This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIS.2020.3027677, IEEE Intelligent Systems

Department: Head
Editor: Name, xxxx@email

Learning Embeddings Based on Global Structural Similarity in Heterogeneous Networks

Wanting Wen
Institute of Automation, Chinese Academy of Sciences,
University of Chinese Academy of Sciences

Daniel D. Zeng
Institute of Automation, Chinese Academy of Sciences,
University of Chinese Academy of Sciences,
Shenzhen Artificial Intelligence and Data Science Institute (Longhua)

Jie Bai
Institute of Automation, Chinese Academy of Sciences

Kang Zhao
The University of Iowa

Ziqiang Li
Institute of Automation, Chinese Academy of Sciences

Abstract—**With different types of nodes and edges, heterogeneous networks have higher levels of structural diversity than homogeneous networks. This paper proposes an unsupervised representation learning model, named gs2vec, to address structural diversity of a node being connected to other types of nodes via different types of edges in heterogeneous networks. The model measures a node's structural roles based on its numbers of neighboring nodes of different types. It also attempts to measure such structural roles beyond the immediate neighborhood of each node by incorporating structural roles of other nodes k-hop away. Experiments based on synthetic and empirical datasets show that gs2vec outperforms state-of-the-art network representation learning models in heterogeneous network analysis tasks such as node classification and node clustering.**

■ **NETWORKS** are natural ways to represent relationships in a wide variety of real-world scenarios, such as social networks, citation networks, telecommunication networks, and chemical reaction networks. Homogeneous networks have one type of nodes and one type of edges. By contrast, heterogeneous networks consist of multiple types of nodes and edges and can better capture the complex relationships among different types of entities.

Numerous studies have attempted to learn latent representations of nodes based on structural proximity to facilitate network inference tasks, such as node classification [1]. The assumption is that a node's structural position in a network can decide its functions or labels. In other words, a node's relationships with other nodes matter. However, the majority of the network embedding research focused on homogeneous networks [1] [2] [3]. The unique challenge for network representation learning in heterogeneous networks is due to the structural diversity caused by multiple types of nodes and edges.

Corresponding author: Daniel D. Zeng (dajun.zeng@ia.ac.cn)

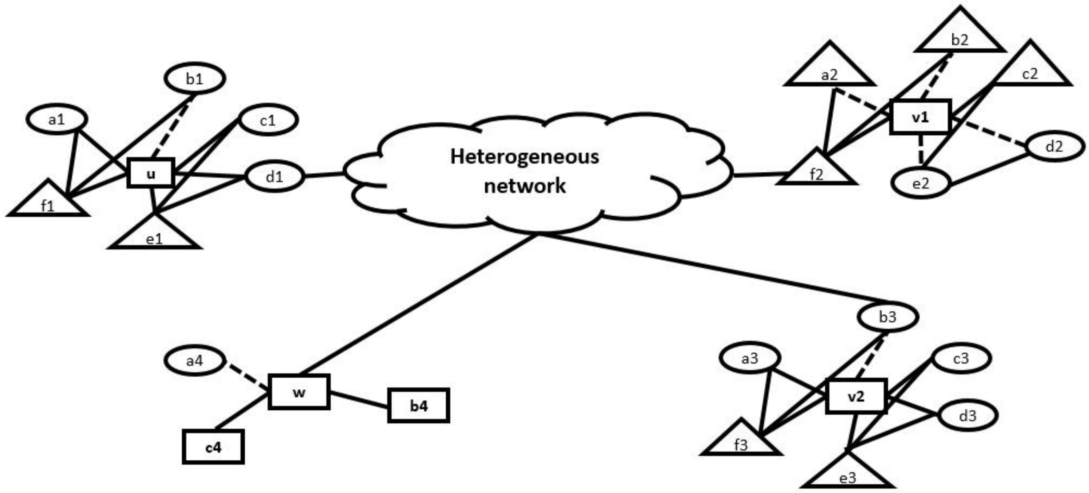


Figure 1. An example of four nodes (u, $v1$, $v2$, w) in four heterogeneous networks. Each node shape represents a node type. Solid lines represent edge type 1 and dotted lines are for edge type 2.

Take a heterogeneous bibliographic network as an example. A bibliographic network can have at least three types of nodes, namely authors, papers and topics. Edges can represent different types of relationships, such as "authors write papers", "authors cover topics", and "papers cite papers. Is one author with four “topic” neighbors equivalent as another author with four “publication” neighbors?

To overcome the challenge, we propose gs2vec, a global-structure based unsupervised representation learning method for heterogeneous networks. The goal of gs2vec is to maximize the likelihood of the structural proximity between nodes by comparing their global structural characteristics in a heterogeneous network. Figure 1 illustrates four different heterogeneous networks, where different shapes represent different node types. From the perspective of structural similarity, node w is different from nodes u, $v1$, and $v2$, because w connects with oval and rectangular nodes, while the other three nodes are all connected with oval and triangular nodes. Meanwhile, u is more structurally similar with $v2$ than with $v1$ because both u and $v2$ have the same numbers of neighbors of each type (two triangular nodes and four oval nodes), while $v1$'s neighbors include four triangular nodes and two oval nodes. Since this structural difference is presented by nodes’ structural characteristics beyond local neighborhood where heterogeneous nodes and edges participated in, we develop a global structure based proximity evaluation strategy, which can quantify the structural similarity of nodes regardless of their topological distance.

