SPINE: Structural Identity Preserved Inductive Network Embedding

Junliang Guo, **Linli Xu*** and **Jingchang Liu**
Anhui Province Key Laboratory of Big Data Analysis and Application,
School of Computer Science and Technology, University of Science and Technology of China
guojunll@mail.ustc.edu.cn, linlixu@ustc.edu.cn, xdjcl@mail.ustc.edu.cn

Abstract

Recent advances in the field of network embedding have shown that low-dimensional network representation is playing a critical role in network analysis. Most existing network embedding methods encode the local proximity of a node, such as the first- and second-order proximities. While being efficient, these methods are short of leveraging the global structural information between nodes distant from each other. In addition, most existing methods learn embeddings on one single fixed network, and thus cannot be generalized to unseen nodes or networks without retraining. In this paper we present SPINE, a method that can jointly capture the local proximity and proximities at any distance, while being inductive to efficiently deal with unseen nodes or networks. Extensive experimental results on benchmark datasets demonstrate the superiority of the proposed framework over the state of the art.

1 Introduction

Network embedding has been successfully applied in a wide variety of network-based machine learning tasks, such as node classification, link prediction, and community detection, etc [Cai *et al.*, 2017; Kipf and Welling, 2016]. Different to the primitive network representation, which suffers from overwhelming high dimensionality and sparsity, network embedding aims to learn low-dimensional continuous latent representations of nodes on a network while preserving the structure and the inherent properties of the network, which can then be exploited effectively in downstream tasks.

Most existing network embedding methods approximate local proximities via random walks or specific objective functions, followed by various machine learning algorithms with specific objective functions to learn embeddings [Perozzi *et al.*, 2014; Grover and Leskovec, 2016]. Specifically, the local proximity of a node is approximated with the routine of learning the embedding vector of a node by predicting its neighborhood, inspired by the word embedding principle [Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b] which learns the embedding vector of a word by predicting its context.

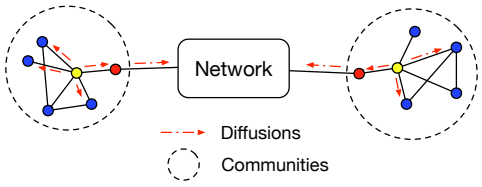

Figure 1: An example of structural identities in the information diffusion process. Red arrows indicate information diffusions between nodes (e.g., retweeting in Twitter), and nodes in the same circle are in the same community.

However, there still exist some potential issues that need further concerns. On the one hand, local proximity preserved methods generally do not model nodes far from each other in practice. Meanwhile, in real-world network mining tasks, nodes that are far apart but close *in structural identity*, or in other words, take similar *roles*, should perform similarly on specific tasks. Figure 1 shows an example of how nodes with different roles perform in the information diffusion processes on social networks. Nodes with different colors indicate different roles in social networks, i.e., *structural hole spanners* (red nodes), *opinion leaders* (yellow nodes) and *ordinary users* (blue nodes) respectively [Lou and Tang, 2013; Yang *et al.*, 2015b]. Intuitively, nodes with same roles behave similarly even with a large distance (yellow nodes in Figure 1), which is the property that should be preserved in the embedding space. In the meantime, the local proximity of a node is also crucial in network embedding. For example in Figure 1, nodes in the same community should be clustered tightly in the embedding space. Therefore, a desirable network embedding method should preserve the local proximity and the global structural identity of a node simultaneously to represent the node precisely. Unfortunately, most existing methods fail to consider the local and global structural information at the same time. In principle, it is challenging to interactively integrate the two kinds of information to obtain comprehensive embeddings rather than a trivial linear combination.

On the other hand, most existing network embedding approaches are *transductive*. To be specific, embeddings are learned on a fixed network, and cannot be directly applied

*Corresponding Author

to new joined nodes or other networks. In contrast, *inductive* methods which are able to generalize to unseen nodes or totally new networks are extensively required in real-world applications, e.g., social recommendation for new users, classification of protein functions in various protein-protein interaction graphs, etc. Unfortunately, traditional network embedding principles such as random walk and matrix factorization are impracticable for unseen nodes or networks, which makes inductive network embedding much more challenging than transductive problems.

In this paper, we propose **SPINE**, an inductive network embedding framework which jointly preserves local proximities and structural identities of nodes. We show that structural similarities between node pairs can be represented by a high-order proximity of the network known as Rooted PageRank (RPR) [Liben-Nowell and Kleinberg, 2007], and by assigning each node a structural feature vector based on RPR, we can encode structural proximities between nodes by measuring the similarities of their structural features. To construct an inductive framework, we learn an embedding generator rather than directly optimizing a unique embedding for each node, through which local proximities are integrated. To further encode structural identities, we propose a biased Skip-Gram Negative Sampling (SGNS) approach with a novel positive sampling strategy guided by structural similarities between nodes. Furthermore, the objective function of SPINE is carefully designed to enhance the structural information contained in the embedding generator.

