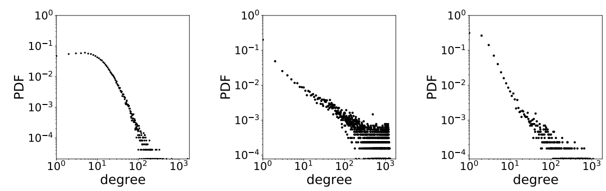
Representation Learning for Scale-free Networks

Rui Feng*, **Yang Yang**†*, **Wenjie Hu, Fei Wu, and Yueting Zhuang**
College of Computer Science and Technology, Zhejiang University, China
†Corresponding author: yangya@zju.edu.cn

Abstract

Network embedding aims to learn the low-dimensional representations of vertexes in a network, while structure and inherent properties of the network is preserved. Existing network embedding works primarily focus on preserving the microscopic structure, such as the first- and second-order proximity of vertexes, while the macroscopic scale-free property is largely ignored. Scale-free property depicts the fact that vertex degrees follow a heavy-tailed distribution (i.e., only a few vertexes have high degrees) and is a critical property of real-world networks, such as social networks. In this paper, we study the problem of learning representations for scale-free networks. We first theoretically analyze the difficulty of embedding and reconstructing a scale-free network in the Euclidean space, by converting our problem to the sphere packing problem. Then, we propose the "degree penalty" principle for designing scale-free property preserving network embedding algorithm: punishing the proximity between high-degree vertexes. We introduce two implementations of our principle by utilizing the spectral techniques and a skip-gram model respectively. Extensive experiments on six data sets show that our algorithms are able to not only reconstruct heavy-tailed distributed degree distribution, but also outperform state-of-the-art embedding models in various network mining tasks, such as vertex classification and link prediction.



(a) Original (b) LE (c) DP-Walker

Figure 1: Scale-free property of real-world networks. (a) is the degree distribution of an academic network. (b) and (c) are respectively the degree distribution of of the network reconstructed based on vertex representations learned by Laplacian Eigenmap (LE) and our proposed method (DP-Walker).

1 Introduction

Network analysis has attracted considerable research efforts in many areas of artificial intelligence, as networks are able to encode rich and complex data, such as human relationships and interactions. One major challenge of network analysis is how to represent network properly so that the network structure can be preserved. The most straightforward way is to represent the network by its adjacency matrix. However, it suffers from the data sparsity. Other traditional network representation relies on handcrafted network feature design (e.g., clustering coefficient), which is inflexible, non-scalable, and requires hard human labor.

In recent years, network representation learning, also known as network embedding, has been proposed and aroused considerable research interest. It aims to automatically project a given network into a low-dimensional latent space, and represent each vertex by a vector in that space. For example, a number of recent works apply advances in natural language processing (NLP), most notably models known as word2vec (Mikolov et al. 2013), to network embedding and propose word2vec-based algorithms, such as DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and node2vec (Grover and Leskovec 2016). Besides, researchers also consider network embedding as part of dimensionality reduction techniques. For instance, Laplacian Eigenmap (LE) (Belkin and Niyogi 2003) learns the low-dimensional representation to expand the manifold where data lie.

Essentially, these methods mainly focus on preserving microscopic structure of network, like pairwise relationship between vertexes. Nevertheless, scale-free property, one of the most fundamental macroscopic properties of networks, is largely ignored.

Scale-free property depicts that the vertex degrees follow a power-law distribution, which is a common knowledge for many real-world networks. We take an academic network as the example, where each edge indicates if a vertex (researcher) has cited at least one publication of another vertex (researcher). Figure 1(a) demonstrates the degree distribution of this network. The linearity on log-log scale suggests a power-law distribution: the probability decreases when the vertex degree grows, with a long tail tending to zero (Faloutsos, Faloutsos, and Faloutsos 1999) (Adamic and Huberman 1999). In other words, there are only a few vertexes of high

*Equal contribution. Ordering determined by dice rolling.

degrees. The majority of vertexes connected to a high-degree vertex are, however, of low degree, and not likely connected to each other.

Moreover, compared with the microscopic structure, the macroscopic scale-free property imposes a higher level constraints on the vertex representations: Only a few vertexes can be close to many others in the learned latent space. Incorporating scale-free property in network embedding can reflect and preserve the sparsity of real-world networks and, in turn, provide effective information to make the vertex representations more discriminative.

In this paper, we study the problem of learning scale-free property preserving network embedding. As the representation of a network, vertex embedding vectors are expected to well reconstruct the network. Most existing algorithms learn network embedding in the Euclidean space by employing the nearest neighbor algorithm. However, we find that most traditional network embedding algorithms will overestimate the number of higher degrees. Figure 1(b) gives an example, where the representation is learned by Laplacian Eigenmap. We analyze and try to understand this theoretically, and study the feasibility of recovering power-law distributed vertex degree in the Euclidean space, by converting our problem to the Sphere Packing problem. Through our analysis, we find that, theoretically, moderately increasing the dimension of embedding vectors can help to preserve the scale-free property. See details in Section 2.