In this paper, we first propose the concept of heterogeneous network transformation, so that different types of nodes in heterogeneous networks share the same vector space. Second, we develop a global heterogeneous structural proximity measure to quantify the structural similarity of nodes and edges in heterogeneous networks. Finally, we propose the framework of gs2vec by utilizing a walk-based method based on structural similarity of heterogeneous network to construct node contexts, and applying the Skip-Gram model [4] to learn node embeddings.

Contributions of this paper are summarized below:

a) We formalize the node representation learning problem based on the global structural similarity in heterogeneous networks, and the objective is to learn node embeddings that preserve nodes’ structure roles in heterogeneous networks.

b) We develop an unsupervised network embedding learning method, gs2vec, which can be used for heterogeneous network analysis effectively.

RELATED WORKS

The past few years have witnessed rapid development of network representation learning approaches. Many methods were inspired by Skip-Gram [4], a technique originally designed for embedding texts. Treating nodes in a network as “words”, several algorithms use random walks and biased random walk to generate node sequences as “sentences”, and then uses skip-gram to obtain network embedding and learn high level proximities [2]. These methods were designed for homogeneous networks with one type of nodes and one type of edges. Heterogeneous network representation learning is an emerging research area. To model nodes and edges of different types, most existing approaches learn node embeddings via jointly minimizing the loss over each node or edge type [5] [6], while some methods for studying heterogeneous graph neural networks have introduced attention mechanisms [7].

Methods mentioned above assumed that nodes are closer to each other are similar in certain ways. Besides similarity based on proximity, in many networks, nodes that are not neighbors may also have high similarity if they play similar structural roles in a network. Such structural similarities have been investigated for homogeneous networks [1], but have not been incorporated into representation learning for heterogeneous networks.

In this paper, we fill this gap by proposing a global-structure-based heterogeneous network embedding learning framework. Compared to models for homogeneous networks, our method treats different types of nodes and edges in different ways. Further, gs2vec also differs from other heterogeneous network embedding models in several ways. For example, although metepath2vec [5] pays attention to structure information in heterogeneous networks, it focuses on a few types of hand-picked meta-paths to learn nodal proximities. By contrast, gs2vec can automatically capture the heterogeneous structural similarity of nodes from a broader range of network contexts, so that the learned structural similarity is not limited by the topological distances between nodes in a network. While optimization-based methods such as HNE [8] attempted to optimize node embeddings to maximize the similarity of adjacent nodes, it is difficult to scale up to capture similarity beyond immediate neighborhood, let alone structural similarity.

It is also worth noting that there is also a stream of network representation methods based on supervised approaches. In these approaches, node embeddings are learned to optimize their performance in a specific task, such as node classification or link prediction [7,9]. Compared to unsupervised approaches we discussed above, supervised approaches often perform better for the tasks they are optimized for, but they also have two disadvantages: (1) they lack generalizability--embeddings learned for one task may perform poorly for another; (2) they need a large amount of training data with ground truth, which may not always exist or can be difficult to obtain. Thus in this paper, we focus on unsupervised approaches, which do not need labeled data and produce embeddings that can be applied to different network inference tasks.

GLOBAL STRUCTURE SPACE CONSTRUCTION

Transforming Heterogeneous Network Representations

Definition 1. Heterogeneous network: A heterogeneous network is defined as a graph $G = (V,E,TV,TE)$, where each node v and each edge e are associated with their type-mapping functions $\varphi(v):V \rightarrow TV$ and $\theta(e):E \rightarrow TE$, respectively. $TV = \{tv_1,tv_2,\ldots,tv_p\}$ and $TE = \{te_1,te_2,\ldots,te_q\}$ denote the set of node types and edge types respectively, where $|TV| + |TE| = p + q > 2$.

The basic idea of our model is that when defining a node's structure role, and hence its embedding, the type and number of its neighboring nodes are important. In this paper, we also would like to construct one vector space that can encode structure characteristics of nodes, regardless of their types and positions in the heterogeneous network. Thus we first transform a heterogeneous network by encoding node information in edges to capture more fined-grained neighborhood information.

Specifically, for an edge $e \in E$, we represent it as $e^{'}:e^{'} = (e,v1,v2)$, where $v1$ and $v2$ are nodes at both ends of e. Then the type mapping function of $e^{'}$ should be:

$$\theta^{'}(e^{'}) = (\theta(e),\varphi(v1),\varphi(v2)) \qquad (1)$$

Now we reconstruct heterogeneous network G as $G^{'} = \{V,E',TV,TE'\}$, where edges are formulated as triples to include node information, and edge types are transformed accordingly as:

$$TE^{'} = TV \times TV \times TE \qquad (2)$$

Where operator $\times$ means a Cartesian product function so that $TV \times TE = \{(x,y)|x \in TV \wedge y \in TE\}$.

