A Survey on Network Embedding

Peng Cui, Xiao Wang, Jian Pei, *Fellow, IEEE*, Wenwu Zhu, *Fellow, IEEE*

Abstract—Network embedding assigns nodes in a network to low-dimensional representations and effectively preserves the network structure. Recently, a significant amount of progresses have been made toward this emerging network analysis paradigm. In this survey, we focus on categorizing and then reviewing the current development on network embedding methods, and point out its future research directions. We first summarize the motivation of network embedding. We discuss the classical graph embedding algorithms and their relationship with network embedding. Afterwards and primarily, we provide a comprehensive overview of a large number of network embedding methods in a systematic manner, covering the structure- and property-preserving network embedding methods, the network embedding methods with side information and the advanced information preserving network embedding methods. Moreover, several evaluation approaches for network embedding and some useful online resources, including the network data sets and softwares, are reviewed, too. Finally, we discuss the framework of exploiting these network embedding methods to build an effective system and point out some potential future directions.

Index Terms—Network embedding, graph embedding, network analysis, data science.

1 Introduction

Many complex systems take the form of networks, such as social networks, biological networks, and information networks. It is well recognized that network data is often sophisticated and thus is challenging to deal with. To process network data effectively, the first critical challenge is to find effective network data representation, that is, how to represent networks concisely so that advanced analytic tasks, such as pattern discovery, analysis and prediction, can be conducted efficiently in both time and space.

Traditionally, we usually represent a network as a graph $G = \langle V, E \rangle$, where V is a vertex set representing the nodes in a network, and E is an edge set representing the relationships among the nodes. For large networks, such as those with billions of nodes, the traditional network representation poses several challenges to network processing and analysis.

- **High computational complexity**. The nodes in a network are related to each other to a certain degree, encoded by the edge set E in the traditional network representation. These relationships cause most of the network processing or analysis algorithms either iterative or combinatorial computation steps, which result in high computational complexity. For example, a popular way is to use the shortest or average path length between two nodes to represent their distance. To compute such a distance using the traditional network representation, we have to enumerate many possible paths between two nodes, which is in nature a combinatorial problem. As another example, many studies assume that a node with links to important nodes tends to be important, and vice versa. In order to evaluate the importance of a node using the traditional network representation, we have to iteratively conduct a stochastic node traversal process until reaching a convergence. Such methods using the traditional network representation result in high computational complexity that prevents them from being applicable to large-scale real-world networks.
- **Low parallelizability**. Parallel and distributed computing is de facto to process and analyze large-scale data. Network data represented in the traditional way, however, casts severe difficulties to design and implementation of parallel and distributed algorithms. The bottleneck is that nodes in a network are coupled to each other explicitly reflected by E. Thus, distributing different nodes in different shards or servers often causes demandingly high communication cost among servers, and holds back speed-up ratio. Although some limited progress is made on graph parallelization by subtly segmenting large-scale graphs [1], the luck of these methods heavily depends on the topological characteristics of the underlying graphs.
- **Inapplicability of machine learning methods**. Recently, machine learning methods, especially deep learning, are very powerful in many areas. These methods provide standard, general and effective solutions to a broad range of problems. For network data represented in the traditional way, however, most of the off-the-shelf machine learning methods may not applicable. Those methods usually assume that data samples can be represented by independent vectors in a vector space, while the samples in network data (i.e., the nodes) are dependant to each other to some degree determined by E. Although we can simply represent a node by its corresponding row vector in the adjacency matrix of the network, the extremely high dimensionality of such a representation in a large graph with many nodes makes the in sequel network processing and analysis difficult.

The traditional network representation has become a bottleneck in large-scale network processing and analysis nowadays. Representing the relationships explicitly using a set of edges in the traditional representation is the upmost barrier.

- *P. Cui, X. Wang, and W. Zhu are with the Department of Computer Science and Technology in Tsinghua University, Beijing 100084, China. E-mail: cuip@tsinghua.edu.cn, wangxiao007@mail.tsinghua.edu.cn, wwzhu@tsinghua.edu.cn*
- *J. Pei is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: jpei@cs.sfu.ca*

Manuscript received April 19, 2005; revised August 26, 2015.

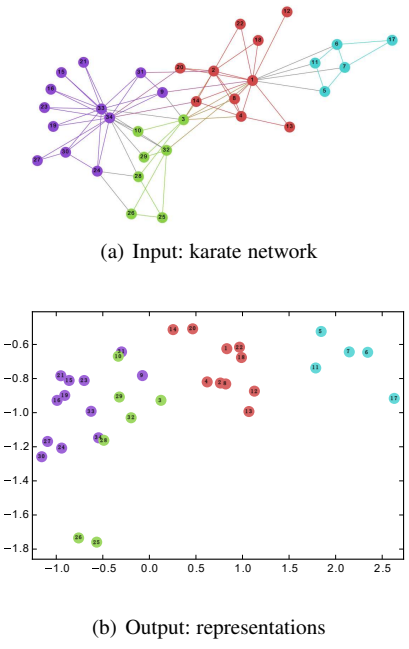
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2849727, IEEE Transactions on Knowledge and Data Engineering

To tackle the challenge, substantial effort has been committed to develop novel network embedding, i.e., learning low-dimensional vector representations for network nodes. In the network embedding space, the relationships among the nodes, which were originally represented by edges or other high-order topological measures in graphs, is captured by the distances between nodes in the vector space, and the topological and structural characteristics of a node are encoded into its embedding vector. An example is shown in Fig. 1. After embedding the karate club network into a two-dimensional space, the similar nodes marked by the same color are close to each other in the embedding space, demonstrating that the network structure can be well modeled in the two-dimensional embedding space.

Network embedding, as a promising way of network representation, is capable of supporting subsequent network processing and analysis tasks such as node classification [2], [3], node clustering [4], network visualization [5], [6] and link prediction [7], [8]. If this goal is fulfilled, the advantages of network embedding over traditional network representation methods are apparent, as shown in Fig. 2. The traditional topology based network representation usually directly uses the observed adjacency matrix, which may contain noise or redundant information. The embedding based representation first aims to learn the dense and continuous representations of nodes in a low dimensional space, so that the noise or redundant information can be reduced and the intrinsic structure information can be preserved. As each node is represented by a vector containing its information of interest, many iterative or combinatorial problems in network analysis can be tackled by computing mapping functions, distance metrics or operations on the embedding vectors, and thus avoid high complexity. As the nodes are not coupling any more, it is convenient to apply mainstream parallel computing solutions for large-scale network analysis. Furthermore, network embedding can open the opportunities for network analysis to be benefited from the rich literature of machine learning. Many off-the-shelf machine learning methods such as deep learning models can be directly applied to solve network problems.

In order to make the embedding space well support network analysis tasks, there are two goals for network embedding. First, the original network can be reconstructed from the learned embedding space. It requires that, if there is an edge or relationship between two nodes, then the distance of these two nodes in the embedding space should be relatively small. In this way, the network relationships can be well preserved. Second, the learned embedding space can effectively support network inference, such as predicting unseen links, identifying important nodes, and inferring node labels. It should be noted that an embedding space with only the goal of network reconstruction is not sufficient for network inference. Taking the link prediction problem as an example, if we only consider the goal of network reconstruction, the embedding vectors learned by SVD tend to fit all the observed links and zero values in the adjacency matrix, which may lead to overfitting and cannot infer unseen links.

In this paper, we survey the state-of-the-art works on network embedding and point out future research directions. In Section 2, we first categorize network embedding methods according to the types of information preserved in embedding, and summarize the commonly used models. We briefly review the traditional graph embedding methods and discuss the difference of these methods with the recent network embedding methods in Section 3. Then, in Sections 4, 5 and 6, we respectively review the methods on



(a) Input: karate network

(b) Output: representations

Fig. 1: An example of network embedding on a karate network. Images are extracted from DeepWalk [3].

structure and property preserving network embedding, network embedding with side information, as well as advanced information preserving network embedding. In Section 7, we present a few evaluation scenarios and some online resources, including the data sets and codes, for network embedding. We conclude and discuss a series of possible future directions in Section 8.

2 Categorization and The Models

To support network inference, more information beyond nodes and links needs to be preserved in embedding space. Most research works on network embedding develop along this line in recent years. There are multiple ways to categorize them. In this paper, according to the types of information that are preserved in network embedding, we categorize the existing methods into three categories, that is, (1) network structure and properties preserving network embedding, (2) network embedding with side information and (3) advanced information preserving network embedding.

2.1 The Categorization of Network Embedding Methods

As mentioned before, network embedding usually has two goals, i.e., network reconstruction and network inference. The traditional graph embedding methods, mainly focusing on network reconstruction, has been widely studied. We will briefly review those methods in Section 3. Fu and Ma [9] present a more detailed survey. In this paper, we focus on the recently proposed network embedding methods aiming to address the goal of network inference. The categorization structure of the related works is shown in Fig. 3.

2.1.1 Structure and property preserving network embedding

Among all the information encoded in a network, network structures and properties are two crucial factors that largely affect

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2849727, IEEE Transactions on Knowledge and Data Engineering

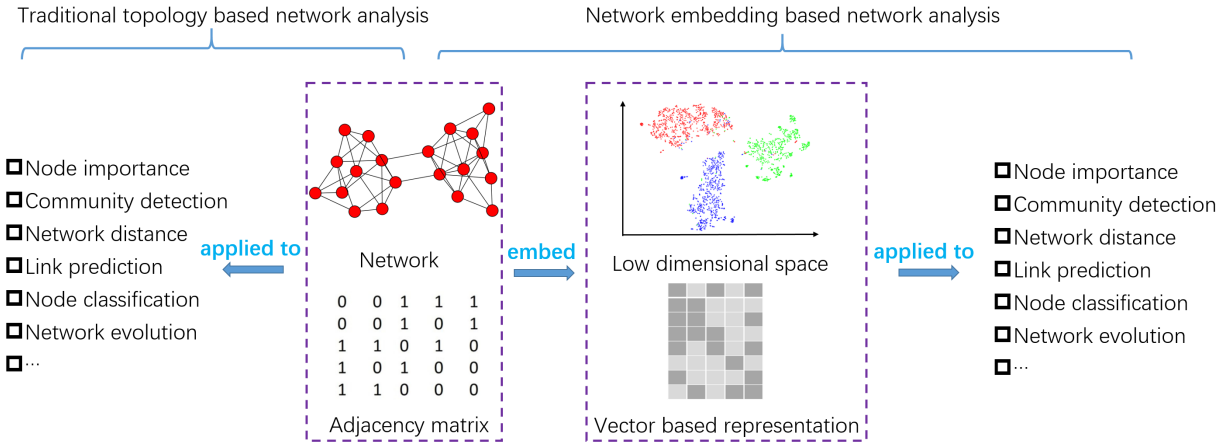


Fig. 2: A comparison between network topology based network analysis and network embedding based network analysis.

