Hierarchical Community Structure Preserving Network Embedding: A Subspace Approach

Qingqing Long*
Key Laboratory of Machine Perception (Ministry of Education), Peking University
qingqinglong@pku.edu.cn

Yiming Wang*
Key Laboratory of Machine Perception (Ministry of Education), Peking University
wangyiming17@pku.edu.cn

Lun Du*†
Microsoft Research
lun.du@microsoft.com

Guojie Song‡
Key Laboratory of Machine Perception (Ministry of Education), Peking University
gjsong@pku.edu.cn

Yilun Jin
Key Laboratory of Machine Perception (Ministry of Education), Peking University
yljin@pku.edu.cn

Wei Lin
Alibaba Group
yangkun.lw@alibaba-inc.com

ABSTRACT

To depict ubiquitous relational data in real world, network data are widely applied in modeling complex relationships. Projecting vertices to low dimensional spaces, quoted as Network Embedding, would thus be applicable to diverse predicative tasks. Numerous works exploiting pairwise proximities, one characteristic owned by real networks, the clustering property, namely vertices are inclined to form communities of various ranges and hence form a hierarchy consisting of communities, has barely received attention from researchers. In this paper, we propose our network embedding framework, abbreviated SpaceNE, preserving hierarchies formed by communities through subspaces, manifolds with flexible dimensionalities and are inherently hierarchical. Moreover, we propose that subspaces are able to address further problems in representing hierarchical communities, including sparsity and space warps. Last but not least, we proposed constraints on dimensions of subspaces to denoise, which are further approximated by differentiable functions such that joint optimization is enabled, along with a layer-wise scheme to alleviate the overhead cause by vast number of parameters. We conduct various experiments with results demonstrating our model's effectiveness in addressing community hierarchies.

CCS CONCEPTS

• **Networks** → *Network structure*; • **Information systems** → *Collaborative and social computing systems and tools*; • **Mathematics of computing** → Graph theory.

*These authors contributed equally to the work.
†Work performed as a student of Peking University.
‡Corresponding Author.

KEYWORDS

Network embedding; subspace; complex networks; community structure; data mining

ACM Reference Format:
Qingqing Long, Yiming Wang, Lun Du, Guojie Song, Yilun Jin, and Wei Lin. 2019. Hierarchical Community Structure Preserving Network Embedding: A Subspace Approach. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357947

1 INTRODUCTION

Network data have been ubiquitous due to its precise depiction of relational data. Traditional network algorithms being not applicable due to its forbidding computational costs, network embedding algorithms, projecting vertices within networks to lower-dimensional vector space while preserving general proximity between nodes, have proved to overcome such complexity and hence being applicable to a wider range of prediction tasks including link prediction and classification [14, 28, 37].

Apart from general proximity between pairwise nodes, a characteristic property real-world networks possess is the clustering property, namely nodes would tend to form communities with varying size and range at a far higher frequency than randomly connected. In addition, community structures are highly informative in that they shed light on how the network is inclined to be, with connections within communities being considerably more probable than those across communities, and hence provide network structures with resistance to noise links that occur randomly. Even more remarkably, not only do vertices form communities, the communities thus formed are commonly organized in a hierarchical manner as well, which proves to be significantly indicative of functional components of the underlying networks [3, 4].

Ubiquitous and instrumental as community structures and their hierarchy are, relatively few attention has been devoted to such properties. Typical attempts trying to preserve communities within vector spaces are MNMF [34] and GNE [7]. MNMF, aiming at preserving community structures within networks and reflect them in embedding networks ignores the hierarchy across communities, while GNE, preserving hierarchies across communities through spherical projection, suffers from drawbacks that, spherical projections tend to propel vertices across communities undesirably far from each other, resulting in incorrect modeling for vertices from different communities. In addition, communities lying deep within the hierarchy are treated with extremely small spheres yet possessing the same dimensionality as their shallower counterparts in GNE, and hence did not sufficiently exploit the inclusion relationship between high-level and lower-level communities.

It is hence concluded that, utilizing communities as well as their hierarchy is a field of pressing importance yet demanding challenges, with relatively few attempts devoted to it. Specifically, we summarize the following challenges yet to be resolved such that hierarchical community structures can be utilized.

(1) **Sparsity** Intuitively, communities lying deep within the hierarchy are less inclusive than their shallower counterparts, and hence possess less variance within themselves. Therefore, the original space in which the whole network is embedded to will be increasingly inappropriate for embedding deeper communities as they require far less variance to be encoded than is enabled by the original space. Consequently, embedding thus learned will suffer from extreme sparsity and as a result, noise arises, which will undermine the representation for deeper communities. We hence conclude that the sparsity issue for low-level communities should be addressed, and that dimensionality of spaces where communities reside should vary corresponding to their depth, a requirement scarcely met by previous research works.
(2) **Space Warps** As previously mentioned, spherical projection used by GNE suffers from restrictions imposed by the radii of the spheres, which decay exponentially as the hierarchy deepens. Consequently, it is common for vertices across communities to be exponentially distant than those within the same community, thus inappropriately underestimating density of links across communities and condensing links within communities. It is hence deduced that a desirable figure to which communities are projected to should be designed to ensure the correct modeling of nodes within and across communities and alleviate space warps.
(3) **"Curse" of Depth** Just like the "Curse of Dimensionality", when encountered with extremely deep hierarchies, it is not trivial that a sensible scale in which communities reside be still maintained, a counter-example of which would be, again, GNE. As explained previously, the radii of spheres to be projected to shrink exponentially with increasing depth, resulting in unduly tiny radii which may cause practical problems including underflow.

To address these challenges, we propose Subspace Network Embedding, abbreviated SpaceNE, to model the community structures in networks along with their hierarchy. Specifically, we observe that subspaces within Euclidean space inherently follow the organization of hierarchy. For example, in three-dimensional space as illustrated in Figure 1, planes and lines, which are subspaces of varying dimensions reside within the 3-d space, while as lines reside within planes and planes within the 3-d space, an inherent hierarchy is illustrated. In addition, natural metrics of measuring distances between pairwise subspaces exist, such as angles between pairwise planes and lines, which can be easily adopted to measure similarities between communities dwelling within subspaces of the same dimensions. Both the aforementioned factors, inherent hierarchy and handy metrics, contribute to our modeling of hierarchical community structure using subspace.

