SINE: Side Information Network Embedding

Zitai Chen[1,2], Tongzhao Cai[1,2], Chuan Chen[1,2](✉), Zibin Zheng[1,2], and Guohui Ling[3]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
chenchuan@mail.sysu.edu.cn

[2] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China

[3] Data Center of Wechat Group, Tencent Technology, Shenzhen, China

Abstract. Network embedding learns low-dimensional features for nodes in a network, which benefits the downstream tasks like link prediction and node classification. Real-world networks are often accompanied with rich side information, such as attributes and labels, while most of the efforts on network embedding are devoted to preserving the pure network structure. Integrating side information is a challenging task since the effects of different attributes vary with nodes and the unlabeled nodes can be influenced by diverse labels from neighbors, not to mention the heterogeneity and incompleteness. To overcome this issue, we propose Side Information Network Embedding (SINE), a novel and flexible framework using multiple side information to learn a node representation. SINE defines a flexible and semantical neighborhood to model the inscape of each node and designs a random walk scheme to explore this neighborhood. It can incorporate different attributes information with particular emphasis depending on the characteristics of each node. And label information can be both explicitly and potentially integrated into the representation. We evaluate our method and existing state-of-the-art methods on the tasks of multi-class classification. The experimental results on 5 real-world datasets demonstrate that our method outperforms other methods on the networks with side information.

Keywords: Network embedding · Random walk · Multilayer network

1 Introduction

Network data are ubiquitous in the real world, ranging from social networks like Wechat and Facebook, marketing networks, airline transportation networks to academic citation networks, to name a few. Abundant useful knowledge is concealed in these networks which can benefit network analysis and applications in reality. For instance, in social networks, link prediction analysis could lower

Z. Chen and T. Cai—Contributed equally to this work.

G. Li et al. (Eds.): DASFAA 2019, LNCS 11446, pp. 692–708, 2019.
https://doi.org/10.1007/978-3-030-18576-3_41

the cost and difficulty for users to seek friends online as well as offer a chance for service providers to improve their user experience. As the size of networks grows, opportunities come with challenges. On the one hand, it enriches the network treasure house and provides ample materials for network researchers. On the other hand, more complex relationships coupled in the networks are increasing the challenges dramatically in the analysis tasks.

Recently, as a novel dimensional reduction technique in analyzing large-scale networks, network embedding is proposed and has attracted a surge of research attention in many researches ranging from data mining, machine learning to mathematics. The main target of network embedding is to preserve as much information as possible from the network with a low dimension representation for each node. To achieve this goal, multiple approaches have been proposed, such as GraRep [2], DeepWalk [14], LINE [17] and SDNE [20]. More importantly, a lot of real-world applications have demonstrated their value in the downstream learning tasks, such as node classification, link prediction and data visualization.

Despite the improvement it gains, current works of network embedding mostly concentrate on preserving the structure of pure networks. In the real world, nodes in a network are usually accompanied with rich side information, such as attributes and labels. The attribute homophily theories [9,10] show the strong connection between node attributes and topological structure. They depend on and influence each other in the network. For instance, articles in Wikipedia might not only cite or be cited by other related articles, but also contain a detailed explanation of the specific object, which helps in link prediction tasks to precisely provide editors with highly related articles. Moreover, labels such as group or community categories also provide useful information to assist in network learning. Even a limited number of labeled nodes can conduct a discriminative embedding. Taking Wechat as an example, users in the same group chat tend to share posts or links of related themes which is informative in precise advertisement targeting. Thus, the importance of side information is self-evident, whilst network embeddings ignoring the side information not only weaken the ability of expression but also blur the representations.

However, it is not easy for the pure network embedding methods like DeepWalk to incorporate additional information during its random walk in the original network since the heterogeneity and incompleteness complicate the situation. Thus, applying the pure network embedding methods directly is problematic. In contrast to the pure network embedding, side information network embedding targets at leveraging the discrepancy of the heterogeneous data sources and distilling the complementary information. What's more, attributes and labels might be sparse, noisy and incomplete. Hence, it is nontrivial to study the problem of fusing labels and attributes into network structure and learning discriminative representations for network nodes. Some recent works have scratched the surface of this topic, yet various problems exist. They either lack careful and specific consideration of side information or are trapped in time-consuming learning models. Exhaustive discussions are given in Sect. 2.

In this paper, we investigate the side information network embedding deeply. Inspired by the groundbreaking work DeepWalk and the constructive follow up work Node2vec on the pure network, we propose an innovative random walk scheme to integrate multiple knowledge on side information network. We aim at answering the following questions: (1) How to incorporate topological structure, attribute information and node labels into a unified representation meanwhile tackling the incomplete, sparse and noisy problem accompanied; (2) How much does this random walk scheme contribute to downstream learning tasks like node classification.

