Chapter 2
Graph Representation Learning

Peng Cui, Lingfei Wu, Jian Pei, Liang Zhao and Xiao Wang

Abstract Graph representation learning aims at assigning nodes in a graph to low-dimensional representations and effectively preserving the graph structure. Recently, a significant amount of progresses have been made toward this emerging graph analysis paradigm. In this chapter, we first summarize the motivation of graph representation learning. Afterwards and primarily, we provide a comprehensive overview of a large number of graph representation learning methods in a systematic manner, covering the traditional graph representation learning, modern graph representation learning, and graph neural networks.

2.1 Graph Representation Learning: An Introduction

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

Liang Zhao
Department of Computer Science, Emory University, e-mail: liang.zhao@emory.edu

Lingfei Wu
JD.COM Silicon Valley Research Center, e-mail: lwu@email.wm.edu

Peng Cui
Department of Computer Science, Tsinghua University, e-mail: cuip@tsinghua.edu.cn

Jian Pei
Department of Computer Science, Simon Fraser University, e-mail: jpei@cs.sfu.ca

Xiao Wang
Department of Computer Science, Beijing University of Posts and Telecommunications, e-mail: xiaowang@bupt.edu.cn

Traditionally, we usually represent a graph as $\mathscr{G} = (\mathscr{V}, \mathscr{E} >)$, where $\mathscr{V}$ is a node set and $\mathscr{E}$ is an edge set. For large graphs, such as those with billions of nodes, the traditional graph representation poses several challenges to graph processing and analysis.

(1) **High computational complexity.** These relationships encoded by the edge set E cause most of the graph processing or analysis algorithms either iterative or combinatorial computation steps. 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 graph representation, we have to enumerate many possible paths between two nodes, which is in nature a combinatorial problem. Such methods result in high computational complexity that prevents them from being applicable to large-scale real-world graphs.

(2) **Low parallelizability.** Parallel and distributed computing is de facto to process and analyze large-scale data. Graph 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 graph 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.

(3) **Inapplicability of machine learning methods.** Recently, machine learning methods, especially deep learning, are very powerful in many areas. For graph 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 graph 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 graph, the extremely high dimensionality of such a representation in a large graph with many nodes makes the in sequel graph processing and analysis difficult.

To tackle these challenges, substantial effort has been committed to develop novel graph representation learning, i.e., learning the dense and continuous low-dimensional vector representations for nodes, so that the noise or redundant information can be reduced and the intrinsic structure information can be preserved. In the learned representation 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 structural characteristics of a node are encoded into its representation vector.

Basically, in order to make the representation space well support graph analysis tasks, there are two goals for graph representation learning. First, the original graph can be reconstructed from the learned representation space. It requires that, if there is an edge or relationship between two nodes, then the distance of these two nodes in the representation space should be relatively small. Second, the learned representation space can effectively support graph inference, such as predicting unseen links, identifying important nodes, and inferring node labels. It should be noted that a representation space with only the goal of graph reconstruction is not sufficient for graph inference. After the representation is obtained, downstream tasks such as node classification , node clustering , graph visualization and link prediction can be dealt with based on these representations. Overall, there are three main categories of graph representation learning methods: traditional graph embedding, modern graph embedding, and graph neural networks, which will be introduced separately in the following three sections.

2.2 Traditional Graph Embedding

Traditional 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. As mentioned before, graph embedding usually has two goals, i.e. reconstructing original graph structures and support graph inference. The objective functions of traditional graph embedding methods mainly target the goal of graph reconstruction.

Specifically, Tenenbaum et al (2000) first constructs a neighborhood graph G using connectivity algorithms such as K nearest neighbors (KNN). Then based on G, the shortest path between different data can be computed. Consequently, for all the N data entries in the data set, we have the matrix of graph distances. Finally, the classical multidimensional scaling (MDS) method is applied to the matrix to obtain the coordinate vectors. The representations learned by Isomap approximately preserve 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) (Roweis and Saul, 2000) 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. Finally, in the low-dimensional space, LLE constructs a neighborhood-preserving mapping based on locally linear reconstruction. Laplacian eigenmaps (LE) (Belkin and Niyogi, 2002) also begins with constructing a graph using ε-neighborhoods or K nearest neighbors. Then the heat kernel (Berline et al, 2003) is utilized to choose the weight of two nodes in the graph. Finally, the node representations can be obtained by based on the Laplacian matrix regularization. Furthermore, the locality preserving projection (LPP) (Berline et al, 2003), a linear approximation of the nonlinear LE, is proposed.