In addition, we are delighted to find that subspaces possess other appealing properties that can further enhance our modeling of community structures and their hierarchy. On one hand, the dimensionality of subspaces is highly flexible, which corresponds to the inconsistency of variance within communities of different depth within the hierarchy and alleviates the sparsity issue, thereby filtering undesirable noise and enhancing our representation. On the other hand, subspaces are flat and possess consistent scale of distance regardless of their dimensionality, which maintains distances within a community and across communities at comparable scales while allowing deep hierarchies to possess a similar scale of distance to their shallower counterparts, facilitating the modeling of arbitrarily deep community hierarchies.

Consequently, we extended DeepWalk [23], which preserves general proximity between pairwise vertices though co-occurence in random walks. In addition to proximity between vertices, with subspaces to model hierarchical community structures, we project vertices according to their communities, into subspaces of corresponding dimensions, during which objectives preserving similarities within a community and across communities are adopted, such that representations for communities, i.e. subspaces can be jointly optimized along with node representation vectors. What is more, we impose constraints on the dimensions of subspaces used to represent communities, keeping them as low as possible, such that a minimal level of redundancy and noise is kept, which was further approximated according to matrix algebra and convex optimization into a differentiable one, such that it can be optimized simultaneously along with proximity between pairwise vertices and communities.

To summarize, we make the following contributions:

- We propose Subspace Network Embedding model, abbreviated SpaceNE, introducing subspaces to the field of community preserving network embedding, which, to the best of our knowledge, is the first attempt of introducing subspaces into network representation learning.
- We design elaborate objectives preserving proximity between pairwise nodes, across communities, along with constraints on subspace dimension which was then approximated by a differentiable one, leading to efficient optimization of our model.
- We conduct extensive experiments on several real-world datasets, where experimental results demonstrate that SpaceNE is significantly more competitive to its various counterparts on various applications.

2 RELATED WORK

Hierarchical Network. Many real-world systems can be mapped into networks with hierarchical community structure [19][21]. Generally, a network community refers to a dense sub-network in which

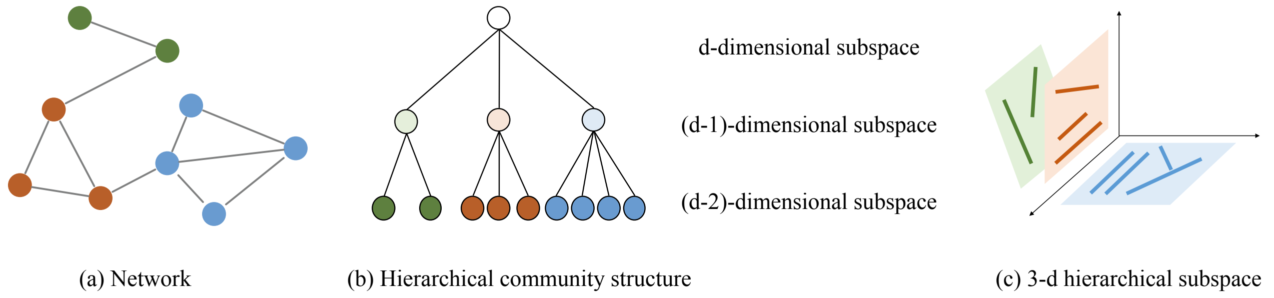
d-dimensional subspace
(d-1)-dimensional subspace
(d-2)-dimensional subspace
(a) Network
(b) Hierarchical community structure
(c) 3-d hierarchical subspace

Figure 1: The correspondence between the community hierarchy and the subspace hierarchy

vertices are densely connected one another [20]. The communities can be recursively divided into sub-communities. Thus, communities on different scales are hierarchically organized like a tree. Exploring the hierarchical community structure of network has proved to possess a wide range of applications, including scientific collaboration analysis [10], protein function prediction [26] and so on. Thus more and more works pay attention to networks containing a hierarchy of communities [3, 25]. [3] claims that knowing the hierarchical structure of complex networks is useful for link prediction tasks.

Network Embedding. Network embedding aims at embedding the network data to a low-dimensional space [5, 8, 33]. Deepwalk [23] learns vertex representations by using truncated random walk and Skip-Gram. LINE [28] first proposes the method to preserve the first-and second- order proximity among nodes. Struc2Vec [24] defines a hierarchy to measure node similarity at different scales, and constructs a multilayer graph to describe structural similarities.

As an important property of the network, the community structure has been extensively studied in network embedding. Recently, as for preserving the hierarchical community structure of the network, several models have been proposed. [34] leverages the matrix factorization to integrate the community structure information into the node embeddings. [22] learns the network representations in the hyperbolic space. Its limitation is the learned representations are hyperbolic vectors, which can not be applied to the majority machine learning algorithms whose inputs are usually Euclidean ones. Inspired by the structure of the Galaxy, [7] presents GNE that embeds the communities at different scales onto the surface of the spheres with different radii. However, this algorithm has obvious shortcomings. With the level increasing, the radii of spheres decrease quickly, which limits the representation spaces for the communities in deep levels. Therefore, the optimizing algorithm with a constant learning rate can not learn the proper representations for them.

Subspace. A subspace is a subset of a topological space endowed with the subspace topology [27]. Subspaces, inherently of lower variance than the original space, can be used to approximate data with higher dimensions such that only the principal features are kept within. Typical and widespread examples of utilizing subspaces are low-rank approximations and subspace clustering. Classical related words include [6, 15, 31], which all have achieved convincing results in their corresponding fields. Nonetheless, though widely explored in other fields, subspaces are largely overlooked in the field of network representation learning.

3 PRELIMINARIES AND PROBLEM STATEMENT

Definition 1 (***Hierarchical Network***)**.** Let $G(V, E)$ denote an undirected network with V as the vertex set and E being the edge set. T denotes the tree representing hierarchy of communities within the network G with depth L and node set C. For a node $c \in C$, $ch(c)$ and $pa(c)$ denotes its set of children and its parent node within T, respectively. We let $C_i^l \subseteq V$ denote the set of the i-th community within the l-th layer. Specifically, $C_1^1 = V$ is the original set of vertices of G.

