ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

Research Status of Graph Embedding in Graph Neural Networks

Qingzhe Meng*, Jie Zhang

School of xijing University, Xi'an, China

*Corresponding author: zmq13326264217@163.com

Abstract

Graph neural networks (GNNS) is a powerful machine learning tool for processing various graph structure data. Graph embedding is one of the key problems in GNNS. The core of this technology is to map the nodes in the graph structure to the low dimensional space, and usually retain some key information of the nodes in the original graph. In order to capture the structure and attribute relationship between nodes, so as to provide high-quality feature representation for various downstream tasks. Many advanced graph embedding algorithms have been proposed and widely used in various fields. In addition, how to combine graph embedding with other machine learning tasks, such as graph classification, graph clustering and link prediction, is also an important research direction. This paper summarizes and compares the related representative technologies involved by combing the research and development context of graph embedding, effectively filling the gap in the application review of related technologies, and providing reference for researchers on related issues.

Keywords

Graph Embedding; GNNs; Graph Classification; Graph Clustering.

1. Introduction

The research progress of graph embedding method is of great significance for modern network analysis and processing. The goal of graph embedding is to map the nodes or edges in the graph to the low dimensional vector space, while maintaining the structure information and node characteristics in the graph.

With the rise of deep learning technology, graph neural network (GNN), especially graph convolution neural network (GCN), has become the mainstream tool for processing graph structure data. Through the innovative design of the model structure, as well as the in-depth mining of node feature information and network structure, researchers are gradually unlocking the potential of graph structure data and providing scientific basis for future network analysis and decision-making.

The overall organizational framework of the paper is shown in Fig 1, in which section 1 introduces the relevant basic knowledge of the diagram, including the commonly used symbol definitions in this paper; Section 2 combs graph embedding based on proximity method, mainly including DeepWalk and node2vec; Section 3 summarizes the graph embedding method, which is the main feature-based method, and basically includes various methods commonly used in modern times; In Section 4, graph embedding method based on graph neural network is sorted out; Section 5 summarizes the main challenges faced by graph embedding from the dynamic graph embedding level, and prospects the future research trends.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

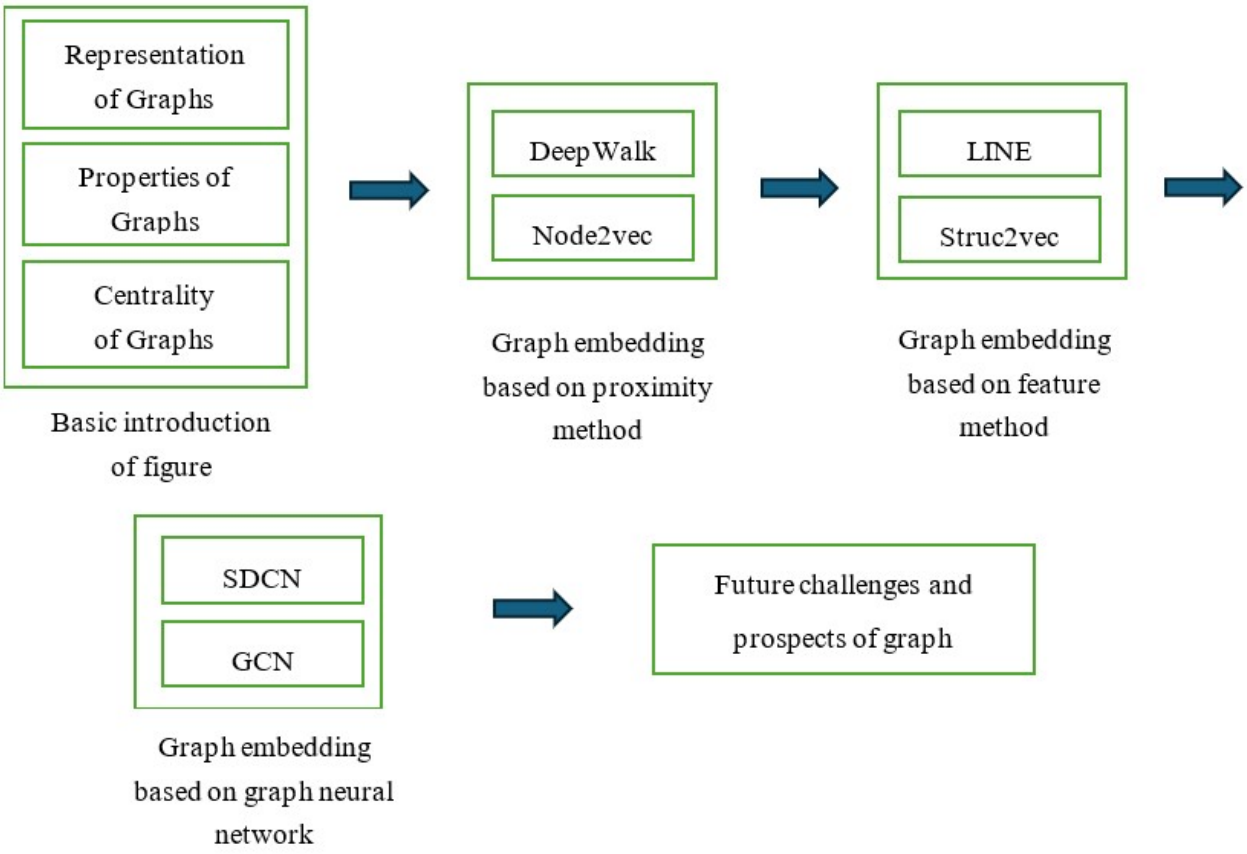
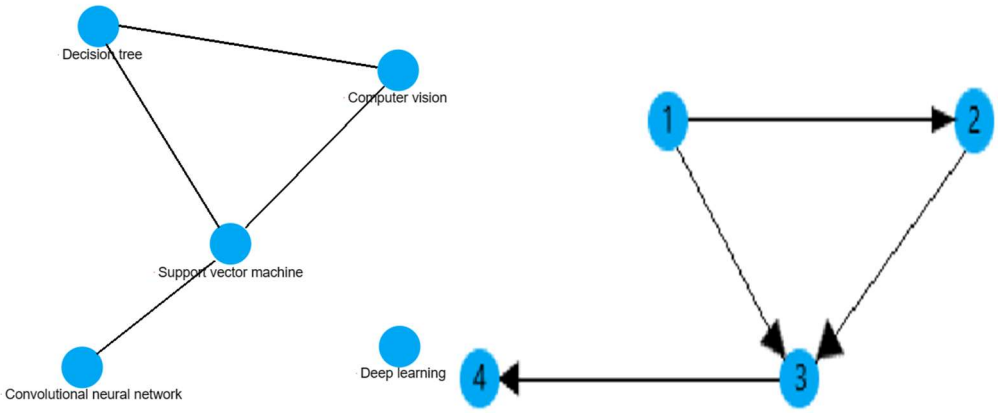