Inspired by our theoretical analysis, we then propose the *degree penalty* principle for designing scale-free property preserving network embedding algorithms in Section 3: punishing the proximity between vertexes with high degrees. We further introduce two implementations of our approach based on the spectral techniques(Belkin and Niyogi 2003) and the skip-gram model(Mikolov et al. 2013). As Figure 1(c) suggests, our approach can better preserve the scale-free property of networks. In particular, the Kolmogorov-Smirnov (aka. K-S statistic) distance between the obtained degree distribution and its theoretical power-law distribution is 0.09, while the value for the degree distribution obtained by LE is 0.2 (the smaller the better).

To verify the effectiveness of our approach, we conduct experiments on both synthetic data and five real-world data sets in Section 4. The experimental results show that our approach is able to not only preserve the scale-free property of networks, but also outperform several state-of-the-art embedding algorithms in different network analysis tasks.

We summarize our contribution of this paper as follows:

- We analyze the difficulty and feasibility of reconstructing a scale-free network based on learned vertex representations in the Euclidean space, by converting our problem to the Sphere Packing problem.
- We propose the *degree penalty* principle and two implementations to preserve the scale-free property of networks and improve the effectiveness of vertex representations.
- We validate our proposed principle by conducting extensive experiments and find that our approach achieves a significant improvement on six data sets and three tasks compared to several state-of-the-art baselines.

2 Theoretical Analysis

In this section, we try to study why most network embedding algorithms will overestimate higher degrees in theoretical, and analyze if there exists a solution of scale-free property preserving network embedding in the Euclidean space. This section also provides intuitions of our approach in Section 3.

2.1 Preliminaries

Notations. We consider an undirected network $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is the vertex set containing n vertexes and E is the edge set. Each $e_{ij} \in E$ indicates an undirected edge between v_i and v_j. We define the adjacency matrix of G as $\mathbf{A} = [A_{ij}] \in \mathcal{R}^{n\times n}$, where $A_{ij} = 1$ if $e_{ij} \in E$ and $A_{ij} = 0$ otherwise. Let $\mathbf{D}$ be a diagonal matrix where $D_{ii} = \sum_j A_{ij}$ is the degree of v_i.

Network embedding. In this paper, we focus on the representation learning for undirected networks. Given an undirected graph $G = (V, E)$, the problem of graph embedding aims to represent each vertex $v_i \in V$ into a low-dimensional space R^k, i.e., learning a function $\mathbf{f} : V \mapsto \mathbf{U}^{n\times k}$, where $\mathbf{U}$ is the embedding matrix, $k \ll n$ and network structures can be preserved in $\mathbf{U}$.

Network reconstruction. As the representation of a network, the learned embedding matrix is expected to well reconstruct the network. In particular, one can reconstruct the network edges based on distances between vertexes in the latent space $\mathcal{R}^k$. For example, the probability of an edge existing between v_i and v_j can be defined as

$$p_{i,j} = \frac{1}{1 + e^{\|\mathbf{u_i} - \mathbf{u_j}\|}} \tag{1}$$

where the Euclidean distance between embedding vectors $\mathbf{u_i}$ and $\mathbf{u_j}$ of the vertex v_i and v_j, in respective, is considered. In practice, a threshold $\varepsilon \in [0, 1]$ is chosen and an edge e_{ij} will be created if $p_{i,j} \geq \varepsilon$. We call the above method as ε-NN, which is geometrically informative and a common method used in many existing work (Shaw and Jebara 2009; Belkin and Niyogi 2003; Alanis-Lobato, Mier, and Andrade-Navarro 2016).

2.2 Reconstructing Scale-free Networks

Given a network, we call it as a scale-free network, when its vertex degrees follow a power-law distribution. In other words, there are only a few vertexes of high degrees. The majority of vertexes connected to a high-degree vertex is, however, of low degree, and not likely connected to each other. Formally, the probability density function of vertex degree D_{ii} has the following form:

$$p_{D_{ii}}(d) = Cd^{-\alpha}, \alpha > 1, d > d_{\min} > 0 \tag{2}$$

where α is the exponent parameter and C is the normalization term. In practice, the above power-law form only applies for vertexes with degrees greater than a certain minimum value $d_{\min}$ (Clauset, Shalizi, and Newman 2009a).

However, it is difficult to reconstruct a scale-free network in the Euclidean space by ε-NN. As we see in Figure 1(b),

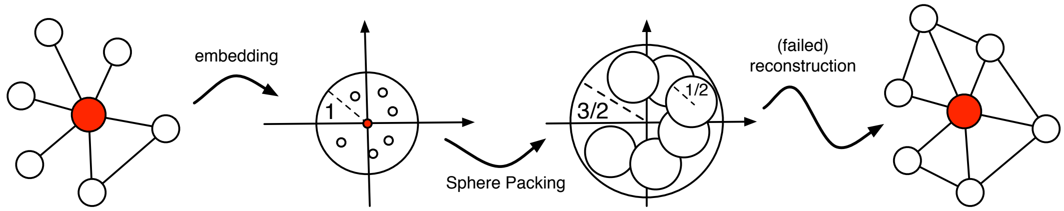


Figure 2: An illustration of an ego-network centered by a hub vertex, a potential embedding solution, which is equivalent to a sphere packing solution, and leads to a failed reconstruction, where higher degrees are overestimated.

the higher degrees will be overestimated. We aim to explain this theoretically.

