PCANE: Preserving Context Attributes for Network Embedding

Danhao Zhu[1,2], Xin-yu Dai[1(✉)], Kaijia Yang[1], Jiajun Chen[1], and Yong He[2]

[1] Nanjing University, Nanjing 210031, Jiangsu, People's Republic of China
{zhudh,yangkj}@nlp.nju.edu.cn, {daixinyu,chenjj}@nju.edu.cn
[2] Jiangsu Police Institute, Nanjing 210093, Jiangsu, People's Republic of China
{zhudanhao,heyong}@jspi.cn

Abstract. Through mapping network nodes into low-dimensional vectors, network embedding methods have shown promising results for many downstream tasks, such as link prediction and node classification. Recently, attributed network embedding obtained progress on the network associated with node attributes. However, it is insufficient to ignore the attributes of the context nodes, which are also helpful for node proximity. In this paper, we propose a new attributed network embedding method named PCANE (Preserving Context Attributes for Network Embedding). PCANE preserves both network structure and the context attributes by optimizing new object functions, and further produces more informative node representations. PCANE++ is also proposed to represent the isolated nodes, and is better to represent high degree nodes. Experiments on 3 real-world attributed networks show that our methods outperform the other network embedding methods on link prediction and node classification tasks.

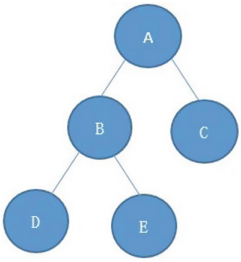
1 Introduction

Recently, some plain network (with only links and nodes) embedding methods were proposed and achieved substantial improvements, such as DeepWalk [12] and Node2Vec [1]. These methods first sampled a number of node sequences from the network based on random walk, which presented structural regularities of the network. Then they tried to preserve the context nodes to the source nodes in the sequences. Besides links, the rich information associated to the nodes can also help to produce more informative network embeddings, including attribute [6], label [10] and so on [16,17]. These auxiliary information can be considered as some types of attributes.

However, the existing attributed network embedding methods [6,10,17] can only utilize the attributes of the source node, so called source attributes, but ignored the attributes of the context nodes, named context attributes. Figure 1 shows some examples of a citation network. With the title attributes of A along, we only acquire that A is about feature learning of network. If the attributes of B, C, D and E are used, we can reasonable infer A may be related to random walk and social networks, and it's true. Note that a more recently research SEANO [5]

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11441, pp. 156–168, 2019.
https://doi.org/10.1007/978-3-030-16142-2_13

can model the attributes of the adjacent nodes, such as B and C. However, they are not able to utilize the attributes of other context nodes, such as D and E.



Node	Attributes(Title words)
A	node2vec: Scalable Feature Learning for Networks
B	Friends and neighbors on the web
C	Supervised random walks: predicting and recommending links in social networks
D	Link prediction in relational data
E	Maximum Consistency Preferential Random Walks

Fig. 1. Samples of a citation network. The citation relationship and the paper are regarded as the link and the node respectively. The title words are the attributes of the node.

In this paper, we propose two models called PCANE (Preserving Context Attributes for Network Embedding) and PCANE++ to learn node representations of attributed network. We first apply random walk based strategy to generate context nodes for the source node, which contain the high order structure information of the network to the source node. Second, we propose PCANE, which has two objective functions to preserve the context nodes and the context attributes respectively. Finally, some networks may contain isolated nodes that have only attributes but no links. These nodes will not be trained by PCANE. We therefore propose PCANE++, which directly encodes attributes to the source vectors. PCANE++ can thus cope with isolated nodes and enhance the effect of the source attributes.

In summary, the contributions of this paper are as follows.

- We propose PCANE and PCANE++ for attributed network embedding. The ability of preserving context attributes can help to produce better node embeddings.
- We conduct extensive experiments on 3 open datasets with two tasks of link prediction and node classification. Empirical results demonstrate the effectiveness and rationality of PCANE and PCANE++.

The rest of the paper is organized as follows. Section 2 discusses related works and our motivation. Section 3 introduces problem definition. In Sect. 4, we present our methods for attribute network embedding. The experiments and analysis are outlined in Sect. 5.