Our main contribution is a flexible framework for learning latent representations for the attributed network with a limited number of labels, called Side Information Network Embedding (SINE). The key ideas of SINE are:

– Measure the node relationships with others on attributes information and then evaluate the importance of attributes and geometric structure for each node individually. In contrast to treating information of each node unanimously, learning on a discriminative data makes the delicate embedding possible.
– Establish *label hubs* and *label hyperlinks* for the labeled nodes to communicate with each other explicitly. And we design a label biased random walk scheme to integrate label information potentially.
– Generate sampled contexts (neighborhoods) for nodes, which contain immediate geometric neighbors, similar nodes in the aspect of attributes and nodes explicitly or latently sharing the same label. Thus, in such all-side neighborhood built with nodes in a heterogeneous relationship, nodes can be modeled with more precise representation. The more frequently two nodes appear in the similar neighborhoods, the more likely they possess similar information.

The rest of this paper is organized as follows: First a brief overview of pure network and side information network embedding is provided, followed by the proposed SINE framework. Then sound experiments are presented. Finally, conclusion and future works are discussed.

2 Related Work

Network embedding can be traced back to the manifold learning, which aims to analyze the structure of manifold and map it into a low dimension Euclidean space to facilitate the machine learning algorithms. However, these methods, such as IsoMap [18], LLE [16], LE [1] and LPP [5], are trapped in the time-consuming eigen-decomposition and not applicable for large scale network embedding.

Recently, inspired by the Skip-Gram [11] learning word representation from its context, [14] propose DeepWalk that generates node neighborhoods with a truncated random walk to simulate the relationship between words and sentences, and bring prosperity to the embedding community. In Node2vec [4], a follow up work of DeepWalk, authors propose a biased random walk which can explore neighborhood under control of extra parameters. To preserve the structure similarity, Struc2vec [15] generates node contexts on the graph which is newly constructed based on structure similarity. On the other line of pure network embedding, a variety of methods [2,15,17,20] are proposed. For example, to preserve first- and second-order proximity, LINE [17] proposes a joint probability and conditional probability model while SDNE [20] adopts an autoencoder model. However, losing sight of labels and attributes may set a limit on the performance of all these topological structure based methods.

Some recent efforts have explored the possibility of integrating side information of the node to learn a better representation. TADW [21] employs an inductive matrix factorization to integrate attributes. SNE [8] proposes a multi-layer perceptron to model the reconstruction error by concatenating attribute record as an input. While they don't model the attribute affinity, which is essential for network analysis. TriDNR [13] learns three kinds of relation node-attribute, inter-node and attribute-label in a coupled deep model. Label information is not used for inter-node relationship modeling, which might weaken its representation power. LANE [7] learns a smooth representation from three individual representations of structure, attribute and label. AANE [6] accelerates the joint learning process of attribute and network structure. However, they equally treat the effect of attribute information on each node, which is too coarse in learning the representations. MMDW [19] only integrates label information by a semi-supervised model, which jointly optimizes the matrix factorization of adjacency matrix and the max-margin classifier of SVM. DANE [3] learns a consistency from the structure and attribute representation which captures the nonlinearity encoded by two autoencoders. However, it suffers from the high computational drawbacks. All in all, the existing methods come across various deficiency. To overcome the problems they meet, we propose a new model with strong pertinence.

3 Framework of SINE

In this section, the problem formulation is firstly given. Then we present the feature learning framework in our method. Next, we introduce attribute embedding module followed by the label embedding module.

3.1 Problem Formulation

We consider the problem of learning node representations in three aspects: structure, attributes and labels. Let $G = \{V, E, W\}$ be a pure network, where V represents the nodes of the network, $E \subseteq (V \times V)$ are the topological connections and W are edge weights (one for unweighted network). With side information, network is further denoted as $G_S = \{V, E, W, X, Y\}$, with multiple attributes $X = \{X^{(i)}\}_{i=1}^{m}$, $X^{(i)} \in \mathbb{R}^{|V| \times s_i}$ where m is the number of attributes and s_i is the size of the i^{th} feature space, and $Y \in \mathbb{R}^{|V| \times |\mathcal{Y}|}$ where $\mathcal{Y}$ is the set of labels. We define a function $\mathcal{L} : V \rightarrow \mathcal{Y}$, and $\mathcal{L}(u) = i$ if node u is labeled with i.

Formally, we aim to learn the low-dimensional representation $H \in \mathbb{R}^{|V| \times d}$ which can incorporate information from three sources of data. As a result, H could achieve better performance in the downstream tasks such as node classification. We denote h_u, a column of H^T, as the representation of node u.