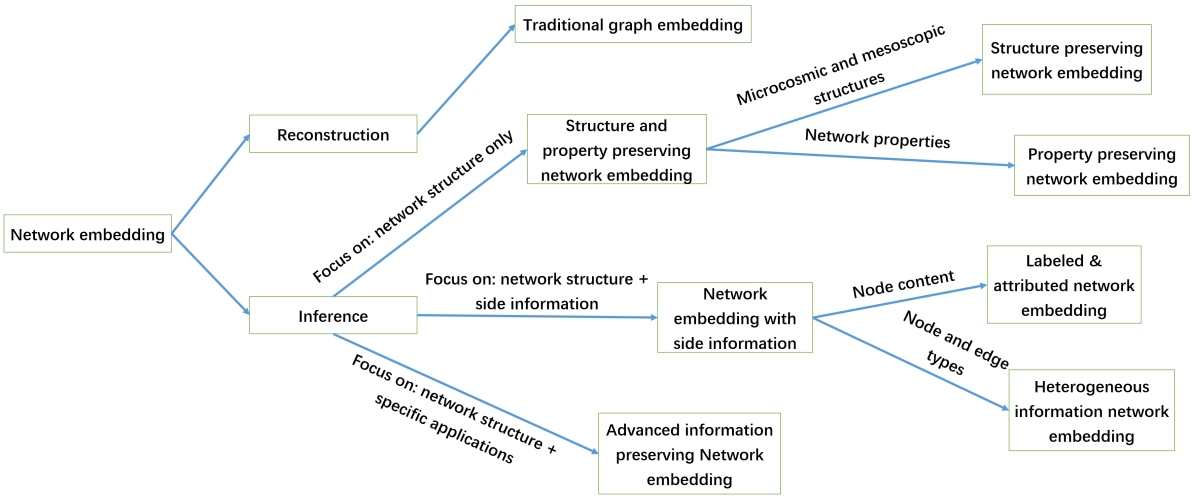


Fig. 3: An overview of different settings of network embedding.

network inference. Consider a network with only topology information. Many network analysis tasks, such as identifying important nodes and predicting unseen links, can be conducted in the original network space. However, as mentioned before, directly conducting these tasks based on network topology has a series of problems, and thus poses a question that whether we can learn a network embedding space purely based on the network topology information, such that these tasks can be well supported in this low dimensional space. Motivated by this, attempts are proposed to preserve rich structural information into network embedding, from nodes and links [10] to neighborhood structure [3], high-order proximities of nodes [6], and community structures [4]. All these types of structural information have been demonstrated useful and necessary in various network analysis tasks. Besides this structural information, network properties in the original network space are not ignorable in modeling the formation and evolution of networks. To name a few, network transitivity (i.e. triangle closure) is the driving force of link formation in networks [11], and structural balance property plays an important role in the evolution of signed networks [12]. Preserving these properties in a network embedding space is, however, challenging due to the inhomogeneity between the network space and the embedding vector space. Some recent studies begin to look into this problem and demonstrate the possibility of aligning these two spaces at the property level [8], [13].

2.1.2 Network Embedding with Side Information

Besides network topology, some types of networks are accompanied with rich side information, such as node content or labels in information networks [14], node and edge attributes in social networks [15], as well as node types in heterogeneous networks [16]. Side information provides useful clues for characterizing relationships among network nodes, and thus is helpful in learning embedding vector spaces. In the cases where the network topology is relatively sparse, the importance of the side information as complementary information sources is even more substantial. Methodologically, the main challenge is how to integrate and balance the topological and side information in network embedding. Some multimodal and multisource fusion techniques are explored in this line of research [15], [17].

2.1.3 Advanced Information Preserving Network Embedding

In the previous two categories, most methods learn network embedding in an unsupervised manner. That is, we only take the network structure, properties, and side information into account,

and try to learn an embedding space to preserve the information. In this way, the learned embedding space is general and, hopefully, able to support various network applications. If we regard network embedding as a way of network representation learning, the formation of the representation space can be further optimized and confined towards different target problems. Realizing this idea leads to supervised or pseudo supervised information (i.e. the advanced information) in the target scenarios. Directly designing a framework of representation learning for a particular target scenario is also known as an end-to-end solution [18], where high-quality supervised information is exploited to learn the latent representation space from scratch. End-to-end solutions have demonstrated their advantages in some fields, such as computer vision [19] and natural language processing (NLP) [20]. Similar ideas are also feasible for network applications. Taking the network node classification problem as an example, if we have the labels of some network nodes, we can design a solution with network structure as input, node labels as supervised information, and embedding representation as latent middle layer, and the resulted network embedding is specific for node classification. Some recent works demonstrate the feasibility in applications such as cascading prediction [18], anomaly detection [21], network alignment [22] and collaboration prediction [23].

In general, network structures and properties are the fundamental factors that need to be considered in network embedding. Meanwhile, side information on nodes and links, as well as advanced information from target problem is helpful to enable the learned network embedding work well in real applications.

2.2 Commonly Used Models in Network Embedding

To transform networks from original network space to embedding space, different models can be adopted to incorporate different types of information or address different goals. The commonly used models include matrix factorization, random walk, deep neural networks and their variations.

2.2.1 Matrix Factorization

An adjacency matrix is commonly used to represent the topology of a network, where each column and each row represent a node, and the matrix entries indicate the relationships among nodes. We can simply use a row vector or column vector as the vector representation of a node, but the formed representation space is N-dimensional, where N is the total number of nodes. Network embedding, aiming to learn a low-dimensional vector space for a network, is eventually to find a low-rank space to represent a network, in contrast with the N-dimensional space. In this sense, matrix factorization methods, with the same goal of learning low-rank space for the original matrix, can naturally be applied to solve this problem. In the series of matrix factorization models, Singular Value Decomposition (SVD) is commonly used in network embedding due to its optimality for low-rank approximation [8]. Non-negative matrix factorization is often used because of its advantages as an additive model [4].

2.2.2 Random Walk

As mentioned before, preserving network structure is a fundamental requirement for network embedding. Neighborhood structure, describing the local structural characteristics of a node, is important for network embedding. Although the adjacency vector of a node encodes the first-order neighborhood structure of a node, it is usually a sparse, discrete, and high-dimensional vector due to the nature of sparseness in large-scale networks. Such a representation is not friendly to subsequent applications. In the field of natural language processing (NLP), the word representation also suffers from similar drawbacks. The development of Word2Vector [24] significantly improves the effectiveness of word representation by transforming sparse, discrete and high-dimensional vectors into dense, continuous and low-dimensional vectors. The intuition of Word2Vector is that a word vector should be able to reconstruct the vectors of its neighborhood words which are defined by co-occurence rate. Some methods in network embedding borrow these ideas. The key problem is how to define "neighborhood" in networks.

To make analogy with Word2Vector, random walk models are exploited to generate random paths over a network. By regarding a node as a word, we can regard a random path as a sentence, and the node neighborhood can be identified by co-occurence rate as in Word2Vector. Some representative methods include DeepWalk [3] and Node2Vec [25].

2.2.3 Deep Neural Networks

By definition, network embedding is to transform the original network space into a low-dimensional vector space. The intrinsic problem is to learn a mapping function between these two spaces. Some methods, like matrix factorization, assume the mapping function to be linear. However, the formation process of a network is complicated and highly nonlinear, thus a linear function may not be adequate to map the original network to an embedding space.

If seeking for an effective non-linear function learning model, deep neural networks are certainly useful options because of their huge successes in other fields. The key challenges are how to make deep models fit network data, and how to impose network structure and property-level constraints on deep models. Some representative methods, such as SDNE [6], SDAE [26], and SiNE [13], propose deep learning models for network embedding to address these challenges. At the same time, deep neural networks are also well known for their advantages in providing end-to-end solutions. Therefore, in the problems where advanced information is available, it is natural to exploit deep models to come up with an end-to-end network embedding solution. For instance, some deep model based end-to-end solutions are proposed for cascade prediction [18] and network alignment [22].

The network embedding models are not limited to those mentioned in this subsection. Moreover, the three kinds of models are not mutually exclusive, and their combinations are possible to make new solutions. More models and details will be discussed in later sections.

3 Network Embedding v.s. Graph Embedding

The goal of graph embedding is similar as network embedding, that is, to embed a graph into a low-dimensional vector space [27]. There is a rich literature in graph embedding. Fu and Ma [9] provide a thorough review on the traditional graph embedding methods. Here we only present some representative and classical methods on graph embedding, aiming to demonstrate the critical differences between graph embedding and the current network embedding.

3.1 Representative Graph Embedding Methods

Graph embedding methods are originally studied as dimension reduction techniques. A graph is usually constructed from a feature represented data set, like image data set. Isomap [28] first constructs a neighborhood graph G using connectivity algorithms such as K nearest neighbors (KNN), i.e., connecting data entries i and j if i is one of the K nearest neighbors of j. Then based on G, the shortest path d_{ij}^G of entries i and j in G can be computed. Consequently, for all the N data entries in the data set, we have the matrix of graph distances $D_G = \{d_{ij}^G\}$. Finally, the classical multidimensional scaling (MDS) method is applied to D_G to obtain the coordinate vector $\mathbf{u}_i$ for entry i, which aims to minimize the following function:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(d_{ij}^G - \|\mathbf{u}_i - \mathbf{u}_j\|)^2. \tag{1}$$

Indeed, Isomap learns the representation $\mathbf{u}_i$ of entry i, which approximately preserves the geodesic distances of the entry pairs in the low-dimensional space.

The key problem of Isomap is its high complexity due to the computing of pair-wise shortest pathes. Locally linear embedding (LLE) [29] is proposed to eliminate the need to estimate the pairwise distances between widely separated entries. LLE assumes that each entry and its neighbors lie on or close to a locally linear patch of a mainfold. To characterize the local geometry, each entry can be reconstructed from its neighbors as follows:

$$\min_{\mathbf{W}} \sum_i \|\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j\|^2, \tag{2}$$

where the weight W_{ij} measures the contribution of the entry $\mathbf{x}_j$ to the reconstruction of entry $\mathbf{x}_i$. Finally, in the low-dimensional space, LLE constructs a neighborhood-preserving mapping based on locally linear reconstruction as follows:

$$\min_{\mathbf{U}} \sum_i \|\mathbf{u}_i - \sum_j W_{ij}\mathbf{u}_j\|^2. \tag{3}$$

By optimizing the above function, the low-dimensional representation matrix $\mathbf{U}$, which preserves the neighborhood structure, can be obtained.

Laplacian eigenmaps (LE) [30] also begins with constructing a graph using ϵ-neighborhoods or K nearest neighbors. Then the heat kernel [31] is utilized to choose the weight W_{ij} of nodes i and j in the graph. Finally, the representation $\mathbf{u}_i$ of node i can be obtained by minimizing the following function:

$$\sum_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2 W_{ij} = tr(\mathbf{U}^T\mathbf{L}\mathbf{U}), \tag{4}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix, and $\mathbf{D}$ is the diagonal matrix with $D_{ii} = \sum_j W_{ji}$. In addition, the constraint $\mathbf{U}^T\mathbf{D}\mathbf{U} = \mathbf{I}$ is introduced to avoid trivial solutions. Furthermore, the locality preserving projection (LPP) [32], a linear approximation of the nonlinear LE, is proposed. Also, it introduces a transformation matrix $\mathbf{A}$ such that the representation $\mathbf{u}_i$ of entry $\mathbf{x}_i$ is $\mathbf{u}_i = \mathbf{A}^T\mathbf{x}_i$. LPP computes the transformation matrix $\mathbf{A}$ first, and finally the representation $\mathbf{u}_i$ can be obtained.

These methods are extended in the rich literature of graph embedding by considering different characteristics of the constructed graphs [9].

3.2 Major Differences

Network embedding and graph embedding have substantial differences in objective and assumptions. As mentioned before, network embedding has two goals, i.e. reconstructing original networks and support network inference. The objective functions of graph embedding methods mainly target the goal of graph reconstruction. As discussed before, the embedding space learned for network reconstruction is not necessarily good for network inference. Therefore, graph embedding can be regarded as a special case of network embedding, and the recent research progress on network embedding pays more attention to network inference.

Moreover, graph embedding mostly works on graphs constructed from feature represented data sets, where the proximity among nodes encoded by the edge weights are well defined in the original feature space. In contrast, network embedding mostly works on naturally formed networks, such as social networks, biology networks, and e-commerce networks. In those networks, the proximities among nodes are not explicitly or directly defined. The definition of node proximities depends on specific analytic tasks and application scenarios. Therefore, we have to incorporate rich information, such as network structures, properties, side information and advanced information, in network embedding to facilitate different problems and applications.