2 Related Works and Our Motivation

2.1 Related Work

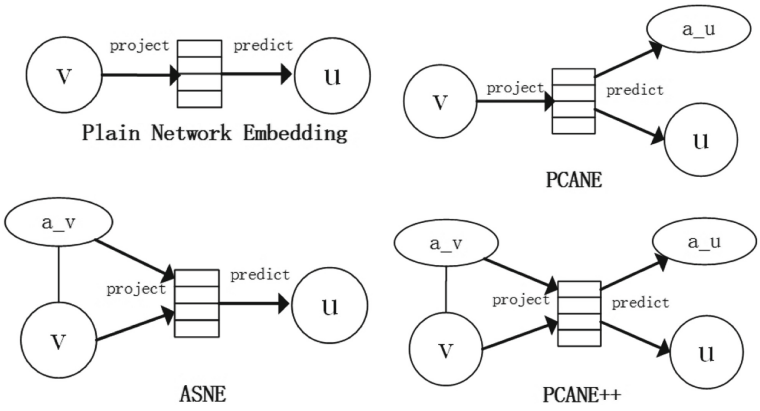
One of the most fundamental problems in network analysis is network embedding, that focuses on embedding a network into a low-dimensional vector

space. The plain network was investigated first. DeepWalk [12] used local node sequences obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences. Node2Vec [1] refined the way to generate node sequence by balancing breadth-first sampling and depth-first sampling. Line [13] designed new object functions by preserving local and global structures. The method is able to scale for real world information networks which usually contain millions of nodes.

Recently, some researchers found attributes of nodes can help to produce more informative representations. SEANO [5] is designed for partially labeled attributed network. They encoded the attributes of the source node and its adjacent neighbor nodes, and then jointly decoded the source label and the neighbor nodes. The most related work to our method is ASNE [6]. In their work, each node was mapped to two vectors, an ID vector for encoding the structure information and an attribute vector for encoding its attributes. Then the two vectors were jointed optimized to maximize the likelihood of preserving neighborhood nodes. However, SEANO [5] cannot utilize high-order context attributes and ASNE [6] cannot use any context attributes at all.

2.2 Our Motivation

Most plain network embedding methods aim to preserve the structure of the original network, by maximizing the likelihood of the context nodes to their source node, as in the top of Fig. 2(a). Afterwards, some attributed network embedding methods (such as ASNE [6] followed the objective, but integrated the source attributes when projecting the source vector, as in the bottom of Fig. 2(a).



(a) Existing methods. (b) Our methods.

Fig. 2. An illustration of the existing network embedding methods and our methods. v and u denotes a source node and one of its context nodes respectively. a_v and a_u denote the attributes of v and u respectively.

We can refer to the underlying idea of network embedding: if the high-dimensional information of the original network can be recovered from the learned embeddings, then the embeddings should contain all necessary network information for downstream tasks. For a plain network, preserving the context nodes indeed aims to recover the node topology structure. ASNE still follows the objective when embedding attributed network. Hence, no matter how strong the learning algorithm is, the global attributes will be forever lost since it is not able to recover any attribute information from the vectors. As in Fig. 2(b), PCANE and PCANE++ try to predict both the context nodes and their attributes. The additional objective to predict the context attributes can be considered as applying some extra constrains on v's embedding. As a result, our methods are able to learn higher quality vectors with richer information.

3 Problem Definition

We formally define the attributed network in Definition 1.

Definition 1 (Attributed Network). *An attributed network is defined as $G = (V, E, T, A)$, where V is the set of nodes, each corresponding to a data object; E is the set of links between the nodes, each corresponding to a relationship between two nodes; T is the attribute type set; A is the set of all discrete attribute values; Each attribute value $a \in A$ is corresponding to a type $T(a) \in T$. Each node $v \in V$ is associated with several attribute values $A(v)$. Each $e \in E$ is an ordered pair $e = (u, v)$ where $u, v \in V$ and is associated with a weight $w_{uv} > 0$, which denotes the strength of the edge.*

The notation differs from [6]'s definition by adding type T, which will be used to preserve the context attributes. PCANE and PCANE++ can be applied to any (un)directed, (un)weighted network. For undirected network, for $e = (u, v) \in E$, there must be an $e' = (v, u) \in E$. For unweighted network, every w_{uv} is constant 1.