3.2 Feature Learning Framework

We extend the Skip-Gram architecture [11] to the side information network. Formally, in network G_S we maximize the log-probability of observing a network neighborhood $N_S(u)$ for node u conditioned on its representation h_u:

$$\max_H \sum_{u \in V} \log Pr(N_S(u)|h_u). \tag{1}$$

With the assumption of conditional independence and symmetry effects from neighbors, Eq. 1 simplifies to:

$$\max_H \sum_{u \in V} \Big[\log \lambda_u + \sum_{v \in N_S(u)} h_v^T \cdot h_u\Big], \tag{2}$$

where $\lambda_u = \sum_{v \in V} \exp(h_v^T \cdot h_u)$. We can see that nodes in a more similar neighborhood would have similar representations. And a semantically rich neighborhood can more precisely describe the intrinsic correlation on the node. In the following subsections, we will propose our method to integrate side information into the neighborhood. As for the problem of the expensive computational cost on λ_u in Eq. 2, negative sampling [12] is adopted. We optimize Eq. 2 using stochastic gradient ascent over the model parameters defining the features h.

3.3 Measure the Attribute Importance

In contrast to assuming attribute information on different nodes is independent like TADW and SNE, we measure the correlation between nodes with respect to attributes. In detail, a kernel method $\mathcal{K}$ is taken to measure the attribute affinity between any pair of nodes: $\mathcal{K}(u, v) = \phi(X_u) \cdot \phi(X_v)$. We construct the attribute network A that encodes the affinity between two nodes. The edge weight between two nodes u and v is then given by:

$$A_{uv} = \mathcal{K}(u, v), \forall u, v \in V. \tag{3}$$

In G_S, now we have a stack of information networks, 1 topological network G and m weighted networks $\{A^{(i)}\}_{i=1}^m$ built from diverse attributes.

Since the attributes information and topological knowledge are concealed in different networks, we ought to learn the representation from each of the networks. A straightforward way is to build different neighborhoods $N_S^{(i)}(u)$ ($N_S^{(0)}(u)$ denote the neighborhood of node u on G) for each node u on each network of $G \cup \{A^{(i)}\}_{i=1}^m$ and then concatenate their representation learned from respective Skip-Gram models as the final representation. Although from an information preserving view point concatenating different representations could maintain characteristics in diverse networks, it neither alleviates the effect of noise nor distills information hidden across representations from the perspective of integrating information. Furthermore, any incomplete attribute information, which is quite common in real-world datasets, can crash it down for the unobserved node in one of the networks. Another way is to combine neighborhoods $\bigcup_i N_S^{(i)}(u)$ extracted from different networks and then learn the representation from a unique Skip-Gram model. Yet the combination that treats all the side information equally without discrimination for the individual node is careless and unacceptable in the analysis, not to mention the expensive computational cost of building multiple neighborhoods for a node. All in all, how to discriminatingly learn the side information and efficiently sample node neighborhood matters.

Supported by the analysis above, we first propose a measurement on the neighborhood of attribute affinity networks to evaluate the local property. Intuitively, the more similar neighbor is, the more attention should be paid to exploring this neighbor. Exploring the neighborhood shares the same principle. To this end, we define the local cohesion of node u on $A^{(i)}$ as follow:

$$\rho_u^{(i)} = \frac{\bar{a}_u^{(i)}}{\bar{a}^{(i)}} = \frac{avg_t(A_{ut}^{(i)})}{avg_{s,t}(A_{st}^{(i)})}, \tag{4}$$

where $\bar{a}^{(i)}$ is the average edge weight of the i^{th} complete affinity network $A^{(i)}$ and $\bar{a}_u^{(i)}$ is the average weight of edges that associate with node u w.r.t. $A^{(i)}$. Thus, the larger $\rho_u^{(i)}$ is, the more informative immediate neighbors are provided. Then, by comparing the strength of node's local cohesion in different networks, we can distinguish the importance of different attributes for a specific node. In other words, if neighbors are more similar with node u in a certain network than in others, this network should undertake more responsibility for exploring neighbors. For the importance of topology, we can calculate in the same way. We denote $A^{(0)} = G$, $A_{uv}^{(0)} = W_{uv}$ and $\rho_u^{(0)} = 1$ for unweighted network, which is also included in Eq. 4.

Then we propose a multi-network random walk strategy to generate node neighbors $N_S(u)$. Walking across $\{A^{(i)}\}_{i=0}^m$ generates a semantically rich node sequence that incorporates diverse node relationships (or similarities) from different networks. After that, we can construct a neighborhood with multi-relation neighbors for each node. In the proposed random walk scheme, we first decide "which network should be traversed" by $\rho_u^{(i)}$, namely choose the more important data source for node u. The probability is proportional to the importance, in particular:

$$P(u, A^{(i)}) = \frac{\rho_u^{(i)}}{\sum_{i=0}^m \rho_u^{(i)}}. \tag{5}$$

And then we carry out the weighted random walk in the chosen network (e.g. $A^{(i)}$) with the probability as follow:

$$P_A(u,v) = \frac{A_{uv}^{(i)}}{\sum_{x \in V} A_{ux}^{(i)}}. \tag{6}$$

3.4 Fuse the Label Information

Modeling label information is entirely different from attributes, the other kind of side information. Labels are much more refined and scarce in networks. Owning to the sparsity, if we treat labels like attributes to construct an independent network, there would be two problems: Firstly, a great number of nodes without labels will be absent in this network; Secondly, the linkage connecting nodes sharing same the label will build information isolated island, which has no assistance to other nodes. In network analysis, it is always assumed that the node's label is highly correlated to the topological structure and could be affected by its labeled neighbors according to their similarity. We propose two ways to explicitly and potentially fuse labels information in the topological neighborhood as shown in Fig. 1.