2 SPINE

In this section, we propose Structural Identity Preserved Inductive Network Embedding (SPINE), a novel inductive approach for *unsupervised* network embedding, which consists of three components: structural feature generation, embedding generation and biased SGNS optimization.

Problem Definition: Given an undirected network $G = \{V, E, \boldsymbol{F}\}$, in which a set of nodes V are connected by a set of edges E, and $\boldsymbol{F} \in \mathbb{R}^{|V|\times f}$ is the content matrix of nodes. The adjacency matrix is $\boldsymbol{A}$ where $A_{i,j} = w_{i,j}$ is the edge weight between node v_i and v_j, and we denote the corresponding transition matrix as $\boldsymbol{P}$, where $P_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{|V|} w_{i,k}}$ represents the transition probability between node v_i and v_j. Our goal is to learn $\boldsymbol{E} \in \mathbb{R}^{|V|\times d}$, where d is a small number of latent dimensions. These low-dimensional representations should well preserve the structural properties of G, including local proximities and structural identities, which can be evaluated with downstream tasks such as node classification.

2.1 Rooted PageRank and Structural Identity

We start with theoretical preliminaries of our structural feature generation algorithm. Here we introduce a well-known high-order proximity of a network named Rooted PageRank (RPR) [Liben-Nowell and Kleinberg, 2007], defined as $\boldsymbol{S}^{\text{RPR}} = (1-\beta_{\text{RPR}})(\boldsymbol{I} - \beta_{\text{RPR}}\boldsymbol{P})^{-1}$, where $\beta_{\text{RPR}} \in (0,1)$ is the probability of the current node randomly walking to a neighbor rather than jumping back to the start node. The (i,j)-th entry of $\boldsymbol{S}^{\text{RPR}}$ is the probability that a random walk from node v_i will stop at v_j in the steady state, which can be used as an indicator of the node-to-node proximity. Therefore, we can use the i-th row of $\boldsymbol{S}^{\text{RPR}}$, denoted as $\boldsymbol{S}_i^{\text{RPR}}$, to represent the global structural information of node v_i. We can further rewrite $\boldsymbol{S}_i^{\text{RPR}}$ in a recursive manner as:

$$S_i^{\text{RPR}} = \beta_{\text{RPR}}\boldsymbol{P}S_i^{\text{RPR}} + (1-\beta_{\text{RPR}})\boldsymbol{g}_i \tag{1}$$

where $\boldsymbol{g}_i$ is the index vector of node v_i whose i-th element is 1 while others are 0.

Next, we are going to verify that $\boldsymbol{S}_i^{\text{RPR}}$ is able to represent the structural identity of node v_i. We first define the *complete structural property* [Batagelj *et al.*, 1992] of a node as:

Definition 1. *A node property $t\colon V \to \mathbb{R}$ is complete structural if for any automorphism φ of every node $v_i \in V$, it always satisfies:*

$$t(v_i) = t(\varphi(v_i))$$

Examples of complete structural properties include $t(v_i)$ = degree of node v_i, $t(v_i)$ = number of nodes at distance k from v_i (k-hop neighbors), $t(v_i)$ = the centrality of v_i, etc. [Batagelj et al.*, 1992].*

Then the following theorem can be directly derived from [Batagelj *et al.*, 1992]:

Theorem 1. *Given a structural description of node v_i defined by a set of complete structural node properties as:*

$$\boldsymbol{T}_i = [t_1(v_i), t_2(v_i), \cdots, t_n(v_i)]$$

where $\boldsymbol{T}_i$ is an n dimensional vector, and $n > 0$ is the number of chosen properties. Let $d(\cdot,\cdot)$ denote standard dissimilarities between vectors, and $v_i \equiv v_j$ indicate node v_i and v_j have equal structural identity, then for $\forall v_i, v_j \in V$:

$$v_i \equiv v_j \iff d(\boldsymbol{T}_i, \boldsymbol{T}_j) = 0$$

From Theorem 1, we can conclude that the structural identities between nodes can be measured through properly designed structural vectors. Next we are going to show that $\boldsymbol{S}_i^{\text{RPR}}$ actually represents a complete structural property.

