International Journal of Data Warehousing and Mining
Volume 19 • Issue 2

CTNRL:
A Novel Network Representation Learning With Three Feature Integrations

Yanlong Tang, Qinghai Normal University, China*
https://orcid.org/0000-0002-8648-977X
Zhonglin Ye, Qinghai Normal University, China
Haixing Zhao, Qinghai Normal University, China
Ying Ji, Qinghai Normal University, China

ABSTRACT

Network representation learning is one of the important works of analyzing network information. Its purpose is to learn a vector for each node in the network and map it into the vector space, and the resulting number of node dimensions is much smaller than the number of nodes in the network. Most of the current work only considers local features and ignores other features in the network, such as attribute features. Aiming at such problems, this paper proposes novel mechanisms of combining network topology, which models node text information and node clustering information on the basis of network structure and then constrains the learning process of network representation to obtain the optimal network node vector. The method is experimentally verified on three datasets: Citeseer (M10), DBLP (V4), and SDBLP. Experimental results show that the proposed method is better than the algorithm based on network topology and text feature. Good experimental results are obtained, which verifies the feasibility of the algorithm and achieves the expected experimental results.

KEYWORDS

Clustering, Network Representation Learning, Node Embedding, Random Walking, Text Features

INTRODUCTION

The data scale of network structures has increased with the advent of the era of big data. So, too, has the difficulty of processing such data. Improved processing of data has, therefore, become a hot issue in current research. It is important to discover hidden information within the network when dealing with the network structure data. Network representation learning aims to learn a vector representation for each node in the network and map it into a vector space. The vector dimension learned is far smaller than the size of the network size, which is conducive to discovering information hidden in the network. By representing the node vector learned by the learning algorithm, it can be directly

DOI: 10.4018/IJDWM.318696 *Corresponding Author

used for the processing of subsequent tasks like node classification (Tang et al., 2016; Zhou et al., 2006) and link prediction (Cao et al., 2010; Yang et al., 2019).

At present, most work is based on network local structure or combined with a kind of feature information to obtain the representation of the node vector. This kind of work has not done much to improve the quality of network embedding. In this paper, we based on DeepWalk algorithm (Perozzi et al., 2014) to make better use of other information within the network. It presents a CTNRL algorithm which can combine multiple feature information in the network, so as to improve the quality of network embedding.

In the first stage of training, the CTNRL algorithm gathers the nodes in the network through an unsupervised clustering algorithm to obtain the clustering information of each node. At the same time, a random walk is carried out on the network to obtain the random walk sequence of nodes. In the case of a given central node, the probability of the occurrence of context nodes is maximized to obtain the topology neighborhood relationship of network nodes to model the topology relationship of nodes in the network. Second, it evaluates the relevance of the network node content, maximizing the probability of the same word appearing at the node, capturing the relevance of text information of a given node, training and constraining the process of representing learning, and adding the clustering information of each node, directly modeling the node clustering information with the context, and learning the input vector and output vector of word clustering. The embedding of the target node is determined by the topology and text characteristics of the node to obtain the optimal network node vector.

RELATED WORKS

The early work of network representation learning is mainly based on matrix eigenvector calculation. This includes local linear representation and Laplace feature table. This method requires high spatial complexity and time complexity, which makes such algorithms unable to be used on large networks.

Inspired by the Word2vec (Mikolov, Sutskever, et al., 2013; Mikolov, Chen et al., 2013; Mikolov et al., 2015) algorithm, the DeepWalk algorithm introduces a neural network into network representation learning. The algorithm first carries out a random walk on the network. Then, it inputs the node sequence into the neural network to obtain the vector representation of nodes. The subsequent Node2vec (Grover & Leskovec, 2016) algorithm and LINE (Tang et al., 2015) algorithm are inspired by DeepWalk algorithm. Unlike methods like DeepWalk, which uses shallow neural networks, the SDNE (Wang et al., 2016) algorithm uses an unsupervised deep autoencoder for the training of network nodes. Different from the DeepWalk algorithm and SDNE algorithm based on the near neighbor hypothesis, the struc2vec (Ribeiro et al., 2017) algorithm believes that two nodes that are not close neighbors may have high similarities.

Matrix-based decomposition is another research focus of network representation learning. It decomposes the target matrix into several matrix multiplication forms to obtain the vector representation of nodes. The GraRep (Cao et al., 2015) algorithm, as a representation learning algorithm of classical matrix decomposition, considers a special relational matrix. It performs SVD decomposition on the special relational matrix to obtain the vector representation of each node. Subsequently, Cheng et al. (2015) first proved that the DeepWalk algorithm based on neural networks is equivalent to matrix decomposition. The TADW algorithm was proposed by combining the text information matrix based on this information.