In the rest of the paper, we mainly focus on the network embedding methods with the goal of supporting network inference.

4 Structure and Property Preserving Network Embedding

In essence, one basic requirement of network embedding is to appropriately preserve network structures and capture properties of networks. Often, network structures include first-order structure and higher-order structure, such as second-order structure and community structure. Networks with different types have different properties. For example, directed networks have the asymmetric transitivity property. The structural balance theory is widely applicable to signed networks.

In this section, we review the representative methods of structure preserving network embedding and property preserving network embedding.

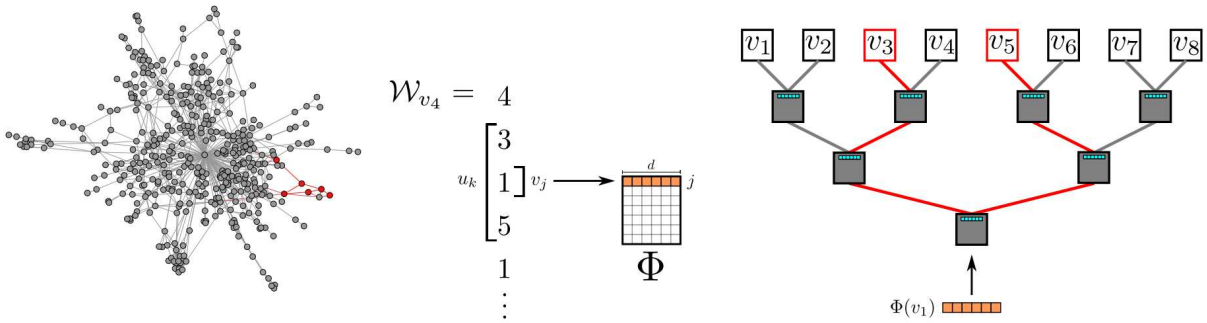
4.1 Structure Preserving Network Embedding

Network structures can be categorized into different groups that present at different granularities. The commonly exploited network structures in network embedding include neighborhood structure, high-order node proximity and network communities.

DeepWalk [3] is proposed for learning the representations of nodes in a network, which is able to preserve the neighbor structures of nodes. DeepWalk discovers that the distribution of nodes appearing in short random walks is similar to the distribution of words in natural language. Motivated by this observation, Skip-Gram model [24], a widely used word representation learning model, is adopted by DeepWalk to learn the representations of nodes. Specifically, as shown in Fig. 4, DeepWalk adopts a truncated random walk on a network to generate a set of walk sequences. For each walk sequence $s = \{v_1, v_2, ..., v_s\}$, following Skip-Gram, DeepWalk aims to maximize the probability of the neighbors of node v_i in this walk sequence as follows:

$$\max_{\phi} Pr(\{v_{i-w}, ..., v_{i+w}\}\backslash v_i | \phi(v_i)) = \Pi_{j=i-w, j\neq i}^{i+w} Pr(v_j|\phi(v_i)), \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2849727, IEEE Transactions on Knowledge and Data Engineering



(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

Fig. 4: Overview of DeepWalk. Image extracted from [3].

where w is the window size, $\phi(v_i)$ is the current representation of v_i and $\{v_{i-w}, ..., v_{i+w}\}\backslash v_i$ is the local context nodes of v_i. Finally, hierarchical soft-max [33] is used to efficiently infer the embeddings.

Node2vec [25] demonstrates that DeepWalk is not expressive enough to capture the diversity of connectivity patterns in a network. Node2vec defines a flexible notion of a node's network neighborhood and designs a second order random walk strategy to sample the neighborhood nodes, which can smoothly interpolate between breadth-first sampling (BFS) and depth-first sampling (DFS). Node2vec is able to learn the representations that embed nodes with same network community closely, and to learn representations where nodes sharing similar roles have similar embeddings.

LINE [10] is proposed for large scale network embedding, and can preserve the first and second order proximities. The first order proximity is the observed pairwise proximity between two nodes, such as the observed edge between nodes 6 and 7 in Fig. 5. The second order proximity is determined by the similarity of the "contexts" (neighbors) of two nodes. For example, the second order similarity between nodes 5 and 6 can be obtained by their neighborhoods 1, 2, 3, and 4 in Fig. 5. Both the first order and second order proximities are important in measuring the relationships between two nodes. The first order proximity can be measured by the joint probability distribution between two nodes v_i and v_j as

$$p_1(v_i, v_j) = \frac{1}{1 + exp(-\mathbf{u}_i^T \mathbf{u}_j)}. \tag{6}$$

The second order proximity is modeled by the probability of the context node v_j being generated by node v_i, that is,

$$p_2(v_j|v_i) = \frac{exp(\bar{\mathbf{u}}_j^T \bar{\mathbf{u}}_i)}{\sum_k exp(\bar{\mathbf{u}}_k^T \bar{\mathbf{u}}_i))}. \tag{7}$$

The conditional distribution implies that nodes with similar distributions over the contexts are similar to each other. By minimizing the KL-divergence of the two distributions and the empirical distributions respectively, the representations of nodes that are able to preserve the first and second order proximities can be obtained.

Considering that LINE only preserves the first-order and second-order proximities, GraRep [34] demonstrates that k-step ($k > 2$) proximities should also be captured when constructing the global representations of nodes. Given the adjacency matrix $\mathbf{A}$, the k-step probability transition matrix can be computed by $\mathbf{A}^k = \underbrace{\mathbf{A}...\mathbf{A}}_{k}$, whose element A_{ij}^k refers to the transition probability $p_k(j|i)$ from a current node i to a context node j and the transition consists of k steps. Moreover, motivated by the Skip-Gram model [24], the k-step loss function of node i is defined as

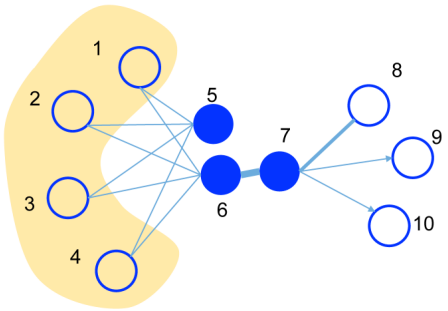


Fig. 5: An example of the first-order and second-order structures in a network. Image extracted from [10].

$$L_k(i) = (\sum_j p_k(j|i) \log \sigma(\mathbf{u}_j^T \mathbf{u}_i)) + \lambda \mathbb{E}_{j' \sim p_k(V)}[\log \sigma(-\mathbf{u}_i^T \mathbf{u}_{j'})], \tag{8}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$, $p_k(V)$ is the distribution over the nodes in the network and j' is the node obtained from negative sampling. Furthermore, GraRep reformulates the loss function as the matrix factorization problem, for each k-step loss function, SVD can be directly used to infer the representations of nodes. By concentrating the representations learned from each function, the global representations can be obtained.

Wang *et al.* [4] propose a modularized nonnegative matrix factorization (M-NMF) model for network embedding, which aims to preserve both the microscopic structure, i.e., the first-order and second-order proximities of nodes, and the mesoscopic community structure [35]. To preserve the microscopic structure, they adopt the NMF model [36] to factorize the pairwise node similarity matrix and learn the representations of nodes. Meanwhile, the community structure is detected by modularity maximization [37]. Then, based on the assumption that if the representation of a node is similar to that of a community, the node may have a high propensity to be in this community, they introduce an auxiliary community representation matrix to bridge the representations of nodes with the community structure. In this way, the learned representations of nodes are constrained by both the microscopic structure and community structure, which contains more structural information and becomes more discriminative.

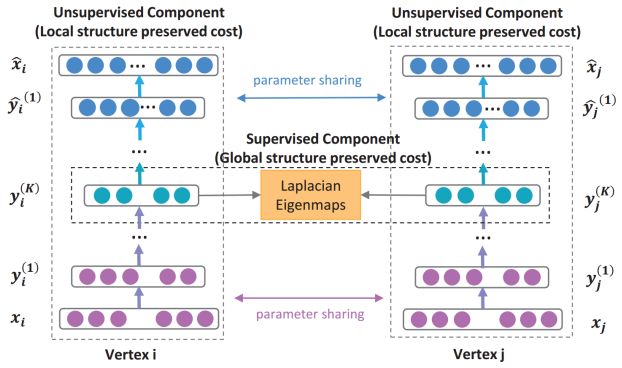


Fig. 6: The framework of SDNE. Image extracted from [6].

The aforementioned methods mainly adopt the shallow models, consequently, the representation ability is limited. SDNE [6] proposes a deep model for network embedding, so as to address the high non-linearity, structure-preserving, and sparsity issues. The framework is shown in Fig. 6. Specifically, SDNE uses the deep autoencoder with multiple non-linear layers to preserve the neighbor structures of nodes. Given the input adjacency nodes $\mathbf{x}_i$ of node i, the hidden representations for each layer can be obtained by

$$\begin{aligned}\mathbf{y}_i^{(1)} &= \sigma(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}) \\ \mathbf{y}_i^{(k)} &= \sigma(\mathbf{W}^{(k)}\mathbf{y}_i^{(k-1)} + \mathbf{b}^{(k)}), k = 2, ..., K.\end{aligned} \tag{9}$$

Then the output representation $\hat{\mathbf{x}}_i$ can be obtained by reversing the calculation process of encoder. To impose more penalty to the reconstruction error of the non-zero elements than that of zero elements, SDNE introduces the penalty vector $\mathbf{b}_i = \{b_{ij}\}_{j=1}^n$ (b_{ij} is larger than a threshold if there is an edge between nodes i and j) and gives rise to the following function that can preserve the second-order proximity

$$\mathcal{L}_{2nd} = \sum_i \|(\hat{\mathbf{x}}_i - \mathbf{x}_i) \odot \mathbf{b}_i\|^2. \tag{10}$$

To preserve the first-order proximity of nodes, the idea of Laplacian eigenmaps [30] is adopted. By exploiting the first-order and second-order proximities jointly into the learning process, the representations of nodes can be finally obtained.

Cao *et al.* [26] propose a network embedding method to capture the weighted graph structure and represent nodes of non-linear structures. As shown in Fig. 7, instead of adopting the previous sampling strategy that needs to determine certain hyper parameters, they considers a random surfing model motivated by the PageRank model. Based on this random surfing model, the representation of a node can be initiatively constructed by combining the weighted transition probability matrix. After that, the PPMI matrix [38] can be computed. Finally, the stacked denoising autoencoders [39] that partially corrupt the input data before taking the training step are applied to learn the latent representations.

In order to make a general framework on network embedding, Chen *et al.* [40] propose a network embedding framework that unifies some of the previous algorithms, such as LE, DeepWalk and Node2vec. The proposed framework, denoted by GEM-D$[h(\cdot), g(\cdot), d(\cdot,\cdot)]$, involves three important building blocks: $h(\cdot)$ is a node proximity function based on the adjacency matrix; $g(\cdot)$ is a warping function that warps the inner products of network embeddings; and $d(\cdot,\cdot)$ measures the differences between h and g. Furthermore, they demonstrate that the high-order proximity for $h(\cdot)$ and the exponential function for $g(\cdot)$ are more important for a network embedding algorithm. Based on these observations, they propose UltimateWalk=GEM-D$[\prod^{(L)}, exp(x), d_{wf}(\cdot,\cdot)]$, where $\prod^{(L)}$ is a finite-step transition matrix, $exp(x)$ is an exponential function and $d_{wf}(\cdot,\cdot)$ is the warped Frobenius norm.

