Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach

Jérôme 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

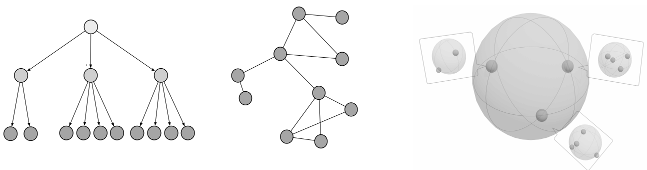
Network embedding is an approach to learn the low-dimensional representation of complex network vertices by neural network [Grover and Leskovec, 2016] or matrix decomposition [Wang *et al.*, 2017] under the condition of preserving different kinds of the structure information of the network. Through the 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[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. Such hierarchical structure networks are most often represented as a tree structure or dendrogram(see Figure 1. (a) (b)). Thus, hierarchical community structure is one of the most prominent features of networks, but existing models can not preserve this information. In a word, compared with the off-the-shelf embedding methods, 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. To the best of our knowledge, the essence of the hierarchical community structure can be interpreted from two perspectives. 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, how to design our optimization with these essence is a challenge as well. Besides, the third challenge is to solve the optimization effectively and efficiently.

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. 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 al-



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

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

mostly; 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 Figure 1(b) (c)), and embed sub-communities on an m-dimensional spherical surface whose spherical center is the parent community they belong to. The radii of the embedded communities indicate the cohesion degree. The embedding process of all communities and vertices is recursively implemented as described above.

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 formally define the problem of hierarchical structure preserving network embedding and introduce the optimization objective of the problem.~~ 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 $\Phi : C \rightarrow R^m$ ($m \ll |C|$) which can preserve the structure property of G.

In our scenario, we hope that 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}, the objective is defined as:

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

where $\sigma(\cdot)$ is the sigmoid function, λ is a hyper-parameter related to negtive sampling indicating the number of negative samples, $\Phi(c)$ is the "current" vector ~~[Cao *et al.*, 2015]~~ of community c we want and $\Phi'(c)$ is the "context" vector [Cao *et al.*, 2015] of community c which is an auxiliary variable.

In order to preserve hierarchical structure properties, we impose constraints on the embedded communities. The constraints can be divided into two aspects: horizontal and vertical. Horizontally, vertices belonging to the same community should be closer to each other than those belonging to different communities(see the first constraint of Eq. 4). Vertically, as the division granularity decreases, the cohesion degree of communities will increase. That means the cohesion degree of low level communities should be less than that of high level ones in T(see the second constraint of Eq. 4). We give the detailed definition of Hierarchical Preserving Network Embedding:

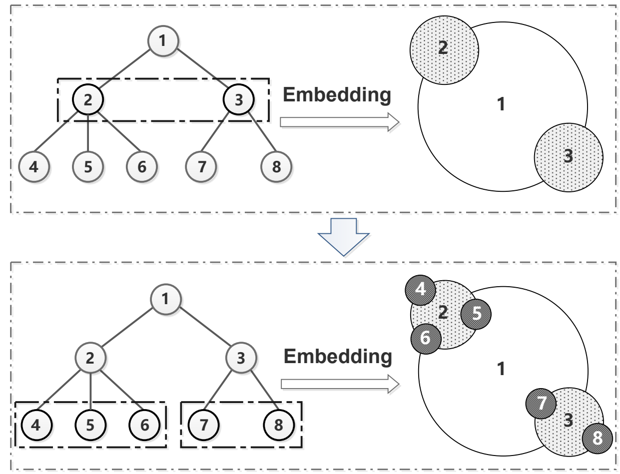


Figure 2: Structure of GNE

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

$$\begin{aligned} &\min_{\Phi,\Phi'} \sum_{l}^{L-1} \sum_{c_l^k \in C^l} O^{(k)}(\Phi, \Phi') \\ s.t. \quad &||\Phi(c_u^l) - \Phi(c_v^l)|| - ||\Phi(c_u^l) - \Phi(c_w^l)|| < 0, \\ &||\Phi(c_i^{l+1}) - \Phi(c_j^l))|| - ||\Phi(c_j^l) - \Phi(c_k^{l-1})|| < 0, \end{aligned} \tag{4}$$

where,

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

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. In the later section, we simplify the problem by imposing stronger constraints.

