Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach

Jêröme Lang

Laboratoire d'Analyse et Modélisation des Systèmes pour l'Aide à la Décision (LAMSADE)
pcchair@ijcai-18.org

Abstract

Network Embedding is a method to learn the low-dimensional vector representations of nodes under the condition of preserving different kinds of the network properties. Previous works mainly focus on preserving structure information of nodes on a particular resolution, like neighbor information or community information, but can not preserve the hierarchical community structure which enables that the network can be easily analyzed on various resolutions. In this paper, we formulate a constrained optimization problem to describe the hierarchical community structure preserving network embedding. Inspired by the galaxy structure with hierarchy, we propose the Galaxy Network Embedding (GNE) model to solve the sophisticated optimization above effectively. In detail, we present an approach to embed communities into an low dimension spherical surface whose center represents the parent community they belong to. The experiments reveal that the node representations from GNE preserves the hierarchical community structure and shows advantages in several applications such as node multi-classification and network visualization. The source code of GNE is available online.

1 Introduction

Network embedding is an approach to learn the low-dimensional representation of complex network vertices by neural network [Grover and Leskovec, 2016] or matrix decomposition [Wang *et al.*, 2017] under the condition of preserving different kinds of the structure information of the network. Through the network embedding, many general machine learning methods can be applied for fast and effective vertices classification, clustering and network visualization [Bhagat *et al.*, 2011] [Yan *et al.*, 2007].

Network embedding works are composed of structure-preserving method and property-preserving method. Our paper focuses on preserving structure information. Thus, in terms of structure-preserving methods, inspired by the word2vec in NLP[Mikolov *et al.*, 2013], some methods consider the node context and represent a node with its nearby nodes[Perozzi *et al.*, 2014][Grover and Leskovec, 2016]. Another methods present more explicit optimization to preserve the pair-wise proximity of node, which take full account of the characteristics of network structures[Tang *et al.*, 2015][Cao *et al.*, 2015][Wang *et al.*, 2016]. In addition to preserving the microscopic structure, the community structure, one important mesoscopic description of network structure, is incorporated into network embedding in MNMF, of which the learned embedding space can well reflect the organizational structures and functional components of networks[Wang *et al.*, 2017].

Essentially, these methods mainly focus on preserving neighborhood information, or community structure on a particular scale. Nevertheless, the community structure of a complex network is hierarchical[Clauset *et al.*, 2006], such as social networks, air transportation networks, and metabolic networks. In the presence of hierarchy, the concept of community structure becomes richer[Sales-Pardo *et al.*, 2007], for instance, in the chemical network, amino acids constitute proteins, which in turn are composed of basic chemical elements. In a word, the hierarchical community preserving network embedding can provide richer structural information, and make it easier to analyze the network on different resolutions.

It is challengeable to formulate the hierarchical community preserving network embedding. Not only the topology relationship between the nodes of the same layer needs to be considered, but the relationship between different levels needs to be considered. In addition to formulate an effective optimization, designing a model to efficiently solve it is a challenge as well.

In this paper, we define a constrained optimization problem to formulate the hierarchical community structure preserving network embedding and propose the Galaxy Network Embedding (GNE) model to solve it. To the best of our knowledge, two type of relationship should be preserved on the hierarchical community network embedding. The first is local information which preserve the microscopic structure(pairwise node similarity) in the same community at a certain layer. The second is hierarchy information. Horizontally, the nodes in the same community are more similar to each other. Vertically, a community in the deeper level has a greater cohesion degree. Thus, an optimization extends from GraRep[Cao *et al.*, 2015], is defined to preserve the pair-wise similarity be-

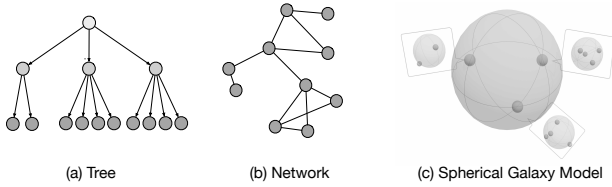(a) Tree   (b) Network   (c) Spherical Galaxy Model

Figure 1: Different display approaches of the network hierarchical structure.

tween communities, with constraints on hierarchy horizontally and vertically.

In order to solve the sophisticated non-convex optimization with constraints efficiently, we propose the Galaxy Network Embedding model. The galaxy, another type of systems with hierarchical structure, has two desirable structural characteristics which ensure that the embedding results preserve the essence of hierarchy: the first is that stars belonging to the same galaxy distribute in the sphere of ball almostly; the other is that the distance between the balls is much greater than the radius of balls. Inspired by the galaxy structure, we regard communities composed of sub-communities as galaxies composed of stellar systems (see Fig.1(b) (c)) In detail, based on the hierarchical tree of a network whose nodes denote a community on a particular resolution, which can be extracted by off-the-shelf models, the temporary representation of each node in the tree is firstly embeded with the optimization without constraints layer by layer. And then, considering the constraints of hierarchy, we embed the temporary representation of sub-communities on an m-dimensional spherical surface recursively whose spherical center is the parent community they belong to. The radii of the embedded communities indicate the cohesion degree.

