2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)

SimWalk: Learning Network Latent Representations with Social Relation Similarity

Shicheng Cui*†, Bin Xia†, Tao Li‡§, Ming Wu†, Deqiang Li†, Qianmu Li*† and Hong Zhang†
†*School of Computer Science & Engineering*
Nanjing University of Science & Technology, Nanjing, China
Email: {cuishicheng, bxia, wuming, lideqiang, qianmu, zhhong}@njust.edu.cn
‡*School of Computer Science*
Nanjing University of Posts & Telecommunications, Nanjing, China
Email: towerlee@njupt@edu.cn
§*School of Computer Science*
Florida International University, Miami, U.S.A.
Email: taoli@cs.fiu.edu

***Abstract*—In this paper, we present a novel method, namely SimWalk, to learn latent representations of networks. SimWalk maps nodes to a continuous vector space which maximizes the likelihood of node sequences. We design a probability-guided random walk procedure based on relation similarity, which encourages node sequences to preserve context-related neighborhoods. Different with previous work which generates rigid node sequences, we believe that relations in social networks, especially similarity, can guide the walk to generate a more linguistic sequence. In this perspective, our model learns more meaningful representations. We demonstrate SimWalk on several multi-label real-world network classification tasks over state-of-the-art methods. Our results show that SimWalk outperforms the popular methods in complex networks.**

Keywords-network latent representations; relation similarity; context-related neighborhoods

I. Introduction

In network analysis, we might be interested in predicting nodes' groups or finding potential correlations. For example, we could recommend people to users who might be willing to acquaint, or in expert mining area [1], experts could be automatically clustered into different groups which are highly related to their expertise.

Traditional ways for analyzing network relations is to use the sparsity of a network representation. It enables the design of efficient discrete algorithms, but can make it harder to generalize in statistical learning [2]. Recently, language modeling techniques have been introduced into network analysis which have proven significant achievement [2, 3]. These algorithms try to utilize random walk models to explore network structures and transform node sequences into sentences by analogy. However, due to the blindness and disorder of random walks, sequences only show context-free relations among nodes, which apparently contradicts with context-related sentences in natural language process (NLP).

Hence, in this paper, we propose SimWalk, a novel method to learn network latent representations. As the network is a domain with object-to-object relationships, we introduce SimRank [5] with Softmax to measure the similarity of the structural context in which objects occur, based on their relationships with other objects, and provides the transition probability from one node to another node. Then, we design a probability-guided random walk procedure based on relation similarity, which attempt to model context-related node sequences. Our method learns node embeddings by maximizing the likelihood of node sequences, using stochastic gradient descent (SGD).

Our key contribution is that we depict context-related graph structures by modeling a probability-guided random walk based on relation similarity. Compared with state-of-the-art methods, SimWalk learns graph structures from both explicit sequences and implicit similarity relationships. Besides, our sequence sampling strategy is more superior, due to the implicit similarity, which not only improves the rigid search strategy, but also reduces hand-crafted features to some extent.

We evaluate SimWalk on several challenging multi-label network classification tasks. Our results show that SimWalk outperforms the popular methods in complex networks.

Overall, our contributions are as follows:

1. We propose SimWalk, a novel method to learn network latent representations with relation similarity.

2. We introduce SimRank to reflect implicit graph information, and use Softmax to provide the transition probability between nodes.

3. We depict context-related graph structures by modeling a probability-guided random walk based on relation similarity, which fits NLP format more appropriately.

The rest of the paper is organized as follows. In Section 2, we briefly introduce related work in network embedding learning. In Section 3, we present SimWalk, a new method that learns network latent representations, in detail. We evaluate SimWalk on several multi-label real-world network classification tasks over state-of-the-art methods and show the detail experiments in Section 4. We conclude our work

978-1-5386-1829-5/17/$31.00

in Section 5.