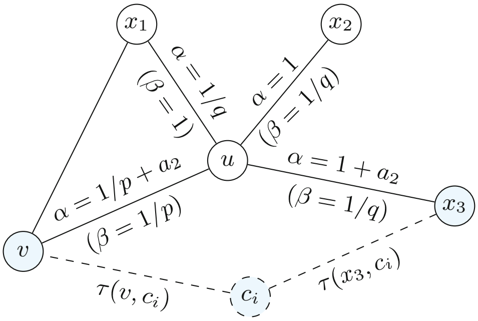


Fig. 1. Illustration of label incorporation way. The nodes colored in cyan are with the same label and the others' label are unknown. The explicit way: The *label hub* c_i is linked with *label hyperlink* (e.g. (v, c_i) and (x_3, c_i)) presented in dashed line. Nodes sharing the same label can walk to each other via their common *label hub*. The potential way: The following example is given to show the influence of node sequence and restrict the influence within 2^{nd} order, which compatible with Node2vec. The walk just transitioned from labeled node v to unlabeled node u and is now evaluating its next step out of node v. Edge notations indicate search bias α for SINE and β for Node2vec. α comprise of bias from topology and label.

To take advantage of label's guidance in gathering node together, we first introduce the notions of *label hyperlink* and *label hub* that help to explicitly learn the label information in the random walk procedure. By building an imaginary *label hub* $c_i, i \in \mathcal{Y}$ for each label on network G, nodes with same label can connect to each other through the *label hyperlink* to the corresponding *label hub*. In particular, the unnormalized probability of walking through *label hyperlink* is defined as follow:

$$\tau(u, c_i) = \begin{cases} \gamma, & \text{if } \mathcal{L}(u) = i, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where γ is a hyperparameter. This explicit method will directly bridge the gap between labeled nodes which are not so close to each other in the topological network. It is also reasonable that nodes with the same label are much closer than those nodes with different labels. In this way, nodes in the neighborhood containing the same label hub are more likely to have similar representations than those who don't.

However, the explicit *label hub* method restricts the influence of label within the labeled nodes, and can not spread the labeled nodes' information to affect the unlabeled neighbors. Thus, we resort to the random walk sequence for a helping hand. Intuitively, a node sequence would be more likely to walk to the related labeled community where it came from. Since nodes in the same community are similar both in topology and label and the node sequence can be regarded as a sampling of the corresponding community, the alternative nodes that are either immediate neighbors of sequence or sharing the same label in the sequence would be more attractive. To measure the attraction of the alternative nodes, we present the biased random walk with two additive parts: topological and labeled parts. Consider a random walk that has a traversed node sequence $T = \{u_i\}_{i=1}^{n}$ with length n and now resides at node u_n (Fig. 1). The walk now defines the unnormalized transition probability of its neighbor x as follow: $\tau(u_n, x) = \alpha \cdot w_{u_n,x}$, where $\alpha = \alpha_{topo} + \alpha_{label}$,

$$\alpha_{topo} = \begin{cases} 1/p, & \text{if } d(\{u_i\}_{n-m}^{n-1}, x) = 0, \\ 1/q, & \text{if } d(\{u_i\}_{n-m}^{n-1}, x) = 1, \\ 1, & \text{otherwise,} \end{cases} \tag{8}$$

$$\alpha_{label} = \begin{cases} \sum_{i=0}^{m} I(\mathcal{L}(u_{n-i}) = \mathcal{L}(x)) a_i, & \text{if } x \text{ is labeled,} \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

and $d(U, x)$ denotes the shortest distance between node x and nodes in set U, I is the indicator function, m is the range of influence of T and $\{a_i\}_{i=1}^{m}$ controls the label influence of different distance. To make it clear, α_{topo} controls the sequence to revisit T with bias $1/p$ and walk around T with bias $1/q$. While the α_{label} controls the probability of traversing the neighbors with label that has been visited in T. We perform the label biased random walk with the probability as follow:

$$P_G(u, v) = \frac{\tau(u, v)}{\sum_{x \in V \cup \{c_i\}_{i=1}^{|\mathcal{Y}|}} \tau(u, x)}. \tag{10}$$

It occurs to us that when we restrict the influence of T within the last two nodes (i.e. $m = 1$) when computing α_{topo}, it is similar to Node2vecWalk defined in [4] with exchanging parameters 1 and $1/q$. We denote the bias as β and explain in Fig. 1. The pseudocode for SINE Walk is given in Algorithm 1. The time complexity analysis of this algorithm is given in the experiment section.