In recent years, researchers have been studying how to use other abundant information in the network to improve the effect of network node classification. Li et al. (2020) combined semantic information to improve the accuracy of node classification. This method integrated hash learning, semantic information, and structural information in the same framework for the first time. The CANE algorithm was proposed to solve the different roles of the target node when interacting with other nodes (Tu et al., 2017). This introduces a mutual attention mechanism that integrates the structural

information and text information of the node. In addition to the textual features, Xu et al. (2021) proposed CAJE in combination with the global attention mechanism. This introduced a convolutional neural network into representation learning to understand the attribute information of neighbor nodes and obtain node vectors.

Chen and Li (2021) made use of the rich information contained in the graph by proposing MAGCN, a network representation learning model that integrates multiorder neighborhood information based on attention mechanism. Ni et al. (2021) proposed the DPBCNE algorithm, which considers both edge perspective and node perspective. It obtains node vector and side vector through coupling learning. Sun et al. (2019) proposed a representation learning method, vGraph, which can detect overlapping communities and nonoverlapping communities and learn nodes and community embedding at the same time. It designs a smooth regularizer in the potential space to make adjacent nodes more similar in the vector space. Wang et al. (2021) proposed the NTF algorithm, inspired by the energy-level dissipation method. The work constructed an influence subgraph for the central node to reduce the influence of noise on the training process. Zhang and Lu et al. (2020) proposed the ANEMF algorithm for matrix decomposition based on cosine similarity. It decomposed first-order structural similarity matrix and second-order structural similarity matrix and attribute matrix, which could preserve the characteristic information of the network.

Zhang and Chai et al. (2020) proposed the ANESC algorithm. At the initial stage of training, the algorithm combines the network structure with attribute information. It restrains the vector representation of nodes by clustering features during training. Zhang and Yin (2021) proposed the HINSC algorithm using meta-path and clustering information on heterogeneous networks. Other network representation learning work can refer to the work of Sun et al. (2021). In the current work, the network structure, node text features, node clustering information cannot be integrated into a framework.

To make better use of the information in the network, this article proposes the CTNRL algorithm. It employs the network structure, text features, and clustering features to improve the effect of node classification. At the network level (to obtain the clustering information of each node), the algorithm first conducts clustering on the whole network. Second, the network structure, which is modeled by a random walk, combines the clustering feature and text feature of nodes to obtain a good node embedding.

DEFINITIONS

Define the network as $G = \left(V, E, W, S\right)$, where $V = \left\{v_1, \ldots\ldots, v_N\right\}$ is the node set of the network, $N = \left|V\right|$ is the total number of nodes in the network, $E = \{e_{ij} \mid 0 < i, j \leq N\}$ is the edge set of the network, e_{ij} is represented as an edge of the node *i* to node *j*, *W* is the external text features of the nodes in the network, and $S = \left\{s_1, \ldots, s_N\right\}$ is the clustering feature of the network.

Given a network $G = \left(V, E, W, S\right)$, network representation learning aims to learn a low-dimensional vector representation $v_i \in V$, where $d << N$. Therefore, in the network, the nodes with close topology, the same text features, and the same cluster information are closer in the vector space.

BASIC KNOWLEDGE

DeepWalk

Word2vec, as a classic algorithm in word representation learning, has a profound impact on the field of representation learning. In this algorithm, there are two classical frameworks, including Skip-Gram

and CBOW. For a word sequence $W = \left\{w_1, \cdots, w_n\right\}$, Skip-Gram maximizes the occurrence probability of context words when the target word is given:

$$L\left(W\right) = \sum_{i=1}^{N} \sum_{-k \le j \le k} logPr(w_{i+j} \mid w_i)$$

where *k* is the sliding window size and w_i is the current word.

Inspired by Word2vec, the DeepWalk algorithm introduces the neural network method into representation learning. In the DeepWalk algorithm, assuming that the embedding of the current node is related to the context node, the algorithm first randomly wanders on the network to generate the node sequence $V = \left(v_1, \cdots, v_n\right)$. It then selects the context node set $c\left(v\right)$ through the sliding window, maximizing the probability of the occurrence of the context node given the current node:

$$L\left(V\right) = \sum_{i=1}^{N} \sum_{-k \le j \le k} logPr(v_{i+j} \mid v_i)$$

where $\Pr(v_{i+j} \mid v_i)$ is the predicted probability of the context node under the current node, formally defined by the softmax function:

$$\Pr(v_{i+j} \mid v_i) = \prod_{v_j \in c(v_i)} \Pr(v_{i+j} \mid v_i) = \prod_{v_j \in c(v_i)} \frac{\exp(v_{i+j}^{\mathrm{T}} \cdot v_i)}{\sum_{v_j' \in V} \exp(v_{i+j}^{'\mathrm{T}} \cdot v_i)}$$

where v_{i+j} and v_i are the embeddings of the context node and the current node, respectively.