Mapping Nodes to the Global Structure Space

After transforming a heterogeneous network, we get a set of edge types that are also based on types of nodes being connected. With such fine-grained edge types, we can represent the structural characteristics of nodes by simply mapping them into a vector space, which we define as the "global structure space".

Definition 2. Global structure space: the global structure space Q of a heterogeneous network G is defined as a D dimensional real space R^D, with a corresponding node mapping function $f:V \rightarrow Q$. In this paper, we build the global structure space on top of $TE^{'}$. For node $v \in V$, its mapping function is then defined as $f(v) = < q_i^v >$, where q_i^v is the number of edges of type i that v has and $i \in [1,|TE^{'}|]$.

Take heterogeneous networks in Figure 1 as an example. There are three types of nodes and two types of edges, so the number of edge types $D = |TE^{'}| = 3^2 \times 2 = 18$. For node w , there are two edges of type 1 to connect it with two rectangular nodes, and one edge of type 2 to connect it with one oval node. Thus the vector representation of node w in the global structure space can be written as a vector with a length of 18:

$$f(w) = (2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$$

In fact, edges may not be possible among certain two types of nodes. In addition, if G is undirected, then the dimension of Q can be reduced by half as the type of edges from one type of node to another would be the same no matter which node type is the source node. In Figure 1, only 8 dimensions exist in the undirected network G, namely (1,1,1) , (2,1,3), (1,1,2), (2,1,2), (1,1,3), (1,2,3) , (1,3,3), and (1,2,2), represented in the format of $\theta^{'}(e^{'}) = (\theta(e),\varphi(v1),\varphi(v2))$). Thus we can reduce the global structure space of G from 18 to 8 dimensions. For example, node w has two edges with dimension (1, 1, 1) and one edge with dimension (2,1,3). Thus $f(w) = (2,1,0,0,0,0,0,0)$. Such a novel and more fine-grained vector representation of nodes will then be used in subsequent calculation of structural proximity.

HETEROGENEOUS NETWORK REPRESENTATION LEARNING BASED ON GLOBAL STRUCTURE PROXIMITY

In a transformed heterogeneous network $G^{'}$, the structural proximity of nodes is treated to be hierarchical and cope with increased neighborhood sizes, capturing more refined notions of structural proximity. Intuitively, two nodes that have the same number of edges for each edge type are structurally similar. In addition, if the two nodes' neighbors also have the same number of edges of the same edge type, then they should be structurally more similar.

The level-k neighborhood of a node u in $G^{'}$ is the set of nodes at a shortest distance of k (hop count) from u, independent of edge types. As the structural proximity of nodes should be hierarchical, we learn the structural proximity between nodes level by level. We start with how to measure structural proximity between two nodes at level 0 in global structure space Q, then generalize it to the level k where $k > 0$. Finally, we show how to use node structural proximity at each level to learn network representations.

Structural Proximity at Level 0

For node u in $G^{'}$, there is only u itself at level 0. Therefore, we only consider the structural similarity of two nodes themselves at this level. In global structure space Q, the structural characteristics of u are represented by its vector $f(u)$ in Q. The similarity of two nodes $p^{u,v}$ can be intuitively defined as the distance between two vectors $f(u)$ and $f(v)$:

$$p^{u,v} = \|f(u) - f(v)\| = \left[\sum_{i\, in\, TE'} (q_i^u - q_i^v)^2\right]^{1/2} \quad (3)$$

where smaller distance means higher proximity.

Structural Proximity at Level k

Next we generalize the method of measuring structural proximity with global structure space from level 0 to level k.

Let $R_k(u) = \{v_i\}$ denotes the set of nodes at the k-th level neighborhood of u in $G^{'}$. Note that $R_0(u) = \{u\}$ has only node u itself, and $R_1(u)$ contains u's one-hop neighbors. Let $\mathrm{Q}_k^u = \mathrm{f}(v_i)$ be the global structure space of u at level k, where $v_i \in R_k(u)$ and $\mathrm{Q}_k^u \subset \mathrm{Q}$. We define $\mathrm{P}(Q_1,Q_2)$ as the proximity of two Global Structure space Q_1 and Q_2 .Then the Global Structure proximity of u and v at level k is defined as:

$$p_k^{u,v} = P\left(Q_k^u,Q_k^v\right) = \left[\sum_{i \in TE'} \left(l_{i,k}^{u,v}\right)^2\right]^{1/2} \quad (4)$$

where $l_{i,k}^{u,v}$ denotes the proximity between u and v at level k by dimension i in Q. When k=0, $p_k^{u,v} = p^{u,v}$ *and* $l_{i,0}^{u,v} = q_i^u - q_i^v$. In other words, the proximity between u and v at level k combines the difference between u and v in each dimension of Q_k.