These methods are extended in the rich literature of graph embedding by considering different characteristics of the constructed graphs (Fu and Ma, 2012). We can find that traditional 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. While, in contrast, modern graph embedding, which will be introduced in the following, 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. For example, an edge between two nodes usually just implies there is a relationship between them, but cannot indicate the specific proximity. Also, even if there is no edge between two nodes, we cannot say the proximity between these two nodes is zero. The definition of node proximities depends on specific analytic tasks and application scenarios. Therefore, modern graph embedding usually incorporates rich information, such as network structures, properties, side information and advanced information, to facilitate different problems and applications. Modern graph embedding needs to target both of the two goals mentioned before. In view of this, traditional graph embedding can be regarded as a special case of modern graph embedding, and the recent research progress on modern graph embedding pays more attention to network inference.

2.3 Modern Graph Embedding

To well support network inference, modern graph embedding considers much richer information in a graph. According to the types of information that are preserved in graph representation learning, the existing methods can be categorized into three categories: (1) graph structure and properties preserving graph embedding, (2) graph representation learning with side information and (3) advanced information preserving graph representation learning. In technique view, different models are 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.

2.3.1 Structure-Property Preserving Graph Representation Learning

Among all the information encoded in a graph, graph structures and properties are two crucial factors that largely affect graph inference. Thus, one basic requirement of graph representation learning is to appropriately preserve graph structures and capture properties of graphs. Often, graph structures include first-order structure and higher-order structure, such as second-order structure and community structure. Graphs with different types have different properties. For example, directed graphs have the asymmetric transitivity property. The structural balance theory is widely applicable to signed graphs.

2.3.1.1 Structure Preserving Graph Representation Learning

Graph structures can be categorized into different groups that present at different granularities. The commonly exploited graph structures in graph representation learning include neighborhood structure, high-order node proximity and graph communities.

How to define the neighborhood structure in a graph is the first challenge. Based on the discovery that the distribution of nodes appearing in short random walks is similar to the distribution of words in natural language, DeepWalk (Perozzi et al, 2014) employs the random walk to capture the neighborhood structure. Then for each walk sequence generated by random walk, following Skip-Gram, DeepWalk aims to maximize the probability of the neighbors of a node in a walk sequence. Node2vec defines a flexible notion of a node's graph 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). Besides the neighborhood structure, LINE (Tang et al, 2015b) is proposed for large scale network embedding, which can preserve the first and second order proximities. The first order proximity is the observed pairwise proximity between two nodes. The second order proximity is determined by the similarity of the "contexts" (neighbors) of two nodes. Both are important in measuring the relationships between two nodes. Essentially, LINE is based on the shallow model, consequently, the representation ability is limited. SDNE (Wang et al, 2016) proposes a deep model for network embedding, which also aims at capturing the first and second order proximites. SDNE uses the deep auto-encoder architecture with multiple non-linear layers to preserve the second order proximity. To preserve the first-order proximity, the idea of Laplacian eigenmaps (Belkin and Niyogi, 2002) is adopted. Wang et al (2017g) propose a modularized nonnegative matrix factorization (M-NMF) model for graph representation learning, which aims to preserve both the microscopic structure, i.e., the first-order and second-order proximities of nodes, and the mesoscopic community structure (Girvan and Newman, 2002). They adopt the NMF model (Févotte and Idier, 2011) to preserve the microscopic structure. Meanwhile, the community structure is detected by modularity maximization (Newman, 2006a). Then, 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.

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 nonlinear models are attempted, demonstrating the large potential of deep models in network embedding.

2.3.1.2 Property Preserving Graph Representation Learning

Currently, most of the existing property preserving graph representation learning methods focus on graph transitivity in all types of graphs and the structural balance property in signed graphs.