To sum up, by generating the node sequences in the newly designed network with the proposed random walk scheme, we can incorporate topology, attributes and labels information into each node's neighborhood $N_S(u)$. Then we can learn the node representation h_u by solving Eq. 2 with stochastic gradient ascent method.

Algorithm 1. The SINE Walk

```
Input: Start node $u$, networks $\{A^{(i)}\}_{i=0}^{m}$, walk length $l$, label hub weight $\gamma$,
       revisit $p$, look-around $q$, label influence $\phi(d)$
Output: node sequence $T$
Initialize $T$ to empty;
Append $u$ to $T$;
for $iter = 1$ to $l$ do
    Let $curr$ be the last node of $T$;
    Sample $Graph$ from $\{A^{(i)}\}$ with Eq. 5 ;
    $V_{curr}$ = GetNeighbors($curr \in Graph$);
    if $Graph$ is $G$ then
        $V_C = \{c_i\}_{i=1}^{|\mathcal{Y}|}$ ;
        Sample $node$ from $V_{curr} \cup V_C$ with Eq. 10;
    else
        Sample $node$ from $V_{curr}$ with Eq. 6;
    end
    Append $s$ to $T$;
end
return $T$;
```

4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our proposed framework SINE. In particular, we want to answer the following questions.

(1) What are the impact of attributes information on network embedding and how effective is the multi-network random walk strategy to incorporate attributes?
(2) How effective is the guidance impact of the label in the label biased random walk scheme?
(3) How effective are the node representations learned by SINE compared with other state-of-the-art methods in the downstream tasks?

Table 1. Statistics of the dataset

Dataset	Node	Edge	Attribute	Label
BC	5,196	171,743	8,189	6
Flickr	7,575	239,738	12,047	9
Cora	2,708	5,429	1,433	7
Citeseer	3,312	4,732	3,703	6
Wiki	2,405	17,981	4,973	19

4.1 Datasets

In our experiments, we employ 5 real-world datasets: **BlogCatalog (BC), Flicker, Cora, Citeseer** and **Wiki**. All of them are publicly available, and specially the first two have been used in [7]. **BlogCatalog** and **Flickr** are social media networks. Each node is a user and links are the interaction between them. We take their descriptions as the attributes and the groups or categories they joined as labels. **Cora**, **Citeseer** and **Wiki** are citation networks. Each node is a publication and the links are citation relationships between them. The attribute of each node is the bag-of-words representation of the corresponding paper. Statistics of the datasets are summarized in Table 1. Note that all these datasets provide only one attribute feature.

4.2 Baseline Methods

We compare our method with 7 baseline methods. To evaluate the contribution of the side information, two pure network embedding methods, four attributed network embedding methods and a labeled attributed network embedding method are used for comparison. The first category contains **DeepWalk** [14] and **Node2vec** [4]. The second category includes **AANE** [6], **TADW** [21], **SNE** [8] and **DANE** [3]. The last one contains **LANE** [7].

4.3 Metric and Parameter Settings

We perform the multi-class node classification task to evaluate the quality of node representations learned by different methods. To be more specific, we randomly select some portion of the nodes as training set and the remaining as a test set. We train a one-vs-rest SVM classifier on the training set and evaluate it on the test set. For each training ratio, we repeat the trial for 10 times and report the average results. To measure the classification result, we employ Micro-F1 and Macro-F1 scores as metrics.

In SINE, we compute the attribute affinity network with $\mathcal{K}(\cdot,\cdot)$ defined as cosine similarity of attributes. In experiments, we only preserve the top 20 similar neighbors for each node, randomly sample 20 neighbors when performing label biased random walk, and restrict the attraction of the sequence nodes within two step with $a_1 = r, a_2 = s$, which is the trade-off between the computational cost and the accuracy. The default parameters of SINE are set as follows: window size $k = 5$, walks per node $t = 20$, walk length $l = 20$, label biased random walk parameters $p = 4$, $q = 4$, $r = s = 4$, label hub weight $\gamma = 0.5$. The label ratio used for embedding is 10%. For fairness of comparison, the dimension of embedding vectors d is set to 100 for all the methods. The parameters of DeepWalk and Node2vec are kept the same with SINE. The rest parameters for other algorithms are set following the suggestion in their original papers or source codes.