The initial representation of node u at level k in dimension i of Q_k is a real set $s_{h_i,k}^u = q_i^t$, where t$\in$ $R_k(u)$, and $q_i^t \in f(t)$. The number of elements in s is equal to the number of nodes in $R_i(u)$, and elements in set s are not ordered. The task of calculate $l_{i,,k}^{u,v}$ depends on the proximity between sets $s_{i,k}^u$ and $s_{i,k}^v$, but there are two challenges: (1) the two sets could contain different numbers of elements and (2) elements are not ordered in each set.

To calculate $l_{i,k}^{u,v}$, we adopt FastDTW [10] to compute the similarity between two sequences:

$$l_{i,k}^{u,v} = FastDTW\left(S_{i,k}^u,S_{i,k}^v\right) \quad (5)$$

where $S_{i,k}^u$ is the positive ordered version of elements in set $s_{i,k}^u$.

FastDTW is a dynamic time warping algorithm that to find the shortest warp path between two times series that may have different lengths. In each greedy search step of FastDTW, the method needs to define a unit cost between two adjacent elements a in time series A and b in time series B to determine where the next shortest path is. Since elements of sequence A and B represent numbers of edges in our model, we adopt the following cost function:

$$D(\mathrm{a},b) = \frac{|\mathrm{a}-b|}{\min\,(a,b)} \qquad (6)$$

Finally, in order to unify $p^{u,v}$ and $\mathrm{p}_k^{u,v}$, we change Equation (3) into the following format:

$$p^{u,v} = \left[\sum\nolimits_{i\;in\;TE'} \left(\frac{\left|q_i^u - q_i^v\right|}{\min\,(q_i^u, q_i^v)}\right)^2\right]^{1/2} \qquad (7)$$

Contexts of Nodes

The usage of levels allows us to impose a hierarchy to measure structural similarity. Let $hk(u,v)$ denote the structural distance between nodes u and v when considering their k-hop neighborhoods on all edge types. Note that the k-hop neighborhood of node u includes all nodes whose shortest distance to u is less than or equal to k (i.e., at level k or less for u). Such a distance is defined as:

$$h_k(u,v) = h_{k-1}(u,v) + \mathrm{p}_k^{u,v}, \text{where } h_{-1} = 0 \quad (8)$$

Note that by definition, $hk(u,v)$ is non-decreasing with regard to k and is applicable only when both u and v have nodes at level k, although it does not require the presence of all edge types at level k.

With the structural distance between u and v at each hop, we can construct a context graph between nodes in $G^{'}$ and generate the context for each node. To do this, we adopt struc2vec [1], which was originally proposed for homogeneous networks. struc2vec constructs a multilayer weighted graph that encodes the structural similarity between nodes. Each layer is a weighted undirected and complete graph with edge weight between nodes defined as

$$w_k(u,v) = e^{-\,h_k(u,v)}, k = 0,\ldots,K \qquad (9)$$

where number of layers K is the diameter of G.

To generate contexts for nodes, struc2vec uses bias random walks according to edge weights that are based on structural similarity between nodes. When walking in the same layer, the probability of stepping from node u to node v is proportional to their edge weight:

$$p_k(u,v) = \frac{w_k(u,v)}{\sum_{\substack{v\in V\\ v\neq u}} w_k(u,v)} \qquad (10)$$

To walk between different layers, the method first defines a weight function for each node u in layer k to its corresponding node in layer *k-1* and layer *k+1* they exist.

$$\mathrm{w}(u_k,u_{k+1}) = \log\left(\rho_k(u) + e\right), k = 0,\ldots,K-1 \;(11)$$

$$\mathrm{w}(u_k,u_{k-1}) = 1, k = 1,\ldots.K$$

where $\rho_k(u)$ is the number of edges that are incident to u and have weight larger than the average edge weight of the complete graph in layer k. Then the probabilities for a walk to move between two layers at node u are defined as:

$$p_k(u_k,u_{k+1}) = \frac{\mathrm{w}(u_k,u_{k+1})}{\mathrm{w}(u_k,u_{k+1}) + \mathrm{w}(u_k,u_{k-1})} \qquad (12)$$

$$p_k(u_k,u_{k-1}) = 1 - p_k(u_k,u_{k+1}) \qquad (13)$$

The contexts of each node generated with struc2vec can then be used as the input of the skip-gram model to learn node embeddings.

EXPERIMENTS

In this section, we evaluated the performance of gs2vec by comparing it with several state-of-the-art unsupervised representation learning methods in different network analysis tasks.

Baseline methods we include in experiments are: (1) DeepWalk [2], which uses local information obtained from truncated random walks to learn latent representations; (2) Struc2vec [1], which discovers structural embeddings at different scales through a sequence of walks on a multi-layered graph; (3) Role2vec [3] which uses a flexible notion of attributed random walks to capture structural similarity (roles); (4) Metapath2vec [5], which learns embeddings for all types of nodes in heterogeneous networks by following a predefined meta-path scheme; and (5) HiWalk [6] which learns embeddings for nodes whose types are predefined in heterogeneous networks.