Fig. 1 overall paper organization framework

2. Basic Introduction of Figure

2.1 Representation of Graphs

Graph usually refers to a graph (such as Fig 2). Especially in computer science and mathematics, a graph is an abstract data structure composed of nodes (or vertices) and edges. Nodes represent objects in the graph, while edges represent relationships between nodes. A graph can be directed (edges have directions) or undirected (edges have no directions) or weighted (edges have weights). A graph can also be represented by adjacency matrix. If a graph is represented by adjacency matrix as shown in Fig 2 (a), and list the adjacency matrix[1].



(a) (b)

Fig. 2 (a)undirected graph (b)PageRank example digraph

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

2.2 Properties of Graphs

2.2.1 Degree

Degree is the number of edges connected to the vertex. For an undirected graph, the sum of the degrees of all vertices is twice the number of edges. This is because each edge in an undirected graph is counted as the degree of two connected vertices[2].

In a directed graph, degree is further divided into in degree and out degree. In degree is the number of edges pointing to the vertex; Out degree refers to the number of edges from this vertex to other vertices.

2.2.2 Connected Graph

For an undirected graph, if any node i can reach node j through an edge, it is called a connected graph. Fig 2(a) is a connected graph, and Fig 2(b) is a non connected graph.

In a directed graph, if any two vertices can reach each other, it is called a strongly connected graph. If the direction of a directed graph is ignored and the resulting undirected graph is connected, the original directed graph is called weakly connected graph[3].

2.3 Graph Centrality

2.3.1 Intermediary Centrality

Betweenness centrality is an index used to measure the importance of nodes in the network. It measures the degree to which a node acts as an intermediary or bridge in the network. Specifically, mediation centrality is defined based on the mediation role of the shortest path between nodes in the network, that is:

$$C_b(v) = \sum_{i \neq v \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \quad (1)$$

σ_{ij} refers to the total number of all shortest paths from node i to node j.$\sigma_{ij}(v)$ is the number of shortest paths passing through node V in the shortest path from node i to node j. However, node V cannot be the start or end of the shortest path, that is, $i \neq v \neq j$,because the requirement here is "through"[4]. The intermediary centrality of a node is the number of times that all the shortest paths between any other two nodes pass through the node.

2.3.2 Connection Centrality

Betweenness centrality is an index used to measure the importance of nodes in the network. It measures the degree to which nodes act as bridges in the network, that is, the number and way of nodes connecting other nodes in the network, that is:

$$C_B(v) = \frac{n-1}{N_{vj}} \quad (2)$$

For each node V, N_{vj} is the sum of the shortest paths from V to all nodes. Connection centrality can help us find the nodes that play a key role as a bridge in the network. These nodes have an important impact on the network connection and information dissemination[5].