The main goal of the aforementioned methods is to learn the embedding for nodes, currently there are some methods proposed to learn the whole-graph embedding. For example, in [41], they define some connected and non-isomorphic induced sub-graphs and different networks consist of these sub-graphs. Then language modeling method, such as SkipGram, can be applied to the edit-distance graph of these sub-graphs. Therefore, the representations of these sub-graphs can be learned and the similarities of different networks can be captured. PATCHY-SAN [42] is proposed to learn the embedding for a whole graph based on convolutional neural network (CNN) [43], so as to deal with the whole graph related tasks. In order to make the traditional CNN compatible with the network data, they elaborately design several network data preprocessing steps, such as node sequence selection and graph normalization. In this way, the network topology can be transformed to the formation for CNN.

In summary, many network embedding methods aim to preserve the local structure of a node, including neighborhood structure, high-order proximity as well as community structure, in the latent low-dimensional space. Both linear and non-linear models are attempted, demonstrating the large potential of deep models in network embedding.

4.2 Property Preserving Network Embedding

Among the rich network properties, the properties that are crucial for network inference are the focus in property preserving network embedding. Specifically, most of the existing property preserving network embedding methods focus on network transitivity in all types of networks and the structural balance property in signed networks.

Ou *et al.* [44] aim to preserve the non-transitivity property via latent similarity components. The non-transitivity property declares that, for nodes A, B and C in a network where (A, B) and (B, C) are similar pairs, (A, C) may be a dissimilar pair. For example, in a social network, a student may connect with his classmates and his family, while his classmates and family are probably very different. To address this, they use a set of linear projection matrices to extract M hash tables, and thus, each pair of nodes can have M similarities $\{S_{ij}^m\}_{m=1}^M$ based on those hash tables. Then the final similarity between two nodes can be aggregated from $\{S_{ij}^m\}_{m=1}^M$. Finally they approximate the aggregated similarity to the semantic similarity based on the observation that if two nodes have a large semantic similarity, at least one of the similarities S_{ij}^m from the hash tables is large, otherwise, all of the similarities are small.

Preserving the asymmetric transitivity property of directed network is considered by HOPE [8]. Asymmetric transitivity indicates that, if there is a directed edge from node i to node j and a directed edge from j to v, there is likely a directed edge from i to v, but not from v to i. In order to measure this high-order proximity, HOPE summarizes four measurements in a general formulation, that is, Katz Index [45], Rooted PageRank [7], Common Neighbors [7], and Adamic-Adar [46]. With the high-order proximity, SVD can be directly applied to obtain the low

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2849727, IEEE Transactions on Knowledge and Data Engineering

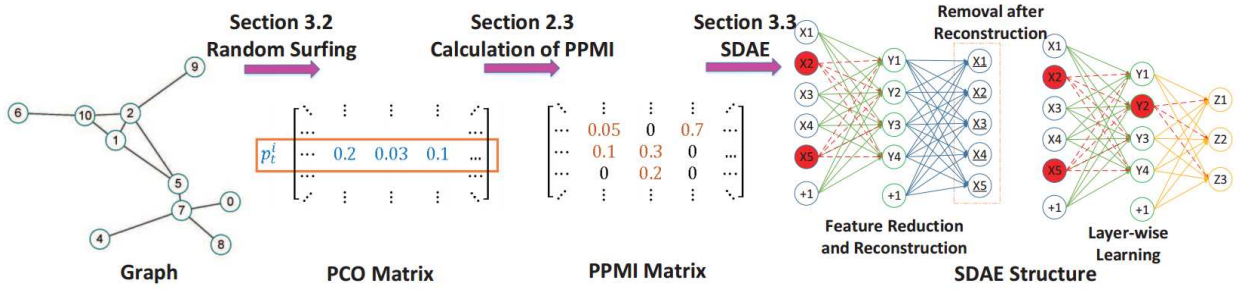


Fig. 7: Overview of the method proposed by Cao *et al.* [26]. Image extracted from [26].

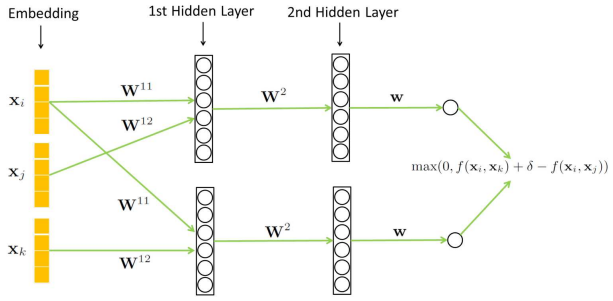


Fig. 8: The framework of SiNE. Image extracted from [13].

dimensional representations. Furthermore, the general formulation of high-order proximity enables HOPE to transform the original SVD problem into a generalized SVD problem [47], such that the time complexity of HOPE is largely reduced, which means HOPE is scalable for large scale networks.

SiNE [13] is proposed for signed network embedding, which considers both positive and negative edges in a network. Due to the negative edges, the social theories on signed network, such as structural balance theory [12], [48], are very different from the unsigned network. The structural balance theory demonstrates that users in a signed social network should be able to have their "friends" closer than their "foes". In other words, given a triplet (v_i, v_j, v_k) with edges $e_{ij} = 1$ and $e_{ik} = -1$, the similarity $f(v_i, v_j)$ between nodes v_i and v_j is larger than $f(v_i, v_k)$. To model the structural balance phenomenon, a deep learning model consisting of two deep networks with non-linear functions is designed to learn the embeddings and preserve the network structure property, which is consistent with the extended structural balance theory. The framework is shown in Fig. 8.

The methods reviewed in this subsection demonstrate the importance of maintaining network properties in network embedding space, especially the properties that largely affect the evolution and formation of networks. The key challenge in is how to address the disparity and heterogeneity of the original network space and the embedding vector space at property level.

4.3 Summary

Generally, most of the structure and property preserving methods take high order proximities of nodes into account, which demonstrate the importance of preserving high order structures in network embedding. The difference is the strategy of obtaining the high order structures. Some methods implicitly preserve high-order structure by assuming a generative mechanism from a node to its neighbors, while some other methods realize this by explicitly approximating high-order proximities in the embedding space. As topology structures are the most notable characteristic of networks, structure-preserving network methods embody a large part of the literature. Comparatively, property preserving network embedding is a relatively new research topic and is only studied lightly. As network properties usually drive the formation and evolution of networks, it shows great potential for future research and applications.

5 Network Embedding with Side Information

Besides network structures, side information is another important information source for network embedding. Side information in the context of network embedding can be divided into two categories: node content and types of nodes and edges. In this section, we review the methods that take side information into network embedding.

5.1 Network Embedding with Node Content

In some types of networks, like information networks, nodes are acompanied with rich information, such as node labels, attributes or even semantic descriptions. How to combine them with the network topology in network embedding arouses considerable research interests.

Tu *et al.* [14] propose a semi-supervised network embedding algorithm, MMDW, by leveraging labeling information of nodes. MMDW is also based on the DeepWalk-derived matrix factorization. MMDW adopts support vector machines (SVM) [49] and incorporates the label information to find an optimal classifying boundary. By optimizing the max-margin classifier of SVM and matrix factorization based DeepWalk simultaneously, the representations of nodes that have more discriminative ability can be learned.

Le *et al.* [50] propose a generative model for document network embedding, where the words associated with each documents and the relationships between documents are both considered. For each node, they learn its low-rank representation $\mathbf{u}_i$ in a low dimensional vector space, which can reconstruct the network structure. Also, they learn the representation of nodes in the topic space based on the Relational Topic Model (RTM) [51], where each topic z is associated with a probability distribution over words. To integrate the two aspects, they associate each topic z with a representation φ_z in the same low dimensional vector space and then have the following function:

$$P(z|v_i) = \frac{exp(-\frac{1}{2}\|\mathbf{u}_i - \varphi_z\|^2)}{\sum_z exp(-\frac{1}{2}\|\mathbf{u}_i - \varphi_z\|^2)}. \quad (11)$$

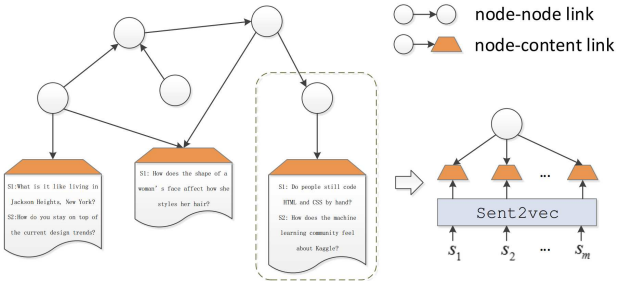


Fig. 9: The augmented network proposed by Sun *et al.* [52]. Image extracted from [52].

Finally, in a unified generative process, the representations of nodes $\mathbf{U}$ can be learned.

Besides network structures, Yang *et al.* [15] propose TADW that takes the rich information (e.g., text) associated with nodes into account when they learn the low dimensional representations of nodes. They first prove that DeepWalk is equivalent to factorizing the matrix $\mathbf{M}$ whose element $M_{ij} = \log([e_i(\mathbf{A} + \mathbf{A}^2 + ... + \mathbf{A}^t)]_j/t)$, where $\mathbf{A}$ is the adjacency matrix, t denotes the t steps in a random walk and e_i is a row vector where all entries are 0 except the i-th entry is 1. Then, based on the DeepWalk-derived matrix factorization and motivated by the inductive matrix completion [17], they incorporate rich text information $\mathbf{T}$ into network embedding as follows:

$$\min_{\mathbf{W},\mathbf{H}} \|\mathbf{M} - \mathbf{W}^T\mathbf{H}\mathbf{T}\|_F^2 + \frac{\lambda}{2}(\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2). \tag{12}$$

Finally, they concatenate the optimal $\mathbf{W}$ and $\mathbf{HT}$ as the representations of nodes.

TADW suffers from high computational cost and the node attributes just simply incorporated as unordered features lose the much semantic information. Sun *et al.* [52] consider the content as a special kind of nodes, and give rise to an augmented network, as shown in Fig. 9. With this augmented network, they are able to model the node-node links and node-content links in the latent vector space. They use a logistic function to model the relationship in the new augmented network, and by combining with negative sampling, they can learn the representations of nodes in a joint objective function, such that the representations can preserve the network structure as well as the relationship between the node and content.

Pan *et al.* [53] propose a coupled deep model that incorporates network structure, node attributes and node labels into network embedding. The architecture of the proposed model is shown in Fig. 10. Consider a network with N nodes $\{v_i\}_{i=1,...,N}$, where each node is associated with a set of words $\{w_i\}$, and some nodes may have $|L|$ labels $\{c_i\}$. To exploit this information, they aim to maximize the following function:

$$\begin{aligned}\mathcal{L} =& (1-\alpha)\sum_{i=1}^{N}\sum_{s\in\mathcal{S}}\sum_{-b\le j\le b, j\neq 0}\log P(v_{i+j}|v_i) \\ & \alpha\sum_{i=1}^{N}\sum_{-b\le j\le b}\log P(w_j|v_i) + \alpha\sum_{i=1}^{|L|}\sum_{-b\le j\le b}\log P(w_j|c_i),\end{aligned} \tag{13}$$

where $\mathcal{S}$ is the random walks generated in the network and b is the window size of sequence. Specifically, function P, which captures the probability of observing contextual nodes (or words)

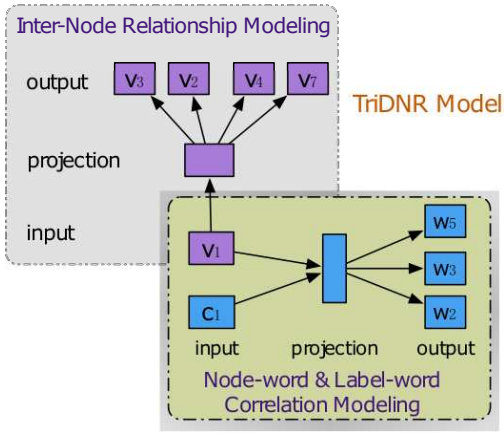


Fig. 10: The framework of TriDNR [53]. Image extracted from [53].

given the current node (or label), can be computed using the softmax function. In Eq. 13, the first term is also motivated by Skip-Gram, similar to DeepWalk, which models the network structure. The second term models the node-content correlations and the third term models the label-node correspondences. As a result, the learned representations is enhanced by network structure, node content, and node labels.