Intuition. While reconstructing the network by ε-NN, we select a certain ε, and for a vertex v_i with embedding vector $\mathbf{u_i}$, we regard all points that fall in the closed ball of radius ε centered at $\mathbf{u}_i$, denoted by $B(\mathbf{u_i}, \varepsilon)$, as ones having an edge with v_i. When v_i is a high-degree vertex, there will be many points in $B(\mathbf{u_i}, \varepsilon)$. We expect these points are far away from each other, keeping more vertexes with low degree and thus in turn keep the power-law distribution. However, intuitively, as these points are placed in the same closed ball $B(\mathbf{u_i}, \varepsilon)$, it will be more likely that their distances from each other are less than ε. As a result, there will be many edges created among those points, which violates the hypothesis of the scale-free property.

Following the above idea, we introduce a theorem below, which is discussed in $\mathcal{R}^k$, and, without loss of generality, we set ε as 1.

Theorem 1 (Sphere Packing)**.** There are m points in $B(0, 1)$ whose distances from each other are larger than or equal to 1, if and only if, there exists m disjoint spheres of radius $1/2$ in $B(0, 3/2)$.

Proof. It follows directly from these two facts: the center of any sphere of radius $1/2$ in $B(0, 3/2)$ falls in $B(0, 1)$ and the distance between any two centers of two different spheres of radius $1/2$ is larger than or equal to 1. $\square$

Remark. Theorem 1 converts our problem of reconstructing a scale-free network to the *Sphere Packing Problem*, which seeks to find the optimal packing of spheres in high dimensional spaces (Cohn and Elkies 2003; Vance 2011; Venkatesh 2012). Figure 2 gives an example, where a network centered by a high-degree vertex is embedded into a two-dimensional space. The embedding result corresponds to an equivalent Sphere Packing solution, which fails to place enough disjoint spheres and leads to a failed network reconstruction (many nonexistent edges are created). Formally, we define the packing density as follows:

Definition 1 (Packing Density)**.** The packing density is the fraction of the space filled by the spheres making up the packing. For convenience, we define the optimal sphere packing density in $\mathcal{R}^k$ as Δ_k, which means no packing of spheres in $\mathcal{R}^k$ achieves a packing density larger than Δ_k.

As Theorem 1 suggests, we aim to find a packing solution with large density so that more points in a closed sphere can keep their distance larger than 1. However, in general cases, finding the optimal packing density remains an open problem. Still, we are able to derive the upper and lower bounds of Δ_k for sufficiently large k.

Theorem 2 (Upper and lower bounds for Δ_k)**.**

$$-1 \leq \lim_{k \to \infty} \frac{1}{k} \log_2 \Delta_k \leq -0.599$$

Specifically, we always have

$$\Delta_k \geq 2^{-k} \tag{3}$$

And the following inequality holds for sufficiently large k:

$$\Delta_k \leq 2^{-0.599k} \tag{4}$$

Eq. 4 is one of the best upper bound when $k \geq 115$ (Cohn and Zhao 2014). The proof of the above theorem can be found in the work of Kabatiansky and Levenshtein.

Theorem 3. Suppose x is in $\mathcal{R}^k$, $\varepsilon > 0$, and there can be at most M_k points in $B(x, \varepsilon)$ whose distances from each other are larger than ε, then

$$M_k \leq \lfloor 3^k 2^{-0.599k} \rfloor \tag{5}$$

$$M_k \geq \left\lfloor \left(\frac{3}{2}\right)^k \right\rfloor \tag{6}$$

where $\lfloor \cdot \rfloor$ means taking the integer part. Eq. 5 holds for sufficiently large k.

Proof. By Theorem 1 we only need to estimate the number of disjoint spheres of radius $\frac{1}{2}\varepsilon$ that can be fitted in $B(x, \frac{3}{2}\varepsilon)$. The volume of a k-dimensional ball of radius R is given by $\frac{\pi^{k/2}}{\Gamma(k/2+1)} R^k$. Plugging in the radius, the volumes of $B(x, \frac{3}{2}\varepsilon)$ and a sphere of radius $\frac{1}{2}\varepsilon$ are respectively $V_1 = \frac{\pi^{k/2}}{\Gamma(k/2+1)} (\frac{3}{2}\varepsilon)^k$ and $V_2 = \frac{\pi^{k/2}}{\Gamma(k/2+1)} (\frac{1}{2}\varepsilon)^k$. Since the optimal packing density is given by Δ_k, we have

$$\begin{aligned} M_k &= \left\lfloor \Delta_k \frac{V_1}{V_2} \right\rfloor \\ &= \lfloor 3^k \Delta_k \rfloor \end{aligned}$$

Combing with Eq. 3 and Eq. 4, we obtain the inequality as desired. $\square$

Discussion. The lower bound of M_k in Eq. 6 suggests the feasibility to reconstruct scale-free network by ε-NN in the Euclidean space, when k, the dimension of embedding vector, is sufficiently large. For instance, when $k = 100$, we have $M_k > 4.06 \times 10^{17}$, which is enough to keep scale-free property holds for most real-world networks.