2.4 Web Page Sorting Algorithm

2.4.1 PageRank

PageRank is used to evaluate the importance of web pages in search engine results. It is a link analysis algorithm, which determines the weight or importance of a web page by analyzing the link relationship between web pages.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

PageRank algorithm can be described mathematically as an iterative calculation process. In each iteration, each page will update its PageRank value according to its current PageRank value and the number and quality of links to other pages[6]. When the damping coefficient=0.85, if the PageRank value of pagerank3 does not reach 4 nodes, it reaches any node in the graph through the damping coefficient.

2.4.2 HITS

HITS(hyperlink induced topic search) is another link analysis algorithm, which aims to identify authoritative web pages and topic related web pages on a specific topic. Unlike PageRank, hits algorithm divides web pages into two categories: Authority page and hub page[7].

HITS algorithm calculates the scores of authority page and hub page iteratively until it converges to a stable state. Therefore, hits algorithm is usually used for topic specific search and information retrieval tasks. For example, in Fig 2 (b)$V1_{hub}=V2_{authority}$, $V2_{hub}=V3_{authority}$, $V3_{hub}=V4_{authority}$, $V3_{authority}=V1_{hub}+V1_{hub}$. After continuous iteration, each node reaches a steady state, and finally the network priority ranking is obtained.

3. Graph Embedding based on Proximity Method

Graph embedding is a technique that maps nodes or edges in a graph to a low dimensional vector space. When embedding a graph, the topological structure, similarity, connection relationship and other information between nodes or edges are usually considered.

3.1 DeepWalk

The traditional method Deepwalk uses random walk to generate a sequence of nodes, and learns the node representation through word embedding model. The DeepWalk algorithm of Bryan perozzi et al. DeepWalk algorithm mainly includes two steps: random walk and skip gram model.

3.1.1 Random Walk

In DeepWalk, random walk refers to a random walk of fixed length starting from each node and generating a series of node sequences. This process is similar to walking randomly on the graph and recording the access order of each node[8]. Each time you walk, the current node will jump to the adjacent node with a certain probability. After the stroll, we can get a node sequence $S=\{v_1,v_2,...,v_n\}$.

3.1.2 Skip Gram Model

The skip gram model is a model for learning the vector representation of words. It is also a neural network model widely used in the field of natural language processing.

In DeepWalk, we use the sequence of nodes generated by random walk as input, and use the skip gram model to learn the vector representation of nodes. Specifically, for a node sequence $S=\{v_1, v_2,..., v_n\}$,Specifically, we can use these vectors for clustering, visualization, recommendation and other applications[9].

3.1.3 Experimental Part

Generate the term sequence through Wikipedia terms, and use the term data for DeepWalk first, select Wikipedia terms as the nodes in the graph. such as the links between terms and other terms jointly linked to (as shown in Fig 3).

The dimensionality of high-dimensional vectors is reduced to two dimensions through t-SNE dimensionality reduction visualization, and the relative position relationship between each node is displayed through the scatter diagram as shown in Fig 4(a). On the basis of dimension reduction visualization of t-SNE, highlight the top 10 nodes of embedding and add annotations at their positions as shown in Fig 4(b).