II. Related Work

Representation learning has been a critical issue in NLP, which aims at learning meaningful embeddings for samples, like words, sentences and documents. Bengio et al. [13] try to fight the curse of dimensionality by learning a distributed representation for words which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. Mikolov et al. [6, 14] propose CBOW and Skip-Gram for computing continuous vector representations of words from very large data sets and try to preserve the linear regularities among words. Pennington et al. [15] propose a specific weighted least squares model that trains on global word-word co-occurrence counts and thus makes efficient use of statistics. Glove leverages statistical information by training on the nonzero elements in a word-word cooccurrence matrix and produces a word vector space with meaningful sub-structure.

Recently, many researchers give much attention to network embedding learning problems, which require models to transform graph vertices into meaningful embeddings. With NLP techniques introduced into network analysis, some extraordinary work achieves great progress. DeepWalk [2] first introduces Skip-Gram [6] framework to solve vertex representation problems. It uses rigid random walk to show graph structures. Node2vec [3] defines a flexible notion of a node's network neighborhoods and designs a biased random walk model to simulate breadth-first sampling and depth-first sampling from both local micro-view and global macro-view. For large-scale networks, Line [4] could preserve local and global network structures, and is designed for overcoming large-scale network embedding problems, which is able to learn the embedding of a network with millions of vertices and billions of edges. Some other work also improves network analysis techniques, such as SDNE [7] that exploits the first-order and second-order proximity jointly to preserve network structures, Deep Graph Kernels [8] that learns latent representations of sub-structures for graphs. Analogous to image-based convolutional networks, Niepert et al. [9] present a general framework to extract locally connected regions from graphs.

The main differences between our proposed method and previous work can be summarized as follows:

1. We learn network latent representations from both explicit sequences and implicit graph information.

2. Our unsupervised sampling strategy is probability-guided, not rigid.

3. We aim to preserve context-related graph information to our network embeddings.

III. Method

A. Overall Framework

Let $G = (V, E)$ be a given network, where nodes in V represent objects of the domain, edges in E represent relations between objects, $E \subseteq (V \times V)$. Our aim is to define a function λ, which maps each node v to a d-dimensional representation $\mathbf{x}$, i.e. $\lambda : v \rightarrow \mathbf{x} \in \mathbb{R}^{d \times |V|}$. Due to the success of NLP techniques in network analysis, we use the Skip-Gram framework to optimize our model.

Skip-Gram maximizes the co-occurrence probability among the words that appear in a sentence [2, 7]. Analogous to nodes in a graph, nodes can be viewed as words, and a graph can be regarded as a corpus. Assume that nodes are constituted into a large amount of sequences $Seq(\cdot)$ by a certain sampling strategy, so we can define the following objective function:

$$\sum_{v_i \in V} \sum_{iter=1}^{N_i} \log \Pr(Seq(v_i)|\lambda(v_i)), \tag{1}$$

where $iter$ is controlled by the probability-guided random walk strategy.

We approximate the conditional probability in Eq. (1) using an independence assumption:

$$\Pr(Seq(v_i)|\lambda(v_i)) = \prod_{u_j \in Seq(v_i)} \Pr(u_j|\lambda(v_i)). \tag{2}$$

Hence, given λ and the assumption, we need to optimize the following objective function:

$$\max_{\lambda} \sum_{v_i \in V} \sum_{iter=1}^{N_i} \log \prod_{u_j \in Seq(v_i)} \Pr(u_j|\lambda(v_i)). \tag{3}$$

Calculating Eq. (3) is expensive for a large network, so instead, we can use Negative Sampling or Hierarchical Softmax solutions [6] to approximate. Meanwhile, λ can be learned by optimizing the likelihood objective using SGD.

B. Relation Similarity

Intuitively, two objects are similar if they are referenced by similar objects [5]. Thus, we introduce SimRank to measure similarity between nodes, which reflects implicit graph information. In order to make our measurement more general, we assume that networks are directed, and the undirected edge can be viewed as the edge which points to both sides.

