Semantic Graph Representation Learning in Attributed Networks

Meng Qin[1] , **Kai Lei**[1,2,*]

[1]ICNLAB, School of Electronics and Computer Engineering (SECE), Peking University
[2]PCL Research Center of Networks and Communications, Peng Cheng Laboratory

mengqin_az@foxmail.com, leik@pkusz.edu.cn
[*]Corresponding Author

Abstract

In this study, we focus on the graph representation learning task in attributed networks. Different from existing embedding methods that treat the incorporation of network structure and semantic as the simple combination of two optimization objectives, we propose a novel Semantic Graph Representation (SGR) model to formulate the joint optimization of the two heterogeneous sources into a common high-order proximity based framework. Concretely, we first construct a new abstracted weighted graph, where the complex (homogeneous and heterogeneous) relations among nodes and attributes (in the original network) are comprehensively encoded. Conventional embedding methods concerned topological proximity can then be easily applied to the newly constructed graph to learn the representations of both node and attribute while capturing the nonlinear high-order intrinsic correlation inside or among network structure and semantic. The learned attribute embeddings can also effectively support some semantic-oriented inference tasks (e.g., semantic community detection), helping to reveal the network's deep semantic. The effectiveness of SGR is further verified on a series of real networks, where it achieves impressive performance over other state-of-the-art competitors.

1 Introduction

Graph (network) is an effective model and data structure to describe the entities and relations of various complex systems (e.g., social networks, communication networks, etc). Graph representation learning (a.k.a. network embedding), which aims to encode the network into a low-dimensional representation with the primary properties preserved, has emerged as an important topic in the research of complex network analysis [Bandy *et al.*, 2018], due to its powerful ability to support the downstream network inference tasks (e.g., community detection, link prediction, etc.) [Cui *et al.*, 2018].

As reviewed in [Khosla *et al.*, 2019; Cui *et al.*, 2018], network structure (e.g., topology) is a significant information source available for most graph representation approaches. For instance, [Tang *et al.*, 2015] explored the observed network topology and the hidden neighborhood similarity (i.e., the first-order and second-order proximity) respectively in two optimization objectives, while the high-order proximities (i.e., neighbor structures with k-step random walk) of the network were comprehensively considered in [Perozzi *et al.*, 2014; Aditya Grover, 2016; Cao *et al.*, 2015]. Besides the microscopic structure (i.e., local neighbor proximity), the mesoscopic community structure [Wang *et al.*, 2017] can further help to reveal the function and organization of the network. Typical community-preserved representation methods include [Liang *et al.*, 2016; Wang *et al.*, 2017].

In this study, we focus on the graph representation learning in attributed networks, where network semantic (e.g., node attribute) is another significant heterogeneous information source. It's strongly believed that network attribute carries orthogonal and complementary knowledge beyond the topology [Jin *et al.*, 2018], which can potentially enhance learned representations and improve the performance of downstream network inference tasks. Several methods have been proposed to integrate such two heterogeneous sources, including the matrix factorization (MF) based methods [Cheng *et al.*, 2015; Bandy *et al.*, 2018; Huang *et al.*, 2017] and the deep learning based approaches [Jin *et al.*, 2018; Cao *et al.*, 2018]. Despite their effectiveness, there remain several limitations.

First, most hybrid methods treat the integration of network semantic as the auxiliary regularization (to existing models only consider topology) or simply combine the respective optimization objectives to learn features of the two sources, where the nonlinear high-order correlation between network structure and semantic is not fully explored. For example, if node v_i's neighbors $N(v_i)$ have more shared attributes $\{a_w\}$, v_i is more likely to be semantically similar with $N(v_i)$ even though it doesn't directly have all the attributes in $\{a_w\}$. Existing methods may fail to capture such high-order correlation between the two heterogeneous sources in a nonlinear manner, which can potentially lead to better performance for the downstream applications.

Moreover, the representations learned by most existing approaches with topology and attribute may only be capable to improve the performances of some simple network inferences (e.g., community detection), but cannot be directly applied to some advanced semantic-oriented downstream tasks, e.g., semantic community detection [Wang *et al.*, 2016b], where one

can obtain the corresponding semantic descriptions of each community simultaneously when the community partition is finished. For most existing embedding methods, additional efforts still need to be taken after the basic inference (e.g., community partition) to generate such descriptions (in order to explore the network's semantic) with the undesired loss of correlations between the two sources.

We introduce a novel Semantic Graph Representation (SGR) model to alleviate the aforementioned limitations, in which both the node and attribute (in original network G) are treated as the entities in a new abstracted weighted graph G'. Moreover, G' comprehensively encodes 3 types of relations (i.e., relations between (i) node pairs $\{(v_i, v_j)\}$, (ii) attribute pairs $\{(a_w, a_s)\}$ and (iii) heterogeneous entity pairs $\{(v_i, a_w)\}$). In this case, conventional high-order topology based embedding methods (e.g., DeepWalk [Perozzi *et al.*, 2014]) can be easily applied to G' to jointly learn the low-dimensional representations of both nodes $\{v_i\}$ and attributes $\{a_w\}$, where the nonlinear high-order intrinsic correlations among network topology and attribute are fully captured. Besides the node representations, the attribute embeddings learned by SGR can also be effectively used to support the semantic-oriented inference (e.g., semantic community detection), revealing the deep semantic of the network.

We summarize our main contributions as follow. **(i)** We formulate the graph representation learning in attributed networks as the embedding task of an abstracted weighted graph with heterogeneous entities, so that conventional high-order proximity based methods can be used to explore the network semantic. **(ii)** We proposed a novel SGR method, which can not only fully utilize the attribute information to improve the representation performance but can also generate the semantic descriptions to support the advanced semantic-oriented network inferences. **(iii)** An enhancement scheme based on graph regularization is also introduced for SGR to explore the effect of other side information (e.g., community structure). **(iv)** To verify the effectiveness of SGR, we conduct extensive experiments on a series of real networks, where SGR consistently outperforms other state-of-the-art approaches.

In the rest of this paper, we first give the formal problem definition regarding graph representation learning in Section 2, and then elaborate the SGR model in Section 3. The experiments are described in Section 5, which includes the performance evaluation on real network datasets and case study about the semantic description. Section 6 concludes this paper and indicates our future work.

2 Problem Definition

In this study, we generally consider the graph representation learning problem of undirected networks with discrete node attributes. Assume that there are n nodes and e edges in the network, and the total number of attributes is m. An attributed network can be formally described as a 4-tuple $G = (V, E, A, F)$, where $V = \{v_1, \cdots, v_n\}$ is the set of nodes; $E = \{(v_i, v_j) | v_i, v_j \in V\}$ is the set of edges; $A = \{a_1, \cdots, a_m\}$ is the set of attributes; $F = \{f(v_1), \cdots, f(v_n)\}$ represents the attribute map from V to A, with $f(v_i) \subset A$ as the attribute set of node v_i.