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

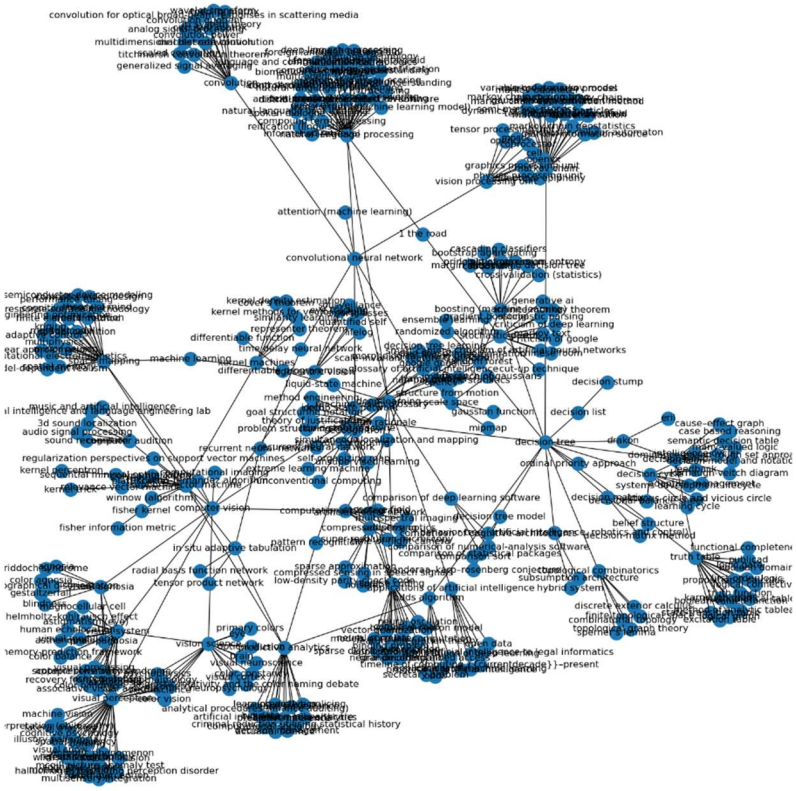
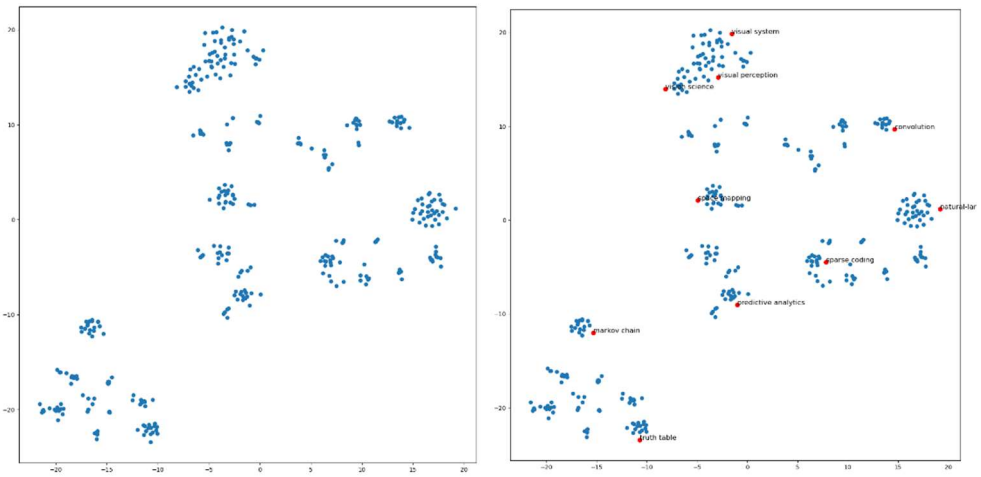


Fig. 3 Wikipedia entry generation entry sequence



(b)

Fig. 4 (a)dimensionality reduction of high-dimensional vector to two-dimensional (b)highlights the top 10 nodes of embedding

The experimental results show that the advantage of DeepWalk algorithm is that it can process large-scale graph data and infer the similarity between nodes from incomplete graphs. In addition, the implementation of DeepWalk algorithm is simple and easy to expand and optimize. However, DeepWalk also has some limitations. First, DeepWalk only considers local information, not global information. Secondly, DeepWalk cannot handle weighted graphs. Finally, for high-dimensional dense graphs, the effect of DeepWalk may be affected.

3.2 node2vec

The core idea of node2vec method proposed by Grover Aditya et al. Is to capture the structural information between nodes by simulating random walks on a graph. In addition, the author also

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

conducted parameter sensitivity analysis and compared it with other node embedding methods, further proving the performance advantages of node2vec method[10].

3.2.1 Experimental Part

Adjusting the two control parameters p and q, node2vec can balance the diversity and bias of exploration in the process of walking, so as to better capture the structural information between nodes. When p>q, DFS depth first search is used to mine homogeneous communities (as shown in Fig 5). When p<q, BFS width first searches, mining the structure and function of nodes (as shown in Fig 6).