Following the examples in Definition 1, the centrality of v_i can be regarded as a complete node property of v_i. There are multiple ways to measure the centrality of a node, while the original PageRank [Brin and Page, 2012] is exactly a variant of eigenvector centrality. At each iteration, the PageRank value $\boldsymbol{\pi}_i$ of node v_i is updated as in [Langville and Meyer, 2011]:

$$\boldsymbol{\pi}_i^T = \boldsymbol{\pi}_i^T\boldsymbol{M},$$

and $\boldsymbol{M}$ is the Google matrix defined as:

$$\boldsymbol{M} = \beta\boldsymbol{P} + (1-\beta)\boldsymbol{g}_i\mathbf{1}^{\mathrm{T}} \tag{2}$$

where $\boldsymbol{g}_i = (\frac{1}{|V|}, \cdots, \frac{1}{|V|})_{|V|}$ and $\mathbf{1} = (1, \cdots, 1)_{|V|}$. According to Equation (1), as a variant of PageRank, the only difference between $\boldsymbol{S}_i^{\text{RPR}}$ and the original PageRank is the choice of $\boldsymbol{g}_i$. Therefore, the target matrix of Rooted PageRank can be written as:

$$\boldsymbol{M}_i^{\text{RPR}} = \beta_{\text{RPR}}\boldsymbol{P} + (1-\beta_{\text{RPR}})\boldsymbol{I} \tag{3}$$

where $\boldsymbol{I}$ is the identity matrix. Thus $\boldsymbol{S}_i^{\text{RPR}}$ is the leading left hand eigenvector of $\boldsymbol{M}_i^{\text{RPR}}$, i.e., $\boldsymbol{S}_i^{\text{RPR}}$ satisfies: $(\boldsymbol{S}_i^{\text{RPR}})^T = (\boldsymbol{S}_i^{\text{RPR}})^T\boldsymbol{M}_i^{\text{RPR}}$. As a consequence, $\boldsymbol{S}_i^{\text{RPR}}$ is also a variant of eigenvector centrality, thus can be further regarded as a complete structural property to represent the structural identity of v_i, i.e., $\boldsymbol{T}_i = \boldsymbol{S}_i^{\text{RPR}}$.

Algorithm 1 Rooted random walk sampling

Input: the graph G, the present node v_i, the continuation probability $\beta_{\text{RPR}} \in (0, 1)$, hyper-parameters k, m, l
Output: the structural feature vector $\boldsymbol{T}_i$ of v_i

```
Initialize a counter C_i ∈ R^{|V|}
repeat
    P_s = RootedRandomWalk(G, v_i, l, β_RPR)
    for v_j in P_s do
        C_i[j] ← C_i[j] + 1
    end for
until m times
C_i ← Sort C_i in a descending order
T_i ← The first k elements of C_i
return T_i ← T_i / sum(T_i)
```

2.2 Structural Feature Generation

In this section we first state our motivation of choosing $\boldsymbol{S}_i^{\text{RPR}}$ as the structural identity instead of others, based on which the structural feature generation method is introduced.

To construct an inductive method, we expect the length of the structural description of a node is independent of the total number of nodes $|V|$ and fixed at k. In this paper, we use the top k values of $\boldsymbol{S}_i^{\text{RPR}}$ as the structural description of v_i. Compared with other high-order proximities or node properties (e.g., the original PageRank), RPR captures the global structural information of the network, while being tailored to encode the local structural information of root nodes according to the definition, and thus can better represent the importance of neighbors at various distances to the present node [Haveliwala, 2002]. Moreover, it has been theoretically and empirically proven [Litvak *et al.*, 2007] that in a network with power-law degrees, the RPR values also follow a power-law distribution. Therefore, the largest k RPR values are able to represent the structural information of a node with a suitably chosen k.

When calculating $\boldsymbol{S}_i^{\text{RPR}}$, considering the *inductive* prerequisite, the transition matrix $\boldsymbol{P}$ which encodes the structure of the network may be unavailable. Alternatively, we approximate $\boldsymbol{S}_i^{\text{RPR}}$ from the local structure around v_i through a Monte Carlo approximation. The complete procedure is summarized in Algorithm 1, where m and l indicate the number of repeats and the length per random walk respectively, and k controls the length of the structural feature vector. The largest k values of $\boldsymbol{S}_i^{\text{RPR}}$ are taken as the structural description of v_i, denoted as $\boldsymbol{T}_i$ in the rest of this paper.

2.3 Embedding Generation

To construct an inductive network embedding framework, we generate embeddings instead of directly optimizing the embedding matrix, through which the structural information and the content information of networks can be jointly incorporated.