To summarize, we make the following contributions:

- We formulate the hierarchical community structure preserved network embedding as an optimization problem with constraints and present a novel method to transform the difficult optimization problem into an unconstrained optimization problem more easily to be solved.

- We propose the Galaxy Network Embedding (GNE) model based on a novel spherical embedding method, which preserves the hierarchical community structure on any resolution while embedding the vertices or communities of a network into low-dimensional vectors in Euclidean space.

- GNE is extensively evaluated on four real networks and four synthetic networks. The experiment results demonstrate that our model can integrally preserve the hierarchical community structure and is significantly superior to other models on vertex classification and network visulization.

2 Related Work

Network Embedding. Network embedding maps the vertices or edges of a network into a low-dimensional vector space, which is beneficial at vertex classification and network visualization. It is well recognized that network representation has the two goals: reconstruction original network and support network inference. Manifold learning aims to reconstruct all the links, causing overfitting and limiting the network inference ability seriously[Tenenbaum *et al.*, 2000][Roweis and Saul, 2000]. In order to support network inference, structure-preserving method and property-preserving method are proposed. Our paper focuses on preserving structure information. Thus, in terms of structure-preserving methods, inspired by the word2vec in NLP[Mikolov *et al.*, 2013], some methods consider the node context and represent a node with its nearby nodes. For example, Deepwalk [Perozzi *et al.*, 2014] combines truncated random walk [Fouss *et al.*, 2007] with the Skip-Gram [Mikolov *et al.*, 2013] to learn vertex representations; Based on DeepWalk, Node2Vec [Grover and Leskovec, 2016] proposes a biased random walk. Another methods present more explicit optimization to preserve the pair-wise proximity of node, which take full account of the characteristics of network structures. For instance, LINE [Tang *et al.*, 2015] formulates a novel optimization problem based on first- and second-order proximity on network embedding. GraRep [Cao *et al.*, 2015] integrate the k-step information to learn vertex representations. In addition to preserving the microscopic structure, the community structure, one important mesoscopic description of network structure, is incorporated into network embedding in MNMF[Wang *et al.*, 2017]. All the methods above mainly focus on preserving the pairwise proximity or community structure on a particular resolution, however, the community structure on different resolutions are not considered.

Hierarchical Network. Actually, many complex networks in the real world have hierarchical community structure. [Newman, 2003] introduces the community structure of the network, and summarizes that complex networks have the small-world property and the scale-free property. [Song *et al.*, 2005] finds that a large number of real networks have the self-similar property. Since the concept of community structure becomes richer in the presence of hierarchy, more and more works pay attention to extract hierarchy of networks. [Clauset *et al.*, 2008] [Yang *et al.*, 2013][Sales-Pardo *et al.*, 2007][Lancichinetti *et al.*, 2008] propose effective approaches to extract hierarchical community structrue from complex networks that we can directly use in our model. Embedding the hierarchical community structure into low-dimensional vector spaces has not been investigated yet.

3 Problem Definition

We first give the definition of the hierarchical clustering tree [Clauset *et al.*, 2008] of a network.

Definition 1 (***Hierarchical Clustering Tree of Network***). Given an undirected network $G = (V, E)$, the hierarchical clustering tree of G is denoted as T with a depth of L. We denote C as the node set, C^l as the node set at the l-th level and c_i^l as the i-th node at the l-th level of T. $\Gamma(c)$ denotes the child node set of the node c and $\delta(c)$ denotes the parent node of c. Meanwhile, c_i^l also represents the i-th community while recursively dividing G at the l-th level. For the l-th level of T,

$$\forall c_i^l, c_j^l \in C^l \quad c_i^l \cap c_j^l = \varnothing$$

$$\bigcup_i c_i^l = V.$$

Especially, c_1^1 is the root node of T and also the node set of G (i.e. $c_1^1 = V$) and $c_i^L = \{v_i\}$, where v_i is the i-th vertex of the G.

In order to preserve the hierarchical structure property of the network explicitly, we embed not only the vertices but also the communities of all levels. Since $c_i^L = \{v_i\}$, for brevity, we use community representation instead of vertex representation in the model. We define the community representation:

Definition 2 (*Community Representation*). Given a network G and its hierarchical clustering tree T_G, the community representation is a function $\Phi : C \to R^m$ ($m \ll |C|$) which can preserve the structure property of G.

In our scenario, the community representation can preserve two kinds of properties of G. One is the local information, i.e. the pairwise proximity between sub-communities in the same community. The other is the hierarchical structure property, i.e. the parent-child relationship and siblings relationship. In order to preserve the local information between communities, we introduce the community proximity extended from the definition of the common neighbour similarity [Libennowell and Kleinberg, 2007]:

Definition 3 (*Community Proximity*). Community proximity is the pairwise similarity between communities in a network. We define the proximity between c_i^l and c_j^l.

$$S_{i,j}^k = \frac{1}{|c_i^l||c_j^l|} \sum_{u \in c_i^l} \sum_{v \in c_j^l} \xi_{u,v} \tag{1}$$

where $c_i^l, c_j^l \in \Gamma(c_k^{l-1})$ and $\xi_{u,v}$ is the similarity between the vertex pair u and v shown as follows :

$$\xi_{u,v} = \frac{A_u^T A_v}{\sqrt{||A_u||_1 ||A_v||_1}}, \tag{2}$$

where A is the adjacency matrix of G, A_u is the u-th column of the A.