A Synthetic Graph

We first evaluate our method on a synthetic graph we create (Figure 2(a)). We generate the graph with different types of nodes, whose labels indicate each node's structural role. Our goal is to evaluate if our method can recover nodes' structural roles. In this graph, different shapes represent different types of nodes and structurally equivalent nodes have the same color. For example, nodes 9, 10, 11, 13 and 14 are considered equivalent in their structure roles.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIS.2020.3027677, IEEE Intelligent Systems

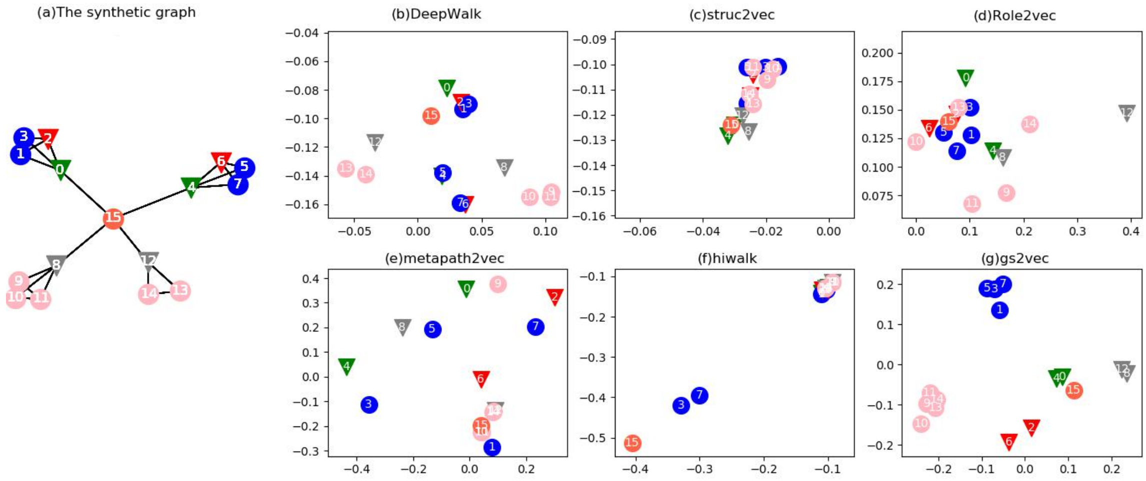


Figure 2. (a)A synthetic graph, where shapes represent node types and colors indicate nodes' structural roles， along with 2D latent vector representations of nodes learned by (b) DeepWalk, (c) struc2vec, (d)Role2vec, (e) Metapath2vec, (f)Hiwalk, and (g) gs2vec, and the horizontal axis and the vertical axis represent values in the first dimension and the second dimension, respectively. Parameters used for all methods: number of walks per node: 40, walk length: 20, skip-gram window size: 10.

To evaluate if gs2vec can learn vector representations that capture the structural equivalence mentioned above, we show embeddings learned by our method along with benchmark methods in Figure 2 (b-g). Note that it is not straightforward to visualize high-dimensional vectors like our node embeddings. Thus we adopt t-SNE [11], a non-linear dimensionality reduction algorithm, to reduce the dimension of node embeddings to 2, and use the 2-D position of each node to represent its embedding. The results reveal that DeepWalk [2] and Hiwalk [6] cannot learn structural equivalence for this network, most likely because they were not designed to detect heterogeneous structural roles. Metapath2vec [5] learns node characteristics on specific meta-paths (like A-B-A) and cannot capture structural roles on a more global level. Struc2vec [1] and Role2vec [3] cannot capture heterogeneous structural roles as they consider all nodes in complete graph as the similar nodes. The proposed gs2vec achieves the best results by effectively generating embeddings where nodes with the same color are close to each other and far from nodes with different colors.

Node Classification

Datasets. We chose BZR-MD [13] and AIDS [14] as the empirical datasets in the following node classification and node clustering experiments. The two data sets consist of molecular compounds which are converted into heterogeneous chemical networks in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. The heterogeneous chemical networks contain a number of subgraphs. Usually the subgraphs are small and isolated from each other, so it is difficult to evaluate the structural proximity of nodes through conventional methods. However, by mapping nodes to the global structure space (as discussed in Section 3), our proposed method can easily capture nodal structural proximity.

Table 1. Details for the two datasets from the four data sets, |TE'| is the dimensions of transformed network

Dataset	BZR-MD-part1	BZR-MD-part2	AIDS1	AIDS2
Nodes	874	897	1028	2907
Edges	957	979	1110	3161
|TE'|	4	3	2	2
Node Label	6	8	7	7

BZR-MD data set is a chemical compound dataset. Nodes correspond to atoms and edges refer to chemical bonds. Node types include C, N, O, F, C1, S, P, and Br. Edge types include single, double, triple or aromatic. We use node types as node labels for multi-class classification. To test the effectiveness of the global structure on network representations, we randomly selected 80 subgraphs from the BZR-MD dataset to construct two subsets of data, named as BZR-MD-part1 and BZR-MD-part2, with 40 subgraphs in each dataset. These two data sets represent two typical chemical compound structures.