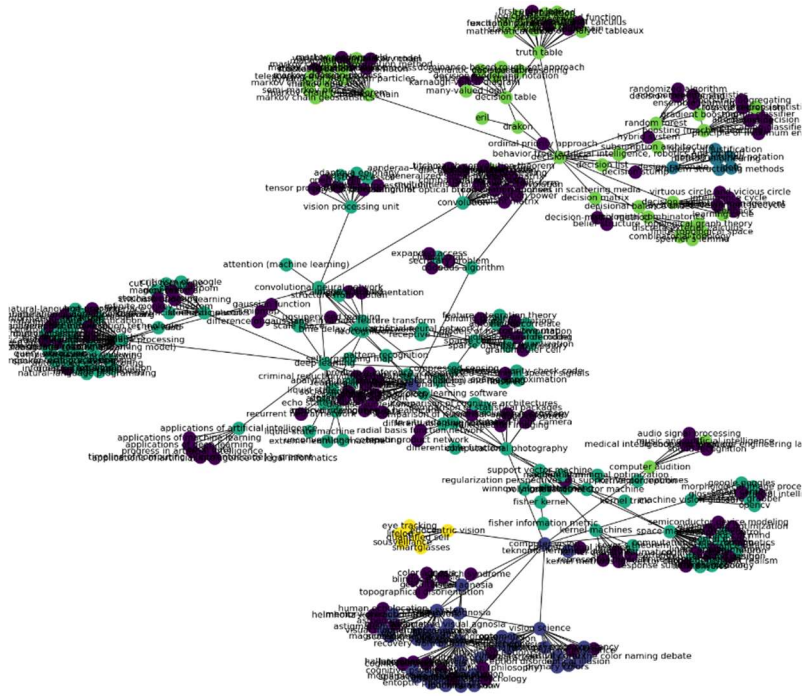


Fig. 5 DFS depth first search

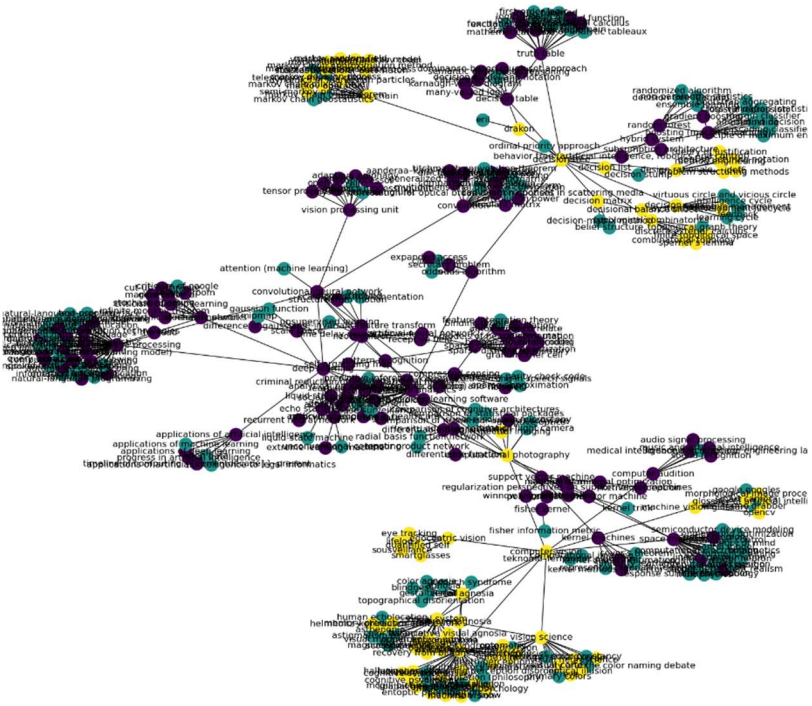
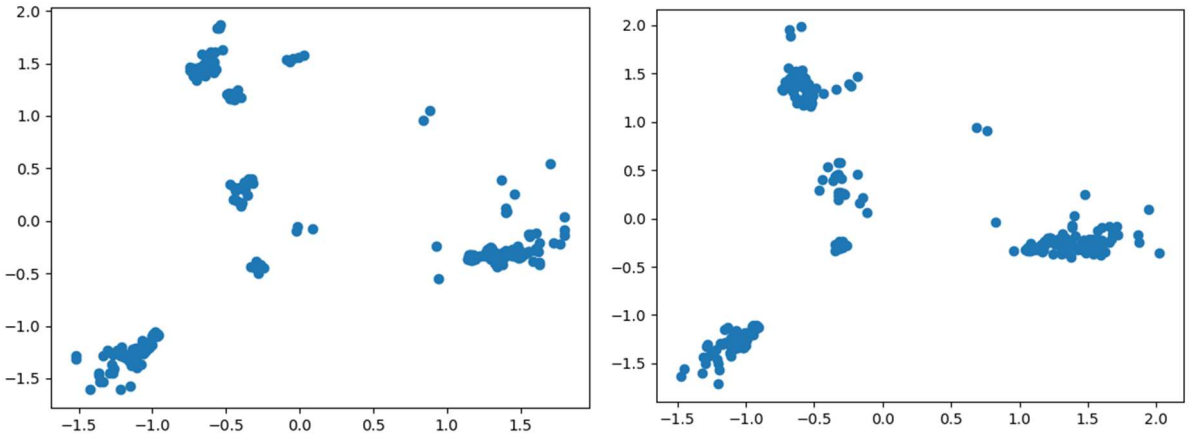


Fig. 6 BFS width first search

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

Reduce the dimension of this module, reduce the data set to two dimensions, and visualize the results. When searching for DFS depth first as shown in Fig 7(a), walkers will explore as deep as possible along a branch in the graph until they reach the leaf node, and then backtrack and explore other branches. DFS tends to explore the deep structure of the graph, so the generated node sequence may more reflect the local structure of the graph. Therefore, using DFS for random walk may lead to more capture of local relationships between nodes.



(a) (b)

Fig. 7 (a)dimensionality reduction of high-dimensional vector to two-dimensional during (b) dimension reduction of high-dimensional vector to two-dimensional during BFS

When DFS searches for BFS width first as shown in Fig 7(b), walkers will successively explore all neighbor nodes of the current node, and then explore the neighbor nodes of these neighbor nodes, and so on. BFS tends to explore the surface structure of the graph, so the generated node sequence may more reflect the global structure of the graph. Therefore, using BFS for random walk may lead to more capture of the global relationship between nodes. If the initial node representation is not good, the algorithm may not converge to the global optimal solution, which will affect the performance of the algorithm.