LANE [54] is also proposed to incorporate the label information into the attributed network embedding. Unlike the previous network embedding methods, LANE is mainly based on spectral techniques [55]. LANE adopts the cosine similarity to construct the corresponding affinity matrices of the node attributes, network structure, and labels. Then, based on the corresponding Laplacian matrices, LANE is able to map the three different sources into different latent representations, respectively. In order to build the relationship among those three representations, LANE projects all these latent representations into a new common space by leveraging the variance of the projected matrix as the correlation metric. The learned representations of nodes are able to capture the structure proximities as well as the correlations in the label informed attributed network.

Huang *et al.* [56] pay more attentions on the scalability of attributed network embedding. The proposed method, named AANE, is based on the decomposition of attribute affinity matrix and the penalty of embedding difference between linked nodes. AANE provides a distributed optimization algorithm to process each node efficiently.

Wei *et al.* [57] study the problem of cross view link prediction (CVLP) based on attributed network embedding, i.e., to recommend nodes with only links to nodes with only attributes (or vice versa). The proposed model learns the link-based and attribute-based representations, and utilize the consensus to establish the relations between them.

Although different methods adopt different strategies to integrate node content and network topology, they all assume that node content provides additional proximity information to constrain the representations of nodes.

5.2 Heterogeneous Information Network Embedding

Different from networks with node content, heterogeneous networks consist of different types of nodes and links. How to unify

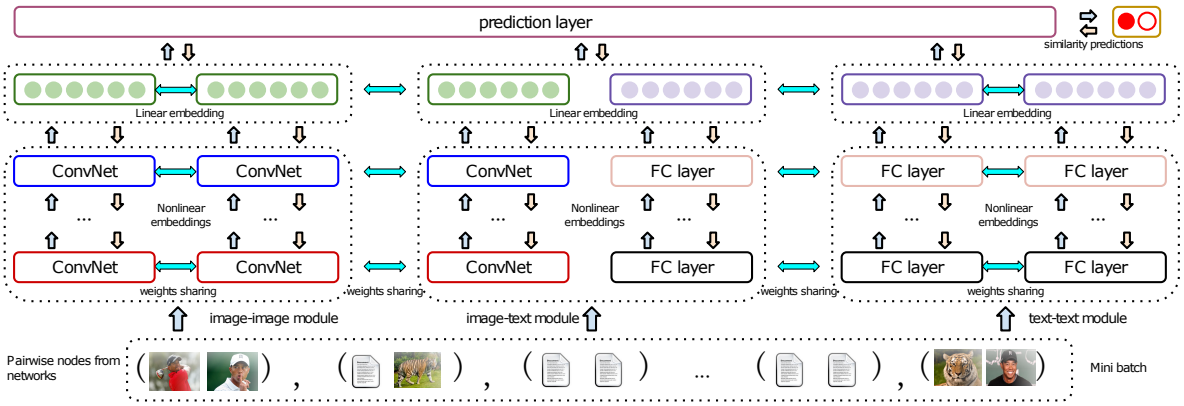


Fig. 11: Overview of the method proposed by Chang *et al.* [16]. Image extracted from [16].

the heterogeneous types of nodes and links in network embedding is also an interesting and challenging problem.

Yann *et al.* [58] propose a heterogeneous social network embedding algorithm for classifying nodes. They learn the representations of all types of nodes in a common vector space, and perform the inference in this space. In particular, for the node $\mathbf{u}_i$ with type t_i, they utilize a linear classification function $f_\theta^{t_i}$ to predict its label and adopt the hinge-loss function Δ to measure the loss with the true label y_i:

$$\sum_{i=1}^{l} \Delta(f_\theta^{t_i}(\mathbf{u}_i), y_i), \tag{14}$$

where l is the number of labeled nodes. To preserve the local structures in the latent space, they impose the following smoothness constraint, which enforces that two nodes i and j will be close in the latent space if they have a large weight W_{ij} in the heterogeneous network:

$$\sum_{i,j} W_{ij} \|\mathbf{u}_i - \mathbf{u}_j\|^2. \tag{15}$$

In this way, different types of nodes are mapped into a common latent space. The overall loss function combines the classification and regularization losses Eq. (14) and Eq. (15). A stochastic gradient descent method is used here to learn the representations of nodes in a heterogeneous network for classifying.

Chang *et al.* [16] propose a deep embedding algorithm for heterogeneous networks, whose nodes have various types. The main goal of the heterogeneous network embedding is to learn the representations of nodes with different types such that the heterogeneous network structure can be well preserved. As shown in Fig. 11, given a heterogeneous network with two types of data (e.g., images and texts), there are three types of edges, i.e., image-image, text-text, and image-text. The nonlinear embeddings of images and texts are learned by a CNN model and the fully connected layers, respectively. By cascading the extra linear embedding layer, the representations of images and texts can be mapped to a common space. In the common space, the similarities between data from different modalities can be directly measured, so that if there is an edge in the original heterogeneous network, the pair of data has similar representations.

Huang and Mamoulis [59] propose a meta path similarity preserving heterogeneous information network embedding algorithm. To model a particular relationship, a meta path [60] is a sequence of object types with edge types in between. They develop a fast dynamic programming approach to calculate the truncated meta path based proximities, whose time complexity is linear to the size of the network. They adopt a similar strategy as LINE [10] to preserve the proximity in the low dimensional space.

Xu *et al.* [61] propose a network embedding method for coupled heterogeneous network. The coupled heterogeneous network consists of two different but related homogeneous networks. For each homogeneous network, they adopt the same function (Eq. (6)) as LINE to model the relationships between nodes. Then the harmonious embedding matrix is introduced to measure the closeness between nodes of different networks. Because the inter-network edges are able to provide the complementary information in the presence of intra-network edges, the learned embeddings of nodes also perform well on several tasks.

5.3 Summary

In the methods preserving side information, side information introduces additional proximity measures so that the relationships between nodes can be learned more comprehensively. Their difference is the way of integrating network structures and side information. Many of them are naturally extensions from structure preserving network embedding methods.

6 Advanced Information Preserving Network Embedding

In this section, we review network embedding methods that take additional advanced information into account so as to solve some specific analytic tasks. Different from side information, the advanced information refers to the supervised or pseudo supervised information in a specific task.

6.1 Information Diffusion

Information diffusion [62] is an ubiquitous phenomenon on the web, especially in social networks. Many real applications, such as marketing, public opinion formation, epidemics, are related to information diffusion. Most of the previous studies on information diffusion are conducted in original network spaces.

Recently, Simon *et al.* [63] propose a social network embedding algorithm for predicting information diffusion. The basic idea is to map the observed information diffusion process into a heat

diffusion process modeled by a diffusion kernel in the continuous space. Specifically, the diffusion kernel in a d-dimensional Euclidean space is defined as

$$K(t,j,i) = (4\Pi t)^{-\frac{d}{2}} e^{-\frac{\|j-i\|^2}{4t}}. \quad (16)$$

It models the heat at location i at time t when an initial unit heat is positioned at location j, which also models how information spreads between nodes in a network.

The goal of the proposed algorithm is to learn the representations of nodes in the latent space such that the diffusion kernel can best explain the cascades in the training set. Given the representation $\mathbf{u}_j$ of the initial contaminated node j in cascade c, the contamination score of node i can be computed by

$$K(t,j,i) = (4\Pi t)^{-\frac{d}{2}} e^{-\frac{\|\mathbf{u}_j-\mathbf{u}_i\|^2}{4t}}. \quad (17)$$

The intuition of Eq. (17) is that the closer a node in the latent space is from the source node, the sooner it is infected by information from the source node. As the cascade c offers a guidance for the information diffusion of nodes, we expect the contamination score to be as closely consistent with c as possible, which gives rise to the following empirical risk function:

$$L(\mathbf{U}) = \sum_c \Delta(K(.,j,.), c), \quad (18)$$

where function Δ is a measure of the difference between the predicted score and the observed diffusion in c. By minimizing the Eq. (18) and reformulating it as a ranking problem, the optimal representations $\mathbf{U}$ of nodes can be obtained.

The cascade prediction problem here is defined as predicting the increment of cascade size after a given time interval [18]. Li *et al.* [18] argue that the previous work on cascade prediction all depends on the bag of hand-crafting features to represent the cascade and network structures. Instead, they present an end-to-end deep learning model to solve this problem using the idea of network embedding, as illustrated in Fig. 12. Similar to DeepWalk [3], they perform a random walk over a cascade graph to sample a set of paths. Then the Gated Recurrent Unite (GRU) [64], a specific type of recurrent neural network [65], is applied to these paths and learn the embeddings for these paths. The attention mechanism is then used to assemble these embeddings to learn the representation of this cascade graph. Once the representation of this cascade is known, a multi-layer perceptron [66] can be adopted to output the final predicted size of this cascade. The whole procedure is able to learn the representation of cascade graph in an end-to-end manner. The experimental results on the Twitter and Aminer networks show promising performance on this task.

6.2 Anomaly Detection

Anomaly detection has been widely investigated in previous work [67]. Anomaly detection in networks aims to infer the structural inconsistencies, which means the anomalous nodes that connect to various diverse influential communities [21], [68], such as the red node in Fig. 13. Hu *et al.* [21] propose a network embedding based method for anomaly detection. In particular, in the proposed model, the k-th element u_i^k in the embedding $\mathbf{u}_i$ of node i represents the correlation between node i and community k. Then, they assume that the community memberships of two linked nodes should be similar. Therefore, they can minimize the following objective function:

$$L = \sum_{(i,j)\in E} \|\mathbf{u}_i - \mathbf{u}_j\|^2 + \alpha \sum_{(i,j)\notin E} (\|\mathbf{u}_i - \mathbf{u}_j\| - 1)^2. \quad (19)$$

This optimization problem can be solved by the gradient descent method. By taking the neighbors of a node into account, the embedding of the node can be obtained by a weighted sum of the embeddings of all its neighbors. An anomaly node in this context is one connecting to a set of different communities. Since the learned embedding of nodes captures the correlations between nodes and communities, based on the embedding, they propose a new measure to indicate the anomalousness level of a node. The larger the value of the measure, the higher the propensity for a node being an anomaly node.

6.3 Network Alignment

The goal of network alignment is to establish the correspondence between the nodes from two networks. Man *et al.* [22] propose a network embedding algorithm to predict the anchor links across social networks. The same users who are shared by different social networks naturally form the anchor links, and these links bridge the different networks. As illustrated in Fig. 14, the anchor link prediction problem is, given source network G^s and target network G^t and a set of observed anchor links T, to identify the hidden anchor links across G^s and G^t.