Clustering Property

In the network, nodes of the same kind are usually clustered together. Therefore, the vector representation of nodes of the same category is closer according to the clustering information of each node. As one of the classical algorithms of clustering algorithms, the *k-means* algorithm has been widely used for its simple implementation and good effect. In the early stage of network representation learning, the clustering algorithm is first used to cluster the network to obtain the clustering information of each node. The corresponding objective function is:

$$L_c = \sum_{i=1}^{N} \min_{c \in C} \left|\left| v_i - \mu_c \right|\right|_2^2$$

where *S* is the number of clusters and μ_s is the center of cluster *s*.

PROPOSED APPROACH

Most of the previous network representation learning work was carried out on the network structure. It ignored other information contained in the network, such as attribute characteristics. To better combine other information of the network to obtain good node vector representation, this article proposes the CTNRL algorithm. This is a good combination of network structure, text features, and

clustering features. In turn, it can better learn vector mapping for nodes in the network. The framework of CTNRL algorithm is shown in Figure 1.

In the CTNRL model, a random walk is carried out on the network structure to model the topological relationship between nodes. At the same time, the relevance of the node textual features is evaluated, and two Skip-Gram frames are linked through the target node. Finally, the clustering information of the node is used as the input to learn the input clustering information vector and the output node vector. The vector of the target node is jointly affected by the topology of the node and the node text characteristics. The objective function of the CTNRL algorithm is as follows:

$$L(V) = \alpha \sum_{i=1}^{N} \sum_{-k \le j \le k} \log \Pr(v_{i+j} \mid v_i) + (1-\alpha) \sum_{i=1}^{N} \sum_{-k \le j \le k} \log \Pr(w_j \mid v_i) + (1-\alpha) \sum_{i=1}^{|S|} \sum_{-k \le j \le k} \log \Pr(w_j \mid s_i)$$

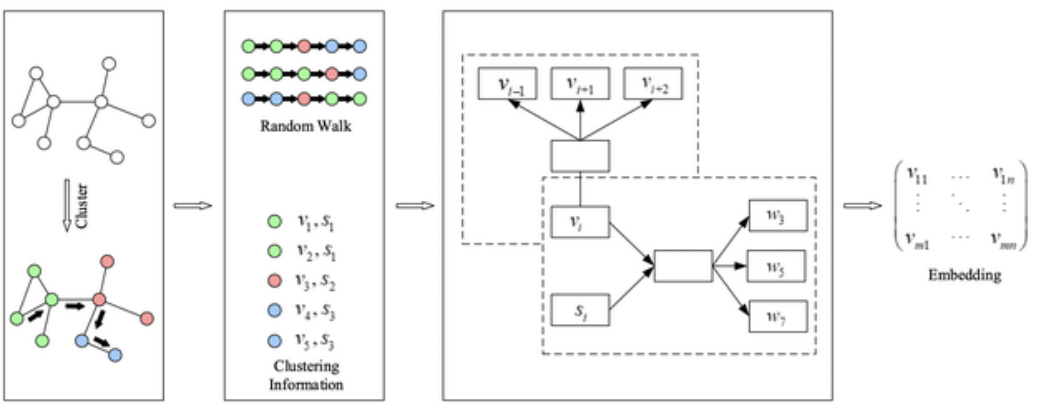
where α is the weight coefficient that balances the network structure, text feature, and clustering feature; *k* is the sliding window size; v_{i+j} is the context node of node v_i; w_j is the *j*th word in the context window; and s_i is the clustering information of node v_i. The probability of the occurrence of the context node under the current node, denoted $\Pr(v_{i+j} \mid v_i)$, is defined by:

$$\Pr(v_{i+j} \mid v_i) = \frac{\exp(v_i^{\mathrm{T}} \cdot v_{i+j})}{\sum_{i=1}^{N} \exp(v_i^{\mathrm{T}} \cdot v)}$$

where v_{i+j} and v_i are the embeddings of the context node and the current node, respectively. The probability of the occurrence of the context node words under the current node, denoted as $\Pr(w_j \mid v_i)$, is defined by:

$$\Pr(w_j \mid v_i) = \frac{\exp(v_i^{\mathrm{T}} \cdot w_j)}{\sum_{w=1}^{W} \exp(v_i^{\mathrm{T}} \cdot w)}$$

Figure 1. CTNRL Model Framework



where w_j and v_i are the embeddings of the context words and the current node, and *W* is the number of different words of the entire network node. The probability of the occurrence of the context node words under the current community, denoted as $\Pr(w_j \mid s_i)$, is defined by:

$$\Pr(w_j \mid s_i) = \frac{\exp(s_i^{\mathrm{T}} \cdot w_j)}{\sum_{w=1}^{W} \exp(s_i^{\mathrm{T}} \cdot w)}$$

where w_j and s_i are the embeddings of the context words and the current community.