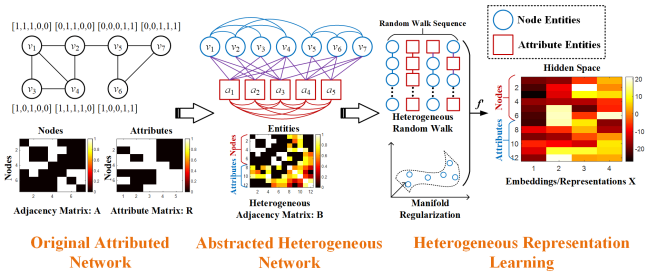


Figure 1: Sketch of the proposed SGR model, where we first (i) construct the abstracted heterogeneous graph according to original network's topology and attribute, and then (ii) use high-order proximity based embedding methods and manifold regularization techniques to learn the representations of both node and attribute.

Given G, the goal of the representation learning task in attributed networks is to learn a function $f : \{v_i\} \mapsto \{\mathbf{x}_i \in \Re^{1\times k}\}$ that maps each node v_i to a k-dimensional vector $\mathbf{x}_i$ (with $k \ll \min\{n, m\}$), in which the significant properties of network structure and semantic (hidden in E and F) are comprehensively preserved. Namely, the node pair (v_i, v_j) with similar properties (e.g., community membership, semantic, etc.) should have similar vector representations $(\mathbf{x}_i, \mathbf{x}_j)$.

Especially, we formulate the above representation learning problem as the embedding task of a corresponding abstracted weighted network G', where each node v_i and attribute a_j (in original network G) are represented as the heterogeneous entities (i.e., abstracted nodes). Concretely, we use a 2-tuple $G' = \{V', E'\}$ to described the abstracted weighted network, where $V' = V \cup A$ is the set of heterogeneous entities and $E' = \{E_1, E_2, E_3\}$ is the set of relations, with $E_1 = \{W(v_i, v_j) | v_i, v_j \in V\}$, $E_2 = \{W(v_i, a_w) | v_i \in V, a_w \in A\}$ and $E_3 = \{W(a_w, a_s) | a_w, a_s \in A\}$ as the set of weighted edges between (original) node pairs $\{(v_i, v_j)\}$, heterogeneous entity pairs $\{(v_i, a_w)\}$ and attribute pairs $\{(a_w, a_s)\}$, respectively. Based on G', the goal of SGR is to learn a function $f' : \{v_i, a_j\} \mapsto \{\mathbf{x'}_{v_i}, \mathbf{x'}_{a_j} \in \Re^{1\times k}\}$ that maps each node/attribute v_i/a_j into a k-dimensional hidden space represented by vector $\mathbf{x'}_{v_i}$/$\mathbf{x'}_{a_j}$, in which the primary properties of G' are comprehensively encoded. Given the learned representations $\{\mathbf{x'}_{v_i}, \mathbf{x'}_{a_i}\}$, one can treat $\{\mathbf{x'}_{v_i}\}$ as the final result (i.e., the node vectors) of the embedding task, while $\{\mathbf{x'}_{a_i}\}$ can also be utilized to generate the semantic descriptions for each node or (node) cluster by selecting the top nearest attribute entities of the specific node or cluster center in the mapped hidden space.

3 The Model

We propose a novel Semantic Graph Representation (SGR) method to tackle the embedding problem of attributed networks. For the convenience of discussion, we first present the sketch of the model in Figure 1 based on the problem definition in Section 2.

As illustrated in Figure 1, the key of SGR is to integrate 3 different types of relations (i.e., node relation E_1, heterogeneous relation E_2 and attribute relation E_3) into a unified weighted graph G' (i.e., construct the *heterogeneous adja-*

cency matrix), where the topology and semantic of the original attributed network G are comprehensively encoded in G''s weighted topology. The deep knowledge of G' is then fully explored and embedded in a low-dimensional hidden space by utilizing some random walk based representation methods and the graph regularization techniques on G'. We elaborate the details of SGR in the rest of this section.

Modeling the Node Relation. The node relation E_1 in G' represents the topological structure of G. Generally, it can be described by an *adjacency matrix* $\mathbf{A} \in \Re^{n \times n}$, where $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 1$ if there is an edge between the node pair (v_i, v_j) and $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 0$, otherwise. For SGR, we utilize $\mathbf{A}$ to represent the homogeneous relations among the nodes entities $\{v_i \in V\}$ (i.e., the node relations E_1) in G'.

Modeling the Attribute Relation. The attribute relation E_3 (i.e., the homogeneous relations among the attribute entities $\{a_w \in A\}$) in G' describe the similarity between each attribute pair (a_s, a_w). In general, the network semantic of G can be described by a *node attribute matrix* $\mathbf{R}_0 \in \Re^{n \times m}$, where $(\mathbf{R}_0)_{iw} = 1$ if node v_i's attribute set has a_w and $(\mathbf{R}_0)_{iw} = 0$, otherwise. $(\mathbf{R}_0)_{iw}$ can also be defined as the occurrence frequency or IF/IDF value of a_w in v_i's attribute set. Particularly, $(\mathbf{R}_0)_{:,w}$ describes the node membership of a certain attribute a_w (i.e., which nodes have a_w in their attribute sets). Hence, we define the similarity between each attribute pair (a_s, a_w) as the normalized similarity between their node memberships $((\mathbf{R}_0)_{:w}, (\mathbf{R}_0)_{:s})$ and introduce the *node similarity matrix* $\mathbf{P} \in \Re^{m \times m}$, where

$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{P}_0\mathbf{D}^{-1/2}, \tag{1}$$

with $(\mathbf{P}_0)_{ws} = [(\mathbf{R}_0)_{:,w}^T(\mathbf{R}_0)_{:,s}] \Big/ [|(\mathbf{R}_0)_{:,w}| \cdot |(\mathbf{R}_0)_{:,s}|]$ and $\mathbf{D} = \mathrm{diag}(\sum_{s=1}^{m}(\mathbf{P}_0)_{1s}, \cdots, \sum_{s=1}^{m}(\mathbf{P}_0)_{ms})$.

Note that there may exist the magnitude difference between $\mathbf{A}$ and $\mathbf{P}$ when integrating them into G', unfairly affecting the learned the heterogeneous representations. To eliminate such difference, we use the Max-Min normalization to rescale the elements in $\mathbf{P}$ into the range $[0, 1]$ (with the result notated as $\tilde{\mathbf{P}}$). Given a specific vector/matrix input $\mathbf{s}$ (e.g., $\mathbf{P}$), the Max-Min normalization (notated as MNorm) is defined as follow:

$$\tilde{\mathbf{s}} = \mathrm{MNorm}(\mathbf{s}_i) = (\mathbf{s}_i - \mathbf{s}_{\min})/(\mathbf{s}_{\max} - \mathbf{s}_{\min}), \tag{2}$$

where $\mathbf{s}_{\min}$ and $\mathbf{s}_{\max}$ are the minimum and maximum elements in $\mathbf{s}$, respectively.

Modeling the Heterogeneous Relation. In this study, we utilize motif [Benson *et al.*, 2016], a substructure that reveals the higher-order organization and function of the network, to represent the heterogeneous relation E_2 between each heterogeneous entity pair (v_i, a_w). Generally, we consider 3 basic motif instances $M_0 = \{(v_i, a_w)|v_i \in V, a_w \in A\}$, $M_1 = \{(v_i, a_w), (v_j, a_w)|v_i, v_j \in V, a_w \in A\}$ and $M_2 = \{(v_i, a_w), (v_i, a_s)|v_i \in V, a_w, a_s \in A\}$, which are illustrated in Figure 2 with circle and square representing node entity and attribute entity, respectively.