As stated above, the k values in $\boldsymbol{T}_i$ indicate k largest structural proximities between v_i and nodes co-occurring with v_i in rooted random walks. We denote the content matrix of the corresponding k nodes as $\boldsymbol{F}_i^k \in \mathbb{R}^{k \times f}$, which is constructed row by row according to the same order of RPR values in $\boldsymbol{T}_i$. Given the structural features $\boldsymbol{T}_i$ and node content $\boldsymbol{F}_i^k$, we propose an embedding generation method for v_i.

Specifically, we first employ a multilayer perceptron (MLP) to map nodes from the content space to the embedding space, then compute a linear combination of the k vectors with respect to the corresponding weights in $\boldsymbol{T}_i$. Formally, denote the dimensionality of embeddings as d, the weight matrix of the MLP is $\boldsymbol{W}_{\text{M}} \in \mathbb{R}^{f \times d}$, then the embedding generation process can be written as:

$$\boldsymbol{e}_i = \sigma(\sum_{j=1}^{k} T_{i,j} \boldsymbol{F}_{i,j}^k \boldsymbol{W}_{\text{M}}), \tag{4}$$

where $\boldsymbol{F}_{i,j}^k \in \mathbb{R}^f$ is the j-th row of $\boldsymbol{F}_i^k$, σ is the non-linear activation function, and $\boldsymbol{e}_i \in \mathbb{R}^d$ is the embedding vector of node v_i.

2.4 Biased SGNS

The Skip-Gram Negative Sampling (SGNS) model is widely used in representation learning, which is based on the principle of learning the embedding vector of a word/node by predicting its neighbors. More formally, given an embedding vector $\boldsymbol{e}_i$, SGNS is minimizing the objective function as:

$$J(\boldsymbol{e}_i \mid \boldsymbol{e}_p) = -\log(\sigma(\boldsymbol{e}_i^T \boldsymbol{e}_p)) - K \cdot \mathbb{E}_{v_n \sim P_n(v_p)} \log(\sigma(-\boldsymbol{e}_i^T \boldsymbol{e}_n)), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function and K controls the negative sampling number. v_p and v_n are positive and negative nodes of v_i respectively, while $\boldsymbol{e}_p$ and $\boldsymbol{e}_n$ are the corresponding embedding vectors. Technically, negative nodes are sampled from a distribution P_n, and for most network embedding methods, positive nodes are defined as nodes that co-occur with v_i in a fixed-size window in random walk sequences.

In SPINE, to encourage the similarity of embeddings to jointly encode the similarity in terms of structural identities and local proximities simultaneously, we design a novel biased positive sampling strategy based on the structural features generated from Algorithm 1. The complete procedure is illustrated in Algorithm 2. Specifically, we define a structural rate $\alpha \in (0, 1)$ to control the ratio of structural sampling and local proximity sampling. With probability α, a positive sample of v_i is sampled according to the similarities between their structural features (starting from line 2). Otherwise, the positive node is sampled from nodes that co-occur near v_i on trivial random walks (starting from line 9), which is preprocessed and stored in L_i. The similarity metric in line 4 can be chosen from Euclidean distance, cosine similarity, Dynamic Time Warping (DTW) [Salvador and Chan, 2007], etc. In our experiments, we use DTW which is designed to compare ordered sequences as the similarity metric.

The structural sampling paradigm alleviates the limitation of distance. In practice, it is redundant to compute the structural similarity between v_i and all the other nodes, since nodes with completely different local structures are nearly impossible to be sampled as positive pairs through structural sampling. Intuitively, nodes with similar degrees are likely to have similar local structures. Based on this intuition, we reduce the redundancy by only considering nodes which have similar degrees with the present node v_i. Specifically, given

Algorithm 2 Biased positive sampling

Input: the structural feature matrix $\boldsymbol{T}$, the present node v_i, a node list L_i, the structural rate α
Output: v_p, which is a positive sample of v_i

Initialize an empty list $P_s = []$
if random$(0, 1) < \alpha$ **then**
  for $j = 1$ to $|V|$, $j \neq i$ **do**
    $t \leftarrow$ Compute the similarity between $\boldsymbol{T}_i$ and $\boldsymbol{T}_j$
    $P_s \leftarrow$ Append t to P_s
  end for
  $P_s \leftarrow$ Normalize P_s to $[0, 1]$
  $v_p \leftarrow$ Sample a node according to P_s
else
  $v_p \leftarrow$ Randomly choose a node from L_i
end if
return v_p

an ordered list of node degrees, we choose the candidates for structural sampling by taking $O(\log |V|)$ nodes in each side from the location of v_i. As a consequence, the time complexity of structural sampling for each node is reduced from $O(|V|)$ to $O(\log |V|)$.