We usually demonstrate that the transitivity usually exists in a graph. But meanwhile, we can find that preserving such property is not challenging, because in a metric space, the distance between different data naturally satisfies the triangle inequality. However, this is not always true in the real world. Ou et al (2015) aim to preserve the non-transitivity property via latent similarity components. The non-transitivity property declares that, for nodes v_1, v_2 and v_3 in a graph where $(v_1;v_2)$ and $(v_2;v_3)$ are similar pairs, $(v_1;v_3)$ 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. The main idea is that they learn multiple node embeddings, and then compare different nodes based on multiple similarities, rather than one similarity. They observe that if two nodes have a large semantic similarity, at least one of the similarities is large, otherwise, all of the similarities are small. In a directed graph, it usually has the asymmetric transitivity property. 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 (Ou et al, 2016) summarizes four measurements in a general formulation, and then utilizes a generalized SVD problem to factorize the high-order proximity (Paige and Saunders, 1981), such that the time complexity of HOPE is largely reduced, which means HOPE is scalable for large scale networks. In a signed graph with both of positive and negative edges, the social theories, such as structural balance theory (Cartwright and Harary, 1956; Cygan et al, 2012), which are very different from the unsigned graph. The structural balance theory demonstrates that users in a signed social network should be able to have their “friends” closer than their “foes”. To model the structural balance phenomenon, SiNE (Wang et al, 2017f) utilizes a deep learning model consisting of two deep graphs with non-linear functions.

The importance of maintaining network properties in network embedding space, especially the properties that largely affect the evolution and formation of networks, has been well recognized. The key challenge is how to address the disparity and heterogeneity of the original network space and the embedding vector space at property level. 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 highorder 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.

2.3.2 Graph Representation Learning with Side Information

Besides graph structures, side information is another important information source for graph representation learning. Side information in the context of graph representation learning can be divided into two categories: node content and types of nodes and edges. Their difference is the way of integrating network structures and side information.

Graph Representation Learning with Node Content. In some types of graphs, 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 graph representation learning arouses considerable research interests. Tu et al (2016) propose a semi-supervised graph 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) (Hearst et al, 1998) and incorporates the label information to find an optimal classifying boundary. Yang et al (2015b) propose TADW that takes the rich information (e.g., text) associated with nodes into account when they learn the low dimensional representations of nodes. Pan et al (2016) propose a coupled deep model that incorporates graph structure, node attributes and node labels into graph embedding. 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.

Heterogeneous Graph Representation Learning. Different from graphs with node content, heterogeneous graphs consist of different types of nodes and links. How to unify the heterogeneous types of nodes and links in graph embedding is also an interesting and challenging problem. Jacob et al (2014) propose a heterogeneous social graph representation learning 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. Chang et al (2015) propose a deep graph representation learning algorithm for heterogeneous graphs, whose nodes have various types(e.g., images and texts). The nonlinear embeddings of images and texts are learned by a CNN model and the fully connected layers, respectively. Huang and Mamoulis (2017) propose a meta path similarity preserving heterogeneous information graph representation learning algorithm. To model a particular relationship, a meta path (Sun et al, 2011) is a sequence of object types with edge types in between.

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.

2.3.3 Advanced Information Preserving Graph Representation Learning

Different from side information, the advanced information refers to the supervised or pseudo supervised information in a specific task. The 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 combination of advanced information and network embedding techniques enables representation learning for networks.

Information Diffusion. Information diffusion (Guille et al, 2013) is an ubiquitous phenomenon on the web, especially in social networks. Bourigault et al (2014) propose a graph representation learning algorithm for predicting information diffusion in social 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. 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. The kernel describes 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. The cascade prediction problem here is defined as predicting the increment of cascade size after a given time interval (Li et al, 2017a). Li et al (2017a) argue that the previous work on cascade prediction all depends on the bag of hand-crafting features to represent the cascade and graph structures. Instead, they present an end-to-end deep learning model to solve this problem using the idea of graph embedding. The whole procedure is able to learn the representation of cascade graph in an end-to-end manner.

Anomaly Detection. Anomaly detection has been widely investigated in previous work (Akoglu et al, 2015). Anomaly detection in graphs aims to infer the structural inconsistencies, which means the anomalous nodes that connect to various diverse influential communities (Hu et al, 2016), (Burt, 2004). Hu et al (2016) propose a graph embedding based method for anomaly detection. They assume that the community memberships of two linked nodes should be similar. An anomaly node 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.