4. Graph Embedding based on Feature Method

With the wide application of graph data in various fields, learning the representation of nodes in graph becomes very important. LINE, struc2vec and other methods focus on preserving the local network structure of nodes to retain the first-order and second-order proximity information of the graph[11].

4.1 LINE

The LINE proposed by Jian Tang et al. Is to solve the problem of learning node representation in large-scale information networks. Traditional graph embedding methods face the challenges of computational complexity and storage overhead when dealing with large-scale networks[12]. LINE aims to improve the scalability of embedding methods by reducing the computational and storage complexity.

4.1.1 First Order Similarity

In the first-order similarity, LINE learns the vector representation of nodes by maximizing the similarity between nodes and their neighbors. Specifically, for each node u and its neighbor node v, the goal is to maximize the similarity between them. This goal can be formalized as the following formula:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

$$p_1(v_i, v_j) = \frac{1}{1+\exp(-\vec{u}_i^T \times \vec{u}_j)} \quad (3)$$

Where, is the conditional probability of edges i and j given node v.

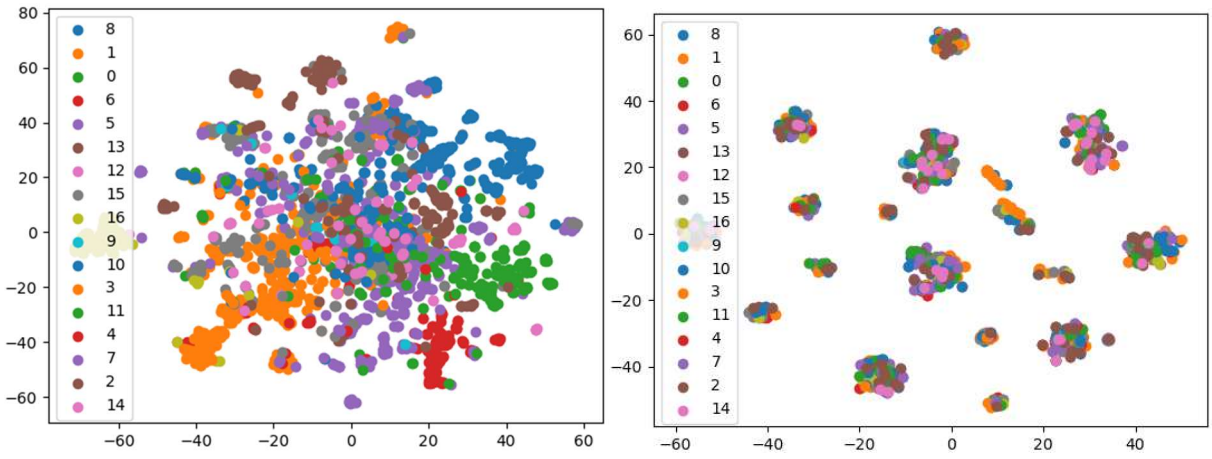
4.1.2 Second Order Similarity

The first-order similarity only considers the similarity between the direct neighbor nodes, while the second-order similarity considers the similarity between the neighbor nodes of the node's neighbor nodes. Two vectors $\vec{u}_i$ and $\vec{u}_i'$ are introduced, where $\vec{u}_i$ is the representation of v_i when it is regarded as a vertex, and $\vec{u}_i'$ is the representation of v_i when it is regarded as a specific "context". For each directed edge (i, j), we first define the probability of the "context" v_j generated by vertex v_i as:

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}_i'^T \times \vec{u}_j)}{\sum_{k=1}^{|V|} \exp(\vec{u}_i'^T \times \vec{u}_j)} \quad (4)$$

4.1.3 Experimental Part

LINE is optimized in two stages: the first stage uses negative sampling to learn the first-order neighbor embedding of nodes, and the second stage uses negative sampling to learn the second-order neighbor embedding in the case of fixed first-order neighbor embedding. The joint optimization of these two stages enables LINE to better retain the neighbor information between nodes. LINE introduces a neighbor sampling strategy, which can better balance the influence of highly similar nodes and low similar nodes as shown in Fig 8(b).



(a) (b)

Fig. 8 (a)dimension reduction of high-dimensional vector to two-dimensional via line
(b) reduce the dimension of high-dimensional vector to two-dimensional through struc2vec

Through different sampling weights, LINE can pay more attention to important neighbor nodes, so as to better learn the embedding of nodes. LINE uses the negative sampling technique to construct the optimization objective function by introducing some negative samples into each positive sample. Because the optimization objective function of LINE can be decomposed into independent parts of each node, LINE has good scalability in large-scale networks. This means that LINE can learn node embedding more efficiently when dealing with large-scale networks.

4.2 struc2vec

Leonardo F.R, et al. Proposed a novel method of struc2vec, which aims to learn the node representation from the node's structural identity to better capture the node's location and structural information in the graph. When its structural importance is greater than the neighbor's importance, it

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

has better results. This method still continues the traditional method in image processing, by performing random walk on the graph and using Skip-gram model to learn the representation of nodes. Different from the traditional method, the representation of nodes is learned according to the combination of different "structural roles" encountered in random walk[13].