Definition 2 (***Subspace***)**.** Let U be the vector space $\mathbb{R}^n$. U_s, a nonempty subset of U, is called a subspace[1] if for any element pair $x, y \in U_s$ and any scalar $\lambda \in \mathbb{R}$, $x + y \in U_s, \lambda x \in U_s$ [27][15]. Subspace has several important properties mentioned as follows to model hierarchical network embedding:

- **Hierarchical structure**. Subspaces within Euclidean space inherently follow the organization of hierarchy. For example, in three-dimensional space as illustrated in Figure 1, communities can be vividly depicted as planes, or 2-d subspaces of the 3-d space, and sub-communities can be consequently modeled as lines, or 1-d subspaces, residing in their corresponding communities, or planes. In addition, vertices, represented by points in Euclidean space, can reside in arbitrary subspaces of arbitrary dimension.
- **Lower Dimensional Representation**. The subspaces has proved to be instrumental in extracting principal features from extremely high-dimensional data, demonstrated distinctively by the famous PCA [35] decomposition algorithm, which selects the subspaces in which highest variances of the original data lie. Consequently, we would anticipate that subspaces would serve as a sieve, preserving those principal characteristics of the original networks and communities.

Definition 3 (***Hierarchical Subspace Network Embedding***)**.** The hierarchical subspace is denoted as T' with a depth of L'. $W =$

[1]There is no loss of generality in assuming that subspaces are linear, i.e., they all contain the origin. For the affine subspaces that do not contain the origin, we can always increase the dimension of the ambient space by one and identify each affine subspace with the linear subspace that it spans. So we always use 'subspace' to denote 'linear subspace' and 'affine subspace' in this work.

$[w_1, w_2, ..., w_n] \in \mathbb{R}^{n\times m}$ denotes a basis of a certain subspace and W_i^l denotes the i-th subspace at the l-th level of T'. The upper bound of the dimension of subspace is a hyper parameter. The descending trend of subspace dimension can be linear (eg. $b - a \cdot l$) or non-linear (eg. $\lfloor \log(b - l) \rfloor$), where b and a are positive integers. Community c_i^l should be corresponding to W_i^l in a hierarchical subspace. The objective of hierarchical community structure preserving network embedding is to learn the subspace representations (corresponding base vectors) of the input hierarchical network.

4 SUBSPACE BASED HIERARCHICAL NETWORK EMBEDDING

Subspaces follow a hierarchy naturally, which inspires us to come across a subspace based hierarchy preserving network embedding approach. SpaceNE preserves pairwise proximity and proximities between hierarchical communities, including structural information within a community and among communities.

4.1 Preserving Pairwise Proximity Between Vertices

We propose that preserving pairwise proximity between vertices should be most fundamental in general network embedding problems. Hence we first address the pairwise proximity between nodes similar to DeepWalk, as mentioned before.

$$\begin{aligned}\min_{\vec{u}} \ell_1 = &\sum_{v_i\in V}\sum_{v_j\in N(v_i)} \log \sigma(\|\vec{u}_j - \vec{u}_i\|_2) \\ &+ k\cdot \mathbb{E}_{v_n\sim P_n(v)}[\log \sigma(-\|\vec{u}_i - \vec{u}_n\|_2)],\end{aligned} \tag{1}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, $\vec{u}_i$ is the representation vector of v_i, $N(v_i)$ denotes the "neighborhood" of v_i, i.e. vertices co-occuring with v_i in random walks, and $P_n(v)$ is the noise distribution for negative sampling.

4.2 Preserving Proximity of Hierarchical Communities

4.2.1 Preservation of Structure within Individual Communities. Vertices from the same community should be closer to each other than those from different communities, thus we project nodes from the same community into the same subspace. To integrate the hierarchical subspace information, for each community c_i^l in layer l we have the following constraint:

$$rank(\boldsymbol{U}_i^l) \le d^l, \tag{2}$$

where d^l is the dimension of the subspace in layer l, a hyperparameter, and $\boldsymbol{U}_i^l$ is a matrix each line of which is a representation vector of a vertex, belonging to community c_i^l. Each vertex v_i in its corresponding community has a new vector representation under base-vectors of the corresponding subspace, which is denoted as $\vec{u}_i^l$. Hence the variable $\vec{u}_i$ in Eq. (1) becomes an alias of $\vec{u}_i^1$.

Considering that such constraints make the problem difficult to solve, we equivalently transform the way of introducing constraints through vertex projection layer by layer. Specifically, we change the decision variables from $\vec{u}_i^1 \in \mathbb{R}^{d_1}$ into $\vec{u}_i^L \in \mathbb{R}^{d_L}$ where $d_L < d_1$, and introduce auxiliary decision variables $S_j^l \in \mathbb{R}^{d^l\times d^{l-1}}$ denoting

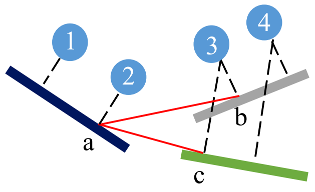

Figure 2: Illustration of relationship preservation among communities. Distance of node embeddings may change before and after projection, as there are angles among the projection subspaces.

the projection matrix, i.e. there exists the following relationship:

$$\vec{u}_i^l = S_j^l \vec{u}_i^{l-1},\ \vec{u}_i^{l-1} = S_j^{l\dagger} \vec{u}_i^l, \tag{3}$$

where j is the index of community c_j^l that the vertex v_i belongs to in the l-th layer, and $S_j^{l\dagger}$ is the pseudo-inverse of the matrix S_j^l.

4.2.2 Preservation of Structure among Communities. However, the relationship among communities may be lost if only the hierarchical projection method above is used. Actually, the relationship among different subspaces is able to reflect the relationship among communities after adding the subspace constraints. For example, as shown in Fig.2, the projected position of node 3 in the grey plane is b, and c in the green plane. The distance between node 2 and node 3 varies with the planes to which node 3 is projected. The main idea of formulating this objective is to minimize the difference of subspace similarity and community similarity between every two communities i, j, namely,

$$\min \quad \ell_2^l = ||\boldsymbol{\Delta}^l - \boldsymbol{\Gamma}^l||_F, \tag{4}$$

where $\boldsymbol{\Delta}^l$ is a matrix, each entry of which $\Delta_{i,j}^l$ is the similarity of two communities c_i^l and c_j^l in the original graph. Similarly, $\Gamma_{i,j}^l$ is the similarity of the two subspaces . The method of calculating Δ can be customized according to the datasets. For example, the community similarity, based on PMI [13] or common neighbors [7], etc, is reasonable. The relationship between two subspaces can be measured by the inner product of projection matrices [27]. Thus the subspace similarity $\Gamma_{i,j}^l$ can be defined as

$$\Gamma_{i,j}^l = ||S_j^l \boldsymbol{U}_i^{l-1}(S_j^l \boldsymbol{U}_i^{l-1})^T||_F. \tag{5}$$

4.2.3 Low Rank Representation. As stated before, inconsistent demand for dimensionality poses a significant threat on the representation of subspaces. On one hand, inappropriate dimensionality for modeling communities, especially those lying deep within the hierarchy, would result in redundancy in dimensionality and henceforth sparsity and noise, which would compromise the quality of representations of vertices and communities. On the other hand, in order to get rid of undesirable noise while conserving the principal features of the original networks, we would prefer a slimmer subspace such that only the variances that are vital enough are kept while leaving those obscure behind.