Graph Alignment. The goal of graph alignment is to establish the correspondence between the nodes from two graphs, i.e., to predict the anchor links across two graphs. The same users who are shared by different social networks naturally form the anchor links, and these links bridge the different graphs. The anchor link prediction problem is, given source graph and target graph and a set of observed anchor links, to identify the hidden anchor links across two graphs. Man et al (2016) propose a graph representation learning algorithm to solve this problem. The learned representations can preserve the graph structure and respect the observed anchor links.

Advanced information preserving graph embedding usually consists of two parts. One is to preserve the graph 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 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.

2.4 Graph Neural Networks

Over the past decade, deep learning has become the "crown jewel" of artificial intelligence and machine learning, showing superior performance in acoustics, images and natural language processing, etc. Although it is well known that graphs are ubiquitous in the real world, it is very challenging to utilize deep learning methods to analyze graph data. This problem is non-trivial because of the following challenges: (1) Irregular structures of graphs. Unlike images, audio, and text, which have a clear grid structure, graphs have irregular structures, making it hard to generalize some of the basic mathematical operations to graphs. For example, defining convolution and pooling operations, which are the fundamental operations in convolutional neural networks (CNNs), for graph data is not straightforward. (2) Heterogeneity and diversity of graphs. A graph itself can be complicated, containing diverse types and properties. These diverse types, properties, and tasks require different model architectures to tackle specific problems. (3) Large-scale graphs. In the big-data era, real graphs can easily have millions or billions of nodes and edges. How to design scalable models, preferably models that have a linear time complexity with respect to the graph size, is a key problem. (4) Incorporating interdisciplinary knowledge. Graphs are often connected to other disciplines, such as biology, chemistry, and social sciences. This interdisciplinary nature provides both opportunities and challenges: domain knowledge can be leveraged to solve specific problems but integrating domain knowledge can complicate model designs.

Currently, graph neural networks have attracted considerable research attention over the past few years. The adopted architectures and training strategies vary greatly, ranging from supervised to unsupervised and from convolutional to recursive, including graph recurrent neural networks (Graph RNNs), graph convolutional networks (GCNs), graph autoencoders (GAEs), graph reinforcement learning (Graph RL), and graph adversarial methods. Specifically, Graroperty h RNNs capture recursive and sequential patterns of graphs by modeling states at either the node-level or the graph-level; GCNs define convolution and readout operations on irregular graph structures to capture common local and global structural patterns; GAEs assume low-rank graph structures and adopt unsupervised methods for node representation learning; Graph RL defines graph-based actions and rewards to obtain feedbacks on graph tasks while following constraints; Graph adversarial methods adopt adversarial training techniques to enhance the generalization ability of graphbased models and test their robustness by adversarial attacks.

There are many ongoing or future research directions which are also worthy of further study, including new models for unstudied graph structures, compositionality of existing models, dynamic graphs, interpretability and robustness, etc. On the whole, deep learning on graphs is a promising and fast-developing research field that both offers exciting opportunities and presents many challenges. Studying deep learning on graphs constitutes a critical building block in modeling relational data, and it is an important step towards a future with better machine learning and artificial intelligence techniques.

2.5 Summary

In this chapter, we introduce the motivation of graph representation learning. Then in Section 2, we discuss the traditional graph embedding methods and the modern graph embedding methods are introduced in Section 3. Basically, the structure and property preserving graph representation learning is the foundation. If one cannot preserve well the graph structure and retain the important graph properties in the representation space, serious information will be lost, which hurts the analytic tasks in sequel. Based on the structure and property preserving graph representation learning, one may apply the off-the-shelf machine learning methods. If some side information is available, it can be incorporated into graph representation learning. Furthermore, the domain knowledge of some certain applications as advanced information can be considered. As shown in Section 4, utilizing deep learning methods on graphs is a promising and fast-developing research field that both offers exciting opportunities and presents many challenges. Studying deep learning on graphs constitutes a critical building block in modeling relational data, and it is an important step towards a future with better machine learning and artificial intelligence techniques.