.32
	DFANE-E	22.4	13.39	20.15	33.71	23.9	32.39	38.86	13.32
	DFANE	**25.1**	**21.78**	30.33	**38.41**	**38.88**	**47.86**	**55.93**	**34.67**

From Table 2, we have the observations that DFANE achieves the best performance for most case than baselines. Specifically, DFANE makes the best classification results on 7 of the 8 datasets in Micro-F1 and Macro-F1, and also performs the best clustering results on 5 of the 8 networks in AC, and 7 of 8 networks in NMI, which further verifies the efficiency of the dual fusion strategies outperforms that of the EF strategy (NANE, DAFNE-E), SF strategy (ANRL, PRRE, DANE), and LF strategy (DANE-L).

5 Conclusion

Integrating heterogeneous information of node attributes and network topology is essential for ANE. In this study, we propose a dual fusion model DFANE for ANE. DFANE integrated the EF and the LF into a united framework, and the unity of the EF and the LF captures the consensus of heterogeneous information. This is the first attempt to adopt dual fusion strategies in a united framework. Furthermore, experiment results on the eight real-world networks, with the tasks of node classification and node clustering, have demonstrated the effectiveness of the DFANE.

References

1. Gao, H., Huang, H.: Deep attributed network embedding. In: Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI 2018, pp. 3364–3370 (2018)
2. Jin, D., Li, B., Jiao, P., He, D., Zhang, W.: Network-specific variational auto-encoder for embedding in attribute networks. In: Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI 2019, pp. 2663–2669 (2019)
3. Mo, J., Gao, N., Zhou, Y., Pei, Y., Wang, J.: NANE: attributed network embedding with local and global information. In: Hacid, H., Cellary, W., Wang, H., Paik, H.-Y., Zhou, R. (eds.) WISE 2018. LNCS, vol. 11233, pp. 247–261. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02922-7_17
4. Pfeiffer III, J.J., Moreno, S., La Fond, T., Neville, J., Gallagher, B.: Attributed graph models: modeling network structure with correlated attributes. In: Proceedings of the 23rd international conference on World wide web, pp. 831–842 (2014)
5. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 385–394 (2017)
6. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp. 1067–1077 (2015)
7. Yang, Y., Chen, H., Shao, J.: Triplet enhanced autoencoder: model-free discriminative network embedding. In: Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI 2019, pp. 5363–5369 (2019)
8. Zhang, Z., Yang, H., Bu, J., Zhou, S., Wang, C.: Anrl: attributed network representation learning via deep neural networks. In: Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI 2018, pp. 3155–3161 (2018)
9. Zhou, S., Yang, H., Wang, X., Bu, J., Wang, C.: Prre: personalized relation ranking embedding for attributed networks. In: the 27th ACM International Conference, pp. 823–832 (2018)