EXPERIMENTAL

To verify the experimental results of the CTNRL algorithm, three datasets—Citeseer (M10), DBLP (V4), and SDBLP—are selected for experimentation. Table 1 shows the metrics associated with the data set.

If there are isolated nodes in the network, the random walk results of CTNRL algorithm and comparison algorithm of the same category will be affected. In order to ensure that the results of the experiment are not affected, isolated nodes in the Citeseer dataset and DBLP dataset were deleted. After the isolated nodes were deleted, 4,610 nodes in the Citeseer dataset and 17,725 nodes in the DBLP dataset remained. To verify the experimental results in a dense network, nodes with fewer than three edges in the SDBLP dataset were deleted. In total, 3,119 nodes remained in the SDBLP dataset after the processing was completed.

Comparison Algorithms

1. **DeepWalk:** The DeepWalk algorithm introduces deep learning into network representation learning. It trains neural networks to obtain vector representation of nodes.
2. **LINE:** The LINE algorithm considers the first- and second-order similarity, which is more suitable for large-scale networks.
3. **Node2vec:** The Node2vec algorithm improves the random walk mode of the DeepWalk algorithm.
4. **GraRep:** The GraRep algorithm considers a special relationship matrix. It uses SVD to decompose the relationship matrix and obtain a vector representation of the node.
5. **Text Feature (TF):** This converts the text feature of the node into a co-occurrence matrix and uses SVD to decompose the co-occurrence matrix to obtain a vector representation of the node.
6. **DW + TF:** This method is a combination of DeepWalk and the Text Feature algorithm. The obtained vectors are spliced in the form of column vector expansion.
7. **MFDW:** MFDW belongs to the representation learning algorithm of matrix decomposition, which obtains the vector representation of nodes through SVD decomposition.

Table 1. Data Description

DS	Original Nodes	Original Edges	Isolated Node	Remaining Nodes	Remaining Edges	Average Degree
Citeseer	10310	5923	5700	4610	5923	2.57
DBLP	60744	105781	43019	17725	105781	11.936
SDBLP	60744	105781	0	3119	39516	25.339

Experimental Parameter Settings

In the experiment, the training set was divided into nine training sets with different proportions. These increased in proportion (0.10 to 0.90). The remaining data serves as a test dataset for the algorithm. The number of random walk sequences was set to 10 times, each walking 40 nodes. The window size was 5, and the balance weight coefficient as 0.2. To ensure the accuracy of the experiment, it was repeated 10 times for all the algorithms. The result of the experiment was the average of the 10 experiments.

Balance Weight Coefficient and Number of Cluster Centers

The balance weight coefficient α is an important parameter among the balance network structure, text features, and clustering features. Setting different balance weight coefficients has a great influence on the overall experiment. The optimal value of the balance weight coefficient is selected through experimental analysis to ensure good performance of the experiment.

The Citeseer dataset was selected as the reference dataset in the experiment. With other parameters unchanged, the balance weight coefficient was set to increase from 0.1 to 0.9, with an interval of 0.2. Table 2 shows the experimental results of different values of balance weight coefficients.

The selection of the number of network clusters will directly affect the vector representation of nodes in the network. In the experiment of observing node classification performance by selecting different numbers of clusters, the most appropriate number of clusters can be selected for the algorithm, which makes the experiment have good performance. Using the Citeseer dataset as the reference dataset, the number of clusters was increased from 5 to 30 at a time. The other parameters were constant. Table 3 shows the influence of different cluster numbers on network node classification performance.

In order to more intuitively observe the impact of different parameter values of balance weight coefficient and number of network clusters on the whole CTNRL algorithm, Figure 2 shows the experimental results of accuracy of network node classification on different proportion training sets and different number of clusters.

From the experimental results of different balance weight coefficient values, it can be found that the effect of network node classification decreases with the increasing value of balance coefficient. According to the experimental results in Figure 2, it can be found more intuitively that when the balance weight coefficient is set between 0.1 and 0.3, the node classification performance is the best. Therefore, to ensure more scientific experimental results, the value of the balance weight coefficient is set between 0.1 and 0.3 on the three datasets.

As can be seen from the experimental results of different cluster number values in Figure 2, the classification performance of network nodes changes with the change of cluster number. When the number of clusters is 20, the experimental results show a high value. This further proves that clustering features play a crucial role in the training process of network representation learning.