Suppose a and b are two nodes which potentially have relation similarity $sim(a, b)$ between each other. $I(a)$ denotes in-nodes set of a and $I(b)$ denotes in-nodes set of b. Thus,

$$sim(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} sim(I_i(a), I_j(b)), \tag{4}$$

where C is a constant between 0 and 1. If $a = b$, $sim(a, b) = 1$.

SimRank is an iteration process calculating on an $N \times N$ matrix, where N is the number of nodes. $R(a, b)$ denotes the similarity score between a and b. At first, we initialize the matrix as follows:

$$R_0(a,b) = \begin{cases} 0, & if \quad a = b \\ 1, & if \quad a \neq b \end{cases}. \tag{5}$$

After k iterations, each node in the matrix can be updated as follows:

$$R_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)). \tag{6}$$

Since we obtain SimRank matrix after enough iterations, we could measure the similarity between a node and its neighborhoods. However, the similarity score is a scalar, we cannot directly construct transition probability by it. Thus, we use Softmax function to construct our transition probability between nodes. Let $Neig(v)$ be the neighborhoods of node v, $u \in Neig(v)$, the transition probability is defined as follows:

$$\Pr(u|v) = \frac{\exp(sim(v,u) \cdot w(v,u))}{\sum_{o \in Neig(v)} \exp(sim(v,o) \cdot w(v,o))}, \tag{7}$$

where $w(v, u)$ represents $edge(v, u)$ weight.

C. Probability-guided Random Walk

The relation similarity process provides the global implicit graph information, here we attempt to use the information to guide the random walk procedure. The rigid random walk model treats each node equally, and uniformly passes by any neighborhoods of a certain node. However, as the Softmax SimRank model generates transition probability for each node, instead of the rigid one, we construct a probability-guided random walk to improve the whole graph-exploring procedure.

Intuitively, nodes have different importance in a graph. Some may reflect their communities or groups structure, but some only attach to their belonging. Hence, our model treats each node with different importance. The simplest way is to evaluate its importance by its neighborhoods and the edge weight.

$$Imp(v) = \begin{cases} |Neig(v)|, & init \\ (1-d) + d \sum\limits_{u \in Neig(v)} \frac{Imp(u) w(v,u)}{|Neig(u)|}, & others \end{cases}. \tag{8}$$

Eq. (8) is an iteration process, where node u represents one of the neighborhoods of node v. $Imp(v)$ represents the importance of node v, and $w(v, u)$ represents the weight of $edge(v, u)$. d is a damping factor between 0 and 1.

Recall $iter$ that we mentioned in Eq. (1), we attempt to make it directly reflect the importance of a node in a graph. So we use Imp to define $iter$.

$$iter(v) = \begin{cases} \lceil Imp_0(v) \rceil + b, & naive \\ \lceil Imp_k(v) \rceil \cdot b, & fine \end{cases}, \tag{9}$$

where $b \geq 1$ is a bias factor. We provide two ways to obtain $iter(v)$, the $naive$ way uses $|Neig(v)|$ to reflect node importance, and the $fine$ way uses Imp after k iterations.

Suppose $\mathcal{W}(v)$ is a node sequence with the start node v and k nodes generated by the probability-guided random walk. After k walks, we arrive at node u which has $Neig(u)$, assume the next node is p, the probability is as follows:

$$\Pr(p|u) = \begin{cases} 0, & p \notin Neig(u) \\ \frac{\exp(sim(u,p) \cdot w(u,p))}{\sum_{o \in Neig(u)} sim(u,o) \cdot w(u,o)}, & p \in Neig(u) \end{cases}. \tag{10}$$

If $\Pr(p|u)$ is larger than any other neighborhoods, p is more likely to be the $k + 1$ node in $\mathcal{W}(v)$.

Based on our method, we can generate a large amount of node sequences that nodes with high-similarity have higher probability to show up in the same sequence and gather close to each other. Moreover, nodes with high importance occupy a larger proportion. Hence, our method not only depicts context-related graph structure, but also pays more attention to the important nodes.