We propose the optimization objective which extends 1-step optimization objective in GraRep [Cao *et al.*, 2015]. To preserve the proximity between communities deriving from the same parent node c_k^{l-1}, we choose to minimize the KL-divergence between the distribution of community similarity from ground truth and learned representation respectively:

$$O^{(k)}(\Phi, \Phi') = \sum_{c_i^l, c_j^l \in \Gamma(c_k^{l-1})} \frac{S_{i,j}^l}{\sum_t S_{i,t}^l} \cdot \log \sigma(\Phi(c_i^l)^T \Phi'(c_j^l))$$

$$+ \frac{\lambda}{|\Gamma(c_k^{l-1})|} \sum_{c_e^l \in \Gamma(c_k^{l-1})} \frac{S_{i,e}^l}{\sum_t S_{i,t}^l} \cdot \log \sigma(-\Phi(c_i^l)^T \Phi'(c_e^l)), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function, $\Phi(c)$ is the 'current' vector of community c we want and $\Phi'(c)$ is the 'context' vector
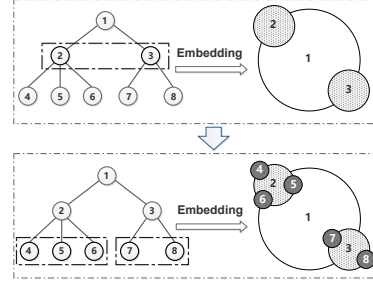

Figure 2: Structure of GNE

of community c which is an auxiliary variable. λ is a hyper-parameter related to negtive sampling indicating the number of negative samples[Cao *et al.*, 2015].

In order to preserve hierarchical structure properties, we impose constraints on the embedded communities.

Definition 4 (*Hierarchical Preserving Network Embedding*). The Hierarchical Preserving Network Embedding problem is formally defined as a constrained optimization problem:

$$\min_{\Phi, \Phi'} \sum_{l}^{L-1} \sum_{c_l^k \in C^l} O^{(k)}(\Phi, \Phi') \tag{4}$$

$$s.t. \quad ||\Phi(c_u^l) - \Phi(c_v^l)|| - ||\Phi(c_u^l) - \Phi(c_w^l)|| < 0,$$
$$||\Phi(c_i^{l+1}) - \Phi(c_j^l))|| - ||\Phi(c_j^l) - \Phi(c_k^{l-1})|| < 0,$$

where,

$$\delta(c_u^l) = \delta(c_v^l), \quad \delta(c_u^l) \neq \delta(c_w^l)$$
$$\delta(c_i^{l+1}) = c_j^l, \quad \delta(c_j^l) = c_k^{l-1}$$

The constraints can be divided into two aspects: Horizontally, vertices belonging to the same community should be closer to each other than those belonging to different communities(see the first constraint). Vertically, the cohesion degree of low level communities should be less than that of high level ones in T(see the second constraint).

It can be seen that the constrained optimization objective Eq.4 is very complex, the number of whose constraints reaches $O(|V|^3)$. It is hard to solve the optimization problem directly.

4 Galaxy Network Embedding Model

Inspired by the structure of galaxies, the original optimization(see Eq.4) is transformed into an optimization with spherical constraints(see Eq.8). In particular, we design a determination strategy of radii and propose a spherical embedding method for sphere of each layer in hierarchy. Finally, the algorithm of the GNE can be found in the last part of this section.

Properties of Galaxy Structure

The galaxy, another type of systems with hierarchical structure, has two desirable structural characteristics which ensure

that the embedding results preserve the essence of hierarchy(see Fig.2): the first is that stars belonging to the same galaxy distribute in the sphere of ball almostly; the other is that the distance between two galaxies is much larger than the radius of each of them, namely,

$$r_i, r_j \ll ||x_i - x_j||_2 \qquad (5)$$

where x is the center of galaxy and r is the radii.

It is evident that if the distribution of the embedded communities is consistent with the structure of galaxies, the constraints of Eq.4 can be satisfied. Therefore, we introduce the galaxy structure into our model.