3 Our Approach

General idea. Inspired by our theoretical analysis, we propose a principle of *degree penalty* for designing scale-free property preserving embedding algorithms: *while preserving first- and second-order proximity, the proximity between vertexes that have high degrees shall be punished.* We give the general idea behind this principle below.

Scale-free property is featured by the ubiquitous existence of "big hubs" which attract most edges in the network. Most of existing network embedding algorithms, explicitly or implicitly, attempts to keep these big hubs close to their neighbors, by preserving first-order and/or second-order proximity(Belkin and Niyogi 2003; Tang et al. 2015; Perozzi, Al-Rfou, and Skiena 2014). However, connecting to big hubs does not imply proximity as strong as connecting to vertexes with mediocre degrees. Taking social network as an example, where a famous celebrity may receive a lot of followers. However, the celebrity may not be familiar or similar to her followers. Besides, two followers of the same celebrity may not know each other at all and can be totally dissimilar. As a comparison, a mediocre user is more likely to known and to be similar to her followers.

From another perspective, a high-degree vertex v_i is more likely to hurt the scale-free property, as placing more disjoint spheres in a closed ball is more difficult (Section 2).

The *degree penalty* principle can be implemented by various methods. In this paper, we introduce two proposed models based on our principle, each of which utilizes the spectral techniques and a skip-gram model respectively.

3.1 Model I: DP-Spectral

Our first model, Degree Penalty based Spectral Embedding (DP-Spectral), mainly utilizes graph spectral techniques. Given a network $G = (V, E)$ and its adjacency matrix $\mathbf{A}$, we define matrix $\mathbf{C}$ to indicate the common neighbors of any two vertexes in G:

$$\mathbf{C} \triangleq \mathbf{A}^T\mathbf{A} - \operatorname{diag}(\mathbf{A}^T\mathbf{A}) \tag{7}$$

where C_{ij} is the number of common neighbors of v_i and v_j. $\mathbf{C}$ can be also regarded as a measurement of second-order proximity, and can be easily extended to further consider first-order proximity by

$$\mathbf{C}' \triangleq \mathbf{C} + \mathbf{A} \tag{8}$$

As we aim to model the influence of vertex degrees in our model, we further extend $\mathbf{C}'$ to consider degree penalty as

$$\mathbf{W} \triangleq (\mathbf{D}^{-\beta})^T\mathbf{C}'\mathbf{D}^{-\beta} \tag{9}$$

where $\beta \in \mathcal{R}$ is the model parameter used to indicate the strength of degree penalty, and $\mathbf{D}$ is a diagonal matrix where D_{ii} is the degree of v_i. Thus $\mathbf{W}$ is proportional to $\mathbf{C}'$ and is inverse proportion to vertex degrees.

Objective. Our goal is to learn vertex representations $\mathbf{U}$, where $\mathbf{u_i} \in \mathcal{R}^k$ is the ith row of $\mathbf{U}$ and represents the embedding vector of vertex v_i, and minimize

$$\sum_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2 W_{ij} \tag{10}$$

under the constraint

$$\mathbf{U}^T\mathbf{D}\mathbf{U} = \mathbf{I} \tag{11}$$

Eq. 11 is provided such that the embedding vectors will not collapse onto a subspace of dimension less than k (Belkin and Niyogi 2003).

Optimization. In order to minimize Eq. 10, we utilize graph Laplacian, which is an analogy to the Laplacian operator in multivariate calculus. Formally, we define graph Laplacian, $\mathbf{L}$, as follows:

$$\mathbf{L} \triangleq \mathbf{D} - \mathbf{W} \tag{12}$$

Observe that

$$\sum_{i,j} W_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|^2 = \operatorname{trace}(\mathbf{U^T L U}) \tag{13}$$

The desired $\mathbf{U}$ minimizing the objective function (10) is obtained by putting

$$\mathbf{U} = [\mathbf{t}_1, \ldots, \mathbf{t}_k] \tag{14}$$

where $\mathbf{t}_i$ is an eigenvector of $\mathbf{L}$. In practice, we use normalized form of $\mathbf{W}$ and $\mathbf{L}$, (i.e., $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{1/2}$ and $\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{1/2}$).

3.2 Model II: DP-Walker

Our second method, Degree Penalty based Random Walk (DP-Walker), utilizes a skip-gram model like word2vec.

We start with a brief description of the word2vec model and its application to network embedding task. Given a text corpus, Mikolov et al. proposed word2vec to learn the representation of each word by facilitating the prediction of its context (Mikolov et al. 2013). Inspired by it, some network embedding algorithms like DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and node2vec (Grover and Leskovec 2016) define a vertex's "context" by co-occurred vertexes in random walk generated paths.

Specifically, the Random Walk rooted at vertex v_i is a stochastic process $\mathcal{V}_{v_i}^k, k = 1, \ldots, m$, where $\mathcal{V}_{v_i}^{k+1}$ is a vertex sampled from the neighbors of $\mathcal{V}_{v_i}^k$ and m is the path length. Traditional methods regard $P(\mathcal{V}_{v_i}^{k+1}|\mathcal{V}_{v_i}^k)$ as a uniform distribution where each neighbor of $\mathcal{V}_{v_i}^k$ has the equal chance to be sampled.