In the 3 motif instances, M_0 is the directly observable relation described by $\mathbf{R_0}$, while M_1 and M_2 represent the higher-order structures that encode the deep knowledge of the heterogeneous relation (e.g., nodes shared with more attributes should be more similar in their properties). To further describe the higher-order structure of M_1 and M_2, we

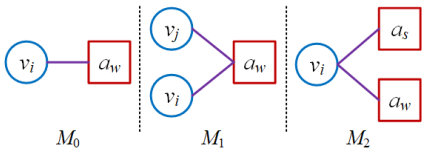


Figure 2: The entity motifs considered in this study with circle and square representing node entity and attribute entity, respectively.

introduce another two *relation matrices* $\mathbf{R}_1 \in \Re^{n \times m}$ and $\mathbf{R}_2 \in \Re^{n \times m}$, where $(\mathbf{R}_t)_{iw}$ is the co-occurrence counts of node v_i and attribute a_w in motif M_t ($t \in \{1, 2\}$) (i.e., the instance number of M_t containing (v_i, a_w)) or the cumulative sum of (v_i, a_w)'s weight (i.e., $(\mathbf{R}_0)_{iw}$) in M_t. In this case, $\mathbf{R}_1$ and $\mathbf{R}_2$ can be considered as the rescaling of $\mathbf{R}_0$'s observed relations, where relation (v_i, a_w) may have large weight $(\mathbf{R}_t)_{iw}$ if it can reflect the significant property of M_t.

We further conduct normalization on $\{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2\}$ (with results notated as $\{\tilde{\mathbf{R}}_0, \tilde{\mathbf{R}}_1, \tilde{\mathbf{R}}_2\}$), since there remains magnitude difference among them. Moreover, we use the combination of $\{\tilde{\mathbf{R}}_0, \tilde{\mathbf{R}}_1, \tilde{\mathbf{R}}_2\}$ to consider the comprehensive effect of different motifs by setting

$$\mathbf{R} = \delta_0\tilde{\mathbf{R}}_0 + \delta_1\tilde{\mathbf{R}}_1 + \delta_2\tilde{\mathbf{R}}_2, \tag{3}$$

with $\{\delta_0, \delta_1, \delta_2\}$ as parameters to control different components. We let $\delta_0 = \delta_1 = \delta_2 = 1$ as the default setting, where $\{M_0, M_1, M_2\}$ have the same contribution in (3) without specific prior knowledge. Better performance can be achieved by fine-tuning $\delta_0, \delta_1, \delta_2 \in \{0, 1\}$ according to our experiments (see Section 4.1), representing the introduction of other auxiliary priors. Similar to $\mathbf{P}$, we also conducted another normalization on $\mathbf{R}$ to further eliminate the magnitude difference between $\mathbf{R}$ and $\{\mathbf{A}, \tilde{\mathbf{P}}\}$ (with result notated as $\tilde{\mathbf{R}}$).

The Unified Model. We can construct the abstracted graph G' with the weighted topology described by the following *heterogeneous adjacency matrix* $\mathbf{B} \in \Re^{(n+m)\times(n+m)}$:

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & \tilde{\mathbf{R}} \\ \tilde{\mathbf{R}}^T & \tilde{\mathbf{P}} \end{bmatrix}, \tag{4}$$

where $\{\mathbf{A}, \tilde{\mathbf{P}}, \tilde{\mathbf{R}}\}$ are considered as different blocks of $\mathbf{B}$, and $\mathbf{B}_{ij}$ is the edge weight of entity pair (e_i, e_j) ($e_i, e_j \in V \cup A$). In this case, conventional high-order topological proximity based graph representation methods (e.g., DeepWalk [Perozzi *et al.*, 2014]) can be easily applied to G' to fully explore both the information sources of network structure and semantic.

In this study, we adopt the following equivalent matrix factorization (MF) objective of DeepWalk [Qiu *et al.*, 2018] as an example to derive the embeddings of SGR:

$$\mathbf{Z} = \log(\mathrm{vol}(G') \cdot (\frac{1}{o}\sum\nolimits_{r=1}^{o}(\mathbf{D}^{-1}\mathbf{B})^r)\mathbf{D}^{-1}) - \log b, \tag{5}$$

where o is the content window size (i.e, step/order of random walk); b is the number of negative sampling; $\mathbf{D} = \mathrm{diag}(d_1, d_2, \cdots, d_{n+m})$ is the degree diagonal matrix with $d_i = \sum_{j=1}^{n+m}\mathbf{B}_{ij}$ as the degree of entity e_i; $\mathrm{vol}(G') = \sum_{i=1}^{n+m} d_i$ is the volume of G'. Based on $\mathbf{Z}$, the graph representation learning task can then be generally represented as

Algorithm 1: Semantic Graph Representation (SRG)

Input: $\mathbf{A}, \mathbf{R}_0$
Output: $\{\mathbf{X}^*, \mathbf{Y}^*\}$

construct *node similarity matrix* $\mathbf{P}$ via (1)
normalize $\mathbf{P}$ via (2) (with the result notated as $\tilde{\mathbf{P}}$)
construct *relation matrices* $\{\mathbf{R}_1, \mathbf{R}_2\}$ according to motifs $\{M_1, M_2\}$
normalize $\{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2\}$ via (2) (with the result notated as $\{\tilde{\mathbf{R}}_0, \tilde{\mathbf{R}}_1, \tilde{\mathbf{R}}_2\}$)
construct *relation matrix* $\mathbf{R}$ via (3)
normalize $\mathbf{R}$ via (2) (with the result notated as $\tilde{\mathbf{R}}$)
construct *heterogeneous adjacency matrix* $\mathbf{B}$ via (4)
construct the optimization objective (5)
get the solution of (6) (i.e., $\{\mathbf{X}^*, \mathbf{Y}^*\}$) by using SVD (i.e., (8))

the following MF-based optimization problem:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \left\|\mathbf{Z} - \mathbf{X}\mathbf{Y}^T\right\|_F^2, \tag{6}$$

with $\mathbf{X} \in \Re^{(n+m)\times k}$ and $\mathbf{Y} \in R^{(n+m)\times k}$ as two low-dimensional matrices. The singular value decomposition (SVD) techniques can be utilized to get the optimal solution of (6), which is defined as follow:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \approx \mathbf{U}_{:,1:k}\mathbf{\Sigma}_k\mathbf{V}_{:,1:k}^T. \tag{7}$$

In (7), $\mathbf{\Sigma} = diag(\theta_1, \theta_2, \cdots, \theta_{n+m})$ is the diagonal matrix of singular values with $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_{n+m}$. Particularly, we use the top-k singular values to approximatively reconstruct $\mathbf{Z}$, so the solution of (6) can be derived by setting

$$\mathbf{X}^* = \mathbf{U}_{:,1:k}\sqrt{\mathbf{\Sigma}_k},\ \mathbf{Y}^* = \mathbf{V}_{:,1:k}\sqrt{\mathbf{\Sigma}_k}, \tag{8}$$

where we adopt $\mathbf{X}^*$ as the final representation result of SGR. In summary, we conclude the above process in Algorithm 1.