Consequently, we are motivated to impose penalties on the dimensionality needed for subspaces, which corresponds to the rank of projection matrices. Therefore, we add the following constraint on projection matrices:

$$\min_{S} \quad \ell^l_{reg} = \sum_{i=1}^{k} rank(S^l_i), \tag{6}$$

where *rank* of a matrix is the number of linearly independent columns which it contains and k is the number of communities at each layer. Since $S \in \mathbb{R}^{m \times n}$ is the projection matrix, the projected space lies within the column space of S. We can minimize the rank of S with Eq.(6) to reduce the dimension of the projected subspace. Eq.(6) can restrict S^l_i to be of rank as low as possible, resulting in desirable low-dimensional subspaces, which further improves the capability of denoising.

Considering all the objectives mentioned above, the overall objective function is:

$$\min_{\boldsymbol{U}^L, S} \ell_3 = \ell_1 + \alpha \sum_{l=1}^{L} \ell^l_2 + \lambda \sum_{l=1}^{L} \ell^l_{reg}, \tag{7}$$

where α and λ are the coefficients regulating the contributions of two community preserving targets.

5 LEARNING PROCEDURE

The optimization of Eq.(7) is extremely difficult due to the following two problems. First, it has a massive number of parameters that need to be optimized at a single iteration. Thus directly optimizing may lead to vanishing gradient [11]. Second, it has a discrete term of ranks, which cannot be solved directly by differentiating. Correspondingly we propose two alternatives to address the above problems.

5.1 From Global to Layer-wise Optimization

It is obvious that the overall objective function has a massive number of complicated parameters, such as matrix inversion, which may lead to an unacceptable inefficiency and vanishing gradient. Inspired by the layer-by-layer training in auto-encoder [12], we consider learning the parameters hierarchically in our method. Specifically, we first optimize pairwise proximity Φ^{local}_i through Eq.(1), which is followed by adding the constraints to Φ^{local}_i of each layer in the subspace hierarchy. On the other hand, unitary transformation does not change the relative position of vectors [36], thus it's reasonable to learn the parameters layer by layer, alleviating the need to project nodes all the way down to the bottom of the hierarchy.

The hierarchy subspace constraints are essentially projections among spaces, while other constraints are naturally disassembled in the layer-wise optimization. This projection is similar to the general idea of PCA [35], which selects the spaces in which maintains the nearest reconstruction of vertices before and after projection with high efficiency. Based on the following Lemma.1, we extend PCA to describe the projection of all inner-community nodes in the subspace:

$$\min \quad \ell^l_4 = \quad \sum_{i=1}^{k} -tr((S^l_i)^T \boldsymbol{U}^l_i (\boldsymbol{U}^l_i)^T S^l_i), \tag{8}$$

where $tr(M)$ is the trace of M. Eq.(8) constructs a low-dimensional representation of the data, which omits complicated and inefficient matrix inversion, and improves the capability of denoising further.

Lemma 1. If Eq.(8) achieves desired optimal solution, the optimal result learned from recursively optimization is the same as that of joint optimization (Eq. (1)) with constraints in Section 4.2.1.

Proof. Let $d(\vec{u}_i, \vec{u}_j)$ be the Euclidean distance of vertices v_i and v_j. The vector $\vec{u}_i$ after projection is denoted as $\vec{u}'_i$. Based on Trigonometric Inequalities [1] in Euclidean Space, here comes

$$d(\vec{u}_i, \vec{u}_j) \le d(\vec{u}_i, \vec{u}'_i) + d(\vec{u}'_i, \vec{u}'_j) + d(\vec{u}_j, \vec{u}'_j),$$

$$d(\vec{u}'_i, \vec{u}'_j) \le d(\vec{u}_i, \vec{u}'_i) + d(\vec{u}_i, \vec{u}_j) + d(\vec{u}_j, \vec{u}'_j).$$

The projection of PCA[35] maintains the nearest reconstruction of vertices before and after projection, which means

$$\lim_{Eq.(8) \to 0} d(\vec{u}_i, \vec{u}'_i) \to 0, d(\vec{u}_j, \vec{u}'_j) \to 0$$

Thus,

$$d(\vec{u}'_i, \vec{u}'_j) \to d(\vec{u}_i, \vec{u}_j).$$

It means that relationships among vertices before and after projection are maintained. □

5.2 From Discrete to Differentiable Optimization

Rank minimization turns out to be an NP-hard combinatorial problem that is computationally intractable in practical cases. Meanwhile, the rank minimization is discontinuous, so the objective function cannot be optimized with SGD. The tightest possible convex relaxation of function (13) is to replace the rank with the nuclear norm equal to the sum of its singular values [2]:

$$rank(M) \approx ||M||_*, \tag{9}$$

where $||M||_*$ is the nuclear norm of matrix M.

As nuclear norm is discontinuous, it cannot be optimized with SGD as well. [18] gives a smooth approximation via conjugate for nuclear norm. It is the sum of the Huber penalties applied to the singular values of M.

$$||M||_* \approx \sum_i \phi_\mu(\Theta_i(M)), \tag{10}$$

where $\Theta_i(M)$ is the singular values of M, and ϕ_μ is

$$\phi_\mu(Z) = \begin{cases} \dfrac{Z^2}{2\mu}, & |Z| \le \mu \\ |Z| - \dfrac{\mu}{2}, & |Z| > \mu \end{cases} \tag{11}$$

where μ controls accuracy and smoothness, Z is the result of $\Theta_i(M)$.