4.1 Model Framework based on Galaxy Structure

First, we simplify the Galaxy model. We suppose that galaxies are located at a spherical surface. In other words, the distances from each galaxy to the center of its galaxy group are equal. Specifically, besides the Euclidean representation $\Phi(c_i^l)$, each embedded community c_i^l has another attribute r_i^l (the radius of the corresponding galaxy group). We have the following constraints:

$$\forall c_i^{l+1} \in \Gamma(c_k^l), \quad l = 1, 2, ..., L-1,$$
$$||\Phi(c_i^{l+1}) - \Phi(c_k^l)||_2 = r_k^l, \qquad (6)$$

$$\forall c_i^{l+1}, c_j^{l+1} \in \Gamma(c_k^l), \quad l = 1, 2, ..., L-1,$$
$$r_i^{l+1}, r_j^{l+1} \ll ||\Phi(c_i^{l+1}) - \Phi(c_j^{l+1})||_2. \qquad (7)$$

The constraints of Eq.4 can be satisfied if Eq.6 and Eq.7 are satisfied(see). For Eq.7, we ensure it is satisfied by desiging the proper determination strategy of r_i. For spherical constraints Eq.6, we combine it with Eq.3 to obtaining a local optimization objective with c_k^{l-1} as the parent community:

$$O^{(k)}(\Phi, \Phi') = \sum_{c_i^l, c_j^l \in \Gamma(c_k^{l-1})} \frac{S_{i,j}^l}{\sum_t S_{i,t}^l} \cdot \log \sigma(\Phi(c_i^l)^T \Phi'(c_j^l))$$
$$+ \frac{\lambda}{|\Gamma(c_k^{l-1})|} \sum_{c_e^l \in \Gamma(c_k^{l-1})} \frac{S_{i,e}^l}{\sum_t S_{i,t}^l} \cdot \log \sigma(-\Phi(c_i^l)^T \Phi'(c_e^l)),$$
$$s.t. \quad \forall c_i^{l+1} \in \Gamma(c_k^l), \quad ||\Phi(c_i^{l+1}) - \Phi(c_k^l)||_2 = r_k^l \qquad (8)$$

In this way, we can optimize the objective Eq.8 recursively from top to bottom in hierarchy. The whole embedding procedure is illustrated in Fig.2. Next, we will introduce how to determine the radius of each community and how to solve the optimization problem Eq.8 level by level.

Strategy of Determining Radius

In order to ensure Eq.7 is satisfied, the radius r and the distance d between two spheres should be subject to:

$$r_i^l = \eta \cdot d_{i,*}^{l-1}, \eta < \frac{1}{2} \qquad (9)$$

$$\sum_{j=l}^{L} r^j < \zeta d_*^{l-1}, \zeta < \frac{1}{2} \qquad (10)$$

where

$$d_{i,*}^{l-1} = \min_{c_j^l \in \Gamma(\delta(c_i^l)), i \neq j} ||\Phi(c_i^l) - \Phi(c_j^l)|| \qquad (11)$$

$$d_{i,*}^{l-1} < 2 \cdot r_i^{l-1} \qquad (12)$$

More precisely, r^l denotes the sphere radius of nodes at layer l and L denotes the depth of T_G. Especially, we assume that r_i^l and $d_{i,*}^{l-1}$ have a linear relationship and η reflects the cohesion degree of the community c_i^l relative to c_*^l (see Eq.9). The constant ζ ensures any two spherical surfaces at any layer will not intersect in Eq.10, such as $\zeta = \frac{1}{3}$.

Lemma 1. When η is less than $\frac{\zeta}{1+2\zeta}$, the constraint Eq.7 can be satisfied, where ζ is a constant to make sure any two spherical surfaces at any layer will not intersect in Eq.10.

Proof. According to Eq.9 and Eq.12, the relationship between r^j and d_*^{l-1} can be derived as follows, where $j \in [l, L]$:

$$r^j = \eta \cdot d_*^{j-1} < \eta \cdot 2r^{j-1}$$
$$= 2\eta^2 \cdot d_*^{j-2} < 2^2\eta^2 \cdot r^{j-2}$$
$$......$$
$$< 2^{k-1}\eta^k \cdot d_*^{j-k} \qquad (13)$$

When $k = j - l + 1$,

$$r^j < 2^{j-l}\eta^{j-l+1} \cdot d_*^{l-1} \qquad (14)$$

thus, we get the upper bound of r^j. The inequation Eq.10 is valid if the upper bound of $\sum r^j$ is less than ζd_*^{l-1}. Substituting Eq.10 for Eq.14, we obtain:

$$\sum_{j=l}^{L} 2^{j-l}\eta^{j-l+1} \cdot d_*^{l-1} < \zeta d_*^{l-1} \Rightarrow \sum_{j=l}^{L} r^j < \zeta d_*^{l-1}$$

Thus, since $\eta < \frac{1}{2}$,

$$\sum_{j=l}^{L} (2\eta)^{j-l+1} < \frac{2\eta}{1 - 2\eta} < 2\zeta$$

$$\eta < \frac{\zeta}{1 + 2\zeta}$$

$\square$

In conclusion, when $\eta < \frac{\zeta}{1+2\zeta}$, Eq.10 is satisfied, simultaneously, the constraint Eq.7 is satisfied. In experiments, we assign $\frac{1}{3}$ to ζ.