IV. Experiment

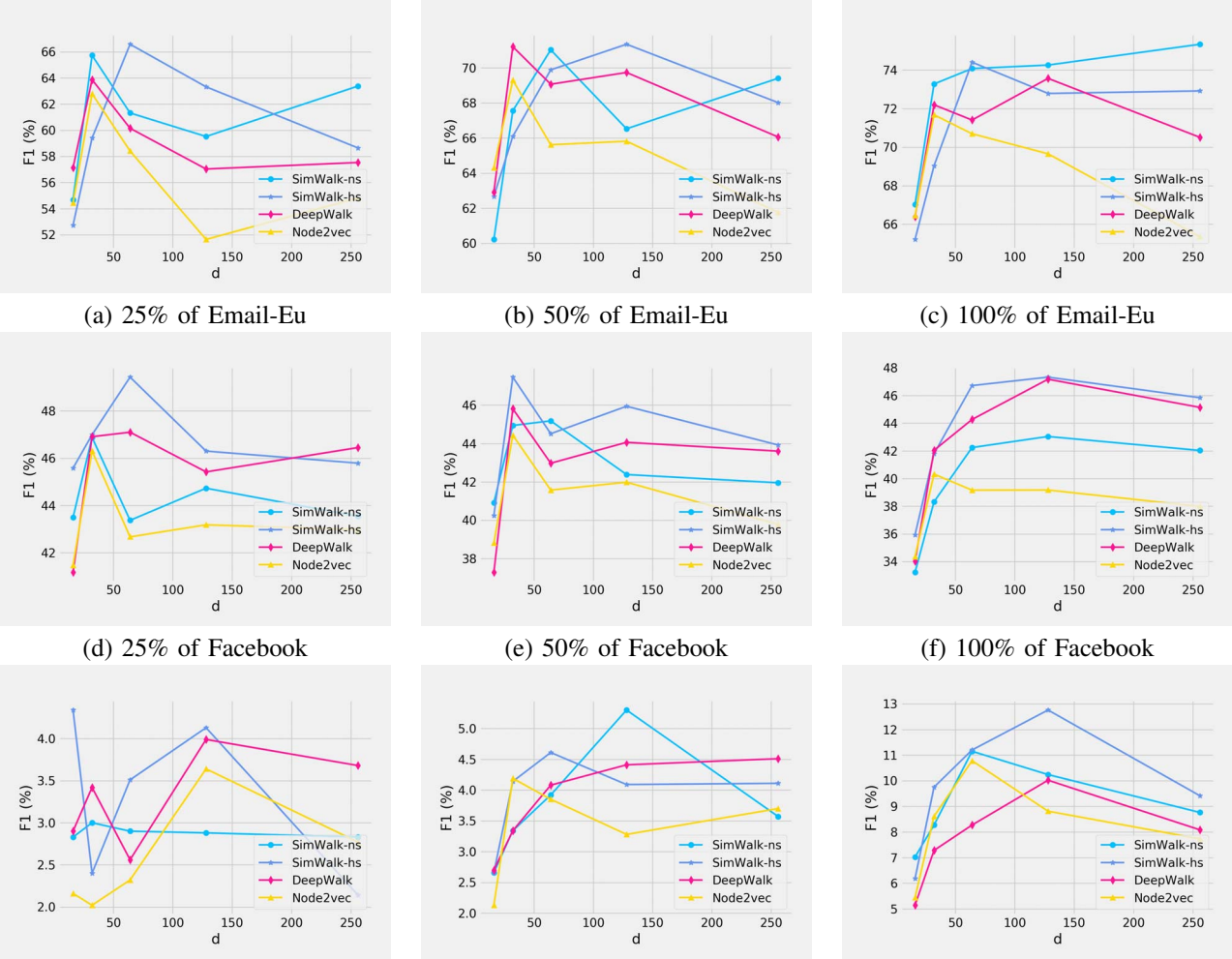
We evaluate the performance of SimWalk against state-of-the-art methods [2, 3] on several multi-label classification tasks. DeepWalk learns low dimensional feature representations by Skip-Gram and random walk. Node2vec uses tunable parameters to guide its random walk model to simulate BFS and DFS.