Table 2. Multi-class node classification results (± stand standard deviations) for the four data sets with different TR. Bold fonts indicate the best performer in each column.

BZR-MD-part1	TR	Deepwalk	struc2vec	role2vec	metapath2vec	HiWalk	gs2vec
Micro-F1(%)	10%	68.34	79.49	71.81	75.96	69.83	**91.41**± 1.39
	30%	72.39	85.56	74.63	75.89	71.77	**94.49**± 1.00
	50%	74.00	88.15	74.44	75.40	72.74	**95.18**± 0.81
	70%	74.6	89.09	75.34	75.68	73.70	**95.59**± 1.33
	90%	78.85	91.84	78.62	78.97	78.16	**95.92**± 0.71
Macro-F1(%)	10%	16.13	26.54	17.14	14.38	14.75	**57.39**± 6.92
	30%	16.12	44.34	15.81	14.38	14.60	**66.79**± 3.27
	50%	15.44	50.63	16.24	14.32	14.23	**70.06**± 4.21
	70%	15.78	54.50	15.17	14.35	14.22	**72.82**± 4.26
	90%	15.01	53.15	14.69	14.70	14.61	**70.64**± 7.62
BZR-MD-part2	**TR**	**Deepwalk**	**struc2vec**	**role2vec**	**metapath2vec**	**HiWalk**	**gs2vec**
Micro-F1(%)	10%	63.81	79.87	70.47	67.58	74.55	**89.94**± 1.64
	30%	70.77	86.04	73.00	69.67	75.35	**94.41**± 0.77
	50%	73.17	87.67	73.62	73.06	75.01	**94.53**± 0.56
	70%	73.56	88.17	73.82	74.15	75.19	**94.42**± 1.37
	90%	73.19	88.70	73.75	74.20	74.42	**95.61**± 2.10
Macro-F1(%)	10%	11.93	24.93	12.40	10.67	11.28	**48.37**± 3.78
	30%	12.55	37.97	12.19	10.74	10.79	**54.66**± 2.45
	50%	11.87	38.12	12.36	10.71	10.77	**56.11**± 3.18
	70%	11.66	40.30	11.19	10.72	10.64	**55.35**± 5.10
	90%	11.51	38.09	12.11	10.66	10.64	**54.69**± 6.74
AIDS1	**TR**	**Deepwalk**	**struc2vec**	**role2vec**	**metapath2vec**	**HiWalk**	**gs2vec**
Micro-F1(%)	10%	65.15	74.65	71.12	74.34	69.64	**80.13**± 1.25
	30%	68.98	77.42	72.87	74.22	70.16	**84.34**± 1.41
	50%	73.19	77.76	72.77	73.98	71.80	**84.83**± 2.06
	70%	76.04	80.85	73.87	73.79	72.67	**86.81**± 2.01
	90%	74.76	79.76	74.15	77.23	72.29	**87.32**± 4.41
Macro-F1(%)	10%	11.04	16.96	11.38	9.47	10.51	**24.01**± 5.36
	30%	11.35	25.93	11.51	9.46	11.08	**36.73**± 5.02
	50%	10.96	29.29	11.40	9.44	11.07	**42.01**± 2.85
	70%	11.23	30.44	12.06	9.43	10.91	**39.64**± 3.5
	90%	10.36	27.50	11.50	9.68	10.24	**36.16**± 6.81
AIDS2	**TR**	**Deepwalk**	**struc2vec**	**role2vec**	**metapath2vec**	**HiWalk**	**gs2vec**
Micro-F1(%)	10%	53.42	88.35	57.7	61.87	53.59	**88.93**± 0.72
	30%	62.02	89.39	59.81	62.1	56.31	**90.82**± 0.43
	50%	65.34	89.72	61	61.59	57.62	**91.08**± 0.39
	70%	67.76	90.39	60.64	61.08	58.5	**91.33**± 0.77
	90%	69.83	89.52	60.79	61.66	59.41	**91.27**± 2.09
Macro-F1(%)	10%	13.87	39.7	13.46	9.54	13.07	**41.77**± 0.74
	30%	14.79	41.13	12.83	9.56	12.89	**42.44**± 0.55
	50%	15.29	41.61	12.19	9.51	12.43	**42.74**± 0.39
	70%	16.06	42.34	11.63	9.46	11.97	**43.11**± 0.70
	90%	16.83	41.19	11.42	9.51	11.64	**43.2**± 0.70

The AIDS dataset consists of graphs representing molecular compounds. Nodes are labeled with the number of the corresponding chemical symbol and edges by the valence of the linkage. We selected two sets of subgraphs from the AIDS data set to construct two subsets of data by the scale of the subgraphs: AIDS1 and AIDS2. AIDS1 consists of all subgraphs containing nodes number between 25 and 30, while

AIDS2 consists of the subgraphs containing nodes number more than 60, in which the subgraphs are much larger than those in AIDS1. These two data sets also represent two typical chemical compound structures. More details of the data sets are given in Table 1.