Lemma 2. The constraints of Eq.4 can be satisfied if Eq.6 and Eq.7 are satisfied

Proof. we first proof the first constraint in Eq.4 can be satisfied among communities in the same layer. we define:

$$c_u^{l+1}, c_v^{l+1} \in \Gamma(c_i^l)$$
$$c_w^{l+1} \in \Gamma(c_j^l)$$

According to the Eq.7,

$$r_i^l, r_j^l \ll ||\Phi(c_i^l) - \Phi(c_j^l)||_2$$
$$\Rightarrow r_i^l, r_j^l < ||\Phi(c_i^l) - \Phi(c_j^l)||_2 - r_i^l - r_j^l$$

Obviously,

$$||\Phi(c_v^{l+1}) - \Phi(c_u^{l+1})||_2 \leq 2 \cdot r_i^l < ||\Phi(c_i^l) - \Phi(c_j^l)||_2 - r_i^l - r_j^l$$
$$||\Phi(c_v^{l+1}) - \Phi(c_w^{l+1})||_2 > ||\Phi(c_i^l) - \Phi(c_j^l)||_2 - r_i^l - r_j^l$$
$$\Rightarrow ||\Phi(c_v^{l+1}) - \Phi(c_u^{l+1})||_2 < ||\Phi(c_v^{l+1}) - \Phi(c_w^{l+1})||_2$$

Thus, the first constraint in Eq.4 can be satisfied.

Next, we proof the second constraint between different layers in Eq.4 can be satisfied. We have

$$c_v^{l+1} \in \Gamma(c_i^l), \quad c_i^l \in \Gamma(c_k^{l-1})$$
$$||\Phi(c_v^{l+1}) - \Phi(c_i^l)||_2 = r_i^l = \eta d_{i,*}^{l-1}$$
$$||\Phi(c_i^l) - \Phi(c_k^{l-1})||_2 = r_k^{l-1} = \eta d_{k,*}^{l-2}$$

according to the derivation in Eq.13, $*$ denotes maximum:

$$d_*^{l-1} \leq 2 \cdot r_*^{l-1}$$
$$r_*^{l-1} = \eta \cdot d_*^{l-2}, \quad \eta < \frac{1}{2}$$

thus, we can obtain

$$d_*^{l-1} < d_*^{l-2}$$
$$\Rightarrow ||\Phi(c_v^{l+1}) - \Phi(c_i^l)||_2 < ||\Phi(c_i^l) - \Phi(c_k^{l-1})||_2$$

In conclusion, the constraints of Eq.4 can be satisfied if Eq.6 and Eq.7 are satisfied. □

4.2 Spherical Embedding

Compared with the optimization objective of the classic neural embedding method, we just add an extra spherical constraint in Eq.8. The neural embedding method can be directly optimized using SGD, which is efficient and whose effect has been verified. Considering these advantages, we transform Eq.8 into a two-step optimization procedure. First, we use the Adam optimizer[Rushing *et al.*, 2005] to optimize Eq.3 without constraint based on neural networks, and obtain the representation of communities $\Omega(c_i^l)$ which preserving the pairwise relative Euclidean distances of communities. Second, we map $\Omega(c_i^l)$ to a spherical surface to get the final representation of communities $\Phi(c_i^l)$. Next, we introduce the spherical projection method.

Optimization Objective of Spherical Projection

To preserve the relative distances in spherical projection, we formally define the following optimization objective:

$$\min_{\Phi} J^{(k)} = \left\| \frac{D}{||D||_F} - \frac{B}{||B||_F} \right\|_F + \mu \exp(-\gamma ||B||_F) \quad (15)$$
$$s.t. \quad ||\Phi(c_i^l) - \Phi(c_k^{l-1})|| = r_k^{l-1}$$

where,

$$c_i^l, c_j^l \in \Gamma(c_k^{l-1})$$
$$D_{ij} = ||\Omega(c_i^l) - \Omega(c_j^l)||$$
$$B_{ij} = ||\Phi(c_i^l) - \Phi(c_j^l)||$$

The first term is to preserve the relative distances, and the second term is a penalty term to make the distances between each pair of points after projection are as large as possible. μ and γ are hyper parameters. As the overall optimization procedure is top-down, $\Phi(c_k^{l-1})$, r_k^{l-1} and D are all constants.

Spherical Projection Optimization Procedure

The constraints of Eq.15 are for arbitrary spherical surfaces. We can perform scaling and translation transformations on the coordinate system to turn them into the unit sphere constraints, namely the following conversion :

$$\Psi(c_i^l) = \frac{\Phi(c_i^l) - \Phi(c_k^{l-1})}{r_k^{l-1}}. \quad (16)$$

Hence the constraints can be transformed as follows:

$$||\Psi(c_i^l)|| = 1. \quad (17)$$

The Euclidean distance between $\Psi(c_i^l)$ and $\Psi(c_j^l)$ is

$$B_{ij}' = ||\Psi(c_i^l) - \Psi(c_j^l)|| \quad (18)$$

Bringing Eq.17 into Eq.18, we have:

$$B_{ij}' = ||\Psi(c_i^l)||^2 + ||\Psi(c_j^l)||^2 - 2\Psi(c_i^l)^T \Psi(c_j^l)$$
$$= 2 - 2\cos(\Psi(c_i^l), \Psi(c_j^l)) \quad (19)$$