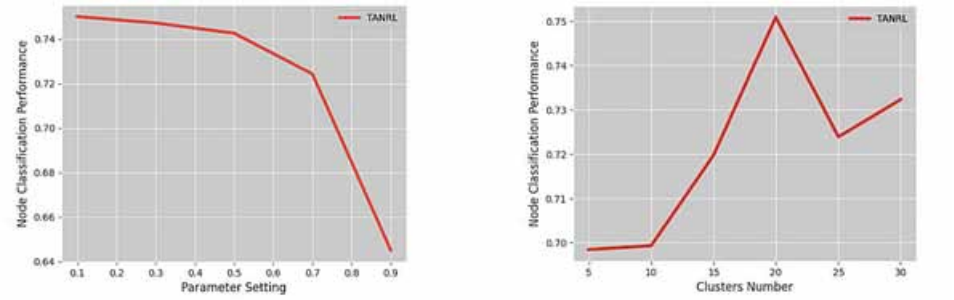
Table 2. Experimental Results Under Different Balance Weight Coefficients

Parameter Setting	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	Avg
$\alpha = 0.1$	**0.7353**	**0.7421**	**0.7511**	**0.7555**	**0.7482**	**0.7533**	**0.7543**	**0.7505**	**0.7609**	**0.7501**
$\alpha = 0.3$	**0.7331**	**0.7435**	**0.7484**	**0.7496**	**0.7534**	**0.7478**	**0.7501**	**0.7491**	**0.7502**	**0.7472**
$\alpha = 0.5$	0.7254	0.7375	0.7440	0.7414	0.7466	0.7458	0.7461	0.7501	0.7467	0.7426
$\alpha = 0.7$	0.6937	0.7136	0.7234	0.7289	0.7255	0.7285	0.7351	0.7315	0.7380	0.7243
$\alpha = 0.9$	0.5650	0.6291	0.6421	0.6541	0.6595	0.6650	0.6563	0.6620	0.6783	0.6457

Table 3. Experimental Results Under Different Cluster Numbers

Clusters Number	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	Avg
$S = 5$	0.6847	0.6979	0.6978	0.7036	0.6959	0.6997	0.7025	0.7081	0.6954	0.6984
$S = 10$	0.6827	0.6940	0.6983	0.7000	0.6988	0.7071	0.7040	0.7087	0.7000	0.6993
$S = 15$	0.7053	0.7159	0.7214	0.7203	0.7222	0.7277	0.7236	0.7240	0.7178	0.7198
$S = 20$	**0.7396**	**0.7487**	**0.7461**	**0.7496**	**0.7538**	**0.7517**	**0.7522**	**0.7625**	**0.7537**	**0.7509**
$S = 25$	0.7083	0.7175	0.7236	0.7247	0.7230	0.7275	0.7316	0.7350	0.7239	0.7239
$S = 30$	0.7164	0.7292	0.7338	0.7333	0.7349	0.7371	0.7397	0.7366	0.7293	0.7323

Figure 2. Average Value of Experimental Results



(a) Experimental results of different balance weight coefficients (b) Experimental results of different cluster numbers

Results and Analysis

Table 4, Table 5 and Table 6 show the experimental results of the algorithm on different data sets.

From the experimental results of the Citeseer dataset, it can be found that the results obtained by the proposed method in this article are better than other comparison algorithms. In addition, the accuracy of network node classification is 0.07 to 0.24 higher than that of other comparison algorithms. Compared with other algorithms based on network structure or matrix decomposition,

Table 4. Average Classification Performance of Nodes on the Citeseer(M10) Dataset

Algorithm	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	Avg
DeepWalk	0.5589	0.5930	0.6089	0.6148	0.6219	0.6230	0.6262	0.6233	0.6395	0.6122
LINE	0.4264	0.4706	0.4804	0.4957	0.5043	0.5102	0.5118	0.5307	0.5363	0.4963
Node2vec	0,6247	0.6561	0.6600	0.6707	0.6740	0.6715	0.6746	0.6807	0.6856	0.6664
GraRep	0.3938	0.5309	0.5785	0.5975	0.5997	0.6105	0.6157	0.6209	0.6089	0.5729
TF	0.5769	0.6130	0.6276	0.6305	0.6348	0.6330	0.6287	0.6219	0.6395	0.6229
DW+TF	0.5831	0.6115	0.6273	0.6337	0.6418	0.6396	0.6550	0.6549	0.6530	0.6333
MFDW	0.5762	0.6079	0.6233	0.6305	0.6296	0.6300	0.6300	0.6348	0.6430	0.6228
CTNRL	**0.7264**	**0.7384**	**0.7430**	**0.7460**	**0.7481**	**0.7463**	**0.7393**	**0.7485**	**0.7446**	**0.7423**