First, Man *et al.* [22] extend the original sparse networks G^s and G^t to the denser networks. The basic idea is that given a pair of users with anchor links, if they have a connection in one network, so do their counterparts in the other network [69], in this way, more links will be added to the original networks. For a pair of nodes i and j whose representations are $\mathbf{u}_i$ and $\mathbf{u}_j$, respectively, by combining the negative sampling strategy, they use the following function to preserve the structures of G^s and G^t in a vector space:

$$\log \sigma(\mathbf{u}_i^T \mathbf{u}_j) + \sum_{k=1}^{K} E_{v_k \propto P_n(v)}[\log(1 - \sigma(\mathbf{u}_i^T \mathbf{u}_k))], \quad (20)$$

where $\sigma(x) = 1/(1 + exp(-x))$. The first term models the observed edges, and the second term samples K negative edges.

Then given the observed anchor links $(v_i^s, u_j^t) \in T$ and the representations $\mathbf{u}_i$ and $\mathbf{u}_j$, they aim to learn a mapping function ϕ parameterized by θ so as to bridge these two representations. The loss function is defined as:

$$\|\phi(\mathbf{u}_i;\theta) - \mathbf{u}_j\|_F. \quad (21)$$

The mapping function can be linear or non-linear via Multi-Layer Perceptron (MLP) [66]. By optimizing Eq. (20) and Eq. (21) simultaneously, the representations that can preserve the network structure and respect the observed anchor links can be learned.

6.4 Summary

Advanced information preserving network embedding usually consists of two parts. One is to preserve the network structure so as to learn the representations of nodes. The other is to establish the connection between the representations of nodes and the target task. The first one is similar to structure and property preserving network embedding, while the second one usually needs to consider the domain knowledge of a specific

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2849727, IEEE Transactions on Knowledge and Data Engineering

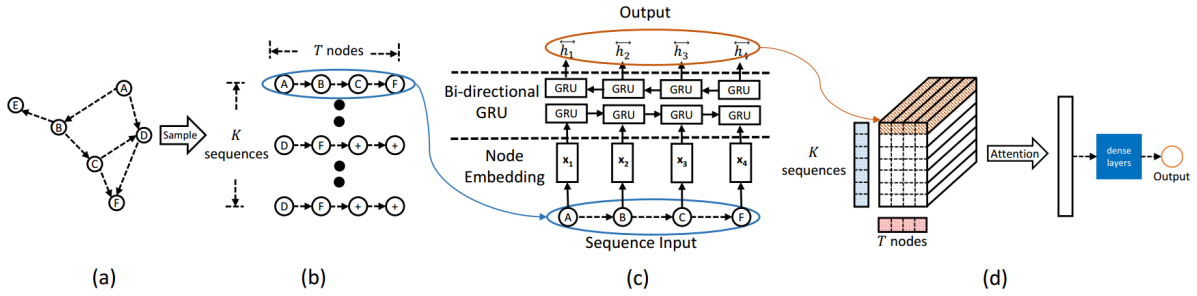


Fig. 12: The end-to-end pipeline of DeepCas proposed by Li *et al.* [18]. Image extracted from [18].

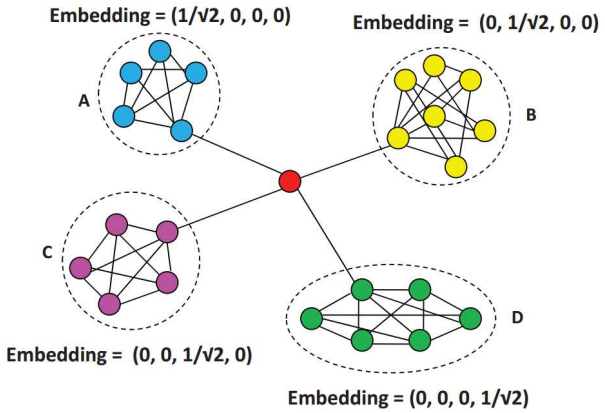


Fig. 13: The anomalous (red) nodes in embedding, and A, B, C, D are four communities [21]. Image extracted from [21].

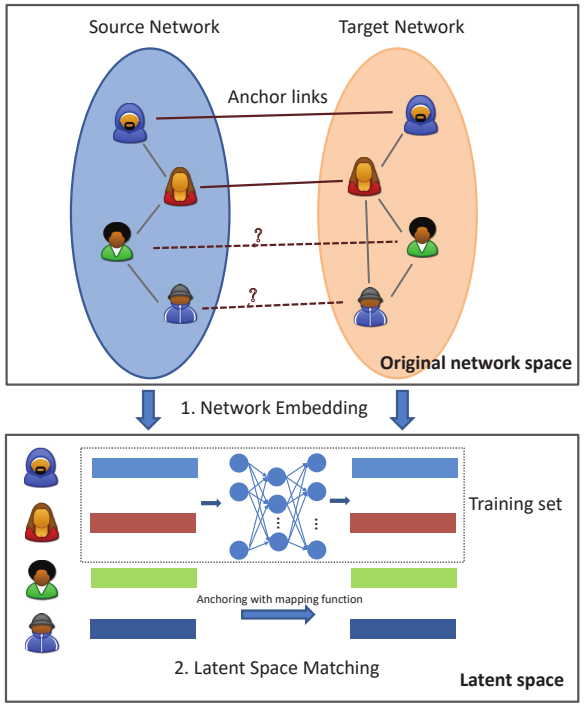


Fig. 14: The illustrative diagram of network embedding for anchor link prediction proposed by Man *et al.* [22]. Image extracted from [22].

task. The domain knowledge encoded by the advanced information makes it possible to develop end-to-end solutions for network applications. Compared with the hand-crafted network features, such as numerous network centrality measures, the combination of advanced information and network embedding techniques enables representation learning for networks. Many network applications may be benefitted from this new paradigm.

All the aforementioned network embedding methods are summarized in Table 1. Because some works lack of the technique details, here we mainly summarize the time complexity of some representative network embedding methods, shown in Table 2. In particular, K is the representation dimension, and N and $|E|$ are the number of nodes and edges, respectively. I is the number of iterations. L is the walk length and w is the number of random-walk trials. e is the number of epochs, and S is the number of sampled training triplets and J is a constant related with the structure of deep neural network. m is the length of node attributes. $nz(\mathbf{X})$ is the number of non-zeros in matrix $\mathbf{X}$. As we can see, the time complexity of some network embedding methods, such as DeepWalk, LINE, and Node2vec, is linear with respect to the number of nodes N. Therefore, these methods are usually scalable for large networks. While, some works, such as GraRep, M-NMF, and LANE, have the quadratic complexity, which may cause the scalability issue and limit them to be applied to the large networks.

7 Network Embedding in Practice

In this section, we summarize the data sets, benchmarks, and evaluation tasks that are commonly used in developing new network embedding methods.

7.1 Real World Data Sets

Getting real network data sets in academic research is always far from trivial. Here, we describe some most popular real world networks currently used in network embedding literature. The data sets can be roughly divided into four groups according to the nature of the networks: social networks, citation networks, language networks, and biological networks. A summary of these data sets can be found in Table 4. Please note that, the same name may be used to refer to different variants in different studies. Here we aim to provide an overview of the networks, and do not attempt to describe all of those variants in detail.

7.1.1 Social Networks

- **BLOGCATALOG** [70]. This is a network of social relationships of the bloggers listed on the BlogCatalog

TABLE 1: A summary of different types of network embedding methods

Method	Categorization			Technique			
	Network alone	Side information	Advanced information	Matrix factorization	Random walk	Deep neural network	others
DeepWalk [3]	√				√		
LINE [10]	√						√
GraRep [34]	√			√			
SDNE [6]	√					√	
Node2vec [25]	√				√		
M-NMF [4]	√			√			
Cao *et al*. [26]	√					√	
Chen *et al*. [40]	√			√			
HOPE [8]	√			√			
SiNE [13]	√					√	
Ou *et al*. [44]		√					√
MMDW [14]		√		√			
Le *et al*. [50]		√					√
TADW [50]		√		√			
Sun *et al*. [52]		√					√
Pan *et al*. [53]		√				√	
LANE [54]		√					√
Yann *et al*. [58]		√					√
Chang *et al*. [16]		√				√	
Huang *et al*. [59]		√					√
Xu *et al*. [61]		√					√
Simon *et al*. [63]			√				√
Li *et al*. [18]			√			√	
Hu *et al*. [21]			√				√
Man *et al*. [22]			√			√	

TABLE 2: The complexity of some representative network embedding methods

Method	Time complexity
DeepWalk [3]	$\mathcal{O}(KN\log N)$
LINE [10]	$\mathcal{O}(K\|E\|)$
GraRep [34]	$\mathcal{O}(N\|E\|+KN^2)$
SDNE [6]	$\mathcal{O}(KI\|E\|)$
Node2vec [25]	$\mathcal{O}(KN)$
M-NMF [4]	$\mathcal{O}(KIN^2)$
Chen *et al*. [40]	$\mathcal{O}(K\max(\|E\|,NLw))$
HOPE [8]	$\mathcal{O}(K^2I\|E\|)$
SiNE [13]	$\mathcal{O}(eNSJ)$
TADW [50]	$\mathcal{O}(N\|E\|+KI\|E\|+KmIN+K^2IN)$
TriDNR [53]	$\mathcal{O}(K*nz(\mathbf{X})\log(m)+KN\log(N))$
LANE [54]	$\mathcal{O}(mN^2+KIN^2)$
Yann *et al*. [58]	$\mathcal{O}(\|E\|)$

website. One instance of this data set can be found at http://socialcomputing.asu.edu/datasets/BlogCatalog3.

- **FLICKR** [70]. This is a network of the contacts between users of the photo sharing websites Flickr. One instance of the network can be downloaded at http://socialcomputing.asu.edu/datasets/Flickr.
- **YOUTUBE** [71]. This is a network between users of the popular video sharing website, Youtube. One instance of the network can be found at http://socialcomputing.asu.edu/datasets/YouTube2.
- **Twitter** [72]. This is a network between users on a social news website Twitter. One instance of the network can be downloaded at http://socialcomputing.asu.edu/datasets/Twitter.

7.1.2 Citation Networks

- **DBLP** [73]. This network represents the citation relationships between authors and papers. One instance of the data set can be found at http://arnetminer.org/citation.
- **Cora** [74]. This network represents the citation relationships between scientific publications. Besides the link information, each publication is also associated with a word vector indicating the absence/presence of the corresponding words from the dictionary. One instance of the data set can be found at https://linqs.soe.ucsc.edu/node/236.
- **Citeseer** [74]. This network, similar to Cora, also consists of scientific publications and their citation relationships. One instance of the data set can be downloaded at https://linqs.soe.ucsc.edu/node/236.
- **ArXiv** [75], [76]. This is the collaboration network constructed from the ArXiv website. One instance of the data set can be found at http://snap.stanford.edu/data/ca-AstroPh.html.

7.1.3 Language Networks

- **Wikipedia** [77]. This is a word co-occurrence network from the English Wikipedia pages. One instance of the data set can be found at http://www.mattmahoney.net/dc/textdata.

7.1.4 Biological Networks

- **PPI** [78]. This is a subgraph of the biological network that represents the pairwise physical interactions between proteins in yeast. One instance of the data set can be downloaded at http://konect.uni-koblenz.de/networks/maayan-vidal.

7.2 Node Classification

Given some nodes with known labels in a network, the node classification problem is to classify the rest nodes into different classes. Node classification is one of most primary applications for network embedding [3], [10]. Essentially, node classification based on network embedding for can be divided into three steps.