Table 2. Micro-F1 score of classification

Datasets	Ratio	SINE	LANE	AANE	TADW	SNE	DANE	DW	Node2vec
BC	10%	**0.8459**	0.5696	0.7036	0.7502	0.5714	0.7404	0.3561	0.5750
	20%	**0.8805**	0.6543	0.7756	0.7623	0.6201	0.7907	0.4982	0.6317
	30%	**0.8959**	0.6915	0.8103	0.7972	0.6515	0.8084	0.5295	0.6477
	40%	**0.8991**	0.6987	0.8261	0.8053	0.6744	0.8171	0.5666	0.6524
	50%	**0.9055**	0.7199	0.8353	0.8378	0.6773	0.8348	0.5836	0.6739
Flickr	10%	**0.7897**	0.6212	0.5663	0.2901	0.1164	0.4297	0.1563	0.3089
	20%	**0.8454**	0.7043	0.6301	0.3674	0.1542	0.5655	0.2475	0.3772
	30%	**0.8617**	0.7444	0.6583	0.4210	0.1938	0.6091	0.2760	0.3929
	40%	**0.8678**	0.7664	0.6834	0.4429	0.2171	0.6354	0.2942	0.4171
	50%	**0.8780**	0.7856	0.7034	0.4510	0.2402	0.6530	0.3098	0.4256
Cora	10%	**0.7263**	0.6966	0.3601	0.7166	0.5806	0.5099	0.7301	0.7098
	20%	**0.7987**	0.7666	0.5539	0.7974	0.6631	0.6102	0.7819	0.7694
	30%	0.8176	0.7836	0.6260	**0.8225**	0.7079	0.6529	0.7961	0.7928
	40%	0.8290	0.8057	0.6728	**0.8356**	0.7350	0.6774	0.8191	0.8166
	50%	0.8350	0.8173	0.7029	**0.8471**	0.7555	0.6978	0.8330	0.8291
Citeseer	10%	**0.6651**	0.4977	0.3575	0.5594	0.2138	0.5366	0.4722	0.4095
	20%	**0.7189**	0.5655	0.5101	0.6316	0.3366	0.6210	0.5432	0.5110
	30%	**0.7282**	0.6073	0.5566	0.6595	0.3937	0.6535	0.5846	0.5563
	40%	**0.7390**	0.6281	0.5825	0.6690	0.4354	0.6541	0.6013	0.5925
	50%	**0.7476**	0.6391	0.5915	0.6862	0.4617	0.6734	0.6139	0.5995
Wiki	10%	**0.6601**	0.5684	0.6159	0.4498	0.5624	0.6501	0.4269	0.4427
	20%	**0.7315**	0.6382	0.7066	0.5664	0.6310	0.7087	0.5448	0.5505
	30%	**0.7647**	0.6629	0.7414	0.6168	0.6612	0.7385	0.5780	0.5803
	40%	**0.7765**	0.6832	0.7551	0.6518	0.6871	0.7471	0.6117	0.6190
	50%	**0.7879**	0.6951	0.7698	0.6688	0.7062	0.7609	0.6311	0.6377

4.4 Performance Evaluation

In this section, we will answer the questions proposed in the beginning of Sect. 5 one by one.

Effectiveness of Multi-network Random Walk Strategy. To answer the first question, we evaluate the proposed multi-network random walk strategy which performs random walk cross multiple networks (including topological network and attribute affinity networks) by conducting a series of experiments. We first perform random walk on the attribute affinity network (**Attribute**) and topological network (**Structure**) respectively and feed the node sequences to Skip-Gram model to get the embeddings for each network. Then we mix the node sequences generated on these two networks and use this mixed corpus to produce embeddings in the same way (**Combine**). Finally, we perform the proposed multi-network random walk strategy without labels. The classification results of these four methods on **BlogCatalog** dataset with different training ratios are shown in Table 3.

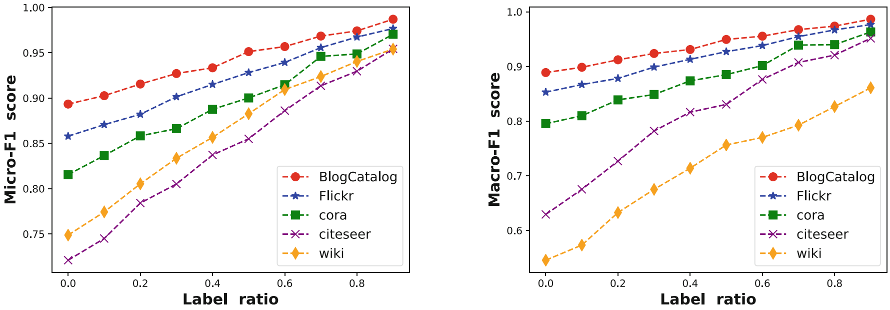
Table 3. F1-score of classification on BlogCatalog

Training ratio		10%	30%	50%	70%
Micro	Structure	0.3520	0.5317	0.5696	0.5987
	Attribute	0.7677	0.8172	0.8360	0.8446
	Combine	0.7866	0.8575	0.8691	0.8833
	SINE	**0.8183**	**0.8770**	**0.8909**	**0.9015**
Macro	Structure	0.3592	0.5428	0.5810	0.6101
	Attribute	0.7797	0.8241	0.8426	0.8498
	Combine	0.7946	0.8624	0.8735	0.8869
	SINE	**0.8251**	**0.8808**	**0.8946**	**0.9044**