A. Datasets

Email-Eu [10]: The network was generated using email data from a large European research institution. The nodes represent members, and edges represent an email connection between any two members. The network contains 1,005 nodes and 22,571 edges.

Facebook [11]: This dataset consists of 'circles' from Facebook. The dataset includes node features (profiles), circles, and ego networks. Nodes represent users and edges represent a friendship relationship between any two users. The network contains 4,039 nodes and 88,234 edges.

Sub-BlogCatalog [12]: This network is provided by blogger authors. The labels represent blogger interests. It has 1,000 nodes and 46,957 edges.



(a) 25% of Email-Eu (b) 50% of Email-Eu (c) 100% of Email-Eu

(d) 25% of Facebook (e) 50% of Facebook (f) 100% of Facebook

(g) 25% of Sub-BlogCatalog (h) 50% of Sub-BlogCatalog (i) 100% of Sub-BlogCatalog

Figure 1. Effectiveness over dimensions, d. n% of a certain dataset denotes n% of labeled nodes and all corresponding edges.

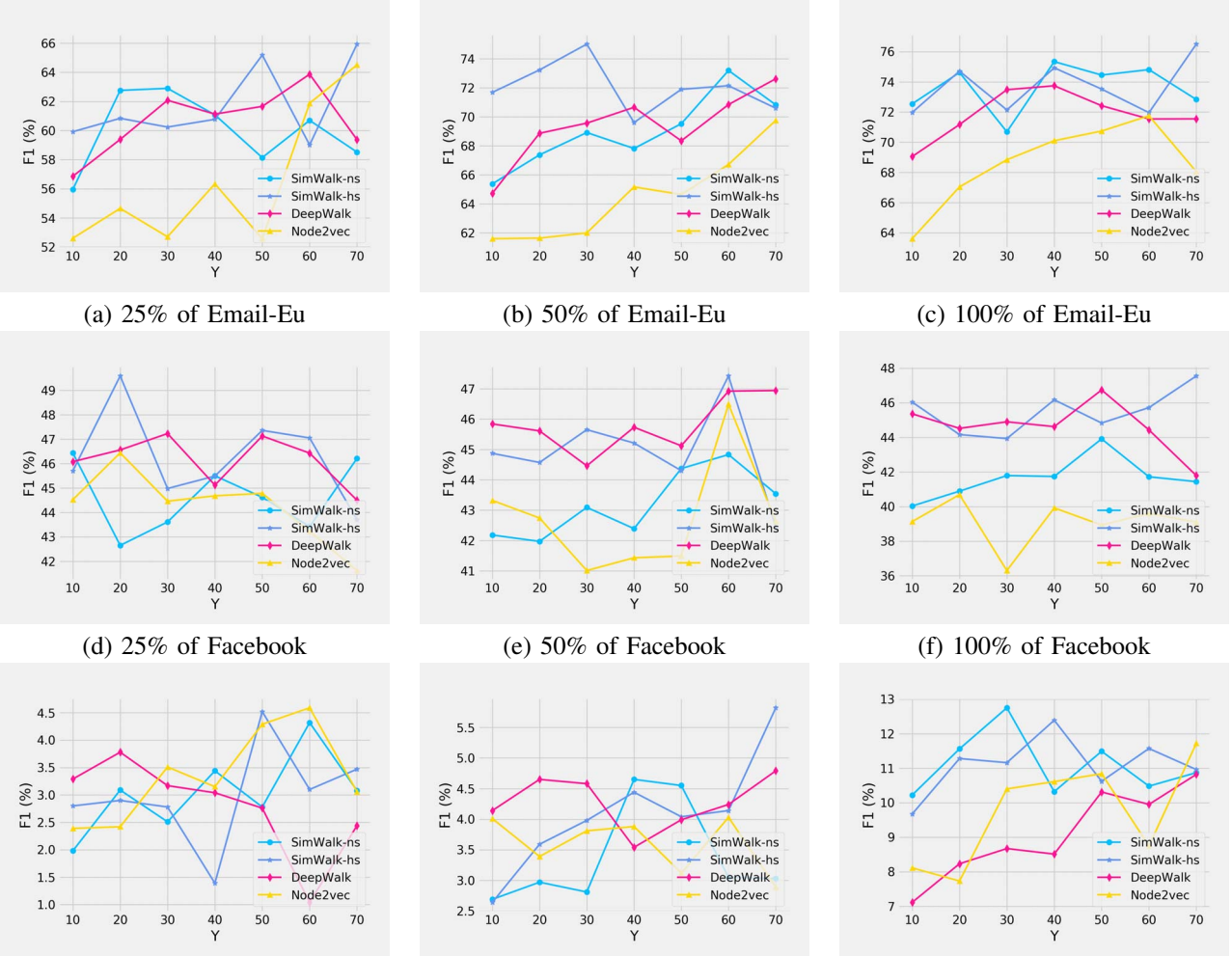
B. Multi-label Classification

We randomly sample a portion of labeled nodes to generate several sub-datasets and evaluate the performance of SimWalk and baselines by the average weighted F-measure (F1) based on 5-fold cross-validation, the process of which is repeated 5 times, with each of the 5 subsamples used exactly once as the testing data, and the remaining 4 subsamples are used as the training data. Besides, the $iter(\cdot)$ is simply obtained by the $naive$ way for SimWalk, and $b = 10$ for all methods. We use the Linear SVM classifier to train all models on multi-label classification tasks. We fix window size $ws = 10$ in Skip-Gram process and only vary the number of latent dimensions (d) and walk length (γ).

We implement both SimWalk with Negative Sampling (SimWalk-ns) and SimWalk with Hierarchical Softmax (SimWalk-hs) to conduct all the following experiments.