Experimental settings. We perform multi-class node classification experiments on the four datasets. First, the embeddings of nodes are learned using gs2vec and the other baseline models. Then node embeddings become feature used to train supervised classifiers based on logistic regression with one-class-vs-the-rest classification. We randomly divide nodes into the training set and the testing set based on a certain ratio of training data (denoted as training ratio-TR). For both datasets, we vary TR from 10% to 90%, and repeat this process 10 times for each TR. Parameters for all methods set as follow: Embedding dimensions =128, number of walks per node= 40, walk length= 20, skip-gram window size= 10. For Metapath2vec, we choose all possible 2-hop paths in the datasets (like A-B-A) as the meta-paths.

Results. Table 2 reports the average performance measured in the multi-class node classification experiment by both Macro-F1 and Micro-F1 for the four datasets. Numbers in bold represent the best performer in each column.

For data sets BZR-MD-part1 and BZR-MD-part2, gs2vec outperforms benchmark methods on each training set. With 10% of all nodes as training data (TR=10%) of BZR-MD-part1 dataset, for example, gs2vec achieves improvements of 116-299% in Macro-F1 and 15-34% in MicroF1 over benchmarks，while for BZR-MD-part2 with 10% of nodes as training data, gs2vec achieves improvements of 94-353% over benchmarks in Macro-F1 and 13-41% in Micro-F1. As TR increases, our method's performance also improves as one would expect, and it still outperforms all benchmark methods.

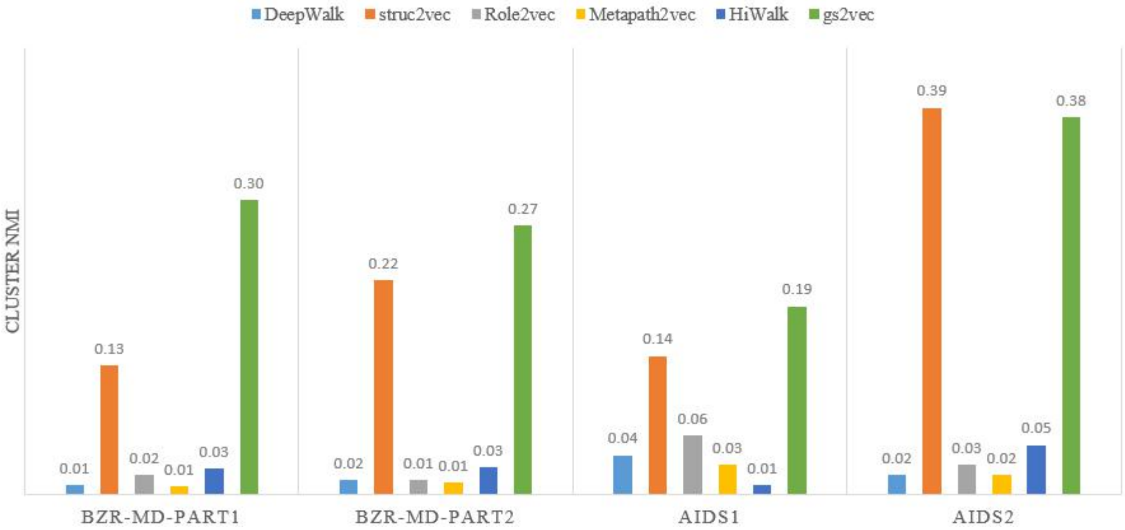
Results for data sets AIDS1 and AIDS2 are similar to the cases of BZR-MD-part1 and BZR-MD-part2, which gs2vec outperforms benchmark methods on each training set. With 10% of all nodes as training data (TR=10%) of AIDS1 dataset, gs2vec achieves improvements of 42-154% in Macro-F1 and 7-23% in Micro-F1 over benchmarks, while for AIDS2 with 10% training data, gs2vec achieves improvements of 5-338% over benchmarks in Macro-F1 and 0.6-66% in Micro-F1. The performance of the proposed method is consistent across different TR. Overall, gs2vec can improve node classification performance on all the four datasets.

In addition, methods designed specifically for heterogeneous networks, such as Metapath2vec [5] and HiWalk[6], and methods designed specifically for learning structural similarity in networks, such as Role2vec [3] and struc2vec [1], generally have better performance than DeepWalk [2], which was designed for homogeneous networks. This highlights the importance of considering structural heterogeneity when learning node representations. Meanwhile, gs2vec can outperform existing heterogeneous network representation methods because our method combines heterogeneity of nodes and edges with structural similarity beyond immediate neighborhood. Our method's superior performance over all benchmarks with low TRs is highly desirable when labeled data is scarce.