Thus the approximated regularization term ℓ'_{reg} is Eq.(12), and the **overall objective function** ℓ in the l-th layer is approximated as Eq.(13):

$$\ell^l_{reg'} \approx \sum_{i=1}^{k} \sum_m \phi_\mu(\Theta_m(S^l_i)), \tag{12}$$

$$\min_{\boldsymbol{U}^L, S} \quad \ell^l = \ell^l_4 + \alpha \ell^l_2 + \lambda \ell^l_{reg'}. \tag{13}$$

It should be noted that, to ensure that the base vectors of the subspace of each level are orthogonal, S_i^l is orthogonalized [1]. Another advantage of layer-by-layer learning is to replace the pseudo-inverse calculation in the origin problem with a matrix orthogonalization calculation over S_i^l of each layer.

Algorithm 1 The SpaceNE algorithm

```
function LearnFeatures(Network $G$, Hierarchical Clustering Tree $T$)
    $\mathbf{U}^{\mathbf{local}}$ = minimize $\ell_1$ with Adam (see Eq.(1))
    $\mathbf{U}$ = RecursiveOptimization ($c_1^1$, $T$, $\mathbf{U}^{\mathbf{local}}$)
return U, S

function RecursiveOptimization(Current node $c_i^l$, Hierachical Clustering Tree $T$, Representation Set $\mathbf{U}^{\mathbf{local}}$)
    if $l = L$ then return U, S
    $\{S_j^{l+1}\}$ = minimize $H_i^{(l)}$ with Adam (see Eq.(13))
    for all $c_j^{l+1} \in Ch(c_i^l)$ do
        $S_j^{l+1} = Orthogonalize(S_j^{l+1})$
        $U_j^{l+1} = S_j^l U_i^l$
    for all $c_j^{l+1} \in Ch(c_i^l)$ do
        U, S = RecursiveOptimization ($c_j^{l+1}$, $T$, $\mathbf{U}^{\mathbf{local}}$)
    return U, S
```

5.3 The Subspace Algorithm

Algorithm 1 is the pseudocode of SpaceNE. In the function *Learn-Features*, we optimize the local objective Eq.(1) to learn the node embedding vectors preserving local information, and then adjust the embedding vectors to add subspace constraints from top to down. In the function *RecursiveOptimization*, the set of projection matrices $\{S_j^{l+1}\}$ can be obtained by optimizing the objective (13) and orthogonalization. We obtain the node representations of the next layer $\boldsymbol{U}_j^{l+1}$ by matrix multiplication.

Time Complexity Analysis Without loss of generality, we assume that T is a k-ary tree and the dimension of subspaces in l-th layer decreases exponentially, i.e. $d^l = \lfloor \log_k(D-l) \rfloor$. SpaceNE should be done on each tree node layer by layer and the learning procedures on the same layer tree nodes can be parallelized. For a certain node c_i^l in T, the complexity of calculating the loss ℓ_2^l is $O(k(\log_k(D-l))^2)$, and the complexity of the loss ℓ_4^l is $O((k \log_k(D-l))^2)$. Therefore, the time complexity of optimizing the overall loss ℓ is $O(\mathcal{E}(k \log_k(D-l))^2)$ where $\mathcal{E}$ is the number of training epochs. Thus the total complexity of SpaceNE is $O(|V|\mathcal{E}(k \log_k D)^2)$. The optimization algorithm is implemented on the Tensorflow platform, thus can be accelerated with GPU.

6 EXPERIMENTS

6.1 Experiment Setup

Datasets. We employ the following real networks in the Facebook social networks [30]. The datasets in our paper and their basic attributes are shown in Table 1. Moreover, synthetic network consisting of a 6-layer hierarchy is used to evaluate the performance of hierarchical structure preservation. We generate a self-similar network with a hierarchical network generation algorithm introduced in [10]. Specifically, we first generate a 6-layer trident tree, and use the leaves of the tree as the nodes of the network. Then we add edges to the node pairs with probability proportional to path length among them in the k-ary tree.

Datasets	Vertices	Edges	Layers	Classes	Labels
Amherst	2314	96394	2	5	✓
Georgetown	9414	425639	2	5	✓
UC	16808	522148	2	5	✓
Sync_6	729	21735	6	×	×
Sync_show	125	982	4	×	×
Amherst_noise	2314	91409	2	5	✓

Table 1: Data set statistics and properties.

Relevant Algorithms. In the experiments, we compare SpaceNE against the following baselines:

- MNMF [34]: a single-layer community structure preserving baseline, which integrates the community information through a matrix factorization.
- GNE [7]: a multi-layer community structure preserving baseline, which embeds communities onto surface of the spheres.
- DeepWalk [23]: a method combines truncated random walks [9] with Skip-Gram [17] model to learn vertex embeddings.
- LINE [28]: a popular baseline based on preserving the first- and second- order relational information among vertices.
- Struc2Vec [24]: measures node similarity at different scales, and uses a multilayer graph to encode structural similarities.
- SpectralClustering [29]: learns the vertex representations by factorizing the Laplacian Matrix.

Parameter Settings. We conducted many experiments, and the optimal default parameters are selected as follows: $\mu = 10$, $\alpha = 1$, $\lambda = 10^3$. In our experiments, the embedding size m of all models is 64.

Besides, the parameter setting of comparison models follows the recommended settings in relevant code packages. Specifically, for DeepWalk, the walk length is set to 40 and window size is set to 10. For GNE, the scaling radius is set to 3.0, λ is set to 0.2, θ is set to 3, the initial radius is set to 10^5, min radius is set to 0.05, max radius is set to 0.25, and the scaling radius is set to 3.0. For LINE, we set the number of negative samples used in negative sampling as 5, the starting value of the learning rate as 0.025, and the total number of training samples as 100M. We consider both fisrt- and second-order information in LINE. For Struc2Vec, the walk length is set to 10, the number of walks is set to 80, the stay probability is set to 0.3. For MNMF, the number of clusters to is set to 10, λ is set to 0.2, β is set to 0.05, η is set to 5.0, the parameter "lower-control" is set to 10^{-15}. For SpectralClustering, we use PCA to reduce dimension of the laplacian matrix.

6.2 Alleviating Sparsity

To evaluate the performance of alleviating sparsity of SpaceNE, we conduct experiments on node classification and its resistance to randomly added noises on edges, with respect to link prediction.