2.5 Learning and Optimization

We introduce the biased SGNS based objective function of our framework in this section. We propose two types of embeddings for each node $v_i \in V$, i.e., a content generated embedding $\boldsymbol{e}_i$ as defined in Equation (4), and a structure based embedding $\boldsymbol{s}_i$ which is the i-th row of an auxiliary embedding matrix $\boldsymbol{W}_{\mathrm{S}} \in \mathbb{R}^{|V| \times d}$. In real-world network-based datasets, the content of nodes is likely to be extremely sparse and weaken the structural information incorporated during the generation process of $\boldsymbol{e}_i$. Therefore we employ a direct interaction between $\boldsymbol{e}_i$ and $\boldsymbol{s}_i$ to strengthen the structural information contained in the learned embeddings.

Formally, given the present node v_i and its positive sample v_j, which is sampled according to Algorithm 2, the pairwise objective function of SPINE can be written as:

$$\begin{aligned} F(v_i, v_j) = \lambda_1 \cdot J(\boldsymbol{e}_i \mid \boldsymbol{e}_j) + \lambda_2 \cdot J(\boldsymbol{s}_i \mid \boldsymbol{s}_j) \\ + (1 - \lambda_1 - \lambda_2) \cdot [J(\boldsymbol{e}_i \mid \boldsymbol{s}_j) + J(\boldsymbol{s}_i \mid \boldsymbol{e}_j)] \end{aligned} \tag{6}$$

where λ_1 and λ_2 control weights of different parts, and $J(\cdot|\cdot)$ denotes the pairwise SGNS between two embeddings defined in Equation (5). Intuitively, the generator and the auxiliary embedding matrix should be well trained through their single loss as well as obtaining each other's information through the interaction loss, where λ_1 and λ_2 determine which one is the primary part.

The structure based embeddings $\boldsymbol{W}_{\mathrm{S}}$ and the parameters $\boldsymbol{W}_{\mathrm{M}}$ of the MLP constitute all the parameters to be learned in our framework. The final objective of our framework is:

$$\min_{\boldsymbol{W}_{\mathrm{S}}, \boldsymbol{W}_{\mathrm{M}}} \sum_{i \neq j}^{|V|} .F(v_i, v_j) \tag{7}$$

By optimizing the above objective function, the embedding generator is supposed to contain the content information and the structural information as well as local proximities and structural identities simultaneously. Therefore, during inference, we drop the auxiliary embedding matrix $\boldsymbol{W}_{\mathrm{S}}$ and only keep the trained embedding generator. In the sequel, embeddings of unseen nodes can be generated by first constructing structural features via Algorithm 1 and then following the paradigm described in Section 2.3.

Method	Citeseer	Cora	Pubmed
node2vec	47.2	69.8	70.3
struc2vec	41.1	64.2	60.7
n+s	42.7	68.3	67.1
n+s+f	57.0	73.7	67.3
SDNE	45.2	68.7	69.1
HOPE	46.1	67.2	69.4
Graphsage	52.6	$79.8^{\dagger}$	75.2
GAT	$72.5^{\dagger} \pm 0.7$	$83.0^{\dagger} \pm 0.7$	$79.0^{\dagger} \pm 0.3$
SPINE	72.6 ± 0.4	$\mathbf{83.7} \pm 0.4$	78.5 ± 0.3
SPINE-p	$\mathbf{73.8} \pm 0.2$	82.2 ± 0.7	$\mathbf{82.2} \pm 0.3$

Table 1: Accuracy of transductive node classification (in percentage). "†" indicates that the results are directly copied from their papers and other results are provided by ourselves.

3 Experiments

3.1 Experimental Setup

We test the proposed model on four benchmark datasets to measure its performance on real-world tasks, and one small scale social network to validate the structural identity preserved in the learned embeddings. For the node classification task, we test our method on Citation Networks [Yang *et al.*, 2016], where nodes and edges represent papers and citations respectively. To test the performance of SPINE while generalizing across networks, we further include PPI [Stark *et al.*, 2006], which consists of multiple networks corresponding to different human tissues. To measure the structural identity preserved in embeddings, we test SPINE on a subset of Facebook dataset [Leskovec and Krevl, 2014], denoted as FB-686, in which nodes and links represent users and their connections, and each user is described by a binary vector.