It can be seen that B_{ij}' is not related to the length of vector $\Psi(c_i^l)$ and $\Psi(c_j^l)$, but only the Cosine distance between them. Therefore, the vector length constraint in the objective can be removed. The final optimization objective on the unit sphere is:

$$\min_{\Psi} H^{(k)} = \left\| \frac{D}{||D||_F} - \frac{B'}{||B'||_F} \right\|_F + \mu \exp(-\gamma ||B'||_F). \quad (20)$$

We use $\Psi_*(c_i^l)$ to denote the minimum point after optimization. After normalization, translation, and scaling, the final results are obtained:

$$\Phi(c_i^l) = \frac{\Psi_*(c_i^l)}{||\Psi_*(c_i^l)||} \times r_k^{l-1} + \Phi(c_k^{l-1}) \quad (21)$$

Obviously, the unconstrained objective has a lower bound. The minimum points can be found using Gradient Descent algorithm.

4.3 The GNE Algorithm

Algorithm 1 The GNE algorithm

---
function LEARNFEATURES(Network G, Hierarchical Clustering Tree T)
 Initialize $\Phi(c_1^1) = 0$ and $\mathbf{r_1^1} = \mathbf{1}$
 Φ = OptimizeRecursively(c_1^1, G, T, $\mathbf{r}$, Φ)
return Φ

---
function OPTIMIZERECURSIVELY(Current node c_i^l, Network G, Hierachical Clustering Tree T, Radius Set $\mathbf{r}$, Reresentation Set Φ)
 if $l = L$ **then return Φ**
 $\Omega(\Gamma(c_i^l))$ = Optimize Objective $O^{(i)}$ with Adam
 $\Psi(\Gamma(c_i^l))$ = Optimize Objective $H^{(i)}$ with Adam
 for all $c_j^{l+1} \in \Gamma(c_i^l)$ **do**
  $\Phi(c_j^{l+1}) = \dfrac{\Psi(c_j^{l+1})}{\|\Psi(c_j^{l+1})\|} \times \mathbf{r_i^l} + \Phi(\mathbf{c_i^1})$
  $\mathbf{r_j^{l+1}} = \eta \cdot \mathbf{d_{j,*}}$
 for all $c_j^{l+1} \in \Gamma(c_i^l)$ **do**
  Φ = OptimizeRecursively(c_j^{l+1}, G, T, $\mathbf{r}$, Φ)
 return Φ

The pseudocode for GNE, is given in Algorithm 1. The whole embedding process is executed from top to down. For all children nodes of a certain node c_i^l, i.e. the nodes in $\Gamma(c_i^l)$, the learned representation $\Psi(\Gamma(c_i^l))$ on the unit sphere can be obtain after two-step optimization procedure with Adam[Rushing *et al.*, 2005], the optimization Eq.3 is used first and the optimization Eq.20 follows. The final representation $\Phi(c_j^{l+1})$ can be obtained by the transformation in Eq.21. Besides, the corresponding r_j^{l+1} can be calculated with minimum distance $d_{j,*}$ and a constant η less than $\frac{1}{5}$ (see Lemma 1).

Time Complexity Analysis
For brevity on description, we assume that the hierarchical clustering tree of network G=(V,E) is a k-ary tree T_G with depth $log_K|V|$. The number of nodes N in T_G is $\frac{K^{log_K|V|+1}-1}{K-1}$. Spherical embedding should be done on each tree node layer by layer. Especially, the embedding procedures on the same layer tree nodes can be parallelized. For spherical embedding, the complexity of computing $\Omega(\Gamma(c_i^l))$ is $O(\mathcal{E} \times K \times Q)$. More precisely, $\mathcal{E}$ is the number of training epoch, K is the size of community to be embedded and Q is the neural network computing complexity of skip-gram model, i.e. $Q = O(C \times (\mathcal{D} + \mathcal{D} \times M))$, where C is the skip window size, $\mathcal{D}$ is the embedding size and M is the negative sampling size; the complexity of computing $\Psi(\Gamma(c_i^l))$ is $O(\mathcal{E} \times K^2 \times \mathcal{D})$. Thus, the spherical embedding complexity is $O(\mathcal{E} \times K \times Q + \mathcal{E} \times K^2 \times \mathcal{D})$. The optimization algorithm is implemented on the Tensorflow platform, which can be accelerated with GPU.

5 Experiments

In this section, An experimental analysis of our method on both synthetic and real networks is presented. The code of our model is available on Github.

5.1 Experiment Setup

Data Sets We employed the following three real datasets in Facebook social networks [Traud *et al.*, 2012] for node classification, which comprises 100 colleges and universities in US. We choose the social networks in Hamilton University(2314 nodes, 96394 edges), Amherst College(2235nodes, 90954edges) and Georgetown Universit(9414 nodes, 425639 edges). Besides, in order to evaluate the performance of hierarchical community structure preservation, four Hierarchical Random Graphs (HRG) with explicit hierarchical community structure are generated by [Clauset *et al.*, 2008], i.e. Syn_with_125nodes(125 nodes, 406 edges), Syn_with_1800nodes(1800nodes, 739637 edges), Syn_with_2560nodes(2560 nodes, 1460147 edges) and Syn_with_3750nodes(3750 nodes, 3066250 edges).