TABLE 3: A summary of real world networks

networks	structure and property preserving network embedding	network embedding with side information	classification	link prediction	clustering	visualization
BLOGCATALOG	√		√	√	√	√
FLICKR	√		√	√	√	√
YOUTUBE	√		√	√	√	√
Twitter	√			√		
DBLP	√	√	√	√	√	√
Cora	√	√	√	√	√	√
Citeseer	√	√	√	√	√	√
ArXiv	√			√		
Wikipedia	√		√	√	√	√
PPI	√			√		

First, a network embedding algorithm is applied to embed the network into a low dimensional space. Then, the nodes with known labels are used as the training set. Last, a classifier, such as Liblinear [79], is learned from the training set. Using the trained classifier, we can infer the labels of the rest nodes. The popularly used evaluation metrics for multi-label classification problem include Micro-F1 and Macro-F1 [70].

The multi-label classification application has been successfully tested on four categories of data sets, namely social networks (BLOGCATALOG [70], FLICKR [70], and YOUTUBE [71]), citation networks (DBLP [73], Cora [74], and Citeseer [74]), language networks (Wikipedia [77]), and biological networks (PPI [78]).

Specifically, a social network usually is a communication network among users on online platforms. DeepWalk [3], GraRep [34], SDNE [6], node2vec [25], and LANE [54] conduct classification on BLOGCATALOG to evaluate the performance. Also, the classification performance on FLICKR has been assessed in [3], [6], [10], [54]. Some studies [3], [6], [10] apply their algorithms to the Youtube network, which also achieves promising classification results. A citation network usually represents the citation relationships between authors or between papers. For example, [10], [53] use the DBLP network to test the classification performance. Cora is used in [14], [15]. Citeseer is used in [14], [15], [53]. The classification performance on language networks, such as Wikipedia, is also widely studied [10], [14], [15], [25]. The Protein-Protein Interactions (PPI) is used in [25]. Based on NUS-WIDE [80], a heterogeneous network extracted from Flickr, Chang *et al.* [16] validated the superior classification performance of network embedding on heterogeneous networks.

To summarize, network embedding algorithms have been widely used on various networks and have been well demonstrated their effectiveness on node classification.

7.3 Link Prediction

Link prediction, as one of the most fundamental problems on network analysis, has received a considerable amount of attention [7], [82]. It aims to estimate the likelihood of the existence of an edge between two nodes based on observed network structure [83]. Since network embedding algorithms are able to learn the vector based features for each node, the similarity between nodes can be easily estimated, for example, by the inner product or the cosine similarity. A larger similarity implies that the two nodes may have a higher propensity to be linked. Generally, precision@k and Mean Average Precision (MAP) are used to evaluate the link prediction performance [6].

The popularly used real networks for the link prediction task can be divided into three categories: citation networks (ARXIV [75], [76] and DBLP[1]), social networks (SN-TWeibo[2], SN-Twitter [72], Facebook [76], Epinions[3], and Slashdot[4]), and biological networks (PPI [78]). Specifically, [6] and [25] test the effectiveness on ARXIV[5]. HOPE [8] applies network embedding to link prediction on two directed networks SN-Twitter, which is a subnetwork of Twitter[6], and SN-TWeibo, which is a subnetwork of the social network in Tencent Weibo[7]. Node2vec [25] tests the performance of link prediction on a social network Facebook and a biological network PPI. EOE [61] uses DBLP to demonstrate the effectiveness on citation networks. Based on two social networks, Epinions and Slashdot, SiNE [13] shows the superior performance of signed network embedding on link prediction.

To sum up, network embedding is able to capture inherent network structures, and thus naturally it is suitable for link prediction applications. Extensive experiments on various networks have demonstrated that network embedding can tackle link prediction effectively.

7.4 Node Clustering

Node clustering is to divide the nodes in a network into clusters such that the nodes within the same cluster are more similar to each other than the nodes in different clusters. Network embedding algorithms learn representations of nodes in low dimensional vector spaces, so many typical clustering methods, such as Kmeans [84], can be directly adopted to cluster nodes based on their learned representations.

Many evaluation criteria have been proposed for clustering evaluation. Accuracy (AC) and normalized mutual information (NMI) [85] are frequently used to assess the clustering performance on graphs and networks.

The node clustering performance is tested on three types of networks: social networks (e.g., Facebook [87] and YELP [59]), citation networks (e.g., DBLP [60]), and document networks (e.g., 20-NewsGroup [88]). In particular, [16] extracts a social network from a social blogging site. It uses the TF-IDF features extracted from the blogs as the features of blog users and the "following" behaviors to construct the linkages. It successfully applies network embedding to the node clustering task. [4] uses the Facebook

1. http://dblp.uni-trier.de/
2. http://www.kddcup2012.org/c/kddcup2012-track1/data
3. http://www.epinions.com/
4. http://slashdot.org/
5. https://arxiv.org/
6. https://twitter.com/
7. http://t.qq.com/

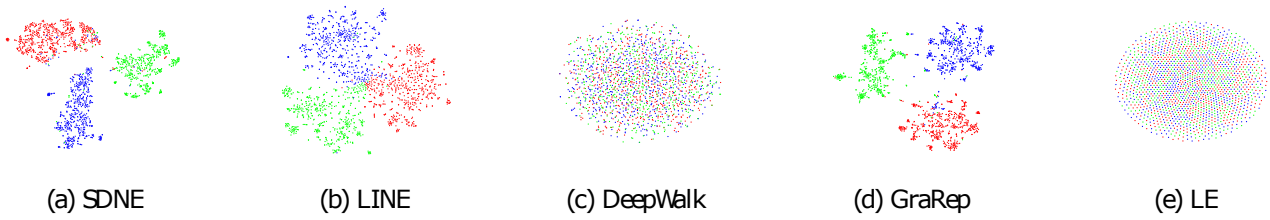


Fig. 15: Network visualization of 20-NewsGroup by different network embedding algorithms, i.e., SDNE [6], LINE [10], DeepWalk [3], GraRep [34], LE [81]. Image extracted from SDNE [6].

social network to demonstrate the effectiveness of community preserving network embedding on node clustering. [59] is applied to more social networks including MOVIE, a network extracted from YAGO [89] that contains knowledge about movies, YELP, a network extracted from YELP that is about reviews given to restaurants, and GAME, extracted from Freebase [90] that is related to video games. [26] tests the node clustering performance on a document network, 20-NewsGroup network, which consists of documents. The node clustering performance on citation networks is tested [59] by clustering authors in DBLP. The results show the superior clustering performance on citation networks.

In summary, node clustering based on network embedding is tested on different types of networks. Network embedding has become an effective method to solve the node clustering problem.

7.5 Network Visualization

Another important application of network embedding is network visualization, that is, generating meaningful visualization that layouts a network on a two dimensional space. By applying the visualization tool, such as t-SNE [91], to the learned low dimensional representations of nodes, it is easy for users to see a big picture of a sophisticated network so that the community structure or node centrality can be easily revealed.

More often than not, the quality of network visualization by different network embedding algorithms is evaluated visually. Fig. 15 is an example by SDNE [6] where SDNE is applied to 20-NewsGroup. In Fig. 15, each document is mapped into a two dimensional space as a point, and different colors on the points represent the labels. As can be seen, network embedding preserves the intrinsic structure of the network, where similar nodes are closer to each other than dissimilar nodes in the low-dimensional space. Also, LINE [10], GraRep [34], and EOE [61] are applied to a citation network DBLP and generate meaningful layout of the network. Pan *et al.* [53] show the visualization of another citation network Citeseer-M10 [92] consisting of scientific publications from ten distinct research areas.

7.6 Open Source Software

In Table 4, we provide a collection of links where one can find the source code of various network embedding methods.

8 Conclusions and Future Research Directions

The above survey of the state-of-the-art network embedding algorithms clearly shows that it is still a young and promising research field. To apply network embedding to tackle practical applications, a frontmost question is to select the appropriate methods. In

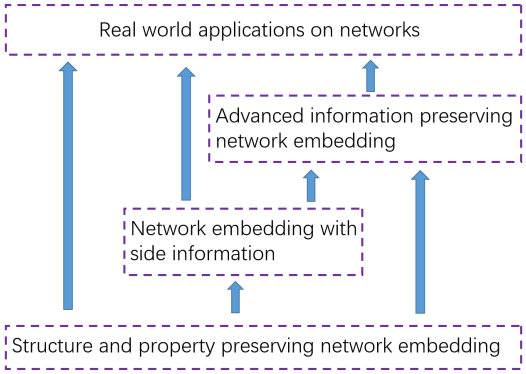


Fig. 16: Relationship among different types of network embedding methods.

Fig. 16 we show the relationship among different types of network embedding methods discussed in this survey.

The structure and property preserving network embedding is the foundation. If one cannot preserve well the network structure and retain the important network properties, in the embedding space serious information is loss, which hurts the analytic tasks in sequel. Based on the structure and property preserving network embedding, one may apply the off-the-shelf machine learning methods. If some side information is available, it can be incorporated into network embedding. Furthermore, the domain knowledge of some certain applications as advanced information can be considered.

In the rest of this section, we discuss several interesting directions for future work.

8.1 More Structures and Properties

Although various methods are proposed to preserve structures and properties, such as first order and high order proximities, communities, asymmetric transitivity, and structural balance, due to the complexity of real world networks, there are still some particular structures that are not fully considered in the existing network embedding methods. For example, how to incorporate network motifs [93], one of the most common higher-order structures in a network, into network embedding remains an open problem. Also, more complex local structures of a node can be considered to provide higher level constraints. The current assumption of network embedding is usually based on the pairwise structure, that is, if two nodes have a link, then their representations are similar. This assumption can work well for some applications, such as link prediction, but it cannot encode the centrality information of nodes, because the centrality of a node is usually related to a more complex structure. As another example, in several real

TABLE 4: A summary of the source code

Structure and property preserving network embedding	
Methods	Source code
DeepWalk [3]	https://github.com/phanein/deepwalk
LINE [10]	https://github.com/tangjianpku/LINE
GraRep [34]	https://github.com/ShelsonCao/GraRep
SDNE [6]	http://nrl.thumedia.org/structural-deep-network-embedding
Node2vec [25]	https://github.com/aditya-grover/node2vec
DNGR [26]	https://github.com/ShelsonCao/DNGR
M-NMF [4]	http://nrl.thumedia.org/community-preserving-network-embedding
GED [40]	https://users.ece.cmu.edu/~sihengc/publications.html
Ou *et al.* [44]	http://nrl.thumedia.org/non-transitive-hashing-with-latent-similarity-components
HOPE [8]	http://nrl.thumedia.org/asymmetric-transitivity-preserving-graph-embedding
Network embedding with side information	
Methods	Source code
MMDW [14]	https://github.com/thunlp/mmdw
TADW [15]	https://github.com/thunlp/tadw
TriDNR [53]	https://github.com/shiruipan/TriDNR
Advanced information preserving network embedding	
Methods	Source code
Information diffusion [63]	https://github.com/ludc/social_network_diffusion_embeddings
Cascade prediction [18]	https://github.com/chengli-um/DeepCas
Anomaly detection [21]	https://github.com/hurenjun/EmbeddingAnomalyDetection
Collaboration prediction [23]	https://github.com/chentingpc/GuidedHeteEmbedding

world applications, an edge may involve more than two nodes, known as a hyperedge. Such a hypernetwork naturally indicates richer relationships among nodes and has its own characteristics. Hypernetwork embedding is important for some real applications.

The power law distribution property indicates that most nodes in a network are associated with a small number of edges. Consequently, it is hard to learn an effective representation for a node with limited information. How this property affects the performance of network embedding and how to improve the embeddings of the minority nodes are still largely untouched.

8.2 The Effect of Side Information

Section 5 discusses a series of network embedding algorithms that preserve side information in embedding. All the existing methods assume that there is an agreement between network structure and side information. To what extent the assumption holds in real applications, however, remains an open question. The low correlation of side information and structures may degrade the performance of network embedding. Moreover, it is interesting to explore the complementarity between network structures and side information. More often than not, each information may contain some knowledge that other information does not have.