4 Galaxy Network Embedding Model 给定一个overview

Inspired by the structure of galaxies in the real world, we propose the GNE model to solve the optimization problem above efficiently through imposing more constraints.

4.1 Overview

Properties of Galaxy Structure

As well known, Galaxy structure has the similar hierarchical property (see Figure 2). For example, the Galaxy includes many stellar systems such as the solar system, while the solar system also consists of several planets. In order to express conveniently, we assume that a galaxy has spherical structure with radius r_i and the center located at x_i. The galaxy structure satisfies that the distance between two galaxies is much larger than the radius of each of them, namely,

段落大意：主要讲galaxy是分层结构的，总结出了球心距远大于球半径，如果保持了galaxy结构属性，就保持了分层的限制

$$r_i, r_j \ll ||x_i - x_j||_2 \tag{5}$$

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.

Model Framework based on Galaxy Structure

~~First, we simplify the Galaxy model.~~ We suppose that the distances from each galaxy to the center of its galaxy group are equal. In other words, all galaxies are located at a spherical surface. ~~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:

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

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

Eq. 6 and Eq. 7 are sufficient conditions for the constraints of Eq. 4 (i.e. the constraints of Eq. 4 can be satisfied if Eq. 6 and Eq. 7 are satisfied). We replace the constraints in optimization objective with two stronger constraints Eq. 6 and Eq. 7. For Eq. 7, we ensure it satisfied by desiging the proper determination strategy of r_i. For spherical constraints Eq. 6, we combine it and Eq. 3, thus obtaining a local optimization objective with c_k^{l-1} as the parent community:

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

为什么要两阶段解这个问题，为什么带r不好求

In this way, we can optimize the objective Eq. 8 from top to bottom. The whole embedding procedure is shown in figure 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.

半径部分放在这个位置说有点突兀

4.2 Strategy of Determining Radius

When determining radius, the following constraint should be satisfied :

$$r_i^l \ll d_{i,*}, \tag{9}$$

where

$$d_{i,*} = \min_{c_j^l \in \Gamma(\delta(c_i^l)), i \neq j} d_{i,j} \tag{10}$$

$$d_{i,j} = \|\Phi(c_i^l) - \Phi(c_j^l)\| \tag{11}$$

Then r_i^l is determined by $d_{i,*}$. We assume that they have a linear relationship:

$$r_i^l = \eta \cdot d_{i,*}, \tag{12}$$

where η reflects the cohesion degree of the community c_i^l relative to c_*^l. We adopt Standard Deviation of the similarity between the community and the vertices to reflect the relationship:

$$\eta = \frac{\theta \cdot \mathrm{std}\left(\xi_{u,c_*^l}\right)}{\sqrt{|c_i^l| - 1} \cdot \mathrm{mean}\left(\xi_{u,c_*^l}\right)}, \quad u \in c_i^l \tag{13}$$

where,

$$\xi_{u,c_*^l} = \frac{1}{|c_*^l|} \sum_{v \in c_*^l} \xi_{u,v}$$

$$\mathrm{mean}\left(\xi_{u,c_*^l}\right) = \frac{1}{|c_i^l|} \sum_{u \in c_i^l} \xi_{u,c_*^l}$$

$$\mathrm{std}\left(\xi_{u,c_*^l}\right) = \sqrt{\frac{1}{|c_i^l|} \sum_{u \in c_i^l} \left(\xi_{u,c_*^l} - \mathrm{mean}\left(\xi_{u,c_*^l}\right)\right)^2}.$$

In the equations above, θ is a adjustable hyper parameter and $\sqrt{|c_i^l| - 1} \cdot \mathrm{mean}\left(\xi_{u,c_*^l}\right)$ is a normalization factor. $\mathrm{std}\left(\xi_{u,c_*^l}\right)$ describes the cohesion degree of the community c_i^l relative to the community c_*^l. It can be seen that the greater the normalized $\mathrm{std}\left(\xi_{u,c_*^l}\right)$ is, the lower the cohesion degree of c_i^l is and the larger r_i^l is.