C. Dimensionality Experiment

In the effectiveness of dimensionality experiment, we fix $\gamma = 40$ and vary d from 16 to 256. Figure 1 presents the summary of the evaluation results. We can conclude that SimWalk outperforms baseline methods in most circumstances. By varying d from 16 to 32, all methods have a big improvement in F1, but when $d = 256$, SimWalk and baselines barely achieve the highest F1 score on most datasets. Except for SimWalk-ns in Figure 1(c), we hardly find the others show an upward trend, instead, most of them behave unsteadily when $d \geq 128$. Somewhat interestingly, when $d = 64$, we can differentiate SimWalk from baseline methods by the F1 score, due to the better performance of our method. Thus, we believe that $d \in \{32, 64, 128\}$ is more suitable for the dimensionality of network embeddings, especially, SimWalk more likely achieves the best F1 score



(a) 25% of Email-Eu (b) 50% of Email-Eu (c) 100% of Email-Eu

(d) 25% of Facebook (e) 50% of Facebook (f) 100% of Facebook

(g) 25% of Sub-BlogCatalog (h) 50% of Sub-BlogCatalog (i) 100% of Sub-BlogCatalog

Figure 2. Effectiveness over walk length, γ. n% of a certain dataset denotes n% of labeled nodes and all corresponding edges.

under such dimensionality.

D. Walk Length Experiment

In the effectiveness of walk length experiment, we fix $d = 64$ based on the dimensionality experiment, and vary γ from 10 to 70. Figure 2 provides the evaluation results of all methods. It shows that SimWalk still outperforms the other methods. In most cases, by increasing γ, all methods show a general improvement. Mostly, SimWalk with Hierarchical Softmax is superior to SimWalk with Negative Sampling, and the highest F1 score is achieved by SimWalk with Hierarchical Softmax in most circumstances.