As for the baselines, we consider unsupervised network embedding methods including node2vec [Grover and Leskovec, 2016], struc2vec [Ribeiro *et al.*, 2017] and their variants. Considering that node2vec and struc2vec are designed to preserve the local proximity and the structural identity respectively, we concatenate the corresponding learned embeddings to form a new baseline, denoted as *n+s*, to illustrate the superiority of SPINE over the linear combination of the two proximities. In addition, *n+s+f* denotes the content-incorporated variant of *n+s*. We also compare with SDNE [Wang *et al.*, 2016] and HOPE with RPR matrix [Ou *et al.*, 2016] to test the performance of our inductive Rooted PageRank approximation. On the other hand, in addition to transductive methods, we also consider the unsupervised variant of Graphsage [Hamilton *et al.*, 2017], an inductive network embedding method which jointly leverages structural and content information. We also report the performance of the state-of-the-art supervised inductive node classification method GAT [Veličković *et al.*, 2017]. Random and raw feature results are also included as baselines in this setting.

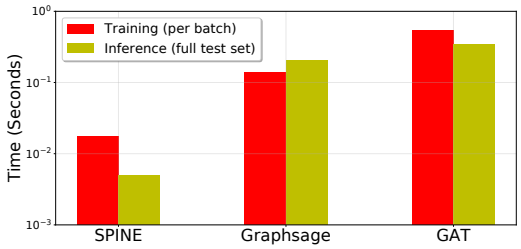

Figure 2: Comparison of running time on Cora, with training batches of size 512 and inference on the full test set (1000 nodes).

For our method, we use SPINE and SPINE-p to indicate the variants with $\boldsymbol{W}_S$ randomly initialized and pretrained with node2vec respectively. To make predictions based on the embeddings learned by unsupervised models, we use one-vs-rest logistic regression as the downstream classifier. For all the methods, the dimensionality of embeddings is set to 200[1].

3.2 Node Classification

We first evaluate the performance of SPINE on node classification, a common network mining task. Specifically, we conduct the experiments in both transductive and inductive settings. For the transductive setting, we use the same scheme of training/test partition provided by Yang *et al.* [2016]. As for the inductive setting, on citation networks, we randomly remove 20%, 40%, 60% and 80% nodes and the corresponding edges, these nodes are then treated as test nodes with the remaining network as the training data. Meanwhile on the PPI network, we follow the same dataset splitting strategy as in [Hamilton *et al.*, 2017], i.e., 20 networks for training, 2 for validation and 2 for testing, where the validation and testing networks remain unseen during training. For both settings we repeat the process 10 times and report the mean score.

Results in the **transductive** setting are reported in Table 1. We can observe that SPINE outperforms all unsupervised embedding methods, and performs comparably with the state-of-the-art supervised framework GAT. In addition, *n+s* performs worse than node2vec, which implies that a simple linear combination of local proximity preserved and structural identity preserved embeddings is incapable of generating a meaningful representation that effectively integrates the two components. The superiority of SPINE over SDNE and HOPE indicates the efficacy of the inductive RPR approximation algorithm as well as the joint consideration of local proximity and structural identity. SPINE also outperforms the content-augmented variant *n+s+f*, which shows that the content aggregation method we propose can better consolidate the content and structure information. Furthermore, the comparison between the two variants of SPINE indicates that while the basic model of SPINE already achieves a competitive performance, we can further enhance the model with initializations of $\boldsymbol{W}_S$ that are well pretrained by focusing only on local proximity, which also justifies the effectiveness of the paradigm of interactive integration proposed in Section 2.5.

As for the comparison on training and test runtime, results are shown in Figure 2. Obviously, SPINE is more efficient in time complexity, especially in the inference stage. In addition, SPINE is also more compact in space complexity, as the parameter scale of SPINE during inference is $O(fd)$, compared to $O(fd + d^2)$ and $O((f + KC)d)$ for Graphsage and GAT with two layers respectively, where K is the number of attention heads and C is the number of classes.

[1]Code is avaliable at https://github.com/lemmonation/spine

	Methods	20%	40%	60%	80%
Citeseer	Random	19.5	20.4	16.7	17.7
	RawFeats	63.9	62.2	60.3	57.7
	Graphsage	58.5	53.9	47.8	41.4
	SPINE	**75.4**	**72.1**	**71.5**	**68.7**
Cora	Random	18.8	22.0	19.1	20.1
	RawFeats	66.6	64.7	64.6	59.6
	Graphsage	73.1	66.4	58.8	48.6
	SPINE	**86.7**	**84.1**	**82.1**	**77.9**
Pubmed	Random	38.5	39.8	39.3	38.9
	RawFeats	75.7	75.4	74.6	72.9
	Graphsage	79.9	79.4	78.2	76.4
	SPINE	**85.7**	**83.7**	**83.0**	**78.8**

Table 2: Accuracy of inductive node classification w.r.t node removal rate (in percentage).