We define the problem of attributed network embedding as follows.

Definition 2 (Attributed Network Embedding). *Given an attributed network $G = (V, E, T, A)$, the problem of attributed network embedding is to embed each $v \in V$ to a low-dimensional vector space R^d, where $d << |V|$. In the mapping, the network structure and network attributes are preserved.*

4 Proposed Method

In this section, we first introduce the PCANE model which is able to preserve the network structure and context attributes. Second, we describe the PCANE++ model, an modification to the PCANE model, which can cope with isolated nodes and enhance the importance of source attributes.

4.1 The PCANE Model

Structure Modeling. First, we apply random walks to the network V to obtain truncated node sequences. The node sequences contain the structure information of the network. In the node sequences, for a source node $v \in V$, its neighborhood nodes within certain steps are regarded as the context nodes of v, denoted as $N(v) \subset V$. We use simple uniform sampling strategy, the same as Deepwalk [12]. Our method can also use more complex biased sampling strategy, such as Node2vec [1]. But in practice, we find minor improvements.

Similar to [1,8,12], we propose to maximize the likelihood of the context nodes given a source node. The underline idea is that the source nodes share similar context nodes should be close in vector space. By assuming conditional independence of the source-context node pairs, we maximize the following objective, as in Eq. 1.

$$O_1 = \prod_{v \in V} \prod_{u \in N(v)} p_1(u|v) \tag{1}$$

We define the conditional probability of source-context nodes with a softmax function.

$$p_1(u|v) = \frac{exp(f(v) \cdot s(u))}{\sum_{k \in V} exp(f(v) \cdot s(k))} \tag{2}$$

f and s are the mapping functions from nodes to the source embeddings and context structure embeddings respectively. Equivalently, f and s are real-value matrices with size $|V| * d$, where each row is corresponding a node and d denotes the dimension of the vectors.

Context Attribute Modeling. We aim to preserve the attributes of the context nodes here. The attributes of the source vector is of course very important too. However, a source node could be the context node of itself during random walk sampling. Hence, for linked nodes, we have already preserve the source attributes implicitly. Since the context nodes is obtained via random walk, our methods can preserve the high-order context attributes.

Similar to structure modeling, we aim to maximize the likelihood of preserving the attributes of all the context nodes.

$$O_2 = \prod_{v \in V} \prod_{u \in N(v)} \prod_{a \in A(u)} p_2(a|v) \tag{3}$$

By optimizing Eq. 2, source nodes with similar context attributes will be closer in vector space. We define the conditional probability of $p(a|v)$ as the softmax function below, where g is the mapping function from attributes to the context attribute embeddings. Equivalently, g is a real-value matrix with size $|A| * d$, where each row is corresponding to a node.

$$p_2(a|v) = \frac{exp(f(v) \cdot g(a))}{\sum_{k \in V \; and \; T(k)=T(a)} exp(f(v) \cdot g(k))} \tag{4}$$

Maximizing Eq. 3 actually has two effects: to enhance the similarity between a source node v and its context attributes, as well as weaken that between v and other attributes. Note that the denominator is not the whole attribute set, but the attributes of the same type to a. In general, to enhance the similarity of one type of attributes should not affect that of the other types. For example, there are two types of attributes in a friendship network, the gender and the career. If a person has the attribute gender-male, then the probability of gender-female should be zero, but the probability of career-teacher should not be influenced. The other advantage of the design is to reduce the calculation cost in the denominator.

4.2 PCANE++: Encoding the Source Attributes Explicitly

Since a source node can be the context node of itself, PCANE has already modeled the source attributes implicitly. However, sometimes it is still necessary to encode the source attributes explicitly. First, some attributed networks may contain some isolated nodes with only attributes but no links. The random walk sampling strategy will produce no context nodes, and hence no context attributes for these nodes. Therefore, PCANE can not even model the source attributes for these nodes. Second, it is difficult to sample the source node as its own context node if the node degree is large or the network is dense, since the walker can easily walks faraway from the source node. Hence, source attributes may not be utilized efficiently. Based on the above considerations, we propose PCANE++, which integrates the source attributes explicitly to the source node. PCANE++ can better cope with isolated nodes and highlight the source attributes.