Table 5. Average Classification Performance of Nodes on the DBLP(V4) Dataset

Algorithm	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	Avg
DeepWalk	0.6226	0.6434	0.6542	0.6598	0.6624	0.6618	0.6660	0.6703	0.6677	0.6565
LINE	0.6449	0.6653	0.6749	0.6787	0.6798	0.6830	0.6903	0.6889	0.6886	0.6772
Node2vec	0.7339	0.7398	0.7525	**0.7561**	**0.7570**	**0.7585**	0.7579	0.7573	**0.7636**	0.7530
GraRep	0.5890	0.6590	0.6726	0.6792	0.6877	0.6888	0.6926	0.6956	0.6979	0.6736
TF	0.6617	0.6946	0.7049	0.7115	0.7129	0.7144	0.7154	0.7157	0.7183	0.7055
DW+TF	0.6261	0.6515	0.6599	0.6622	0.6637	0.6660	0.6703	0.6691	0.6761	0.6605
MFDW	0.6502	0.7468	0.7488	0.7502	0.7505	0.7513	0.7522	0.7457	0.7551	0.7390
CTNRL	**0.7475**	**0.7530**	**0.7546**	0.7554	**0.7570**	0.7550	**0.7588**	**0.7586**	0.7558	**0.7551**

Table 6. Average Classification Performance of Nodes on the SDBLP Dataset

Algorithm	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	Avg
DeepWalk	0.7976	0.8065	0.8188	0.8149	0.8256	0.8235	0.8273	0.8271	0.8337	0.8194
LINE	0.7379	0.7701	0.7811	0.7828	0.7931	0.7897	0.7963	0.7882	0.7877	0.7808
Node2vec	**0.8217**	0.8287	0.8309	0.8451	0.8445	0.8401	0.8484	0.8473	**0.8498**	0.8396
GraRep	0.8099	0.8252	0.8414	**0.8478**	**0.8497**	0.8417	**0.8536**	**0.8527**	0.8495	**0.8413**
TF	0.6503	0.7123	0.7264	0.7386	0.7454	0.7507	0.7514	0.7600	0.7533	0.7320
DW+TF	0.7939	0.8095	0.8160	0.8144	0.8255	0.8222	0.8210	0.8258	0.8315	0.8178
MFDW	0.7979	**0.8308**	**0.8438**	0.8412	0.8453	**0.8429**	0.8470	0.8455	0.8453	0.8377
CTNRL	0.7651	0.7786	0.7879	0.7900	0.7931	0.7915	0.7942	0.8016	0.7887	0.7879

the CTNRL algorithm also considers the external text features and clustering to obtain better results and expected experimental results.

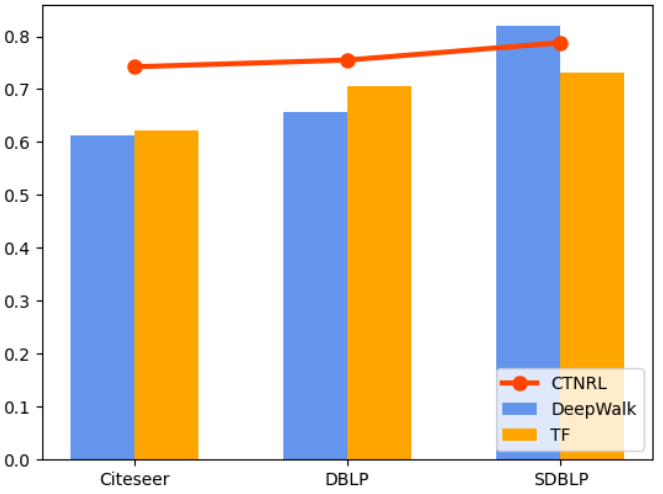
Compared with the Citeseer dataset, the number of nodes and edges in the DBLP dataset is greater than that of the Citeseer dataset. The average degree of the DBLP dataset is also higher than that of the Citeseer dataset, which is a larger and denser dataset. On the DBLP dataset, the proposed algorithm achieves the desired results on different proportions of training sets compared with other comparison algorithms. Experimental results of CTNRL algorithm on DBLP dataset show that the algorithm can be well adapted to large networks.

The SDBLP dataset is a dense dataset with more than three edges reserved. It is a smaller and denser dataset compared with other datasets. In the SDBLP dataset, the CTNRL algorithm has better node classification performance than the partial comparison algorithm in different proportions of training sets. Overall, the CTNRL algorithm has a poor effect. Due to the excessive density of the network, some nodes are incorrectly clustered in the clustering process. This influences the experimental results and causes errors in the node classification effect.

The experimental results of the CTNRL algorithm are compared with those of the DeepWalk algorithm based on local feature structure and the Text Feature(TF) algorithm based on text features, as shown in Figure 3.

As seen from the results in Figure 3, the CTNRL algorithm has better experimental results than the DeepWalk algorithm and TF algorithm on the Citeseer dataset and DBLP dataset due to the combination of network structure, text features, and clustering features. On the SDBLP dataset, the experimental results of the CTNRL algorithm are worse than the DeepWalk algorithm. Because SDBLP data set is a dense data set, DeepWalk algorithm gets better experimental results.