Enhancement of Side Information. Besides the observed network topology and attribute (described by $\{\mathbf{A}, \tilde{\mathbf{R}}\}$), some other latent side information (e.g., community structure) can be utilized to further enhance the representations learned by SGR, potentially resulting in better performance for the downstream network inference applications. For the convenience of discussion, we call such effect as side-enhancement in the rest of this paper.

We use the graph regularization framework to leverage the side information based on the objective (6). For a certain information source (notated as I_l), the corresponding regularization term is defined as follow:

$$\mathrm{Reg}_l(\mathbf{X}, \mathbf{T}_l) = \frac{1}{2}\sum_{i,j=1}^{n} (\mathbf{T}_l)_{ij} \left\|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\right\|_2^2 = \mathrm{tr}(\mathbf{X}^T\mathbf{L}_l\mathbf{X}), \tag{9}$$

where $\mathbf{T}_l \in \Re^{(n+m)\times(n+m)}$ is the matrix encoding the primary properties of I_l and $\mathbf{L}_l = (\mathbf{D}_l - \mathbf{T}_l)$ is $\mathbf{T}_l$'s Laplacian matrix with $\mathbf{D}_l = \mathrm{diag}(\sum_{j=1}^{n+m}(T_l)_{1j}, \sum_{j=1}^{n+m}(T_l)_{2j}, \cdots)$. In fact, (9) can be considered as the penalty given by I_l, in which $\{\mathbf{X}_{i,:}, \mathbf{X}_{j,:}\}$ are regularized to have similar representations if $(\mathbf{T}_l)_{ij}$ has a relatively large value.

In this study, *community structure* and *attribute similarity* are adopted as two available sources of side information.

We utilize the *modularity matrix* $\mathbf{Q} \in \Re^{n\times n}$ [Jin *et al.*, 2018] to represent the community structure of G, where

$$\mathbf{Q}_{ij} = \mathbf{Q}_{ji} = \mathrm{A}_{ij} - d_i d_j/(2e), \tag{10}$$

with $d_i = \sum_{j=1}^{n}\mathbf{A}_{ij}$ as the degree of node v_i and e as the number of edges in G. $\mathbf{Q}$ encodes the primary properties of G's community structure by measuring the difference between the exact edge numbers and the expected number of such edges over all node pairs. Concretely, $\mathbf{Q}_{ij}$ ($\mathbf{Q}_{ji}$) with larger value indicates that edge (v_i, v_j) is more likely to be preserved in a certain community (but not to be cut) when conducting a graph-cut (node clustering) process.

Furthermore, we utilize the cosine similarity between the attribute lists of each node pair (v_i, v_j) to describe the *attribute similarity* of G, where we introduce the *attribute similarity matrix* $\mathbf{S} \in R^{n\times n}$ with

$$\mathbf{S}_{ij} = \mathbf{S}_{ji} = [(\mathbf{R}_0)_{i,:}(\mathbf{R}_0)_{j,:}^T] \Big/ [|(\mathbf{R}_0)_{i,:}| \cdot |(\mathbf{R}_0)_{j,:}|]. \tag{11}$$

The overall optimization objective of side-enhancement can then be generally formulated as follow:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \mathrm{O}(\mathbf{X}, \mathbf{Y}) = \left\|\mathbf{Z} - \mathbf{X}\mathbf{Y}^T\right\|_F^2 + \sum_{l=1}^{L}\lambda_l \mathrm{Reg}_l(\mathbf{X}, \mathbf{T}_l), \tag{12}$$

where λ_l is the parameter to adjust the effect of the l-th side information. In this study, we set $L = 2$ and let

$$\mathbf{T}_1 = \begin{bmatrix} \tilde{\mathbf{Q}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix} \tilde{\mathbf{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{13}$$

with $\{\tilde{\mathbf{S}}, \tilde{\mathbf{Q}}\}$ as the normalized results of $\{\mathbf{S}, \mathbf{Q}\}$ via (4).

To obtain the solution of (12) (notated as $\{\mathbf{X}'^*, \mathbf{Y}'^*\}$) in a relatively fast way, we first use (6)'s result (i.e., (8)) to initialize $\{\mathbf{X}, \mathbf{Y}\}$, and use certain rules to update their values.

For $\mathbf{X}$, we first derive the partial derivative of $\mathrm{O}(\mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{X}$:

$$\partial \mathrm{O}(\mathbf{X}, \mathbf{Y})/\partial \mathbf{X} = 2(\mathbf{X}\mathbf{Y}^T\mathbf{Y} - \mathbf{Z}\mathbf{Y} + \mathbf{L}\mathbf{X}), \tag{14}$$

with $\mathbf{L} = \sum_{l=1}^{L}\lambda_l\mathbf{L}_l$. By setting $\partial \mathrm{O}(\mathbf{X}, \mathbf{Y})/\partial \mathbf{X} = 0$, we have

$$\mathbf{X}'^* = (\mathbf{I}_n + \mathbf{L})^\dagger \mathbf{Z}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y} + \mathbf{I}_k)^\dagger, \tag{15}$$

in which $\mathbf{M}^\dagger$ denotes the pseudo-inverse of matrix $\mathbf{M}$, while $\mathbf{I}_n$ represents an n-dimensional identity matrix.

With regard to $\mathbf{Y}$, we first derive $\mathrm{O}(\mathbf{X}, \mathbf{Y})$'s partial derivative with respect to the $\mathbf{Y}$:

$$\partial \mathrm{O}(\mathbf{X}, \mathbf{Y})/\partial \mathbf{Y} = 2(\mathbf{Y}\mathbf{X}^T\mathbf{X} - \mathbf{Z}^T\mathbf{X}). \tag{16}$$

Similarly, we can obtain $\mathbf{Y}$'s updating rule by setting $\partial \mathrm{O}(\mathbf{X}, \mathbf{Y})/\partial \mathbf{Y} = 0$:

$$\mathbf{Y}'^* = (\mathbf{Z}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^\dagger. \tag{17}$$

In general, the solution $\{\mathbf{X}'^*, \mathbf{Y}'^*\}$ can be obtained by continuously update $\{\mathbf{X}, \mathbf{Y}\}$ via (15) and (17) (after initialization) until converge. According to our pre-experiments, the relative error of objective function (12) (with respect to previous iteration) is less than 10^{-6} just after the first iteration, and (14)'s value keep stables in the subsequent iterations on most real networks. Hence, we just use (15) and (17) to update $\{\mathbf{X}, \mathbf{Y}\}$ once after initialization to get the solution in a fast way. Similar to (8), we utilize $\mathbf{X}'^*$ as the final embedding result. As a conclusion, we summarize the overall process of side-enhancement in Algorithm 2.

Algorithm 2: Side-Enhancement of SGR

Input: $\mathbf{A}, \mathbf{R}_0, \{\mathbf{X}^*, \mathbf{Y}^*\}$
Output: $\{\mathbf{X}'^*, \mathbf{Y}'^*\}$

construct the *modularity matrix* $\mathbf{Q}$ via (10)
normalize $\mathbf{Q}$ via (2) (with the result notated as $\tilde{\mathbf{Q}}$)
construct the *attribute similarity matrix* $\mathbf{S}$ via (11)
normalize $\mathbf{S}'$ via (2) (with the result notated as $\tilde{\mathbf{S}}'$)
construct $\{\mathbf{T}_1, \mathbf{T}_2\}$ via (13) (according to $\{\tilde{\mathbf{S}}', \tilde{\mathbf{Q}}\}$)
use the result of (6) (i.e., (8)) to initialize $\{\mathbf{X}, \mathbf{Y}\}$
update $\mathbf{X}$ via (15) (with the result notated as $\mathbf{X}'^*$)
update $\mathbf{Y}$ via (17) (with the result notated as $\mathbf{Y}'^*$)

Table 1: Statistics details of the real attributed networks.