Model	Amherst				Georgetown				UC			
	30%	50%	70%	90%	30%	50%	70%	90%	30%	50%	70%	90%
SpaceNE	92.52	93.11	**93.74**	**95.09**	**56.12**	**56.42**	**56.92**	**56.54**	**88.69**	**89.02**	**89.23**	**90.07**
GNE	**93.17**	**93.33**	93.26	93.52	52.19	53.53	53.75	53.12	87.78	88.42	88.42	87.57
MNMF	87.11	88.04	89.23	89.96	51.52	51.69	51.60	53.25	87.89	87.95	88.09	88.10
DeepWalk	91.09	91.26	91.71	92.03	51.45	53.25	53.76	54.03	88.35	88.42	88.51	88.63
LINE	91.11	91.53	91.89	91.67	51.35	51.93	52.18	52.38	87.71	87.88	87.95	87.53
Struc2Vec	72.72	73.35	73.92	77.23	46.85	47.44	48.33	47.59	87.96	87.89	88.11	88.25
SpectralClustering	72.88	73.51	73.89	74.41	49.67	50.02	50.79	51.23	84.23	84.35	84.31	84.21

Table 2: The multi-label classification results on different percentages of train datasets

6.2.1 Node Classification. Three real-world social networks of the Facebook datasets with a four-layer hierarchical tree (including root and leaves) are used in the vertex classification experiment. The two intermediate layers are divided by the enrollment year and major, respectively. For MNMF, we use enrollment year as an indicator for community division. The learned representations are used to classify the vertices into a set of labels. The classifier we used is Logistic Regression, 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 2 shows that SpaceNE performs well in most cases. Although MNMF, GNE and SpaceNE all take community information into consideration, SpaceNE still performs better, which means the low-rank representation of SpaceNE plays an important role. Although considering some global network structure, Struc2Vec does not work very well. Compared with Amherst, our model performs better on the Georgetown dataset and the UC dataset. It is because our model is more stable on larger datasets after integrating the hierarchical community structure information.

6.2.2 Resistance to Random Noise. Besides preserving the deep hierarchical community structure, SpaceNE tends to be more resistant to random noises due to the inherent properties of subspaces, hence generating node embeddings robust enough regardless of noise. On one hand, the reconstruction optimization term (Eq.(8)) of SpaceNE is similar to PCA, which maintains the noise reduction feature of it. On the other hand, the low rank optimization term (Eq.(6)) of SpaceNE can learn the most important features of the network and then improve the resistance to noise[15]. In order to verify the resistance to noise of SpaceNE, we conduct several experiments and compare the results of SpaceNE with other community or global structure preserving methods (MNMF, GNE and Struc2Vec). We construct the noisy data according to the method introduced by [32]. Specifically, the dataset we used is Amherst_noise (see Table 1). We contaminate Amherst by adding 5% the number of total edges and delete 5% of them in Amherst randomly. Then, we conduct link prediction tasks using SpaceNE and its counterparts on Amherst and Amherst_noise datasets for comparison.

The results are shown in Table 3. The values in Table 3 represent the reduction in percentage in precision and recall of the algorithms after adding the noise. "Gain of SpaceNE" is the improvement of SpaceNE with respect to the best of other baseline methods. It can be seen from Table 3 that all the algorithms go worse after contamination, but SpaceNE still performs better than its counterparts. Moreover, we can find that compared with other community preserving methods (MNMF and GNE), SpaceNE can not only preserve a deeper hierarchical community structure, but it is also less affected by noise in a complex network.

Model	precision	precision	recall	recall
	70%	90%	70%	90%
SpaceNE	**-0.54**	**-0.24**	**-0.01**	**-0.01**
GNE	-1.26	-0.96	-0.03	-0.26
MNMF	-2.71	-2.05	-0.02	-0.06
Struc2Vec	-1.01	-0.99	-0.02	-0.02
Gain of SpaceNE [%]	46.53	75	50	50

Table 3: The link prediction results on different percentages of the training dataset, Amherst_noise. The values are reduction in precision (or recall) of the algorithms after adding the noise.

6.3 Alleviating Space Warps

To evaluate how SpaceNE alleviates space warps, we conduct experiments on link prediction, verifying whether it relieves the problem owned by its community-aware counterparts. In addition, we present an illustration of how GNE and SpaceNE preserve distances between pairwise, intra-community nodes as the depth increases.

6.3.1 Link Prediction. We conduct link prediction experiments on all three datasets. We sample a proportion of edges from the initial network, which are used as positive samples, along with an identical number of random negative edges. We take the inner product of embedding vectors as the score for each sample, which are then ranked where the samples with their score in top 50% are predicted as "positive" edges. We report precision as the metric, which is equivalent to recall as we use an identical number of positive and negative samples.

The results of are reported in Table 4. Intuitively, community-aware methods are inherently compromised in link prediction tasks as, compared to Skip-Gram based model which preserves primarily proximity between pairwise nodes, their community-aware counterparts tend to sacrifice such pairwise properties for better modeling of clustering properties and hence warping the space where vertices reside, which can be exemplified by GNE and MNMF being gravely defeated by Skip-Gram models including DeepWalk. In contrast, our model, SpaceNE, is able to address more complex community structures without compromising its performance compared with its Skip-Gram counterparts. It is hence concluded that our modeling of communities using subspaces possesses a wider range of applicable fields in that it does not enhance community structures at the expense of other key properties.

6.3.2 Distance between pairwise, intra-community nodes. In order to demonstrate why SpaceNE performs better on the task of link prediction than other community preserving methods, here we show the average distance between pairwise, intra-community nodes, which will be affected most significantly by space warps in GNE.

As SpaceNE is able to preserve the distances among nodes reasonable without making them too close as GNE, we calculate the average distance of intra-community node pairs on different layers. Specifically, the average distance of intra-community node pairs on layer l is calculated as

$$\frac{1}{|M^l|}\sum_{i=1}^{|M^l|}\left(Avg\_Dis(C_i^l)\right), \tag{14}$$

where M^l is the number of communities on the l-th layer. Avg_Dis is the average distance of all nodes pairs belonging to the i-th community on the l-th layer C_i^l, which can be calculated by

$$Avg\_Dis(C_i^l) = \frac{1}{|C_i^l||C_i^l - 1|}\sum_{x,y\in C_i^l, x\neq y} D(x,y). \tag{15}$$

$D(x,y)$ is the Euclidean distance between vector representations of node x and node y. We use Sync_6, introduced previously as an artificial network consisting of a 6-layer community. Fig.3 shows the results with varying numbers of layers, where the y-coordinate scale is **logarithmic**. Fig. 3 shows that the average distance decays exponentially in GNE, while linearly in SpaceNE. This result partly explains why SpaceNE is superior to GNE in essence: the nodes are too close to each other so that they can not be distinguished.