The basic idea is to project the source attributes of v to a separate vector $c(v) \in \mathbb{R}^{d_2}$, and then concatenate it with original source vector $f(v)$ to obtain the new source vector $f_{++}(v)$.

$$f_{++}(v) = \begin{bmatrix} f(v) \\ c(v) \end{bmatrix}$$

Then we replace $f(v)$ with $f_{++}(v)$ in Eqs. 2 and 4, and leave Eqs. 1 and 3 unchanged. To make the dot product in Eqs. 2 and 4 plausible, we define the dimension of f and c in PCANE++ as d_1 and d_2 where $d_1 + d_2 = d$.

To build $c(v)$, we first define a mapping h from attribute $a \in A$ to source attribute vector $h(a) \in \mathbb{R}^{d_2}$. Equivalently, h is a real-value matrix with dimension $|A| * d_2$. $c(v)$ is define as the summation of v's source attribute vectors.

$$c(v) = \sum_{a \in A(v)} h(a)$$

Considering a node t without any links, $f_{++}(t)$ will not be trained either. Hence, $f(t)$ will be the same as it was initialized for ever. However, $c(t)$ is built based on the t's attributes, which provides the attribute information of t. Moreover, the other training instances will project similar attributes to similar $h(.)$, which makes $c(t)$ more reasonable.

4.3 Optimization

Optimization for PCANE. For PCANE, the optimization of O_1 can be simplified to:

$$\arg\max_{f,s} \sum_{v \in V} \sum_{u \in N(v)} \log p_1(u|v) \tag{5}$$

The calculation of the denominator in $p_1(u|v)$ is computational expensive since it is required to traverse the entire node set. We approximate it using negative sampling [9]. For each sampled context node, we randomly select several other nodes as negative context nodes. The optimization of O_2 can be simplified to:

$$\arg\max_{f,g} \sum_{v \in V} \sum_{u \in N(v)} \sum_{a \in A(u)} \log p_2(a|v) \tag{6}$$

and we will also approximate it with negative sampling if the denominator of $p_2(a|v)$ requires too much calculation. We optimize the two objective functions using stochastic gradient ascent over the model parameters defining the features f, s, g. Specifically, we apply the Adaptive Moment Estimation (Adam) [3], which adapts the learning rate according to parameter frequency.

Optimizing the two objective functions is indeed a kind of multi-task learning. In practice, we will alternatively train the two objectives. Once a batch of source-context nodes are trained, we will feed the model the corresponding batch of source-context attributes. Since a node has multiple attributes in general, the latter batch size is larger and is not fixed.

Optimization for PCANE++. The optimization of PCANE++ is similar to PCANE. The only difference is that the learned parameters are f, h, s, g.

Final Embeddings. After optimization, previous wisdom shows using $f + s$ as the final embeddings will bring improvement [4,6,11]. However, we optimize an additional context attribute objective. f receives more training opportunities than s in PCANE, since both objective functions will update f. f is thereby expected to be more informative than g. We propose $f + \alpha s$ as the final embeddings, where α is a real value weight parameter between [0, 1] and hence can turn down the impact of g. With the same consideration, we propose $f_{++} + \alpha s$ as the final embeddings of PCANE++.

5 Experiments

5.1 Experiment Setup

Dataset. We use one social network: the friendship network of students from University of North Carolina at Chapel Hill (UNC) [14], and two citation

networks: DBLP[1] and CITESEER[2]. Code and preprocessed data to reproduce our results is in github page[3].

UNC data has 7 types of discrete attributes, 18163 nodes and 766800 links. For the citation, 4732 edges, and only the title is provided. We apply TF-IDF to extracted 5 most important words for the titles, and the words are used as discrete attributes. Each paper is labeled with research area, which we will used for node classification task later.

Baseline Methods. We compare our methods with several state of art network embedding methods. We set the final embedding size of all the methods to 128 and the parameters generally follow the settings in the original papers.