Besides, in a heterogeneous information network, to measure the relevance of two objects, the meta path, a sequence of object types with edge types in between, has been widely used. However, meta structure [89], which is essentially a directed acyclic graph of object and edge types, provides a higher-order structure constraint. This suggests a huge potential direction for improving heterogeneous information network embedding.

8.3 More Advanced Information and Tasks

In general, most of network embedding algorithms are designed for general purposes, such as link prediction and node classification. These network embedding methods mainly focus on general network structures and may not be specific to some target applications. Another important research direction is to explore the possibility of designing network embedding for more specific applications. For example, whether network embedding is a new way to detect rumors in social network [94], [95]? Can we use network embedding to infer social ties [96]? Each real world application has its own characteristics, and incorporating their unique domain knowledge into network embedding is a key. The technical challenges here are how to model the specific domain knowledge as advanced information that can be integrated into network embedding in an effective manner.

8.4 Dynamic Network Embedding

Although many network embedding methods are proposed, they are mainly designed for static networks. However, in real world applications, it is well recognized that many networks are evolving over time. For example, in the Facebook network, friendships between users always dynamically change over time, e.g., new edges are continuously added to the social network while some edges may be deleted. To learn the representations of nodes in a dynamic network, the existing network embedding methods have to be run repeatedly for each time stamp, which is very time consuming and may not meet the realtime processing demand. Most of the existing network embedding methods cannot be directly applied to large scale evolving networks. New network embedding algorithms, which are able to tackle the dynamic nature of evolving networks, are highly desirable.

8.5 More embedding spaces

The existing network embedding methods embed a network into the Euclidean space. In general, the principle of network embedding can be extended to other target spaces. For example, recently some studies [97] assume that the underlying structure of a network is in the hyperbolic space. Under this assumption, heterogeneous degree distributions and strong clustering emerge naturally, as they are the simple reflections of the negative curvature and metric property of the underlying hyperbolic geometry. Exploring other embedding space is another interesting research direction.

REFERENCES

[1] C. Staudt, A. Sazonovs, and H. Meyerhenke, "Networkit: A tool suite for large-scale network analysis," *Network Science To appear*.

[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[3] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[4] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," 2017.

[5] I. Herman, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on visualization and computer graphics*, vol. 6, no. 1, pp. 24–43, 2000.

[6] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[7] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *journal of the Association for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[8] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 672–681.

[9] Y. Fu and Y. Ma, *Graph embedding for pattern analysis*. Springer Science & Business Media, 2012.

[10] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 1067–1077.

[11] H. Huang, J. Tang, S. Wu, L. Liu *et al.*, "Mining triadic closure patterns in social networks," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 499–504.

[12] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[13] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed network embedding in social media," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 327–335.

[14] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-margin deepwalk: discriminative learning of network representation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016, pp. 3889–3895.

[15] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*, 2015, pp. 2111–2117.

[16] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 119–128.

[17] N. Natarajan and I. S. Dhillon, "Inductive matrix completion for predicting gene–disease associations," *Bioinformatics*, vol. 30, no. 12, pp. i60–i68, 2014.

[18] C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: an end-to-end predictor of information cascades," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 577–586.

[19] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[20] X. Yang, Y.-N. Chen, D. Hakkani-Tür, P. Crook, X. Li, J. Gao, and L. Deng, "End-to-end joint learning of natural language understanding and dialogue manager," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5690–5694.

[21] R. Hu, C. C. Aggarwal, S. Ma, and J. Huai, "An embedding approach to anomaly detection," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 385–396.

[22] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach." IJCAI, 2016.

[23] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 295–304.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[26] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 1145–1152.

[27] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 830–837.

[28] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[30] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[31] N. Berline, E. Getzler, and M. Vergne, *Heat kernels and Dirac operators*. Springer Science & Business Media, 2003.

[32] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, 2004, pp. 153–160.

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[34] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 891–900.

[35] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[36] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[37] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[38] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, 2014, pp. 2177–2185.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[40] S. Chen, S. Niu, L. Akoglu, J. Kovačević, and C. Faloutsos, "Fast, warped graph embedding: Unifying framework and one-click algorithm," *arXiv preprint arXiv:1702.05764*, 2017.

[41] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1365–1374.

[42] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, 2016, pp. 2014–2023.

[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[44] M. Ou, P. Cui, F. Wang, J. Wang, and W. Zhu, "Non-transitive hashing with latent similarity components," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 895–904.

[45] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[46] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[47] C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398–405, 1981.

[48] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk, "Sitting closer to friends than enemies, revisited," *Theory of computing systems*, vol. 56, no. 2, pp. 394–405, 2015.

[49] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[50] T. M. Le and H. W. Lauw, "Probabilistic latent document network embedding," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 270–279.

[51] J. Chang and D. M. Blei, "Relational topic models for document networks," in *International conference on artificial intelligence and statistics*, 2009, pp. 81–88.

[52] X. Sun, J. Guo, X. Ding, and T. Liu, "A general framework for content-enhanced network representation learning," *arXiv preprint arXiv:1610.02906*, 2016.

[53] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Network*, vol. 11, no. 9, p. 12, 2016.

[54] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proceedings of 10th ACM International Conference on Web Search and Data Mining (WSDM)*, 2017.

[55] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[56] X. Huang, J. Li, and X. Hu, "Accelerated attributed network embedding," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 633–641.

[57] X. Wei, L. Xu, B. Cao, and P. S. Yu, "Cross view link prediction by learning noise-resilient representation consensus," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1611–1619.

[58] Y. Jacob, L. Denoyer, and P. Gallinari, "Learning latent representations of nodes for classifying in heterogeneous social networks," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 373–382.

[59] Z. Huang and N. Mamoulis, "Heterogeneous information network embedding for meta path based proximity," *arXiv preprint arXiv:1701.05291*, 2017.

[60] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[61] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 741–749.

[62] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM Sigmod Record*, vol. 42, no. 2, pp. 17–28, 2013.

[63] S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari, "Learning social network embeddings for predicting information diffusion," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 393–402.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[66] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[67] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[68] R. S. Burt, "Structural holes and good ideas," *American journal of sociology*, vol. 110, no. 2, pp. 349–399, 2004.

[69] M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang, "Algorithms for large, sparse network alignment problems," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE, 2009, pp. 705–710.

[70] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 817–826.

[71] L. tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1107–1116.

[72] M. De Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, A. Kelliher *et al.*, "How does the data sampling strategy impact the discovery of information diffusion in social media?" *ICWSM*, vol. 10, pp. 34–41, 2010.

[73] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 990–998.

[74] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[75] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[76] J. Leskovec and A. Krevl, "Snap datasets: Stanford large network dataset collection (2014)," *URL http://snap. stanford. edu/data*, 2016.

[77] M. Mahoney, "Large text compression benchmark," 2011.

[78] B.-J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood *et al.*, "The biogrid interaction database: 2008 update," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D637–D640, 2007.

[79] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[80] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *Proceedings of the ACM international conference on image and video retrieval*. ACM, 2009, p. 48.

[81] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[82] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[83] L. Getoor and C. P. Diehl, "Link mining: a survey," *Acm Sigkdd Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[84] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[85] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[86] L. Lovász and M. D. Plummer, *Matching theory*. American Mathematical Soc., 2009, vol. 367.

[87] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of facebook networks," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 16, pp. 4165–4180, 2012.

[88] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering." in *AAAI*, 2014, pp. 1293–1299.

[89] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, "Meta structure: Computing relevance in large heterogeneous information networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1595–1604.

[90] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 2008, pp. 1247–1250.

[91] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[92] K. W. Lim and W. Buntine, "Bibliographic analysis with the citation network topic model," *arXiv preprint arXiv:1609.06826*, 2016.

[93] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[94] E. Seo, P. Mohapatra, and T. Abdelzaher, "Identifying rumors and their sources in social networks," *SPIE defense, security, and sensing*, pp. 83 891I–83 891I, 2012.

[95] Q. Zhang, S. Zhang, J. Dong, J. Xiong, and X. Cheng, "Automatic detection of rumor on social network," in *Natural Language Processing and Chinese Computing*. Springer, 2015, pp. 113–122.

[96] J. Tang, T. Lou, and J. Kleinberg, "Inferring social ties across heterogenous networks," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 743–752.

[97] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, p. 036106, 2010.

PLACE PHOTO HERE

Peng Cui received his Ph.D. degree in computer science in 2010 from Tsinghua University and he is an Associate Professor at Tsinghua. He has vast research interests in data mining, multimedia processing, and social network analysis. Until now, he has published more than 20 papers in conferences such as SIGIR, AAAI, ICDM, etc. and journals such as IEEE TMM, IEEE TIP, DMKD, etc. Now his research is sponsored by National Science Foundation of China, Samsung, Tencent, etc. He also serves as Guest Editor, Co-Chair, PC member, and Reviewer of several high-level international conferences, workshops, and journals.

PLACE PHOTO HERE

Xiao Wang received his Ph.D. degree from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2016. He is a postdoctoral researcher in Department of Computer Science and Technology, Tsinghua University, Beijing, China. He got the China Scholarship Council Fellowship in 2014 and visited Washington University in St. Louis, USA, as a joint training student from Nov. 2014 to Nov. 2015. His current research interests include data mining, social network analysis, and machine learning. Until now, he has published more than 20 papers in conferences such as AAAI, IJCAI, etc. and journals such as IEEE Trans. on Cybernetics, etc. Now his research is sponsored by National Science Foundation of China.

PLACE PHOTO HERE

Jian Pei is a professor in the School of Computing Science, Simon Fraser University, Canada. His research interests can be summarized as developing effective and efficient data analysis techniques for novel data intensive applications. He is currently interested in various techniques of data mining, Web search, information retrieval, data warehousing, online analytical processing, and database systems, as well as their applications in social networks, health-informatics, business, and bioinformatics. His research has been supported in part by government funding agencies and industry partners. He has published prolifically and served regularly for the leading academic journals and conferences in his fields. He is an associate editor of the ACM Transactions on Knowledge Discovery from Data. He is a fellow of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). He is the recipient of several prestigious awards.

PLACE PHOTO HERE

Wenwu Zhu is currently a Professor and the Vice Chair of the Department of Computer Science, Tsinghua University, Beijing, China. Prior to his current position, he was a Senior Researcher and a Research Manager with Microsoft Research Asia, Beijing, China. He was the Chief Scientist and the Director with Intel Research China, Beijing, China, from 2004 to 2008. He was at Bell Labs, Murray Hill, NJ, USA, as a member of technical staff from 1996 to 1999. He received the Ph.D. degree from the New York University, New York, NY, USA, in 1996. His current research interests include the areas of multimedia computing, communications, and networking, as well as big data.

Wenwu is an IEEE Fellow, AAAS Fellow, an SPIE Fellow, and an ACM Distinguished Scientist. He has been serving as the Editor-in-Chief for the IEEE TRANSACTIONS ON MULTIMEDIA (T-MM) since January 1, 2017. He served on the Steering Committee for T-MM in 2016 and the IEEE TRANSACTIONS ON MOBILE COMPUTING (T-MC) from 2007 to 2010. He has served on various Editorial Boards, such as the Guest Editor for the PROCEEDINGS OF THE IEEE, the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (T-CSVT), as well as an Associate Editor for T-MM, the ACM Transactions on Multimedia, Communications, and Applications, T-CSVT, T-MC, and the IEEE TRANSACTIONS ON BIG DATA. He was the recipient of 5 Best Paper Awards including T-CSVT in 2001 and ACM Multimedia 2012.