Although this method has achieved some success, it faces the limitations of scalability and expression ability when dealing with large-scale complex graphs. To overcome these limitations, graph convolution neural network (GCN) is introduced as a graph embedding learning framework. However, the simple graph convolution model has limitations in capturing long-distance dependence, which has stimulated the development of depth graph convolution neural network[14].

5. Graph Embedding based on Graph Neural Network

In addition to the previous methods, there are also methods based on self encoder. Previous DeepWalk, LINE, node2vec, and struc2vec all used shallow structures, and shallow models often failed to capture highly nonlinear network structures.The proposed goal is to fully mine the high-order structure information of the network through the deep learning method, so as to improve the quality of network representation[15].

5.1 SDNE

SDNE (structural deep network embedding) proposed by Daixin Wang et al. Is a network representation learning method based on deep learning, which uses multiple nonlinear layers to capture the first-order and second-order similarity. SDNE works in parallel with node2vec. It can also be seen as an extension based on LINE[16]. SDNE uses an automatic encoder structure to optimize the first-order and second-order similarity at the same time. The learned vector representation can retain the local and global structure, and is robust to sparse networks[17].

SDNE mainly includes two parts: autoencoder and objective function. The self encoder is used to learn the low dimensional representation of nodes in the network, and the objective function is used to optimize the learning process, so that the learned node representation can retain the structure information of the network as much as possible. Use self encoder to learn the representation of nodes[18]. The regularization term is used to constrain the learned representation to ensure its smoothness and sparsity.

5.2 GCN

When discussing the research progress of graph embedding, we have witnessed the birth and evolution of a variety of technologies. In particular, the introduction of graph convolution neural network (GCN) has brought revolutionary changes to the processing of graph structured data. This method can effectively capture the complex patterns and relationships in the graph structure by aggregating the neighborhood information of nodes to generate the embedded representation of nodes[19]. The depth GCN expands the receptive field by stacking multi-layer convolution, so it can learn more abstract and complex feature representation. But what follows is the over smoothing problem, that is, with the increase of the number of layers, the representation of different nodes tends to be homogeneous, which damages the discrimination ability of the model[20].

In addition to the above methods, graph contrast learning is also used to improve the performance of graph embedding. A multi view graph contrast learning neural network model generates multiple local sub views by edge perturbation strategy, and uses graph diffusion technology to generate global views. Through comparative learning, the model can better learn the feature representation of nodes, especially in the case of few labeled nodes, and improve the generalization ability of the model. The research of graph embedding is developing towards deeper and more complex structure. Scholars not only pay attention to the representation learning ability of the model, but also try to solve the problems of over smoothness, robustness and weak generalization[21].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

6. Future Challenges and Prospects of Graph Embedding

Graph embedding technology, as an important branch of deep learning, has shown its powerful ability in processing complex graph structure data in recent years. Especially in social network analysis, chemical molecular analysis, traffic network analysis and other fields, the application of graph embedding technology has achieved remarkable results[22].

In the future, there are still many potential development directions in the research direction of graph embedding. For example, dynamic graph embedding is an important research direction for dynamic graph data, that is, the graph structure changes with time. Future research can focus on how to effectively capture the evolution process of nodes and edges in dynamic graph data, realize the embedded representation of nodes at different time points, and improve the temporal performance of embedding. Future research can explore how to combine depth generation model and graph embedding technology to generate more representative and rich graph data representation[23].

Through in-depth research and exploration in these directions, we can promote the application and further development of graph embedding technology in various fields, and bring more innovations and breakthroughs in the field of graph data mining and analysis.

7. Summary

As a powerful tool for analyzing complex network data, graph embedding technology has received extensive attention and research in recent years. With the increasing popularity of Internet, cloud computing, Internet of things and other technologies, the application of graph structure data in social network analysis, chemical molecular analysis, traffic network analysis and other fields has become particularly critical. The core of graph embedding method is how to efficiently capture the complex relationships between nodes, and map these relationships to low dimensional space for easy calculation and analysis.

Depth map convolutional neural network has become a research hotspot because of its powerful feature extraction ability. By increasing the convolution depth, graph convolution neural network can learn more abstract and complex feature representation, so as to improve the accuracy of node classification. However, the depth model faces the problem of over smoothing and the consequent rise in computational complexity and storage costs. In order to solve these problems, researchers have proposed a variety of innovation models.

The research progress of graph embedding technology shows that the continuous innovation and optimization of depth map convolutional neural network can effectively improve the accuracy and efficiency of network analysis. These research results not only promote the development of graph embedding technology, but also provide strong technical support for processing large-scale graph structure data. In the future, with the further improvement of algorithm and computing power, graph embedding technology is expected to play its great potential in more fields.