The results in Table 3 illustrate the improvement of our multi-network random walk strategy. Specifically, compared to the first two methods which only utilize either attribute information or network structure, the **Combine** and **SINE** methods always achieve significantly better performance, showing that attribute information is valuable on network embedding. More importantly, our method outperforms other methods in all situations, which proves that our proposed multi-network random walk strategy is effective. In contrast to treating attribute and structure separately, we consider the correlation and interaction between them by a unified random walk sequence to effectively incorporate attribute information, leading to much better node representations.

Effect of Label Information. To answer the second question, we investigate the guidance effect of the label by varing the ratio of labeled nodes from 10% to 90% when performing labeled biased random walk. The training ratios of SVM classifier is fixed to 50%. The result is presented in Fig. 2.

From Fig. 2, we can see that with the increase of label ratio, both metrics are rising, which validates the guidance effect of label on embedding.



(a) Micro-F1 score (b) Macro-F1 score

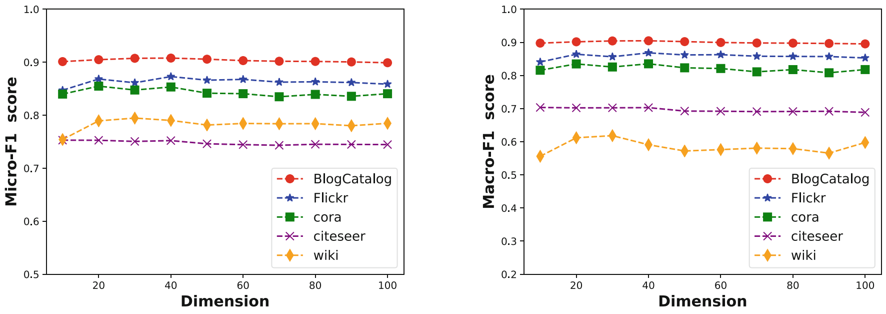
Fig. 2. Classification results of different label ratios

Effectiveness of SINE. To study the effectiveness of our SINE framework which is mentioned in the third question, we compare its performance with all baseline methods with varing the training ratio from 10% to 50%. The classification results of eight methods on five datasets are shown in Table 2. Due to the limitation of space, we only show the result of micro-F1 score and the result of macro-F1 score is similar. From Table 2, we find that our method achieves better performance in most situations with the following observations.

- First, by incorporating the attribute information, most attributed network embedding methods achieve significant improvement compared to the pure network embedding methods.
- Second, with the proposed framework, SINE outperforms other baseline methods in most situations. This is because SINE can effectively integrate the side information and get much more valuable node representations, resulting in better classification results.
- Third, our method performs fairly well when the training ratio is quite small while other baseline methods degrade quickly as the training ratio decreases due to that their representations are noisy and inconsistent in training set and test set. Compared to other algorithms, SINE learns node representations from three data sources, including network structure, attributes information and node labels, which makes the representations more consistent and less noisy.

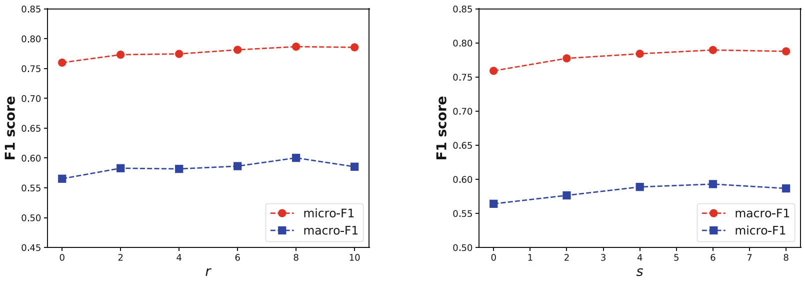
4.5 Parameter Analysis

In this section, we investigate the effects of parameters, including embedding dimension d, label hub weight γ, and labeled bias r and s. We fix the training ratio to 50% and test the classification F1 scores with different parameters. For dimension d, we vary it from 10 to 100 and conduct experiments on five datasets. Figure 3 shows the variations of classification results with different d. The result suggests that our method is stable when d within a reasonable range. As for label