Model	Amherst	Georgetown	UC
SpaceNE	85.61	89.28	91.32
GNE	62.07	68.97	51.25
MNMF	48.89	49.76	50.05
DeepWalk	86.40	89.16	91.39
LINE	74.37	76.58	71.22
Struc2Vec	51.77	49.94	46.83
SpectralClustering	37.76	40.63	38.68

Table 4: The link prediction results on different datasets.

6.4 Eliminating the "Curse" of Depth

To evaluate the performance of modeling extremely deep hierarchies, experiments are conducted on deep hierarchical community detection task. The basic properties of our dataset "Sync_6" are shown in Table 1. "Sync_6" is a 6-layer synthetic dataset. We applied K-means to the learned node representations. We compare the differences between the clustering results and the prior community division at different layers of the hierarchical community tree. Adjusted rand index (ARI), ranging from -1 to 1, is an index reflecting the similarity between two sets, thus it is applied as an index to

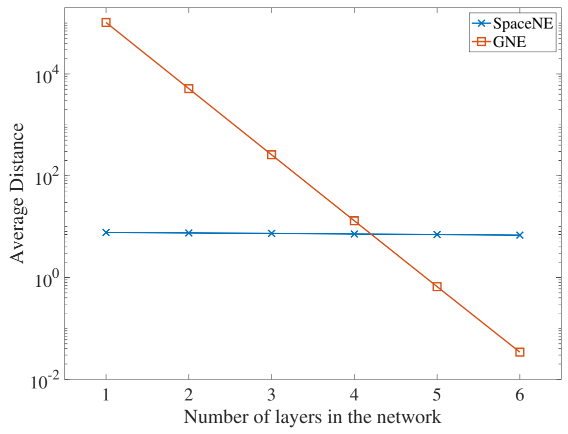

Figure 3: Average Distance with the increasing of layers

evaluate the performance of community preservation. The closer ARI is to 1, the better the performance of community detection is.

Fig.5 illustrates the effect of the number of layers on the hierarchical community detection (excluding the root, for there is only one community in the root layer). The results show that the hierarchical community structure can be integrally preserved by SpaceNE, no matter how deep the hierarchical community or how many communities exist within the network. Obviously, when the number of layers reaches 4, the radii of spheres in GNE are unduly small, thus the accuracy of community detection begins to decline, which is probably cause by underflow as the number of layers increase and the spheres shrink.

Besides, the experimental results illustrate that DeepWalk and LINE are incompetent at structure preservation. While DeepWalk can preserve part of the hierarchical information in the process of random walks, the results show when the number of layers exceeds 4, the performance of DeepWalk drastically declines. LINE performed even worse compared to DeepWalk, as LINE only considers the local structure as far as 2 steps away of each node and does not model the hierarchical community information. MNMF only preserves community at some layers but not all, leading to poor performance after a few layers. It is worth mentioning that as a global structure preserving algorithm, Struc2Vec performs the best at layer 5, while becomes worse at the deeper layers.

6.5 Efficiency

To demonstrate the efficiency advantage of SpaceNE, we compared the running time of SpaceNE, GNE, Struc2Vec and MNMF, methods capturing the community or global structure. The three datasets from small to large are described in Table 1. All efficiency experiments were conducted on a single machine with a 12GB memory GPU. Results are presented in Fig. 6. MNMF only considers a single-layer community, thus the running time is shorter. Although Struc2Vec captures some global structural information, its optimization is too slow, which is not suitable for large networks. Although GNE and SpaceNE have similar time complexity in theory, the convergence of SpaceNE is faster. The reason is that GNE preserves the

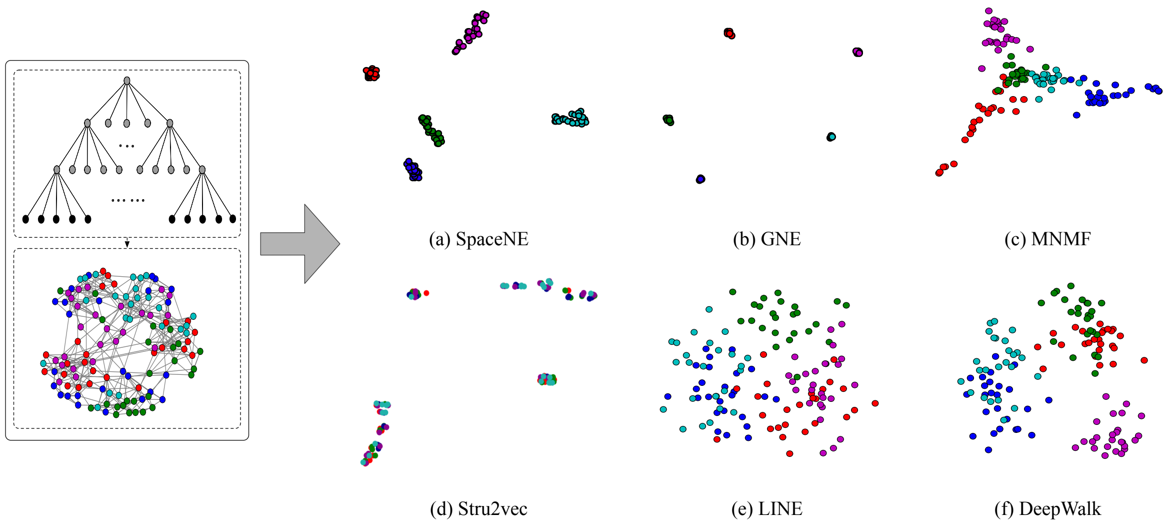

Figure 4: The visualization of vertex representations in 2-D space from different models.

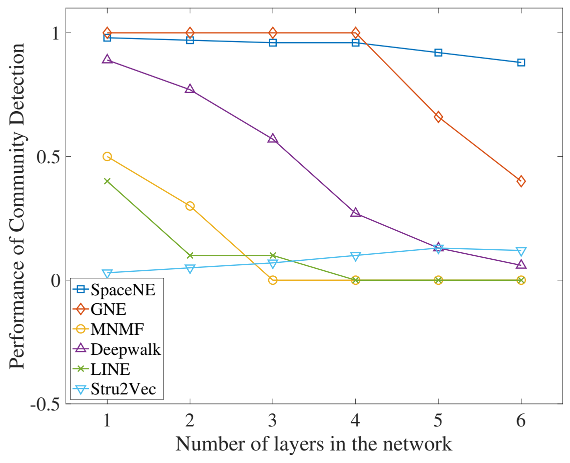

Figure 5: The comparison of hierarchical community preservation on different models. A 6-layer generated hierarchical networks is used.