References

[1] Li P ,Hu Y . Deep graph embedding based on Laplacian eigenmaps for MR fingerprinting reconstruction [J]. Medical Image Analysis, 2025, 101 103481-103481.

[2] Mastrolorito F ,Gambacorta N ,Ciriaco F , et al. Chemical Space Networks Enhance Toxicity Recognition via Graph Embedding. [J]. Journal of chemical information and modeling, 2025,

[3] Noori A ,Balafar A M ,Bouyer A , et al. A novel contrastive multi-view framework for heterogeneous graph embedding [J]. Knowledge and Information Systems, 2025, (prepublish): 1-27.

[4] Han J ,Jang H Y ,Moon W J , et al. Vertical Memristive Crossbar Array for Multilayer Graph Embedding and Analysis. [J]. Advanced materials (Deerfield Beach, Fla.), 2025, e2416988.

[5] Alam T M ,Ahmed F C ,Samiullah M , et al. Supervised graph embedding for classification using discriminating frequent patterns [J]. International Journal of Data Science and Analytics, 2025, 19 (2): 1-13.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202507_11(7).0028

[6] Pan J ,Wang R ,Liu W , et al. Predicting phage-host interaction via hyperbolic Poincaré graph embedding and large-scale protein language technique [J]. iScience, 2025, 28 (1): 111647-111647.

[7] Hu Z ,Kui X ,Liu C , et al. Improving cancer driver genes identifying based on graph embedding hypergraph and hierarchical synergy dominance model [J]. Expert Systems With Applications, 2025, 267 126173-126173.

[8] Xu Y ,Huang B ,Zhou C , et al. Video Anomaly Detection with Hyperbolic Graph Embedding and Masked Normalizing Flows [J]. Electronics, 2024, 13 (24): 5013-5013.

[9] Wang R ,Guo Q ,Dai S , et al. An OD time prediction model based on adaptive graph embedding [J]. Physica A: Statistical Mechanics and its Applications, 2025, 657 130217-130217.

[10] Perozzi B,Al-Rfou' R,Skiena S.DeepWalk:Online Learning of Social Representations.[J].CoRR,2014, abs/1403.6652

[11] Mach M ,Amico E ,Liégeois R , et al. Connectome embedding in multidimensional graph spaces [J]. Network Neuroscience, 2024, 8 (4): 1129-1148.

[12] Zan S ,Ji W ,Zhou G . Knowledge graph embeddings based on 2d convolution and self-attention mechanisms for link prediction [J]. Applied Intelligence, 2024, 55 (2): 104-104.

[13] Xu G ,Rao G ,Zhang L , et al. Entity-relation aggregation mechanism graph neural network for knowledge graph embedding [J]. Applied Intelligence, 2024, 55 (1): 43-43.

[14] Mamani A E ,Buettner F ,Castanon B A C , et al. Exploratory analysis of metabolic changes using mass spectrometry data and graph embeddings [J]. Scientific Reports, 2024, 14 (1): 29570-29570.

[15] He C ,Cheng H ,Yang J , et al. Signed graph embedding via multi-order neighborhood feature fusion and contrastive learning [J]. Neural Networks, 2025, 182 106897-106897.

[16] Cao Y ,Lin X ,Wu Y , et al. A data-centric framework of improving graph neural networks for knowledge graph embedding [J]. World Wide Web, 2024, 28 (1): 2-2.

[17] Hou G ,Yu Q ,Chen F , et al. Directed Knowledge Graph Embedding Using a Hybrid Architecture of Spatial and Spectral GNNs [J]. Mathematics, 2024, 12 (23): 3689-3689.

[18] Zheng F ,You L ,Zhao X , et al. Predicting Tandem Mass Spectra of Small Molecules Using Graph Embedding of Precursor-Product Ion Pair Graph. [J]. Analytical chemistry, 2024,

[19] Cai L ,Ma X ,Ma J . Integrating scRNA-seq and scATAC-seq with inter-type attention heterogeneous graph neural networks. [J]. Briefings in bioinformatics, 2024, 26 (1):

[20] Chen D ,Zhang S ,Zhao Y , et al. An Attribute Graph Embedding Algorithm for Sensing Topological and Attribute Influence [J]. Mathematics, 2024, 12 (23): 3644-3644.

[21] Choi I ,Kim C W . Unlocking ETF price forecasting: Exploring the interconnections with statistical dependence-based graphs and xAI techniques [J]. Knowledge-Based Systems, 2024, 305 112567-112567.

[22] Liu J ,Ji L ,Chen K , et al. Railway operational hazard prediction and control based on knowledge graph embedding and topological analysis [J]. Reliability Engineering and System Safety, 2025, 258 110917-110917.

[23] Zhou Y ,Xie C ,Wang J G , et al. Forecasting cryptocurrency volatility: a novel framework based on the evolving multiscale graph neural network [J]. Financial Innovation, 2025, 11 (1): 87-87.