Node2vec [1]. Node2vec is a plain network embedding method. We set $p = 1$, $q = 0.25$, $window\_size = 10$, $walk\_length = 10$ and $number\_walks = 80$.

Line [13]. Line is also an embedding method for plain network. We set $order = 3$.

TADW [16]. TADW is an embedding method for network where the nodes are associated with text. We set $lambda = 0.2$.

ASNE [6]. ASNE is for attributed network embedding. We set the ID vector size to 100 and attribute vector size to 28. The rest of parameters are the same as Node2vec.

Some other methods related to attributed network embedding, e.g. AANE [2], TriDNR [10] and SLR [7], are excluded from comparison, as [6] has demonstrated that they were outperformed by ASNE. We excluded SEANO [5] since it is designed for partial labeled attributed networks. TADW is applied to the two citation datasets, while publication venue is not used on CITESEER, since TADW can utilize only text attributes.

Training Details. The random walk sampling parameters of PCANE and PCANE++ are similar to those used in Node2vec and Line. Specifically, we set $k = 10$, $l = 10$ and $r = 80$. The dimension of both PCANE and PCANE++ vectors is 128. The dimension of source structure vector and source attribute vector in PCANE++ are 100 and 28 respectively. We randomly initialize the parameters of the matrices with a Gaussian distribution whose mean is 0.0 and standard deviation is 0.01. We train the models with mini-batch Adam [3] whose batch size is 1024. The number of negative sampling is 64. We set $\alpha \in \{0, 0.05, 0.1, 0.15, 0.2\}$ when the best result is obtained on the validation set of link prediction task. We repeated our experiments for 10 random seed initializations and our results are statistically significant with a p-value of less than 0.01.

[1] https://www.aminer.cn/citation (V4 version).
[2] http://citeseerx.ist.psu.edu/.
[3] https://github.com/zhudanhao/PCANE.

5.2 Link Prediction

Task Description. The link prediction task aims to predict whether two nodes are linked in the test set, when they are not linked in the training set. Each dataset of links is divided to a training set and a test set with training ratio in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. We use normalized Cosine angle to measure the similarity between two vectors, and Area Under the ROC Curve (AUROC) [18] to evaluate the similarities. We train the models until the best results are obtained on the test set. Since the datasets have only positive edges, for the test set, we have to add the same number of random fake links as negative samples. It is worth noting that we design two test sets: a standard test set and a filtered test set. The design of the standard test set is the same as previous studies [1,6,15]. The standard test set may contain some isolated nodes with no links appearing in the training set. In the filtered test set, we filter out all links with isolated nodes.

Results. The results of link prediction task is shown in Fig. 3. Note that on UNC, the results on the standard and the test sets make minor differences. The reason is that even 10% training links has already covered almost the entire node set in UNC. Hence, the filtered and the standard test sets on UNC are nearly the same.

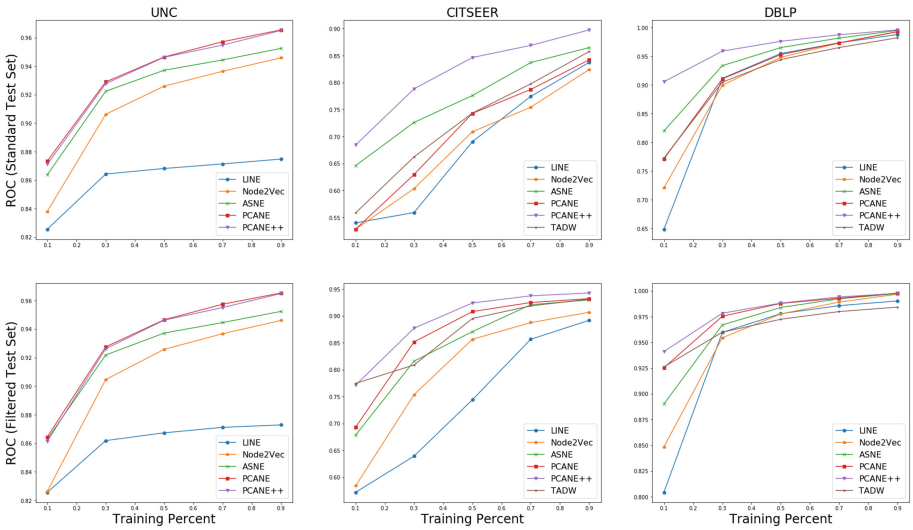


Fig. 3. The results of link prediction. The x axis denotes the fraction of training links, whereas the y axis in the top and bottom rows denote the ROC value on the standard test set and the filtered test set respectively.