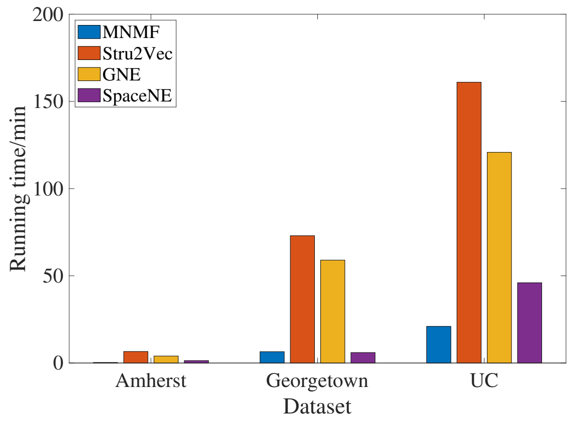

Figure 6: The running time on different datasets.

community structure through projections to the sphere. Compared with the linear matrix multiplication of subspace, the optimization of GNE is more complex and requires more epochs. The result shows that SpaceNE is scalable to large networks.

6.6 Visualization

We visualize the "Sync_show" network used in [7]. Fig.4 shows the visualization experiments. For all experiments in the comparison, we first embed the network into low-dimensional spaces, and then map the low-dimensional vectors of the vertices to a 2-D space with t-SNE[16] package. Note that [7] maps GNE **directly into two-dimensional space**, which is different from the other methods. Thus in order to unify the experimental standards, we do not use the method in this work. We use t-SNE to conduct dimensionality reduction for all methods.

Fig.4 shows that the SpaceNE preserves both relationship among communities and relationship within the community. Although GNE keeps the relationship among communities, the nodes from the same community are too close to each other, which means the relationship among the nodes in the same community is ignored. This proves the space warp of GNE again, from a new view. Additionally, SpaceNE has an outstanding performance on clustering vertices compared with other methods.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed SpaceNE, introducing subspaces to the field of community network embedding. To the best of our knowledge, this work is the first attempt of introducing subspaces into network representation learning. Specifically, we design elaborate objectives preserving proximity between pairwise nodes, across communities, along with constraints on subspace dimension which was then approximated by a differentiable one, leading to efficient optimization of our model. Empirically, we verify SpaceNE in a variety of datasets and applications. Extensive experimental results demonstrate the advantages of SpaceNE, especially on link prediction, hierarchical community preservation.

Here we focus on the hierarchical network embedding by using subspace theory. Nevertheless, the theory of subspace is largely overlooked in the field of network representation learning. For future work, one intriguing direction is utilizing subspace theory to deal with the heterogeneity of complex networks. Also, in real-life scenes, such as neighborhood-based recommendation, when searching for the nearest neighbor of an item, the search engine only needs to search in the lower-dimensional subspace, which can improve the efficiency greatly.

ACKNOWLEDGMENTS

We are thankful to Yizhou Zhang for his helpful suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 61876006 and No. 61572041).

REFERENCES

[1] Åke Björck. 1994. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications* 197 (1994), 297–316.

[2] Emmanuel J Candes and Benjamin Recht. 2009. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717–772.

[3] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* 453, 7191 (2008), 98.

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

[5] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A Survey on Network Embedding. *IEEE Transactions on Knowledge and Data Engineering* (2018), 1–1.

[6] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. 2006. R 1-PCA: rotational invariant L 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 281–288.

[7] Lun Du, Zhicong Lu, Yun Wang, Guojie Song, Yiming Wang, and Wei Chen. 2018. Galaxy network embedding: a hierarchical community structure preserving approach. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2079–2085.

[8] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. 2018. Dynamic network embedding: an extended approach for skip-gram based network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2086–2092.

[9] Francois Fouss, Alain Pirotte, Jeanmichel Renders, and Marco Saerens. 2007. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 19, 3 (2007), 355–369.

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

[11] Robert Hechtnielsen. 1988. Theory of the backpropagation neural network. *Neural Networks* 1 (1988), 445–448.

[12] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[13] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*. 2177–2185.

[14] Ziyao Li, Liang Zhang, and Guojie Song. 2019. Sepne: Bringing separability to network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4261–4268.

[15] Guangcan Liu, Zhouchen Lin, and Yong Yu. 2010. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 663–670.

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

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

[18] Yurii Nesterov. 2005. Smooth minimization of non-smooth functions. *Mathematical Programming* 103, 1 (2005), 127–152.

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

[20] M E J Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* 74, 3 (2006), 036104.

[21] M E J Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113–026113.

[22] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*. 6338–6347.

[23] Bryan Perozzi, Rami Alrfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. *Knowledge Discovery and Data mining* (2014), 701–710.

[24] Leonardo Filipe Rodrigues Ribeiro, Pedro H P Saverese, and Daniel R Figueiredo. 2017. struc2vec : Learning Node Representations from Structural Identity. *knowledge discovery and data mining* (2017), 385–394.

[25] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao Bin Hu. 2009. Detect overlapping and hierarchical community structure in networks. *Physica A Statistical Mechanics & Its Applications* 388, 8 (2009), 1706–1712.

[26] Victor Spirin and Leonid A Mirny. 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America* 100, 21 (2003), 12123–12128.

[27] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. 1993. *Introduction to Linear Algebra*. Vol. 3. Wellesley-Cambridge Press Wellesley, MA.

[28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *International Conference on World Wide Web*. 1067–1077.

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

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

[31] René Vidal. 2011. Subspace clustering. *IEEE Signal Processing Magazine* 28, 2 (2011), 52–68.

[32] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierreantoine Manzagol. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* 11 (2010), 3371–3408.

[33] Junshan Wang, Zhicong Lu, Guojia Song, Yue Fan, Lun Du, and Wei Lin. 2019. Tag2Vec: Learning Tag Representations in Tag Networks. In *The World Wide Web Conference*. ACM, 3314–3320.

[34] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding. In *Association for the Advancement of Artificial Intelligence Conference*.

[35] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[36] Zi Yin and Yuanyuan Shen. 2018. On the Dimensionality of Word Embedding. *neural information processing systems* (2018), 895–906.

[37] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. 2019. DANE: Domain Adaptive Network Embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press.