为什么要用

4.3 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 the problem into a two-step optimization procedure. First, we optimize Eq.3 with no constraints based on neural networks and obtain the Euclidean space representation of communities $\Omega(c_i^l)$. Second, with the pairwise relative Euclidean distances preserved, we project $\Omega(c_i^l)$ to a spherical surface to get the final representation of communities $\Phi(c_i^l)$. In the first step, we use the Adam optimizer [Rushing *et al.*, 2005] to solve the neural embedding problem. As the specific implementation has been well developed, we will not repeat it in detail. We mainly introduce the spherical projection method which can preserve the relative Euclidean distances between points.

Optimization Objective of Spherical Projection

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

$$\begin{aligned} \min_{\Phi} J^{(k)} = &\left\| \frac{D}{\|D\|_F} - \frac{B}{\|B\|_F} \right\|_F + \mu \exp(-\gamma \|B\|_F) \\ s.t. \quad &||\Phi(c_i^l) - \Phi(c_k^{l-1})|| = r_k^{l-1} \end{aligned} \tag{14}$$

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

$$\Psi(c_i^l) = \frac{\Phi(c_i^l) - \Phi(c_k^{l-1})}{r_k^{l-1}}. \tag{15}$$

Hence the constraints can be transformed as follows:

$$\|\Psi(c_i^l)\| = 1. \tag{16}$$

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

$$B'_{ij} = \|\Psi(c_i^l) - \Psi(c_j^l)\| \tag{17}$$

Bringing Eq. 16 into Eq. 17, we have:

$$\begin{aligned} B'_{ij} =& \|\Psi(c_i^l)\|^2 + \|\Psi(c_j^l)\|^2 - 2\Psi(c_i^l)^T\Psi(c_j^l) \\ =& 2 - 2\cos(\Psi(c_i^l), \Psi(c_j^l)) \end{aligned} \tag{18}$$

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

$$\min_{\Psi} H^{(k)} = \left\| \frac{D}{\|D\|_F} - \frac{B'}{\|B'\|_F} \right\|_F + \mu \exp(-\gamma\|B'\|_F). \tag{19}$$

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

$$\Phi(c_i^l) = \left(\frac{\Psi_*(c_i^l)}{\|\Psi_*(c_i^l)\|} + \Phi(c_k^{l-1}) \right) \times r_k^{l-1} \tag{20}$$

Obviously, the unconstrained objective has a lower bound. The minimum points can be found using Gradient Descent algorithm. In the experiment, we use Adam [Rushing *et al.*, 2005] to optimize the objective and find an ideal minimum point. The optimization algorithm is implemented on the Tensorflow platform, which can be accelerated with GPU.

5 Experiments

In this section, we first provide an overview of the datasets and methods which we use in our experiments. Besides, an experimental analysis of our method on both synthetic and real networks is presented. The code of our model is available on Github.

Algorithm 1 The GNE algorithm

function LEARNFEATURES(Network G, Hierarchical Clustering Tree T)
  Initialize $r_1^1 = 1$ and $\Phi(c_1^1) = 0$
  Φ = OptimizeRecursively(c_i^l, G, T, **r**, **Φ**)
return Φ

function OPTIMIZERECURSIVELY(Current node c_i^l, Network G, Hierachical Clustering Tree T, Radius Set **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**
    $\Phi(c_j^{l+1}) = \left(\frac{\Psi(c_j^{l+1})}{\|\Psi(c_j^{l+1})\|} + \Phi(c_i^1) \right) \times r_i^l$
    $r_j^{l+1} = \eta \cdot d_{j,*}$
  for all $c_j^{l+1} \in \Gamma(c_i^l)$ **do**
    Φ = OptimizeRecursively(c_j^{l+1}, G, T, **r**, **Φ**)
  return Φ

5.1 Experiment Setup

Data Sets An overview of networks we consider in experiments is given in Table 2 and Table 3.