Datasets	*N*	*E*	*M*	*C*	Datasets	*N*	*E*	*M*	*C*
Cornell(CO)	195	283	1,588	5	**Gplus(GP)**	700	28,055	887	4
Texas(TE)	185	280	1,501	5	**Cora**	2,708	5,278	1,432	7
Washington(WA)	217	366	1,578	5	**Citeseer(Cite)**	3,264	4,598	3,703	6
Wisconsin(WI)	262	459	1,623	5	**UAI2010(UAI)**	3,061	28,308	4,973	19
Twitter(TW)	155	3,442	1,470	7	**BlogCatalog(BL)**	5,196	171,743	8,189	6
Facebook(FA)	475	10,066	507	9	**Flikr(FL)**	7,575	239,738	12,047	9

4 Experimental Evaluation

4.1 Real Network Evaluation

The Datasets. To verify the effectiveness of SRG, we applied it to 12 real attributed networks. The statistics details of the datasets after necessary pre-processing are shown in Table 1, where n, e, m and c are the number of nodes, edges, (node) attributes and clusters/categories, respectively.

Cornell (CO), *Texas* (TE), *Washington* (WA) and *Wisconsin* (WI) are 4 sub-networks of the WebKB dataset[1], which contains the hyperlinks and content of the web pages collected from the computer science departments in 4 American universities. *Twitter* (TW), *Facebook* (FA) and *Gplus* (GP) are respectively the subsets of the attributed ego-networks (a.k.a. social circles) Twitter[2], Facebook[3] and Google+[4] in the Stanford Network Analysis Project (SNAP). *Cora*[5] [Sen *et al.*, 2008] and *Citeseer*[6] (Cite) [Sen *et al.*, 2008] are 2 science publication networks with the citation relations and paper content, while *UAI2010* (UAI) [Sen *et al.*, 2008] is a Wikipedia article citation network including the reference relations and feature lists. *BlogCatalog*[7] (BL) [Huang *et al.*, 2017] is a dataset collected from the blogger community BlogCatalog[8] containing the interactive relations and interest tags of users. *Flickr*[9] (FL) [Huang *et al.*, 2017] is a social network of the online photo sharing platform Flickr[10], which includes the friend relation among the users and the photos tags of each node.

Baseline Methods. We utilized 11 state-of-the-art graph representation approaches as the baselines, which can be classified into 3 different types. First, *DeepWalk*[11] (DW) [Perozzi *et al.*, 2014], *node2vec*[12] (N2V) [Aditya Grover, 2016], *LINE*[13] [Tang *et al.*, 2015], *SDNE*[14] [Wang *et al.*, 2016a], *GraRep*[15] [Cao *et al.*, 2015] and *AROPE*[16] [Zhang *et al.*, 2018] are methods that explore the high-order proximity of the network topology. Second, *DNR* [Liang *et al.*, 2016] and *M-NMF*[17] [Wang *et al.*, 2017] are approaches that integrate the community structure of networks. Moreover, *TADW*[18] [Cheng *et al.*, 2015], *AANE*[19] [Huang *et al.*, 2017] and *FSC-NMF*[20] (FSC) [Bandy *et al.*, 2018] are the embedding methods which incorporate the network structure and attribute.

To ensure the fairness of comparison, we set the dimension of the representation vector to be 64 (i.e., $k = 64$) for all the methods to be evaluated. Furthermore, we utilized the open source implementation given by the authors for each competitor and adopted its default parameter setting.

With regard to SGR, we adopt *community structure* and *attribute similarity* (represented by $\mathbf{Q}$ and $\mathbf{S}$, respectively) as the available side information (see Section 3). Also, we utilize motifs $\{M_0, M_1, M_2\}$ (see Figure 2) to formulate the heterogeneous relations $E_2 = \{(v_i, a_w) \,|\, v_i \in V, a_w \in A\}$. To effectively illustrate the effect of the hyper-parameters (i.e, $\{\delta_1, \delta_2, \delta_3\}$, $\{\lambda_1, \lambda_2\}$) and the side-enhancement effect, we respectively recorded the evaluation metrics with (i) the default parameter setting (i.e., δ_0=δ_1=δ_2=1), (ii) the fined-tuned parameters (i.e., adjust $\delta_0, \delta_1, \delta_2 \in \{0, 1\}$) and (iii) the side-enhancement (i.e., adjust $\lambda_1, \lambda_2 \in \{0, 1\}$). Corresponding results are notated as SRG(0), SRG(1) and SRG(R).

Performance Evaluation. In the evaluation, we adopted node clustering (a.k.a. community detection) and node classification as the testing downstream applications.

For node clustering, we applied the k-means++ algorithm to the representations learned by all the methods and utilized the normalized mutual information (NMI) [Cao *et al.*, 2018] as well as accuracy (AC) [Cao *et al.*, 2018] as the evaluation metrics. Moreover, we set the number of clusters in k-means++ according to the ground-truth of each dataset. With regard to node classification, we utilized the support vector machine (SVM) implemented by the LibLinear package[21] [Fan *et al.*, 2008] as the downstream classifier. For each dataset, we randomly select 10% of the nodes to train the classifier, with the rest nodes as the test data. Accuracy (AC) [Cui *et al.*, 2018] and Macro-F1 [Cui *et al.*, 2018] are adopted as the quality metrics.

Both the clustering and classification process were repeated 100 times for each method and dataset. The average results in terms of NMI, (clustering) AC, (classification) AC and Macro-F1 are shown in Table 2, 3, 4 and 5 respec-

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/
[2] http://snap.stanford.edu/data/ego-Twitter.html
[3] http://snap.stanford.edu/data/ego-Facebook.html
[4] http://snap.stanford.edu/data/ego-Gplus.html
[5] http://www.cs.umd.edu/ sen/lbc-proj/data/cora.tgz
[6] http://www.cs.umd.edu/ sen/lbc-proj/data/citeseer.tgz
[7] http://github.com/xhuang31/AANE_MATLAB/blob/master/BlogCatalog.mat
[8] http://www.blogcatalog.com
[9] http://github.com/xhuang31/AANE_MATLAB/blob/master/Flickr.mat
[10] https://www.flickr.com
[11] https://github.com/phanein/deepwalk
[12] https://github.com/adityagrover/node2vec
[13] https://github.com/tangjianpku/LINE
[14] https://github.com/shenweichen/GraphEmbedding
[15] https://github.com/ShelsonCao/GraRep
[16] https://github.com/ZW-ZHANG/AROPE
[17] https://github.com/thumanlab/M-NMF
[18] https://github.com/albertyang33/TADW
[19] https://github.com/xhuang31/AANE_MATLAB
[20] https://github.com/benedekrozemberczki/FSCNMF
[21] https://www.csie.ntu.edu.tw/c̆jlin/liblinear

Table 2: Evaluation result of node clustering in terms of NMI(%)