Overall, the proposed PCANE++ consistently outperforms all baseline methods on both the standard test set and the filtered test set. For example, given

10% of training links on the standard test set of DBLP, PCANE++ achieves 9.95% improvement of ROC value. PCANE can not outperform TADW and ASNE on the standard test of CITESEER and DBLP, since the standard test set contains a lot of isolated nodes where PCANE cannot make use of. However, on the filtered test set without isolated nodes, both PCANE and PCANE++ can achieve better results than the baseline methods. The results show the modeling of context attributes enables our methods to learn better representations.

Since PCANE++ can cope with the isolated nodes, PCANE++ easily beats PCANE on the standard test sets of all the networks except UNC. However, the advantage is not that strong as on the filtered test sets. With training percent larger than 30% on DBLP, PCANE and PCANE++ achieve similar performance. On UNC, PCANE even slightly outperforms PCANE++. On Citeseer, the advantage is weaken as the training percent arising. The results indicate that for networks without isolated nodes, PCANE and PCANE++ performs similarly.

5.3 Node Classification

Task Description. Node classification aims to predict the label of node in the test set. Therefore, the task can assess if the learned vectors contains sufficient useful information for the downstream tasks. We conduct node classification on Citeseer and DBLP where the research area is used as labels. We exclude UNC for evaluation since no labels consistently exist on all nodes. We train each entire network for one epoch. Hence there are no isolated nodes in both networks. The trained node embeddings are split to training, development and test set with ratio 8:1:1. A simple softmax classifier is trained on the training set until the best result is obtained on the validation set, and we report the results on the test set.

Results. The results of the node classification task is in Table 1. From the results, we can find:

(1) PCANE++ significantly outperforms other baseline methods. On CITESEER, PCANE++ get an improvement of 2.2% on Macro-F1. On DBLP, PCANE++ achieves an improvement of 7.01% (with 62.23% relative error reduction) on Macro-F1. The result shows modeling context attributes can help to produce more informative network embeddings.
(2) PCANE achieves similar performance to PCANE++ on Citeseer. However, PCANE performs even slightly weaker than ASNE on DBLP. The key reason is that DBLP has much more high degree nodes and is denser, which makes PCANE more difficult to utilize the source attributes. We give detail analysis next.

Table 1. Results of node classification.

	CITESEER		DBLP	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
LINE	0.4796	0.4354	0.7932	0.7445
Node2vec	0.4887	0.4359	0.7712	0.7039
TADW	0.5746	0.5105	0.8299	0.783
ASNE	0.5053	0.453	0.8691	0.8401
PCANE	0.5822	0.5321	0.8578	0.8212
PCANE++	**0.5897**	**0.5335**	**0.9520**	**0.9396**

5.4 Discussion

To understand why our methods can produce better node representations, we present the classification performance w.r.t. the degree of nodes in Fig. 3.

The precisions of both Node2vec and ASNE are sharply increasing when the degree is getting larger. However, the precisions of PCANE and PCANE++ are relatively more stable w.r.t. different degrees. The result shows that utilizing only source attributes and network structure is not sufficient enough for embedding the sparser parts of networks. Preserving context attributes can greatly alleviate the data sparse problem, and enhance the learning of low degree nodes.

ASNE outperforms PCANE when degree is larger than 3. We believe the reason is that PCANE cannot sample source attributes as context attributes efficiently for high degree nodes. Since DBLP is dense, PCANE falls behind PCANE++ even on low degree nodes. Hence, it is necessary to use PCANE++ rather than PCANE when the network is dense or contains many high degree nodes.