- Dolphins data set [Lusseau *et al.*, 2003] is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand.
- Polbooks data set [pol,] is a network of books about recent US politics sold by the online bookseller Amazon.com. Edges between books represent frequent copurchasing of books by the same buyers.
- the Facebook networks [Traud *et al.*, 2012] comprise 100 colleges and universities. We just choose the social networks in Hamilton University and Amherst College.

Besides, in order to evaluate the performance of hierarchical community structure preservation, Hierarchical Random Graphs (HRG) with explicit hierarchical community structure are generated by [Clauset *et al.*, 2008], of which nodes are originated from the leaves of a tree and edges are derived from the sampling of connected paths between leaves in the tree.

Relevant Algorithms To validate the performance of our model, we compare it against with state-of-the-art models:

- DeepWalk [Perozzi *et al.*, 2014] is to map network into low-dimensional vector spaces by truncated random walks. The sampling strategy in DeepWalk can be seen as a special case of node2vec with $p = 1$ and $q = 1$.
- LINE [Tang *et al.*, 2015] is to map network into low-dimensional vector spaces by preserving the first- and second-order proximities of nodes.
- MNMF [Wang *et al.*, 2017] is to map network into low-dimensional vector spaces by incorporating the community structure into network embedding.

简单overview，做了什么实验，对比环境

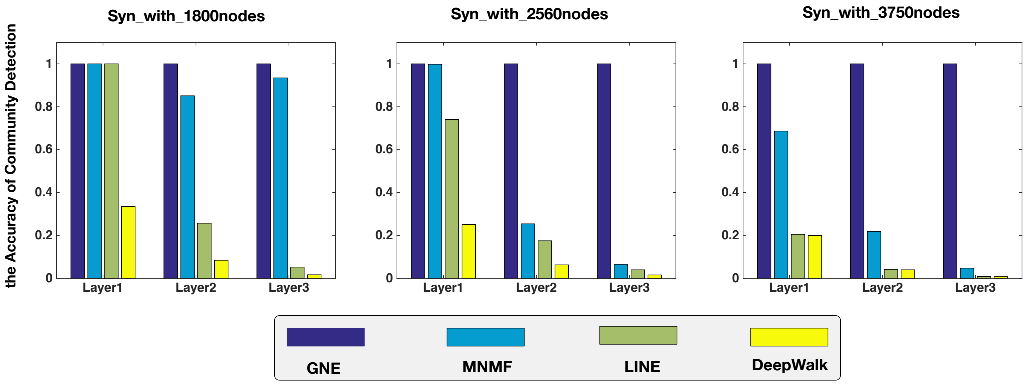
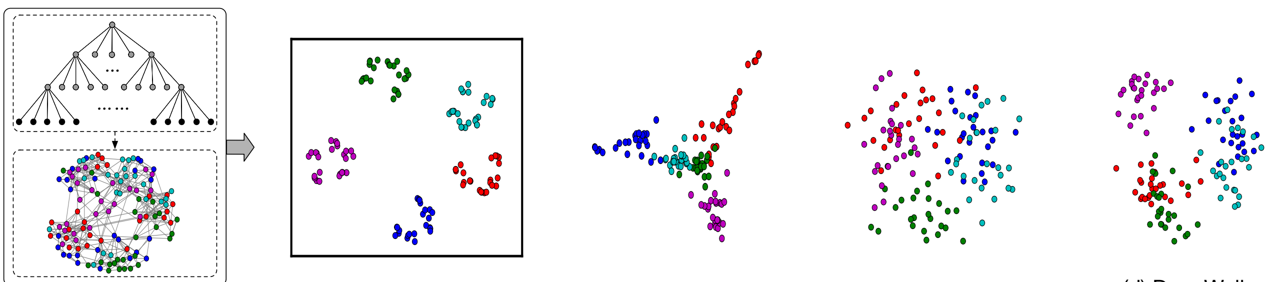


Figure 3: The comparision of hierarchical community preservation on different models. Three different structure of HRG but with the same number of layers are used. For each histogram, the horizontal ordinate denotes layer number, and the vertical ordinate denotes the Jaccard's coefficient.