	CO	TE	WA	WI	TW	FA	GP	Cora	Cite	UAI	BL	FL
DW	6.79	5.79	7.22	7.41	33.98	58.37	32.38	36.91	14.20	33.38	19.22	16.64
N2V	6.56	5.55	6.13	6.81	32.34	56.49	33.12	40.67	21.03	34.31	20.42	17.27
LINE	12.24	18.56	21.19	10.79	35.18	42.84	34.97	25.08	10.80	12.12	4.10	0.65
SDNE	13.78	16.85	24.42	8.98	28.02	28.22	27.73	10.96	4.04	11.44	9.62	3.96
GraRep	8.64	12.12	5.03	8.41	34.34	53.80	38.92	36.66	11.54	33.83	22.08	16.39
AROPE	9.08	10.29	9.63	6.36	29.32	25.42	19.45	8.85	4.54	13.63	14.37	8.40
DNR	11.91	18.70	24.29	9.46	32.45	31.29	25.47	16.09	6.48	5.66	13.28	4.42
MNMF	12.90	18.07	22.88	8.94	32.86	40.53	40.38	10.34	5.18	19.36	17.25	14.76
TADW	11.49	8.45	10.67	13.15	25.34	46.34	5.56	29.71	19.29	25.33	7.85	2.27
AANE	30.63	30.32	38.11	40.91	33.72	49.76	37.72	17.78	19.82	41.18	28.26	39.30
FSC	10.67	14.72	11.25	15.06	9.94	29.71	25.69	13.89	18.83	44.24	1.46	0.37
SGR(0)	31.67	33.97	40.23	40.29	36.75	**61.45**	20.01	49.33	**39.62**	47.70	31.87	20.66
SGR(1)	35.59	36.57	45.19	41.63	**37.06**	-	-	**50.09**	-	**47.97**	**32.80**	20.47
SGR(R)	**36.80**	**38.59**	**46.78**	**45.72**	31.93	-	**54.35**	-	38.48	-	-	**55.20**

Table 3: Evaluation result of node clustering in terms of AC(%)

	CO	TE	WA	WI	TW	FA	GP	Cora	Cite	UAI	BL	FL
DW	38.31	50.37	44.60	43.65	41.89	68.53	56.96	51.55	40.79	36.49	35.44	30.87
N2V	37.50	47.02	40.97	38.92	36.74	57.93	54.53	54.95	44.00	37.69	36.81	31.42
LINE	38.98	54.79	56.16	43.74	42.36	37.58	56.58	42.37	26.96	16.49	25.24	13.09
SDNE	42.12	54.95	62.35	47.43	36.06	26.83	55.27	31.25	22.51	19.17	26.88	15.52
GraRep	32.25	35.39	31.25	33.21	44.79	51.90	53.18	50.43	33.17	37.47	38.79	29.18
AROPE	42.95	56.02	48.91	46.51	37.63	27.58	55.87	32.68	23.05	21.24	28.18	18.27
DNR	37.59	52.42	55.34	42.71	44.57	31.57	53.31	33.56	23.67	12.89	32.57	18.52
MNMF	39.26	54.82	60.19	45.98	40.02	35.81	54.28	32.44	23.30	24.27	33.90	28.35
TADW	47.79	57.63	50.71	50.34	43.81	56.68	43.37	46.32	38.26	28.01	23.26	14.15
AANE	51.49	53.76	52.76	59.23	43.48	69.88	65.34	36.44	43.07	40.72	**45.14**	38.57
FSC	45.73	58.41	50.62	51.95	35.52	41.37	60.55	35.14	41.38	41.15	18.92	11.81
SGR(0)	54.56	59.16	66.00	64.87	45.41	**71.37**	54.64	60.92	**62.78**	45.61	41.85	33.56
SGR(1)	56.25	**63.50**	66.65	60.32	46.95	-	-	**62.44**	-	**46.04**	43.10	33.43
SGR(R)	**59.11**	60.97	**71.41**	**70.85**	**52.68**	-	**83.22**	-	59.92	-	-	**61.91**

Table 4: Evaluation result of node classification in terms of AC(%)

	CO	TE	WA	WI	TW	FA	GP	Cora	Cite	UAI	BL	FL
DW	32.31	47.34	38.85	40.66	48.42	84.44	85.91	68.24	45.32	45.71	59.63	44.54
N2V	31.85	47.40	39.83	40.60	48.54	85.38	87.10	72.53	50.18	48.50	59.94	45.99
LINE	37.19	57.39	53.07	48.64	46.88	83.56	90.08	71.06	44.90	35.05	32.35	12.39
SDNE	38.01	56.62	59.36	48.25	45.22	75.04	89.32	38.66	23.35	26.55	56.37	31.73
GraRep	36.00	50.99	41.74	46.03	**51.12**	84.98	88.45	73.96	48.75	52.61	65.87	50.24
AROPE	37.69	55.37	50.50	46.30	48.67	81.78	93.90	65.14	43.13	45.42	67.12	57.17
DNR	38.46	57.37	58.63	50.24	46.88	72.09	91.88	44.49	27.31	17.90	40.79	17.80
MNMF	37.94	57.85	46.86	46.67	50.35	83.56	91.03	69.57	46.94	45.26	66.45	53.29
TADW	45.07	52.14	48.73	50.42	46.74	79.57	83.77	67.26	56.89	55.43	89.76	56.65
AANE	**60.91**	65.01	69.33	72.34	37.18	71.55	84.85	72.62	65.60	61.29	82.19	85.97
FSC	54.14	65.84	64.14	68.11	41.63	72.45	81.85	61.01	62.57	69.86	73.11	49.45
SGR(0)	55.61	63.89	67.11	69.26	48.69	**85.76**	**94.74**	80.03	67.62	70.72	90.11	84.25
SGR(1)	57.36	65.07	**71.20**	72.10	50.73	-	-	**80.29**	-	**70.91**	**90.18**	84.36
SGR(R)	58.82	**71.17**	69.64	**74.14**	50.82	-	90.17	-	**71.31**	-	-	**89.02**

Table 5: Evaluation result of node classification in terms of F1(%)

	CO	TE	WA	WI	TW	FA	GP	Cora	Cite	UAI	BL	FL
DW	20.05	21.15	20.79	24.22	25.87	54.86	53.69	66.86	41.95	37.68	59.11	43.61
N2V	19.55	20.32	21.03	24.61	18.29	54.46	51.44	70.82	45.96	39.42	59.30	44.71
LINE	24.10	27.36	28.17	28.72	25.18	54.19	55.85	69.20	40.51	24.33	27.89	10.42
SDNE	22.63	25.69	30.08	26.02	24.56	44.88	64.81	22.41	10.32	20.18	55.63	27.59
GraRep	25.48	29.50	24.48	28.76	**27.41**	54.97	59.48	72.76	45.20	43.10	65.22	49.73
AROPE	24.44	28.89	26.89	28.33	25.74	46.28	71.52	63.47	39.36	35.09	65.79	55.58
DNR	19.18	25.70	28.32	22.59	24.57	30.01	63.09	29.26	17.28	7.71	39.61	13.40
MNMF	23.20	30.50	24.77	27.94	27.10	54.89	65.39	67.20	42.81	35.73	65.54	52.46
TADW	29.02	21.89	27.00	29.56	26.60	45.90	50.15	65.02	52.79	43.87	89.59	55.90
AANE	39.95	31.50	36.68	44.02	11.29	26.53	43.91	69.16	59.12	40.99	81.75	85.63
FSC	36.18	36.01	38.78	44.64	20.67	28.58	42.86	58.29	55.95	55.21	72.61	46.16
SGR(0)	38.58	37.77	40.33	43.41	25.13	**57.83**	**73.38**	78.54	62.52	57.85	89.96	84.06
SGR(1)	40.08	38.40	42.42	46.94	25.99	-	-	**78.74**	-	**58.24**	**90.04**	84.16
SGR(R)	**44.34**	**45.86**	**44.43**	**52.78**	27.12	-	66.25	-	**63.05**	-	-	**88.64**