Node Clustering

We also demonstrate how node embeddings learned with gs2vec can help with node clustering in heterogeneous networks. We remove node label and examine if our clustering can put nodes with the same label in the same cluster. The embeddings learned by each method are input to a clustering model based on the classic k-means algorithm. The number of clusters is the same as the number of node types. Clustering results are evaluated with normalized mutual information (NMI). Figure 3 shows the clustering results for the four Datasets. With data sets BZR-MD-part1 and BZR-MD-part2, gs2vec outperforms all the benchmark methods by a large margin: improvements of 131%-2900% on BZR-MD-part1 and 23%-2600% on BZR-MD-part2. With data set AIDS1, gs2vec achieves improvements of 36%-1800%. With data set AIDS2, gs2vec achieves highly competitive, if not better, results with struc2vec. The experimental results show that gs2vec is effective in unsupervised network analysis tasks.

Figure 3. Node clustering results (measured by NMI) in the four data sets



CONCLUSION

In this paper, we propose gs2vec, a novel method to learn network representations for heterogeneous networks based on structural similarity at a global level. Experiment results show that gs2vec excels in capturing node structural characteristics from heterogeneous networks and has superior performance in node classification and node cluster tasks where node labels are more dependent on their structural roles or identities in a heterogeneous network. Our results highlight that structural roles of nodes have important implications in learning node representations for the heterogeneous network.

ACKNOWLEDGMENT

This work is supported in part by the Ministry of Science and Technology of China Major Grant #2016QY02D0305, NSFC Grant #71621002, 71902179 and CAS Key Grant #ZDRW-XH-2017-3. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Ministry of Science and Technology of China Major Grant, NSFC and CAS Key Grant.

■ REFERENCES

1. L. Ribeiro, P. Saverese, and D. Figueiredo. struc2vec: Learning node representations from structural identity. In KDD. 385–394,2017.
2. B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’14). ACM, New York, NY, USA, 701–710, 2014.
3. N.K. Ahmed, R. Rossi, J.B. Lee, T.L. Willke, and R. Zhou. Learning Role-based Graph Embeddings// Statistical Relational AI Workshop - IJCAI-18. 2018.
4. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013..
5. Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp.135–144, 2017.
6. J. Bai, L. Li, D. Zeng. HiWalk: Learning node embeddings from heterogeneous networks. Information Systems, 81:82-91, 2019.
7. X. Wang, H.Y. Ji, C. Shi, B. Wang, P. Cui, P. Yu, and Y.F. Ye. Heterogeneous Graph Attention Network. In Proceedings of WWW2019,.ACM, pp. 4-11,2019.
8. Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 119–128. ACM, 2015.
9. Ting Chen and Yizhou Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In WSDM ’17. ACM.
10. Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis, 2007, 11(5):561-580.
11. L. Van der Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research,9(2579-2605):85, 2008.
12. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Largescale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, pages 1067–1077. ACM, 2015.
13. A.R. Ryan and K.A. Nesreen. The network data repository with interactive graph analytics and visualization. In AAAI, pages 4292–4293, 2015.
14. Riesen K , Bunke H . IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning// Structural, Syntactic, and Statistical Pattern Recognition,

Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings. DBLP, 2008.

Wanting Wen is currently working toward the PhD degree at the Chinese Academy of Sciences, Beijing, China. She received her B.S. degree and M.S. degree in the Beijing University of Posts and Telecommunications. Her research interests include text mining, machine learning, and social computing. Contact her at wanting.wen@ia.ac.cn.

Daniel Zeng is a Research Professor with the Chinese Academy of Sciences, Beijing, China. His research interests include intelligence and security informatics, infectious disease informatics, social computing, recommender systems, software agents, spatial-temporal data analysis, and business analytics. He received the Ph.D. degree in industrial administration from Carnegie Mellon University, Pittsburgh, PA, USA in 1998. He has authored or coauthored one monograph and more than 330 peer-reviewed articles. He served as the Editor-in-Chief of IEEE INTELLIGENT SYSTEMS from 2013–2016 and currently serves as a President of the IEEE Intelligent Transportation Systems Society. Contact him at dajun.zeng@ia.ac.cn.

Jie Bai is an assistant research professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. She received her PhD degree in the Institute of Automation, Chinese Academy of Sciences. Her research interests include cognitive based text analysis, machine learning, and social computing. Contact her at baijie2013@ia.ac.cn.

Kang Zhao is an Associate Professor of Business Analytics, with a joint appointment in Informatics, at the University of Iowa. His current research focuses on data science and business intelligence, especially the mining, modeling, and simulation of social media, online communities, and social/business networks. He has published more than 30 journal papers and his research has been featured in public media from more than 25 countries, such as Washington Post, USA Today, Forbes, Yahoo News, New York Public Radio, BBC and Agence France-Presse. He served as the Chair for INFORMS Artificial Intelligence Section (2014-2016). Contact him at kang-zhao@uiowa.edu.

Ziqiang Li is an assistant engineer with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He received the master degree in management science and engineering from Beijing University of Chemical Technology in 2019. His research interests include bug location, recommender systems, and machine learning. Contact him at ziqiang.li@ia.ac.cn.