(a) GNE (b) MNMF (c) LINE (d) DeepWalk

Figure 4: The visualization of node representations on different models

Model	Dolphins					Polbooks				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
GNE	**100**	**100**	**99.35**	**97.73**	**97.14**	**92.73**	**88.12**	**88.11**	**86.22**	**85.89**
MNMF	78.57	73.16	67.74	68.18	67.86	89.09	85.31	83.52	84.46	70.00
LINE	90.00	97.37	95.16	92.27	81.96	82.73	80.31	81.67	79.19	72.32
DeepWalk	100	99.47	98.71	97.73	95.00	87.27	83.44	85.56	84.19	84.42
Model	Amherst					Hamilton				
	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**
MNMF	89.82	89.06	88.04	86.43	78.44	91.42	90.32	89.12	87.02	81.19
LINE	90.76	91.82	91.48	91.09	89.42	92.33	92.72	92.52	92.62	91.73
DeepWalk	90.62	91.65	91.32	91.13	90.41	92.89	92.33	92.52	92.18	91.55

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

Additionally, we extract the hierarchical community structure of networks with [Tsvetovat and Kouznetsov, 2011] as the input of GNE.

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 Table 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.

Name	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert Y\rvert$
Dolphins	62	159	2
Polbooks	105	441	3
Hamilton	2314	96394	5
Amherst	2235	90954	5

Table 2: Real networks used in experiments. Y denotes the number of classes.

Name	$\lvert V\rvert$	$\lvert E\rvert$	Hierarchy
Syn_with_125nodes	125	406	[5,5,5]
Syn_with_1800nodes	1800	739637	[3,4,5,30]
Syn_with_2560nodes	2560	1460147	[4,4,4,40]
Syn_with_3750nodes	3750	3066250	[5,5,5,30]

Table 3: Hierarchical random networks used in experiments. The hierarchy denotes the number of children per node in each layer, e.g. the tree deriving the HRN with [3,4,5,30] indicates the root node in layer 0 has 3 children, and each node in layer1 has 4 children, and so on.

$$JC = \frac{|SS|}{|SS| + |SD| + |DS|} \tag{21}$$

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

It can be seen from Figure 4. that our model GNE embeds nodes on the different-scaled spherical surface hierarchically. It is evident that the node representations of GNE are consistent 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 Multi-label Classification

In order to verify the effectiveness of GNE on multi-label node classification, four real social networks with hierarchical community structure are used. The learned representations are used to classify the nodes into a set of labels. 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 has a better performance than other models on different percentage of test data size. Besides, our model is robust no matter how much the percentage of test data accounts for. Specifically, even with 10% of the samples training, 90% of the samples testing, the accuracy of our model can still reach 92.57% on average, 1.98% higher than DeepWalk, 8.71% higher than LINE, and 18.20% higher than MNMF.

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.

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, and Daniel Pagenstecher. These instructions are the same as the ones for IJCAI–05, prepared by Kurt Steinkraus, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab.

A LaTeX and Word Style Files

The LaTeX and Word style files are available on the IJCAI–18 website, `http://www.ijcai-18.org/`. These style files implement the formatting instructions in this document.

The LaTeX files are `ijcai18.sty` and `ijcai18.tex`, and the BibTeX files are `named.bst` and `ijcai18.bib`. The LaTeX style file is for version 2e of LaTeX, and the BibTeX style file is for version 0.99c of BibTeX (*not* version 0.98i). The `ijcai18.sty` file is the same as the `ijcai07.sty` file used for IJCAI–07.

The Microsoft Word style file consists of a single file, `ijcai18.doc`. This template is the same as the one used for IJCAI–07.

These Microsoft Word and LATEX files contain the source of the present document and may serve as a formatting sample.

Further information on using these styles for the preparation of papers for IJCAI–18 can be obtained by contacting pcchair@ijcai-18.org.

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.

[Lusseau *et al.*, 2003] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[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.

[pol,] Books about us politics. http://networkdata.ics.uci.edu/data.php?d=polbooks.

[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 *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.