V. Conclusion

In this paper, we present a novel method, namely SimWalk, to learn latent representations of networks. SimWalk maps nodes to a continuous vector space which maximizes the likelihood of node sequences. We use SimRank with Softmax to measure similarity of the structural context. We depict context-related graph structures by modeling a probability-guided random walk based on relation similarity, which encourages node sequences to preserve context-related neighborhoods. Compared with state-of-the-art methods, SimWalk learns graph structures from both explicit sequences and implicit similarity relationships. We demonstrate SimWalk on several multi-label real-world network classification tasks over state-of-the-art methods. Our results show that SimWalk outperforms the popular methods in complex networks.

Our future work will focus on learning network relation (edge) representations based on SimWalk model. Besides, treating the edge representation as a bridge between nodes,

we will attempt to figure out how to combine both node embeddings and edge embeddings to design a more general SimWalk model.

ACKNOWLEDGMENT

This research has been supported by the Fundamental Research Funds for the Central Universities (No. 30916015104), the National key research and development program: key projects of international scientific and technological innovation cooperation between governments (No. 2016YFE0108000), CERNET Next Generation Internet Technology Innovation Project (NGII20160122), the Project of ZTE Cooperation Research (2016ZTE04-11), and Jiangsu province key research and development programs: social development project (BE2017739) and industry outlook and common key technology projects (BE2017100).

REFERENCES

[1] S. Lin, W. Hong, D. Wang, and T. Li. "A survey on expert finding techniques." Journal of Intelligent Information Systems, 2017, pp. 1- 25, doi: 10.1007/s10844-016-0440-5.

[2] P. Bryan, R. Al-Rfou, and S. Skiena. "DeepWalk: online learning of social representations." Acm Sigkdd International Conference on Knowledge Discovery & Data Mining (KDD), 2014, pp. 701-710, doi: 10.1145/2623330.2623732.

[3] A. Grover, and J. Leskovec. "node2vec: Scalable Feature Learning for Networks." Acm Sigkdd International Conference on Knowledge Discovery & Data Mining (KDD), 2016, pp. 855-864, doi: 10.1145/2939672.2939754.

[4] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. "LINE: Large-scale Information Network Embedding." International Conference on World Wide Web (WWW), 2015, pp. 1067-1077, doi: 10.1145/2736277.2741093.

[5] G. Jeh, and J. Widom. "SimRank: a measure of structural-context similarity." Acm Sigkdd International Conference on Knowledge Discovery & Data Mining (KDD), 2002, pp. 538-543, doi: 10.1145/775047.775126.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space." International Conference on Learning Representations (ICLR), 2013, arXiv:1301.3781v3.

[7] D. Wang, P. Cui, and W. Zhu. "Structural Deep Network Embedding." Acm Sigkdd International Conference on Knowledge Discovery & Data Mining (KDD), 2016, pp. 1225-1234, doi: 10.1145/2939672.2939753.

[8] P. Yanardag, and S.V.N. Vishwanathan. "Deep Graph Kernels." Acm Sigkdd International Conference on Knowledge Discovery & Data Mining (KDD), 2015, pp. 1365-1374, doi: 10.1145/2783258.2783417.

[9] M. Niepert, M. Ahmed, and K. Kutzkov. "Learning Convolutional Neural Networks for Graphs." International Conference on Machine Learning (ICML), 2016, arXiv:1605.05273v4.

[10] J. Leskovec, J. Kleinberg, and C. Faloutsos. "Graph Evolution: Densification and Shrinking Diameters." ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007, doi: 10.1145/1217299.1217301.

[11] J. McAuley, and J. Leskovec. "Learning to Discover Social Circles in Ego Networks." Conference on Neural Information Processing Systems (NIPS), 2012.

[12] R. Zafarani, and H. Liu "Social Computing Data Repository at ASU.", 2009.

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. "Neural Probabilistic Language Models." Journal of Machine Learning Research (JMLR), 2003, pp. 1137-1155.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality." Conference on Neural Information Processing Systems (NIPS), 2013, pp. 3111-3119.

[15] J. Pennington, R. Socher, and C. Manning. "Glove: Global Vectors for Word Representation." Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.