However, as our proposed Degree Penalty principle suggests, a neighbor v_i of a vertex v_j with high degree may not be similar to v_j. In other words, v_j shall have less chance to be sampled as one of v_i's context. Thus, we define the probability of the random walker jump to v_j from v_i as

$$\mathrm{Pr(v_j|v_i)} \propto \frac{\mathrm{C'_{ij}}}{(\mathrm{D_{ii}D_{jj}})^{\beta}} \tag{15}$$

where $\mathbf{C}'$ can be found in Eq. 8 and β is the model parameter. According to Eq. 15, we find that v_j will have a greater chance to be sampled when it has more common neighbors with v_i and has a lower degree. After obtaining random walk generated paths, we enable skip-gram to learn effective vertex representations for G by predicting each vertex's context. This results in an optimization problem

$$\arg\min_{\mathbf{U}} - \log \Pr\left(\{\mathrm{v_{i-w}}, \ldots, \mathrm{v_{i+w}}\} \; \mathrm{v_i} | \mathbf{u_i}\right) \tag{16}$$

where $2 \times w$ is the path length we consider. Specifically, for each random walk $\mathcal{V}_{v_i}$, we feed it to skip-gram algorithm (Mikolov et al. 2013) to update the vertex representation. For implementation, we use Hierarchical Softmax (Morin and Bengio 2005; Mnih and Hinton 2009) to estimate the probability distribution.

4 Experiments

4.1 Experiment Setup

Datasets. We use four datasets, whose statistics are summarized in Table 1, for the evaluations.

- Synthetic: We generate the synthetic data by the Preferential Attachment model (Vazquez 2003), which can generate power-law networks.
- Facebook (Leskovec and Mcauley 2012): This dataset is a subnetwork of Facebook [1], where vertexes indicate users of Facebook, and edges denote friendship between users.
- Twitter (Leskovec and Mcauley 2012): This dataset is a subnetwork of Twitter [2], where vertexes indicate users of Twitter, and edges denote following relationships.
- Coauthor (Leskovec, Kleinberg, and Faloutsos 2007): This is a collaboration network that covers scientific collaborations between authors. Vertexes are authors. If au author v_i coauthored at least one paper with author v_j, the network contains an undirected edge between v_i and v_j.
- Citation (Tang et al. 2008): Similar to Coauthor, this is also an academic network, where vertexes are authors. The difference is, in this network, each edge indicates citation relationship instead of coauthor relationship.
- Mobile: This is a mobile network provided by PPDai[3]. Vertexes are PPDai registered users. An edge between two users indicates that one of the users has called the other. Overall, it consists of over one million calling logs.

Baseline methods. We compare the following four network embedding algorithms in our experiments:

[1] http://facebook.com

[2] http://twitter.com

[3] The largest unsecured micro-credit loan platform in China.

	Synthetic	Facebook	Twitter	Coauthor	Citation	Mobile
$\|V\|$	10000	4039	81306	5242	48521	198959
$\|E\|$	399580	88234	1768149	28980	357235	1151003

Table 1: Statistics of datasets. $|V|$ indicates the number of vertexes and $|E|$ denotes the number of edges.

- Laplacian Eigenmap (LE) (Belkin and Niyogi 2003): This represents spectral-based network embedding algorithms. It aims to learn the low-dimensional representation to expand the manifold where data lie.
- DeepWalk (Perozzi, Al-Rfou, and Skiena 2014): This represents skip-gram model based network embedding algorithms. It first generates random walks on the network, and defines the context of a vertex by its co-occurred vertexes in paths. Then, it learns vertex representations by predicting each vertex's context. Specifically, we perform 10 random walks starting from each vertex, and each random walk will have a length of 40.
- DP-Spectral: This is a spectral-technique-based implementation of our degree penalty principle.
- DP-Walker: This is another implementation of our approach. It is based on a skip-gram model.

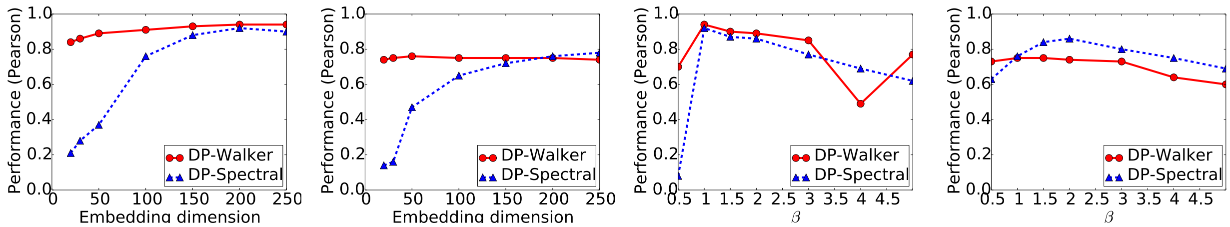
Unless otherwise specified, the embedding dimension for our experiments is 200.