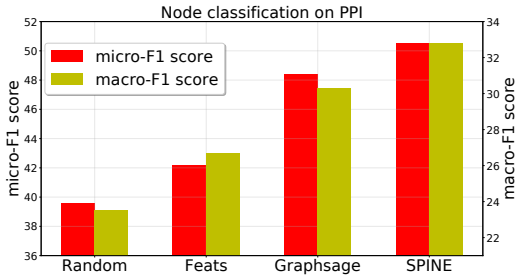

Figure 3: Node classification results on PPI. The left vertical axis indicates the micro-F1 score while the right indicates the macro-F1 score. Both are in percentage.

Results in the **inductive** setting are reported in Table 2 and Figure 3. Note that in this setting we use all the remaining nodes as training data during classification, thus the results are generally larger than that under the transductive setting. One can observe that SPINE outperforms all the baselines, indicating the generalization capability of the embedding generator learned by optimizing our carefully designed objective function. In addition, with the increasing node removal rate which leads to greater loss of local proximities, Graphsage can perform worse than raw features, indicating the limitation of the methods that only preserve local proximities. In contrast, SPINE alleviates the sparsity of local information by incorporating structural identities.

3.3 Structural Identity

We proceed to investigate the structural identity on the FB-686 dataset here. We consider the transductive setting here, and the results under inductive setting can be found in the supplementary material. Specifically, for the original network, we construct a mirror network and relabel the nodes, and consider the union of two networks as the input. As a consequence, node pairs between original nodes and their mirror nodes are obviously structurally equivalent, thus should be projected close in the embeddings space. We then evaluate the Euclidean distance distribution between embeddings of the mirror node pairs and all the node pairs connected by

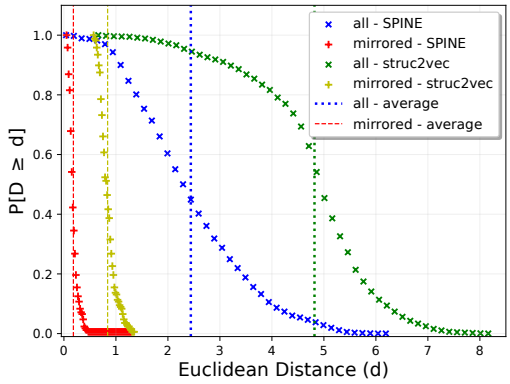

Figure 4: Euclidean distance distribution between mirrored node pairs and connected node pairs on the FB-686 dataset.

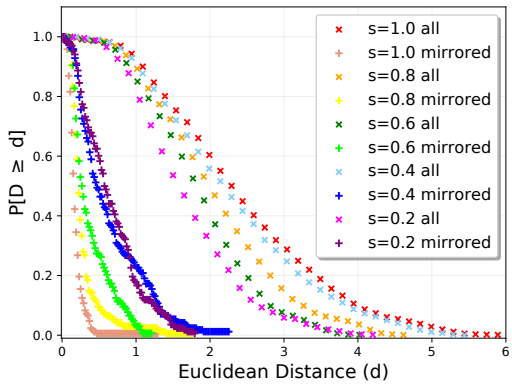

Figure 5: Euclidean distance distribution between mirrored node pairs and connected node pairs on the FB-686 datasets with varying degrees of edge removal and changing content.

edges, denoted as P_m and P_a respectively. Intuitively, if embeddings successfully preserve structural identities of nodes, $\mathbb{E}[P_m]$ should be much smaller than $\mathbb{E}[P_a]$.

Results of SPINE and struc2vec with respect to the two distributions are shown in Figure 4. Obviously, compared to struc2vec, embeddings learned by SPINE yield smaller distances between both mirrored node pairs and connected node pairs, indicating the structural identity and local proximity are jointly preserved better. In addition, the ratio between $\mathbb{E}[P_a]$ and $\mathbb{E}[P_m]$ is 13.40 and 5.72 for SPINE and struc2vec respectively, which means SPINE distinguishes the two proximities more clearly.

Further to test the robustness of SPINE to edge removal and changing content, we randomly sample two new networks from the original FB-686 network. Specifically, we preserve each edge in the original network with probability s, and randomly exchange a 1's location with another 0's location in each node's content vector. Consequently, from the view of structure, the probability for an original edge contained both in the two generated networks is s^2, and smaller s indicates less structure correlation between the two generated networks. From the view of content, mirrored nodes are nearly impossible to have identical content due to the sparsity of content vectors. As can be observed in Figure 5, the ratio between $\mathbb{E}[P_a]$ and $\mathbb{E}[P_m]$ is not significantly affected by the degree of structure perturbation s, which indicates that SPINE can robustly distinguish and preserve structural identity as well as local proximity even with severe perturbations.