Relevant Algorithms We compared the GNE against the following six network embedding algorithm: SpectralClustering[Tang and Liu, 2011], DeepWalk[Perozzi *et al.*, 2014], Node2Vec[Grover and Leskovec, 2016], LINE[Tang *et al.*, 2015], GraRep[Cao *et al.*, 2015] and MNMF[Wang *et al.*, 2017]. Additionally, we extract the hierarchical community structure of networks with [Tsvetovat and Kouznetsov, 2011] as the input of GNE.

Parameters Settings The hyper-parameters of GNE is $\boldsymbol{\theta}$, i.e. $\boldsymbol{\theta} = (\lambda, \mu, \gamma)$. GNE is not very sensitive to the hyper-parameters, whose artificial settings can achieve ideal results with GridSearch as auxiliary on a small range. Besides, the parameter setting of comparison models follow the recommended settings in relevant code packages.

5.2 Hierarchical Community Detection

In this section, we verify the ability of hierarchical community preservation of our model GNE. We consider synthetic data sets in this experiment, including three different structure of HRG but with the same number of layers(see Fig.3). Besides, Jaccard's coefficient [Halkidi *et al.*, 2001] is applied as an external intex for evaluating the performance of community preservation at each hierarchy of networks.
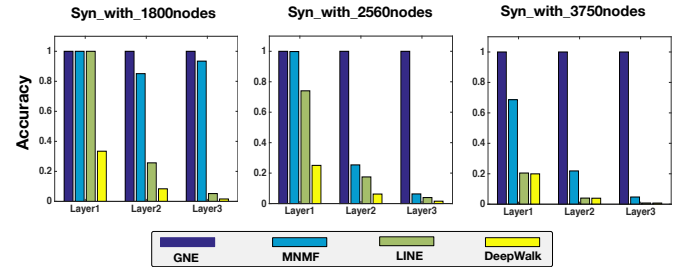

Figure 3: The comparision of hierarchical community preservation on different models. Three different structure of HRG but with the same number of layers are used. For each histogram, the horizontal ordinate denotes layer number, and the vertical ordinate denotes the Jaccard's coefficient.

Fig.3 shows that the content of the hierarchical community structure can be integrally preserved with our model, no

Model	Amherst					Hamilton					Georgetown				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
GNE	**94.91**	**94.25**	**93.94**	**93.97**	**93.39**	**94.66**	**94.55**	**94.49**	**94.49**	**93.88**	53.57	**53.79**	53.06	**52.75**	**52.11**
SpectralClustering	72.89	73.49	73.94	74.32	72.82	78.16	77.60	77.21	77.59	74.92	49.26	50.87	50.79	50.60	48.53
DeepWalk	90.62	91.65	91.32	91.13	90.41	92.89	92.33	92.52	92.18	91.55	54.07	**53.79**	53.35	51.69	50.92
Node2Vec	91.29	91.24	91.04	90.44	90.02	92.09	91.03	91.18	90.06	89.56	52.86	53.73	53.16	52.70	51.28
LINE	90.76	91.82	91.48	91.09	89.42	92.33	92.72	92.52	92.62	91.73	54.64	53.45	53.81	52.71	51.28
GraRep	92.13	92.25	91.78	91.56	91.48	93.67	93.04	92.30	92.40	91.00	**54.80**	53.24	**53.95**	51.87	51.74
MNMF	89.82	89.06	88.04	86.43	78.44	91.42	90.32	89.12	87.02	81.19	53.43	52.63	52.10	51.52	50.35

Table 1: The multi-label classification results on different percentages of test data sets.

matter how many the number of communities is. However, MNMF preserving community information on a particular resolution is inferior to GNE, and the other model could not perfectly deal with such cases with multi-layer and complex community structure.

5.3 Network Visualization

Network visualization is an important application of network embedding which maps a network into two-dimensional space. We visualize a synthetic network with 125 nodes, 406 edges and 5 communities. Fig.4 presents the visualization experiments. We firstly generate a self-similar network of which nodes are derived from the leaves of five-ary tree with four layers and edges are derived from the sampling of connected paths between leaves in the tree. Additionally, the nodes in the network are classified into different communities with Girvan–Newman algorithm [Girvan and Newman, 2002]. We compare our method against other models.

For other models, we layout the network into low-dimensional space, and then further map the low-dimensional vectors of the nodes to a 2-D space with t-SNE package [Maaten and Hinton, 2008]. For our model, we can directly embed network into 2-D vector space according to hyper spherical embedding we proposed.
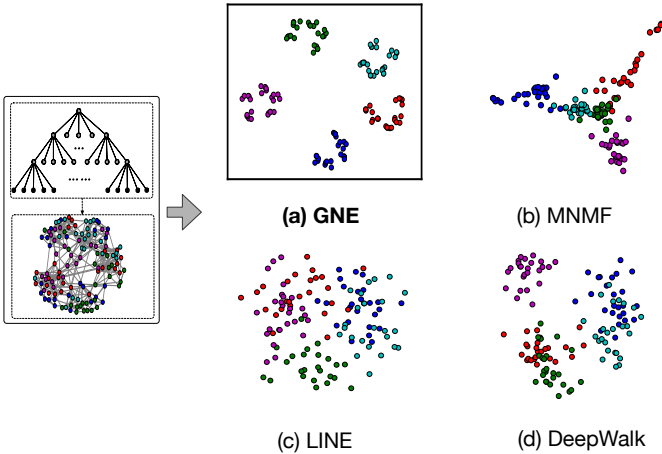