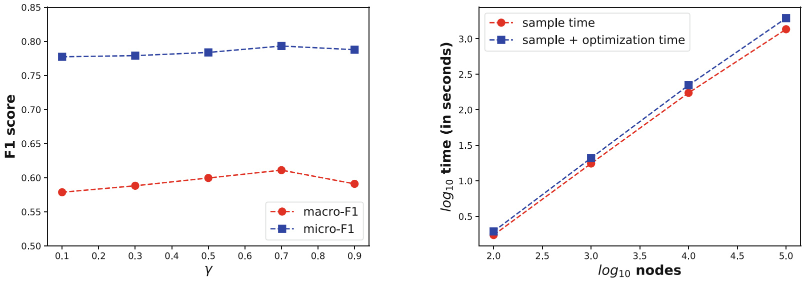


(a) Micro-F1 score (b) Macro-F1 score

Fig. 3. Classification results of different dimensions



(a) Results of different r (b) Results of different s



(c) Results of different γ (d) Average execution time

Fig. 4. Results of parameters analysis and scalability

related parameters γ, r, and s, we set the other two parameters to zeros when analyzing one of them. We vary each of them in different range on the **Citeseer** dataset and the result is presented in Fig. 4. We can find out that the influences of these three parameters are similar. As they increase, the performance becomes better due to the guidance effect of label. However, when they are larger than a threshold, random walk method will always walk to labeled node without walking to its neighbors, losing the information of topological neighborhoods, which reduces the quality of node representations.

4.6 Scalability

The time complexity of our random walk scheme is $O(tle \cdot |V|)$ where e is the average number of edges. In practice, we set $e = 20$ as mentioned in parameter settings so it can be regarded as a constant. The time complexity of Skip-Gram is $O(k(d + d \cdot \log |V|))$ where the window size k and the embedding vector size d are constants so the total complexity is still $O(|V|)$. To test for scalability, we learn node representations using SINE with default parameter values for Erdos-Renyi graphs with node sizes increasing from 10^2 to 10^5. We compute the average running time for 10 independent executions. The result of running time (in log scale) is shown in Fig. 4d. We observe that SINE scales linearly with the size of nodes, which is acceptable in practice. Thus, SINE can be applied to large-scale networks.

5 Conclusion

In this paper, we propose a novel network embedding framework SINE, which can learn high-quality node representations for networks with side information, including attributes and labels. We design a flexible random walk scheme to generate semantically rich neighborhoods for nodes, which contains the proximity in topological structure, node attributes and node labels. The extensive experiments on 5 real-world datasets validate its effectiveness and efficiency.

Acknowlegment. This research was supported by the National Key R&D Program of China (2018YFB1004804), the National Natural Science Foundation of China (11801595, U1811462), the Natural Science Foundation of Guangdong (2018A030 310076), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2016ZT06D211) and the CCF Opening Project of Information System.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction anddata representation. Neural Comput. **15**(6), 1373–1396 (2003). https://doi.org/10.1162/089976603321780317
2. Cao, S., Lu, W., Xu, Q.: Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 891–900. ACM, New York (2015). https://doi.org/10.1145/2806416.2806512
3. Gao, H., Huang, H.: Deep attributed network embedding. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 3364–3370. International Joint Conferences on Artificial Intelligence Organization, July 2018. https://doi.org/10.24963/ijcai.2018/467

4. 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, New York (2016). https://doi.org/10.1145/2939672.2939754
5. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems, vol. 16, pp. 153–160. MIT Press, Cambridge (2003). http://dl.acm.org/citation.cfm?id=2981345.2981365
6. 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). https://doi.org/10.1137/1.9781611974973.71
7. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, pp. 731–739. ACM, New York (2017). https://doi.org/10.1145/3018661.3018667
8. Liao, L., He, X., Zhang, H., Chua, T.: Attributed social network embedding. IEEE Trans. Knowl. Data Eng. 1 (2018). https://doi.org/10.1109/TKDE.2018.2819980
9. Marsden, P.V.: Homogeneity in confiding relations. Soc. Netw. **10**(1), 57–76 (1988). https://doi.org/10.1016/0378-8733(88)90010-X
10. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily insocial networks. Ann. Rev. Sociol. **27**(1), 415–444 (2001). https://doi.org/10.1146/annurev.soc.27.1.415
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119. Curran Associates, Inc. (2013). http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
13. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 1895–1901. AAAI Press (2016). http://dl.acm.org/citation.cfm?id=3060832.3060886
14. 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, New York (2014). https://doi.org/10.1145/2623330.2623732
15. 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. ACM, New York (2017). https://doi.org/10.1145/3097983.3098061
16. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000). https://doi.org/10.1126/science.290.5500.2323
17. 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, Republic and Canton of Geneva, Switzerland (2015). https://doi.org/10.1145/2736277.2741093

18. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000). https://doi.org/10.1126/science.290.5500.2319
19. Tu, C., Zhang, W., Liu, Z., Sun, M.: Max-margin deepwalk: discriminative learning of network representation. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 3889–3895. AAAI Press (2016). http://dl.acm.org/citation.cfm?id=3061053.3061163
20. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM, New York (2016). https://doi.org/10.1145/2939672.2939753
21. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 2111–2117. AAAI Press (2015). http://dl.acm.org/citation.cfm?id=2832415.2832542