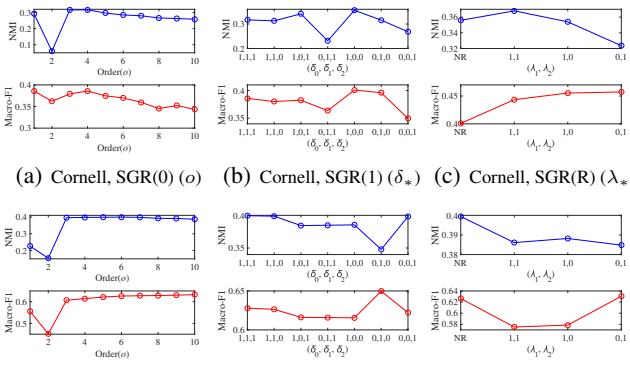
tively, where the best metric is in **bold** and the second-best is underlined. Especially, '-' denotes there is no performance improvement (for the parameter adjustment or side-enhancement) in comparison with the basic version of SGR.

For the application of node clustering, SGR (including SGR(0), SGR(1) and SGR(R)) has the best performance on all the 12 datasets in terms of NMI and outperforms other baselines on 11 datasets in terms of AC (with the average improvement of **25.31**% and **21.22**% compared to the second-best baselines). With regard to node classification, SGR achieves the best performance on 10 and 11 datasets for the metrics of AC and Macro-F1, respectively (with the average improvement of **3.74**% and **9.39**% compared to the second-best baselines).

In comparison with SGR(0) and SGR(1), the performance of both node clustering and classification can be further improved on most of the datasets via the side-enhancement (i.e., SGR(R)), but such improvement cannot be ensured for all the datasets (e.g., *Facebook*, *UAI2010*, *Cora* and *BlogCatalog*). It can be reasonably interpreted that the incorporation of the side information (e.g., community structure) may not only bring complementary knowledge of the network, but also introduce inconsistent features or noise (relative to the application's ground-truth) into the learned representations, affecting the performance of the downstream application. We intend to further explore such effect in our future work.

In the experiments, we first fixed $\delta_1 = \delta_2 = \delta_3 = 1$ and adjusted the proximity order $o \in \{1, 2, \cdots, 10\}$ for each dataset, with the best metric reported for SGR(0). Based on the selected setting of o, we tuned $\delta_0, \delta_1, \delta_2 \in \{0, 1\}$ for SGR(1). Moreover, the side-information is further incorporated (based on the parameter setting of SGR(0) and SGR(1)) by adjusting $\lambda_1, \lambda_2 \in \{0, 1\}$ for SGR(R). Due to the space limit, we demonstrate the parameter adjustment (in terms of NMI and Macro-F1 for node clustering and classification) of *Cornell* and *Citeseer* as 2 examples in Figure 3, where 'NR' represents the baseline performance of SGR(1) (compared with SGR(R)).

According to Figure 3 and our records, different settings of o may result in significantly different performances for a certain dataset, indicating that the order of random walk on G' is a primary factor that affects the incorporation of network topology and semantic. Moreover, the best performance in one application (e.g., node clustering or classification) doesn't mean the best result in other applications with respect to a certain parameter setting (e.g., $(\delta_0, \delta_1, \delta_2) = (0, 1, 0)$ for *Citeseer* and $(\lambda_1, \lambda_2) = (1, 1)$ for *Cornell*). For each dataset, we determine the best parameter setting by comprehensively considering the performances (or performance improvement)



(a) Cornell, SGR(0) (o) (b) Cornell, SGR(1) (δ_*) (c) Cornell, SGR(R) (λ_*)

(d) Citeseer, SGR(0) (o) (e) Citeseer, SGR(1) (δ_*) (f) Citeseer, SGR(R) (λ_*)

Figure 3: Parametric Analysis on *Cornell* and *Citeseer*

Table 6: Average Running Time(s) of SGR

	CO	TE	WA	WI	TW	FA	GP	Cora	Cite	UAI	BL	FL
SGR	1.57	1.34	1.61	1.81	1.20	0.35	1.19	16.39	66.46	100.06	462.93	1259.32
SGR(R)	2.46	1.89	3.22	4.26	3.12	0.51	1.62	27.87	118.40	270.37	1146.97	9997.84

of both node clustering and classification. Although it's hard to find a fixed parameter setting that can ensure the best performance for all the datasets, we recommend to set $o = 4$ and $\delta_0{=}\delta_1{=}\delta_2{=}1$, in which SGR can achieve a relatively high performance (despite not the best) for most datasets. Better result can be obtained by fine-tuning $\delta_0, \delta_1, \delta_2 \in \{0, 1\}$ and utilizing the side-enhancement (i.e., adjust $\lambda_1, \lambda_2 \in \{0, 1\}$).

We used MATLAB to implement SGR. The average running time with and without the side-enhancement on a server (Intel Xeon CPU E5-2680 v4 @2.40GHz and 32GB main memory) are shown in Table 6. To further speed up the computation, some distributed fast SVD approaches (e.g., [Iwen and Ong, 2016]) and optimized matrix operation libraries (e.g., OpenBLAS[22]) can be utilized. We intend to consider such improvement in our future work.

4.2 Case Study

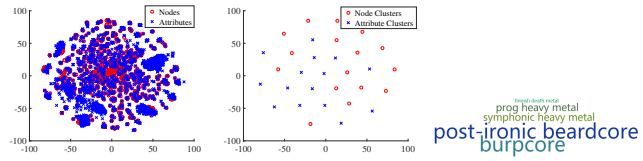
We utilized the *LastFM* dataset[23] [Cantador *et al.*, 2011] to illustrate SGR's ability to generate the semantic description of network and adopted semantic community detection [Wang *et al.*, 2016b] as the testing application, where we can generate one or more wordclouds for each community when the community partition is finished. The dataset was collected from the online music platform last.fm[24], including the friend relations and interest tags of the users. After pre-processing, the dataset contains $1,892$ users (nodes), $12,717$ friend relations (edges) and $9,749$ tags (attributes).