Figure 3. Experimental Results



The experimental results on Citeseer, DBLP, and SDBLP datasets show that the CTNRL algorithm has better experimental results on the larger network with smaller network averageness. However, the training results on the smaller network with denser network are worse. The main reason is that in the process of clustering, the nodes belonging to the same class are divided into different classes of nodes, resulting in the deviation of experimental results.

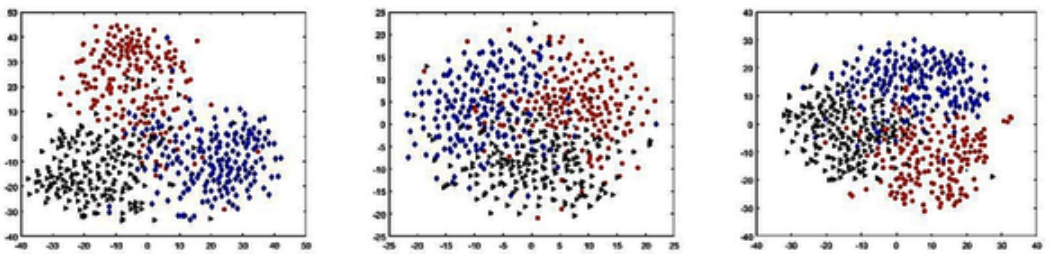
Visualization

Network visualization is an important means to evaluate the classification of network nodes. In network visualization analysis, the classification effect of network nodes can be presented more intuitively. If the node classification effect is good, there will be an obvious clustering phenomenon and obvious clustering boundary in the visualization task. This will facilitate subsequent tasks.

In the Citeseer dataset, DBLP dataset, and SDBLP dataset, three categories are randomly selected. Two hundred nodes are randomly selected in each category for visualization task analysis. Figure 4 shows the visualization results of the CTNRL algorithm on three datasets.

It can be seen from the visualization results that CTNRL algorithm has obvious clustering effects on the Citeseer, DBLP, and SDBLP datasets. Among them, the clustering effect is more pronounced on the Citeseer dataset. In addition, the Citeseer dataset has obvious

Figure 4. CTNRL Visualization



clustering boundaries. The DBLP datasets has a clustering of nodes of the same category; however, the clustering boundaries are not clear. The SDBLP dataset has a clustering effect and a clustering boundary. The clustering boundary is obvious compared to the DBLP dataset. There are four categories in the DBLP dataset. While there are many nodes in each category, the clustering is unsupervised. This results in the incorrect classification of nodes of the same kind in the process of clustering. Hence, the visualization results are worse than the Citeseer dataset.

Case Studies

The Citeseer dataset is selected for case analysis to further verify the experimental effect. First, we randomly select a node in the dataset, titled "Topological Quantum Information Theory" and class labeled "8." In the CTNRL algorithm, the five nodes with the highest cosine similarity to the selected target node are obtained for analysis. The results obtained are shown in Table 7.

As seen in Table 7, the titles of the five returned nodes have the same words as those of the target node. The class labeled of these nodes is the same. For example, in the first node returned, the node is titled "Topological Quantum Field Theory for Calabi-Yau Three Folds and G2 manifolds." The same words exist as the target node ("Topological," "Quantum," "Theory") and the class label of the returned node is the same as the class label of the target node. This shows that the algorithm considers the external text features of nodes when learning node vector representation. It also proves that the clustering information of nodes play a key role in the final node vector representation, which further verifies the feasibility of the algorithm.

CONCLUSION

In this paper, we propose a network representation learning algorithm CTNRL which can combine multiple feature information. The algorithm first conducts clustering on the network to obtain the clustering characteristics of the network. Then, it conducts a random walk on the network structure to model the relationship between nodes. Simultaneously, the nodes and clustering information are modeled to learn the input clustering vector and the output word vector. The vector of the target node is affected by both the network structure and node text feature. The experimental results show that the quality of node vector can be improved by adding text feature and clustering feature based on network structure. The experiment was validated on three datasets. The expected results were obtained.

Table 7. Case Analysis

Algorithm	Vertex Title	Cosine Similarity	Class Label
CTNRL	Topological Quantum Field Theory for Calabi-Yau Three folds and G2 manifolds	0.9668	8
	Ed Nelson's Work in Quantum Theory	0.9139	8
	Quantum Information Theory	0.9052	8
	The Search for the Holy Grailin Quantum Cryptography	0.8947	8
	Quantum Cryptography	0.8868	8

ACKNOWLEDGMENT

This article was supported by the National Key Research and Development Program of China (No. 2020YFC1523300); Qinghai Natural Science Foundation of China (No. 2021-ZJ-946Q); and the Key Laboratory of Tibetan Information Processing and Machine Translation of Qinghai Province (No. 2020-ZJ-Y05).