Tasks. We first utilize different algorithms to learn vertex representations for a certain dataset, then apply the learned embeddings in three different tasks:

- Network reconstruction: This task aims to validate if our algorithm is able to preserve the scale-free property of networks. We evaluate the performance of different algorithms by the correlation coefficients between the reconstructed degrees and the degrees in the given network.
- Link prediction: Given two vertexes v_i and v_j, we feed their embedding vectors, $\mathbf{u_i}$ and $\mathbf{u_j}$, to a linear regression classifier and determine whether there exists an edge between v_i and v_j. Specifically, we let $\mathbf{u}_i - \mathbf{u}_j$ as the feature, and randomly select about 1% pairs of vertexes for training and evaluation.
- Vertex classification: In this task, we consider the vertex labels. For instance, in Citation, each vertex has a label to indicate the researcher's research area. Specifically, given a vertex v_i, we define its feature vector as $\mathbf{u_i}$, and train a linear regression classifier to determine its label.

4.2 Network Reconstruction

Comparison results. In this task, after obtaining vertex representations, we use the ε-NN algorithm, introduced in Section 2, to reconstruct the network. We then evaluate the performance of different methods on reconstructing power-law distributed vertex degrees by considering three different correlation coefficients of the reconstructed degrees and original degrees: Pearson's r (Shao 2007), Spearman's ρ (Spearman 1904), and Kendall's τ (Kendall 1938). All of these statistics are used to evaluate some relationship between



(a) Synthetic. (b) Facebook. (c) Synthetic. (d) Facebook.

Figure 3: Model parameter analysis. (a) and (b) demonstrate the sensitivity of the embedding dimension k in Synthetic and Facebook dataset respectively. (c) and (d) present the sensitivity of the degree penalty parameter β. We omit the results on other datasets due to space limitation.

Dataset	Method (ε)	Pearson	Spearman	Kendall
Synthetic	LE (0.55)	0.14	0.054	0.039
	DeepWalk (0.91)	0.47	-0.22	-0.18
	DP-Spectral (0.52)	0.92	**0.79**	**0.63**
	DP-Walker (0.95)	**0.94**	0.63	0.52
Facebook	LE (0.52)	0.48	0.18	0.12
	DeepWalk (0.81)	0.73	0.65	0.49
	DP-Spectral (0.52)	**0.87**	**0.67**	0.51
	DP-Walker (0.84)	0.75	0.73	**0.57**
Twitter	LE (0.81)	0.17	0.19	0.17
	DeepWalk (0.51)	0.08	0.21	0.26
	DP-Spectral (0.93)	**0.50**	**0.34**	**0.27**
	DP-Walker (0.087)	0.40	0.33	**0.27**
Coauthor	LE (0.50)	0.32	0.04	0.03
	DeepWalk (0.92)	0.66	0.31	0.25
	DP-Spectral (0.51)	0.64	**0.69**	**0.55**
	DP-Walk (0.93)	**0.75**	0.44	0.35
Citation	LE (0.99)	0.11	-0.27	-0.20
	DeepWalk (0.97)	0.51	0.28	0.20
	DP-Spectral (0.50)	0.45	**0.72**	**0.54**
	DP-Walker (0.98)	**0.62**	0.65	0.50
Mobile	LE (0.51)	0.10	0.05	0.04
	DeepWalk (0.71)	0.77	0.22	0.20
	DP-Spectral (0.50)	0.40	**0.68**	**0.60**
	DP-Walker (0.78)	**0.93**	0.22	0.20

Table 2: Performance of different methods on scale-free property reconstruction. For each method, ε (indicated after the corresponding method) is chosen so that Pearson is maximized.

paired samples. Pearson's r is used to detect linear relationships, while Spearman's ρ and Kendall's τ are capable of finding monotonic relationships.

To select ε, for each algorithm, we roll over all values of ε ranging from 0.01 to 1 with step 0.01, and choose the value which maximizes the Pearson's correlation coefficient between the original and recovered degrees. Our selection of Pearson's r as evaluation metric is because of the scale-free property of degree distribution.

From Table 2, we see that our algorithms outperform the baselines significantly. Especially, the Pearson of DP-Walker achieves at least 44.5% improvement, and the Spearman of DP-Spectral achieves at least 84.3% improvement. The good fitness of the vertex degree reconstructed by our algorithms suggests that we can better preserve the scale-free property of the network. Besides, we see that the best ε to maximize Pearson for Deep-Spectral is more stable (i.e., around 0.51) than other methods. Thus one can tune Deep-Spectral's parameters more easily in practical.

Preserving Scale-free Property. After reconstructing the network, for each embedding algorithm, we validate its effectiveness to preserve the scale-free property, by fitting the reconstructed degree distribution (e.g., Figure 1(b) and (c)) to a theoretical power-law distribution (Alstott, Bullmore, and Plenz 2014). We then calculate the Kolmogorov-Smirnov distance between this two distributions and find that the proposed methods can always obtain a better result compared to baselines (0.115 vs 0.225 averagely).