Figure 4: The visualization of node representations on different models

It can be seen from Fig.4 that our model GNE embeds nodes on the different-scaled spherical surface hierarchically. It is evident that the node representations of GNE are consis-

tent with modularity property at each hierarchy, i.e. higher intra-cluster similarity but lower inter-cluster similarity. That is to say, GNE integrally preserves the hierarchical community structure. Additionally, GNE has an outstanding performance on clustering nodes compared with others.

5.4 Node Classification

In order to verify the effectiveness of GNE on node classification, three real social networks with hierarchical community structure are used. The learned representations are used to classify the nodes into a set of labels. The classifier we used is Logistic Regression with sklearn package, and the evaluation metric is Accuracy. Different percentage of nodes are sampled randomly for evaluation, and the rest are for training. The results are averaged over 10 different runs.

Table 1. shows that GNE almost has a better performance than other models on different percentage of test data size. Since it's challengeable to extract the hierarchical clustering tree of big networks with the on-the-shelf works, for the networks of Georgeous University(9414nodes, 425639edges), our model uses the two-layer tree structure with the enrolment year as an indicator of data division, same with MNMF. In general, our model is robust no matter how much the percentage of test data accounts for, specifically on 10% of the samples training, 90% of the samples testing.

6 Conclusion

In this paper, we propose Galaxy Network Embedding (GNE) for network embedding to preserve the hierarchical community structure of a network. Specifically, we introduce an optimization problem with constraints and transform it into an unconstrained optimization problem more easily to be solved. Besides, we propose a spherical embedding method to maintain the hierarchical community structure from top to down. Empirically, we verify GNE in a variety of network datasets and applications. The extensive experimental results on vertex clustering and classification, as well as network visualization, demonstrate the advantages of GNE, especially on these networks with hierarchical community structure.

References

[Bhagat *et al.*, 2011] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. *arXiv: Social and Information Networks*, pages 115–148, 2011.

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

[Clauset *et al.*, 2006] Aaron Clauset, Cristopher Moore, and M E J Newman. Structural inference of hierarchies in networks. *international conference on machine learning*, pages 1–13, 2006.

[Clauset *et al.*, 2008] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *arXiv preprint arXiv:0811.0484*, 2008.

[Fouss *et al.*, 2007] Francois Fouss, Alain Pirotte, Jean-michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369, 2007.

[Girvan and Newman, 2002] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[Grover and Leskovec, 2016] A Grover and J Leskovec. node2vec: Scalable feature learning for networks. page 855, 2016.

[Halkidi *et al.*, 2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering algorithms and validity measures. In *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on*, pages 3–22. IEEE, 2001.

[Lancichinetti *et al.*, 2008] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11(3):19–44, 2008.

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

[Maaten and Hinton, 2008] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2605):2579–2605, 2008.

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

[Newman, 2003] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Alrfou, and Steven Skiena. Deepwalk: online learning of social representations. pages 701–710, 2014.

[Roweis and Saul, 2000] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, 2000.

[Rushing *et al.*, 2005] John Rushing, Udaysankar Nair, Udaysankar Nair, Ron Welch, Ron Welch, and Hong Lin. Adam: a data mining toolkit for scientists and engineers. *Computers and Geosciences*, 31(5):607–618, 2005.

[Sales-Pardo *et al.*, 2007] Marta Sales-Pardo, Roger GuimerÃ, AndrÃ© A. Moreira, and LuÃs A. Nunes Amaral. Extracting the hierarchical organization of complex systems. *arXiv preprint arXiv:0705.1679*, 104(39):15224–15229, 2007.

[Song *et al.*, 2005] Chaoming Song, Shlomo Havlin, and Hernan A Makse. Self-similarity of complex networks. *arXiv preprint cond-mat/0503078*, 2005.

[Tang and Liu, 2011] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

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

[Tenenbaum *et al.*, 2000] J. B. Tenenbaum, Silva V De, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[Traud *et al.*, 2012] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *Social Science Electronic Publishing*, 391(16):4165–4180, 2012.

[Tsvetovat and Kouznetsov, 2011] Maksim Tsvetovat and Alexander Kouznetsov. Social network analysis for startups. 2011.

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

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *The AAAI Conference on Artificial Intelligence*, 2017.

[Yan *et al.*, 2007] Shuicheng Yan, Dong Xu, Benyu Zhang, Hongjiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[Yang *et al.*, 2013] Bo Yang, Jin Di, Jiming Liu, and Dayou Liu. Hierarchical community detection with applications to real-world network analysis. *Data and Knowledge Engineering*, 83(4):20–38, 2013.