REFERENCES

Cao, B., Liu, N. N., & Qiang, Y. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 159–166). ACM Press.

Cao, S., Wei, L., & Xu, Q. (2015). GraRep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM Int on Conf on Information and Knowledge Management* (pp. 891–900). ACM Press. doi:10.1145/2806416.2806512

Chen, B., & Li, J. L. (2021). Attention-based network representation learning model using multi-neighboring information. *Journal of Chinese Computer Systems*, *42*(4), 761–765.

Cheng, Y., Liu, Z., Zhao, D., Sun, M., & Chang, E. (2015). Network representation learning with rich text information. In *Proceedings of the 24th Int Joint Conf on Artificial Intelligence* (pp. 2111–2117). AAAI Press.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). ACM Press. doi:10.1145/2939672.2939754

Li, Z., Tang, J., Zhang, L., & Yang, J. (2020). Weakly-supervised semantic guided hashing for social image retrieval. *International Journal of Computer Vision*, *128*(8-9), 2265–2278. doi:10.1007/s11263-020-01331-0

Mikolov, T., Karafiát, Burget, L., Cernock, J., & Khudanpur, S. (2015). Recurrent neural network based language model. In *Interspeech, Conference of the International Speech Communication Association.* DBLP.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Computer Science*. Advance online publication. doi:10.48550/arXiv.1301.3781

Mikolov, T., Sutskever, I., Kai, C., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conf on Neural Information Processing Systems* (pp. 3111–3119). MIT Press.

Ni, Q. X., Zhang, X., & Pu, Z. (2021). Coupled network embedding method based on dual perspectives. *Computer Systems and Applications*, *30*(9), 247–255.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 701–710). ACM Press. doi:10.1145/2623330.2623732

Ribeiro, L., Saverese, P., & Figueiredo, D. R. (2017). Struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining* (pp. 385–394). ACM Press. doi:10.1145/3097983.3098061

Sun, F. Y., Qu, M., Hoffmann, J., Huang, C. W., & Tang, J. (2019). vGraph: A generative model for joint community detection and node representation learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 514–524). ACM Press.

Sun, J. Q., Zhou, H., & Zhao, Z. Y. (2021). A survey of network representation learning methods. *Journal of Shandong University of Science and Technology*, *40*(01), 177–128.

Tang, J., Aggarwal, C., & Liu, H. (2016). Node classification in signed social networks. In *Proceedings of the 2016 SIAM International Conference on Data Mining* (pp. 54–62). SIAM Press. doi:10.1137/1.9781611974348.7

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on Worldwide Web* (pp. 1067–1077). Springer Press. doi:10.1145/2736277.2741093

Tu, C., Han, L., Liu, Z., & Sun, M. (2017). CANE: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1722–1731). Association for Computational Linguistics. doi:10.18653/v1/P17-1158

Wang, D., Peng, C., & Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1225–1234). ACM Press. doi:10.1145/2939672.2939753

Wang, W., Ma, D. Y., Xin, G., Han, Y., & Wang, B. (2021). A network representation learning method based on topology. *Information Sciences*, *571*, 443–458. doi:10.1016/j.ins.2021.04.048

Xu, Y. K., Ma, F. N., Yang, X. H., & Ye, L. (2021). Attribute network representation learning based on global attention. *Computer Science*, *48*(12), 188–194.

Yang, Y. L., Ye, Z. L., Zhao, H. X., & Meng, L. (2019). Link prediction algorithm based on high-order proximity approximation. *Jisuanji Yingyong*, *39*(8), 2366–2373.

Zhang, D. Y., & Yin, L. J. (2021). Clustering-preserving representation learning on heterogeneous network. *Computer Engineering and Applications*, *57*(7), 144–150.

Zhang, J., Chai, B. F., Zhang, P., & Li, W. B. (2020). Clustering-preserving representation learning in attributed network. *Jisuanji Yingyong Yanjiu*, *37*(06), 1647–1651.

Zhang, P., Lu, G. Y., Lv, S. Q., & Zhao, X. L. (2020). Attributed network representation learning based on matrix factorization. *Computer Engineering*, *46*(10), 67–73.

Zhou, D., Huang, J., & Schlkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems* (pp. 1601–1608). MIT Press.

Yanlong Tang (1997-), M.S. His research interests include network representation learning and deep learning.

Zhonglin Ye (1989-), Ph.D, associate professor. His research interests include question answering systems, presentation learning, and social network data mining.

Haixing Zhao (1969-), Ph.D, professor. His research interests include complex network modeling and analysis, network reliability analysis, etc.

Ji Ying (1997-), M.S. Her research interests include graph theory and its application.