Parameter analysis. We further study the sensitivity of model parameters: embedding dimension and degree penalty weight β. We only present the results on Synthetic and Facebook and omit others due to space limitation. Figure 3(b) and 3(a) shows the Pearson correlation coefficient resulted by our algorithm with different embedding dimensions. When the embedding dimensions grow, the performance increases significantly. Correlation does not change drastically for DP-Walker as the dimension increases, as the figure suggests. Besides, on Facebook, the correlation even decreases slightly as the embedding increases. The figure also shows that DP-Walker is able to obtain fairly high Pearson correlation even in a lower dimension.

We also study how β influences the performance. Figure 3(d) and 3(c) shows that DP-Spectral is more sensitive to the choice of β. This is largely due to the fact that in DP-Spectral, the influence of β is manifested in the objective function (Eq. 10), which is stronger than its influence in DP-Walker embodied in the sampling process. Figure 3(d) and 3(c) shows that the optimal choice of β varies from graph to graph. It makes sense, since β can be viewed as a punishment on the degrees, and the influence of β is therefore supposedly related to the topology of the original network.

4.3 Link Prediction

In this task, we consider the following evaluation metrics: Precision, Recall, and F1-score. Table 3 demonstrates the performance of different methods on the link prediction task. For our methods, we let the model parameter β as optimized in Table 2. We see that, in most cases, DP-Spectral obtains the best performance, which suggests that with the help of

Dataset	Method	Precision	Recall	F1
Synthetic	LE	0.52	0.53	0.53
	Deepwalk	0.51	0.51	0.51
	DP-Spectral	**0.64**	**0.68**	**0.66**
	DP-Walker	0.61	0.63	0.62
Facebook	LE	0.75	0.92	0.83
	Deepwalk	**0.84**	0.97	**0.90**
	DP-Spectral	0.76	**0.98**	0.85
	DP-Walker	0.82	0.95	0.89
Twitter	LE	0.58	0.35	0.43
	Deepwalk	**0.65**	0.77	0.70
	DP-Spectral	0.59	**0.98**	**0.73**
	DP-Walker	0.54	0.58	0.56
Coauthor	LE	0.61	0.83	0.70
	Deepwalk	0.55	0.58	0.56
	DP-Spectral	**0.62**	**0.89**	**0.73**
	DP-Walker	0.56	0.66	0.61
Citation	LE	0.54	0.56	0.55
	Deepwalk	0.54	0.56	0.55
	DP-Spectral	0.52	**0.99**	**0.68**
	DP-Walker	**0.55**	0.57	0.56
Mobile	LE	**0.75**	0.36	0.48
	Deepwalk	0.55	0.60	0.57
	DP-Spectral	0.63	**0.89**	**0.74**
	DP-Walker	0.54	0.58	0.56

Table 3: Experimental results of link prediction.

Method	Acrh	CN	CS	DM	THM	GRA	UNK
LE	0.36	**0.75**	0.14	0.37	0.46	0.13	**0.86**
Deepwalk	0.54	0.54	0.52	0.56	0.56	0.56	0.85
DP-Walker	0.56	0.57	0.54	0.58	0.58	0.55	0.85
DP-Spectral	**0.71**	0.74	**0.78**	**0.76**	**0.74**	**0.75**	0.85

Table 4: Accuracy of Multi-classification Task. The labels are abbreviations for Architecture, Computer Network, Computer Science, Data Mining, Theory, Graphics, and Unknown, respectively, signifying the study field of the vertexes.

the proposed principle, we can not only preserve the scale-free property of networks, but also improve the effectiveness of the embedding vectors.

4.4 Vertex Classification

Table 4 lists the accuracy of vertex classification task on Citation. Our task is to determine an author's research area, which is a multi-classification problem. We define features as vertex representation obtained by the four different embedding algorithms. Generally, from the table, we see that DP-Walker and DP-Spectral beat respectively Deepwalk and Laplacian Eigenmap. In particular, DP-Spectral achieves the best result for 5 out of 7 labels. Besides, we can also observe the stability of the performance. DP-Spectral's result on all labels is more stable than other methods. In comparison, LE achieves a satisfactory result for two labels, but for others the result can be poor. Specifically, the standard deviation of DP-Spectral is 0.04, while the value is 0.26 for LE and 0.1 for DeepWalk.

5 Related Work

Network embedding. Network embedding aims to learn representations for vertexes in a given network. Some researchers regard network embedding as part of dimensionality reduction techniques. For example, Laplacian Eigenmaps (LE) (Belkin and Niyogi 2003) aims to learn the vertex representation to expand the manifold where data lie. As a variant of LE, Locality Preserving Projections (LPP) (He et al. 2005) learns a linear projection from feature space to embedding space. Besides, there are other linear (Jolliffe 2002) and non-linear (Tenenbaum, De Silva, and Langford 2000) network embedding algorithms for dimensionality reduction. Recent network embedding work takes advancements in natural language processing, most notably models known as word2vec (Mikolov et al. 2013), which learns the distributed representations of words. Building on word2vec, Perozzi et al. define a vertex's "context" by their co-occurrence in a random walk path (Perozzi, Al-Rfou, and Skiena 2014). More recently, Grover et al. propose a mixture of width-first and breadth-first search based procedure to generate paths of vertexes (Grover and Leskovec 2016). Dong et al. further develop the model to handle heterogeneous networks (Dong, Chawla, and Swami 2017). LINE (Tang et al. 2015) decomposes a vertex's context into first-order (neighbors) and second-order (two-degree neighbors) proximity. Wang et al. preserve community information in their vertex representations (Wang et al. 2017). However, all of above methods focus on preserving microscopic network structure and ignore macroscopic scale-free property of networks.