In the experiments, we obtained the node embeddings and attribute embeddings (notated as $\{\mathbf{x'}_{v_i}\}$ and $\{\mathbf{x'}_{a_w}\}$) by setting $o = 4$, $\delta_0{=}\delta_1{=}\delta_2{=}1$ for the basic version of SGR (i.e., SGR(0)). We further utilized t-SNE [Der Maaten and Hinton, 2008] to map the representations $\{\mathbf{x'}_{v_i}, \mathbf{x'}_{a_w}\}$ into a 2D space with the result visualized in Figure 4 (a), where $\{x'_{v_i}, x'_{a_w}\}$ are well mapped into a common hidden space, indicating the comparability of the heterogeneous embeddings. Namely, the distance between $(\mathbf{x'}_{v_i}, \mathbf{x'}_{a_w})$ can be used to measure the property similarity between node v_i and attribute a_w. Furthermore, we applied the advanced X-means algorithm [Ishioka, 2000] to determine the proper number of clusters and the corresponding clustering membership respectively for $\{\mathbf{x'}_{v_i}\}$ and $\{\mathbf{x'}_{a_w}\}$. Finally, we set the number of topology clusters and attribute clusters (i.e., K_1 and K_2) to be 16 and 15. The cluster centers of both $\{\mathbf{x'}_{v_i}\}$ and $\{\mathbf{x'}_{a_w}\}$ are also mapped into a 2D space via t-SNE. The corresponding visualization result is shown in Figure 4 (b), in which each community (i.e., node cluster) may have one or more nearest attribute cluster centers (with relatively close distance in the hidden space), indicating that it's possible for a community to have more than one relevant semantic topics.

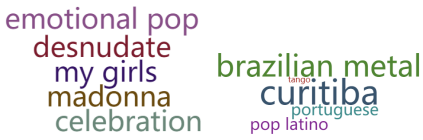
[22] http://www.openblas.net

[23] https://grouplens.org/datasets/hetrec-2011

[24] http://www.lastfm.com



(a) Visualization of Node & Attribute Embeddings (b) Visualization of Cluster Centers (c) Description of Community #15



(d) Top-1 Description of Community #1 (e) Top-2 Description of Community #1

Figure 4: Visualization Results of the Case Study on *LastFM*

We adopted two different strategies to generate the descriptions. First, we selected the top-5 relevant keywords for each community by calculating and ranking the distance between each node cluster center and attribute embedding $\mathbf{x'}_{a_w}$, where each community only have one comprehensive description. The single wordcloud of community #15 is illustrated in Figure 4 as an example. Second, we selected the most relevant topics for each community by measuring the distance between the cluster centers of node and attribute, and then generated the top-5 keywords for each topic. In this case, each community may have more than one descriptions, with each one reflecting a specific aspect of semantic (i.e., topic). As an instance, 2 relevant descriptions of community #1 are presented in Figure 4 (d) and (e).

To verify the semantic relation among the top words in each wordcloud, we used them as the query words of Wikipedia[25] to refer to other related materials. In Figure 4 (c), *metal music* is the primary topic, just as "finnish death metal", "prog heavy metal" and "symphonic heavy metal" infer. "post-ironic beardcore" and "burpcore" are both words used to describe the genre of metalcore music, which is a fusion genre of hardcore punk and extreme metal. For Figure 4 (d), *Pop music* should be the hidden topic, where "emotional pop" is directly related to it. "desnudate" and "my girls" may refer to the songs of American pop singer Christina Aguilera in album Bionic, which is characterized with the genres of electropop and futurepop. Moreover, "madonna" may indicate the American singer Madonna Louise Ciccone, who is referred to as the "Queen of Pop". The topic of Figure 4 (d) is *Latin music*, where "pop latino", "brazilian metal" and "portuguese" are related to the music and language in Latin America. Furthermore, "curitiba" is a city in Brazil, while "tango" is a popular dance originated in Latin America.

5 Conclusion

In this paper, we introduced a novel SGR model to generally formulate the graph representation learning in attributed networks as a high-order (topology) proximity based embedding task of an abstracted weighted graph with heterogeneous entities. The proposed model could not only comprehen-

[25] https://en.wikipedia.org/

sively capture the high-order proximity inside and among network structure and semantic, but also jointly learn the low-dimensional representations of both node and attribute, effectively supporting the advanced semantic-oriented downstream applications (e.g., semantic community detection). The effectiveness of SGR was also verified on a series of real attributed networks for several network inference tasks.

In our future work, we intend to explore a more comprehensive but simpler parameter adjustment strategy to effectively reduce the hyper-parameters' search space, with the guarantee of performance. Moreover, to further reduce SGR's computation time via the distributed SVD techniques and optimized matrix operation libraries is also our next focus.

6 Acknowledgement

This work has been financially supported by Guangdong Natural Science Fund Project (Grant No. 2018A030313017) and Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20170412150946024).

References

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

[Bandy *et al.*, 2018] Sambaran Bandy, Harsh Kara, Aswin Kannan, and M Narasimha Murty. Fscnmf: Fusing structure and content via non-negative matrix factorization for embedding information networks. *arXiv: Social and Information Networks*, 2018.

[Benson *et al.*, 2016] A. R. Benson, D. F. Gleich, and J Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163, 2016.

[Cantador *et al.*, 2011] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *ACM RecSys*, 2011.

[Cao *et al.*, 2015] Shaosheng Cao, Lu Wei, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM*, pages 891–900, 2015.

[Cao *et al.*, 2018] Jinxin Cao, Jin Di, Yang Liang, and Jianwu Dang. Incorporating network structure with node contents for community detection on large networks using deep learning. *Neurocomputing*, 297:S0925231218300985, 2018.

[Cheng *et al.*, 2015] Yang Cheng, Deli Zhao, Deli Zhao, Edward Y. Chang, and Edward Y. Chang. Network representation learning with rich text information. In *AAAI*, 2015.

[Cui *et al.*, 2018] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE TKDE*, pages 1–1, 2018.

[Der Maaten and Hinton, 2008] Laurens Van Der Maaten and Geoffrey E Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008.

[Fan *et al.*, 2008] Rong En Fan, Kai Wei Chang, Cho Jui Hsieh, Xiang Rui Wang, and Chih Jen Lin. Liblinear: a library for large linear classification. *JMLR*, 9(9):1871–1874, 2008.

[Huang *et al.*, 2017] Xiao Huang, Jundong Li, and Xia Hu. Accelerated attributed network embedding. In *SDM*, pages 633–641, 2017.

[Ishioka, 2000] Tsunenori Ishioka. Extended k-means with an efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.

[Iwen and Ong, 2016] M. A. Iwen and B. W. Ong. A distributed and incremental svd algorithm for agglomerative data analysis on large networks. *SIMAX*, 37(4):1699–1718, 2016.

[Jin *et al.*, 2018] Di Jin, Meng Ge, Liang Yang, Dongxiao He, Longbiao Wang, and Weixiong Zhang. Integrative network embedding via deep joint reconstruction. In *IJCAI*, pages 3407–3413. AAAI Press, 2018.

[Khosla *et al.*, 2019] Megha Khosla, Avishek Anand, and Vinay Setty. A comprehensive comparison of unsupervised network representation learning methods. *arXiv: Learning*, 2019.

[Liang *et al.*, 2016] Yang Liang, Xiaochun Cao, Dongxiao He, Chuan Wang, and Weixiong Zhang. Modularity based community detection with deep learning. In *IJCAI*, pages 2252–2258, 2016.

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

[Qiu *et al.*, 2018] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*, pages 459–467. ACM, 2018.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassirad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[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.

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

[Wang *et al.*, 2016b] Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. Semantic community identification in large attribute networks. In *AAAI*, pages 265–271, 2016.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.

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