4 Related Work

Most network embedding methods consider to preserve local proximity between nodes with frameworks based on random walk [Perozzi *et al.*, 2014; Grover and Leskovec, 2016], skip-gram [Tang *et al.*, 2015; Cao *et al.*, 2015], matrix factorization [Yang *et al.*, 2015a; Guo *et al.*, 2017] and deep learning [Wang *et al.*, 2016; Gao and Huang, 2018] respectively. However, it is worth noting that few of the existing works consider the structural identity between nodes, and fail to handle proximities between nodes at distances. Struc2vec [Ribeiro *et al.*, 2017] preserves the structural identity by constructing a multi-layer complete graph and execute random walk on it. HOPE [Ou *et al.*, 2016] captures structural identity through factorizing a global Rooted PageRank matrix. However, while preserving the structural identity, they ignore the basic local proximities of nodes, which limits its applicability on real-world network mining tasks. Similar problems also occur in two recent methods [Tu *et al.*, 2018; Zhang *et al.*, 2018]. SDNE [Wang *et al.*, 2016], a deep learning based method, is only able to take the first- and second-order proximities into account. Furthermore, most of the methods mentioned above are transductive. Inductive methods [Hamilton *et al.*, 2017; Veličković *et al.*, 2017] tackles this challenge by recursively training a set of aggregators for each node to integrate its neighbors' content as the embedding of the current node in every iteration. As nodes at the k-th iteration contain the structural information from their neighbors within k hops, they cannot deal with nodes at arbitrary distances unless with sufficient iterations, which is costly for real-world tasks.

5 Conclusion

In this paper, we propose SPINE, a network embedding approach which is able to jointly preserve structural identities and local proximities of nodes while being generalized to unseen nodes or networks. We assign a structural feature vector to each node based on Rooted PageRank, and we learn an embedding generator leveraging the structural features of each node to incorporate the structural and content information of nearby nodes. In addition, we propose a biased SGNS algorithm with a novel positive sampling procedure, based on which a carefully designed objective function is proposed to enhance the structural information contained in the embedding generator. Extensive experiments demonstrate the superiority of SPINE over the state-of-art baselines on both transductive and inductive tasks. In future work, we are interested in introducing structural identity to other network-based tasks such as social recommendation.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61673364, No. 91746301) and the Fundamental Research Funds for the Central Universities (WK2150110008).

References

[Batagelj *et al.*, 1992] Vladimir Batagelj, Anuška Ferligoj, and Patrick Doreian. Direct and indirect methods for structural equivalence. *Social networks*, 14(1):63–90, 1992.

[Brin and Page, 2012] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.

[Cai *et al.*, 2017] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques and applications. *arXiv preprint arXiv:1709.07604*, 2017.

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015.

[Gao and Huang, 2018] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *IJCAI*, 2018.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864. ACM, 2016.

[Guo *et al.*, 2017] Junliang Guo, Linli Xu, Xunpeng Huang, and Enhong Chen. Enhancing network embedding with auxiliary information: An explicit matrix factorization perspective. *arXiv preprint arXiv:1711.04094*, 2017.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[Haveliwala, 2002] Taher H Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526. ACM, 2002.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Langville and Meyer, 2011] Amy N Langville and Carl D Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[Leskovec and Krevl, 2014] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[Liben-Nowell and Kleinberg, 2007] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[Litvak *et al.*, 2007] Nelly Litvak, Werner RW Scheinhardt, and Yana Volkovich. In-degree and pagerank: why do they follow similar power laws? *Internet mathematics*, 2007.

[Lou and Tang, 2013] Tiancheng Lou and Jie Tang. Mining structural hole spanners through information diffusion in social networks. In *WWW*, pages 825–836. ACM, 2013.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, pages 1105–1114. ACM, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710. ACM, 2014.

[Ribeiro *et al.*, 2017] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *KDD*, pages 385–394. ACM, 2017.

[Salvador and Chan, 2007] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[Stark *et al.*, 2006] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1):D535–D539, 2006.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[Tu *et al.*, 2018] Ke Tu, Peng Cui, Xiao Wang, Philip S Yu, and Wenwu Zhu. Deep recursive network embedding with regular equivalence. In *KDD*, pages 2357–2366. ACM, 2018.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234. ACM, 2016.

[Yang *et al.*, 2015a] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, 2015.

[Yang *et al.*, 2015b] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. Rain: Social role-aware information diffusion. In *AAAI*, 2015.

[Yang *et al.*, 2016] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *preprint arXiv:1603.08861*, 2016.

[Zhang *et al.*, 2018] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *KDD*, pages 2778–2786. ACM, 2018.