Scale-free Networks. The scale-free property has been discovered to be ubiquitous over a variety of network systems (Mood 1950) (Newman 2005) (Clauset, Shalizi, and Newman 2009b), such as the Internet Autonomous System graph (Faloutsos, Faloutsos, and Faloutsos 1999), the Internet router graph (Govindan and Tangmunarunkit 2000), the degree distributions of subsets of the world wide web (Barabási and Albert 1999). Newman (Newman 2005) provides a comprehensive list of such work. However, investigating the scale-free property in a low-dimensional vector space and establishing its cooperation with network embedding have not been fully considered.

6 Conclusion

In this paper, we study the problem of learning the scale-free property preserved network embeddings. We first analyze the feasibility of reconstructing a scale-free network, based on learned vertex representations, in the Euclidean space, by converting our problem to the Sphere Packing problem. We then propose the *degree penalty* principle as our solution and introduce two implementations by leveraging the spectral techniques and a skip-gram model respectively. The proposed principle can also be implemented using other methods, which is left to our future work. We at last conduct extensive experiments on both synthetic data and five real-world data sets to verify the effectiveness of our approach. Experimental results show that our approach outperforms several state-of-the-art baselines on three different tasks.

Acknowledgements. The work is supported by the Fundamental Research Funds for the Central Universities, 973 Program (2015CB352302), NSFC (U1611461, 61625107, 61402403), and key program of Zhejiang Province (2015C01027).

References

Adamic, L., and Huberman, B. A. 1999. The nature of markets in the world wide web. *Q. J. Econ.*

Alanis-Lobato, G.; Mier, P.; and Andrade-Navarro, M. A. 2016. Efficient embedding of complex networks to hyperbolic space via their laplacian. In *Scientific reports*.

Alstott, J.; Bullmore, E.; and Plenz, D. 2014. powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions. *PLoS ONE* 9:e85777.

Barabási, A.-L., and Albert, R. 1999. Emergence of scaling in random networks. *science* 286(5439):509–512.

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15(6):1373–1396.

Clauset, A.; Shalizi, C. R.; and Newman, M. E. J. 2009a. Power-law distributions in empirical data. *SIAM Review* 51(4):661–703.

Clauset, A.; Shalizi, C. R.; and Newman, M. E. 2009b. Power-law distributions in empirical data. *SIAM review* 51(4):661–703.

Cohn, H., and Elkies, N. 2003. New upper bounds on sphere packings i. *Annals of Mathematics* 689–714.

Cohn, H., and Zhao, Y. 2014. Sphere packing bounds via spherical codes. *Duke Mathematical Journal* 163(10):1965–2002.

Dong, Y.; Chawla, N.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD'17*, 135–144.

Faloutsos, M.; Faloutsos, P.; and Faloutsos, C. 1999. On power-law relationships of the internet topology. In *COMPUT COMMUN REV*, volume 29, 251–262.

Govindan, R., and Tangmunarunkit, H. 2000. Heuristics for internet map discovery. In *INFOCOM'00*, volume 3, 1371–1380.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD'16*, 855–864.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H. 2005. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(3):328–340.

Jolliffe, I. 2002. *Principal component analysis*. Wiley Online Library.

Kabatiansky, G. A., and Levenshtein, V. I. 1978. On bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii* 14(1):3–25.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika* 30(1/2):81–93.

Leskovec, J., and Mcauley, J. 2012. Learning to discover social circles in ego networks. In *NIPS'12*, 539–547.

Leskovec, J.; Kleinberg, J. M.; and Faloutsos, C. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery From Data* 1(1):2.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *NIPS'13*, 3111–3119.

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc. 1081–1088.

Mood, A. M. 1950. Introduction to the theory of statistics.

Morin, F., and Bengio, Y. 2005. Hierarchical probabilistic neural network language model. In *AISTATS05*, 246–252.

Newman, M. E. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics* 46(5):323–351.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD'14*, 701–710.

Shao, J. 2007. *Mathematical Statistics*. Springer, 2 edition.

Shaw, B., and Jebara, T. 2009. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 937–944. New York, NY, USA: ACM.

Spearman, C. 1904. The proof and measurement of association between two things. *The American Journal of Psychology* 15(1):72–101.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. In *KDD'08*, 990–998.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW'15*, 1067–1077.

Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290(5500):2319–2323.

Vance, S. 2011. Improved sphere packing lower bounds from hurwitz lattices. *Advances in Mathematics* 227(5):2144–2156.

Vazquez, A. 2003. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E* 67(5):056104.

Venkatesh, A. 2012. A note on sphere packings in high dimension. *International mathematics research notices* 2013(7):1628–1642.

Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; and Yang, S. 2017. Community preserving network embedding. In *AAAI'17*.