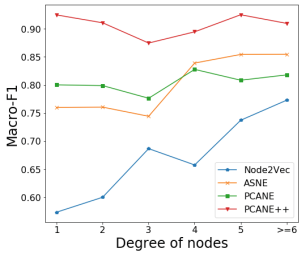
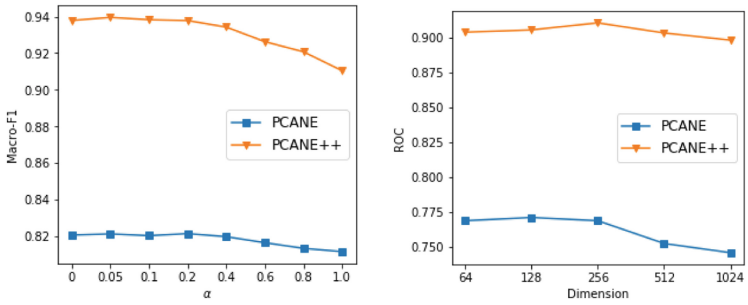


Fig. 4. Classification performance w.r.t. the degree of nodes on DBLP.

5.5 Parameter Sensitivity

Next, we investigate the parameter sensitivity of α and vector dimension. Figure 4(a) presents classification result on UNC w.r.t. different α, and the best

result is obtained when $\alpha = 0.1$, which gives evidence to the necessity of turning down the impact of s. Figure 4(b) gives the result of link prediction w.r.t. different dimensions. The training ratio is 10% and we set the source attribute size of PCANE++ as 28/128% of the total dimension. The result indicates that it is not effective to use too large dimensions (Fig. 5).



(a) α (b) Dimension

Fig. 5. Sensitivity w.r.t. α and dimension.

6 Conclusion

We introduced novel methods to preserve context attributes to improve attributed network embeddings. Our methods can outperform state of art attributed network embedding methods on link prediction and node classification tasks. A number of extensions and potential improvements are possible, such as sampling the context attributes to reduce training time, and improving the walking strategy to balance the effect of high degree nodes.

Acknowledgement. This work is sponsored, in part, by The Natural Science Foundation of the Jiangsu Higher Education Institutions of China under grant number 18KJB510010 and National Nature Science Foundation of China (NSFC) under grant number 61472183.

References

1. 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, pp. 855–864. ACM (2016)
2. Huang, X., Li, J., Hu, X.: Accelerated attributed network embedding. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 633–641. SIAM (2017)
3. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)

4. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Bulletin De La Socit Botanique De France **75**(3), 552–555 (2015)
5. Liang, J., Jacobs, P., Sun, J., Parthasarathy, S.: Semi-supervised embedding in attributed networks with outliers. In: Proceedings of the 2018 SIAM International Conference on Data Mining, pp. 153–161. SIAM (2018)
6. Liao, L., He, X., Zhang, H., Chua, T.S.: Attributed social network embedding. IEEE Trans. Knowl. Data Eng. **30**, 2257–2270 (2018). (Early access)
7. Liao, L., Ho, Q., Jiang, J., Lim, E.P.: SLR: a scalable latent role model for attribute completion and tie prediction in social networks. In: 2016 IEEE 32nd International Conference on Data Engineering, ICDE, pp. 1062–1073. IEEE (2016)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: International Conference on Neural Information Processing Systems, pp. 3111–3119 (2013)
10. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: International Joint Conference on Artificial Intelligence, pp. 1895–1901 (2016)
11. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543 (2014)
12. 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, pp. 701–710. ACM (2014)
13. 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. International World Wide Web Conferences Steering Committee (2015)
14. Traud, A.L., Mucha, P.J., Porter, M.A.: Social structure of facebook networks. Phys. A: Stat. Mech. Appl. **391**(16), 4165–4180 (2012). Social Science Electronic Publishing
15. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234 (2016)
16. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI, pp. 2111–2117 (2015)
17. Zhang, D., Yin, J., Zhu, X., Zhang, C.: User profile preserving social network embedding. In: Proceedings of IJCAI, pp. 3378–3384 (2017)
18. Zou, K., O'Malley, A.J., Mauri, L.: Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. Circulation **115**(5), 654–657 (2007)