Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research Paper

Link Prediction Using Network Embedding Based on Global Similarity

S.F. Mirmousavi, S. Kianian*

SRTTU Software Department of Computer Engineering Faculty, Shahid Rajaee Teacher Training University, Tehran, Iran.

Article Info

Article History:
Received 17 April 2019
Revised 10 August 2019
Accepted 11 December 2019

Keywords:
Complex network
Link prediction
Node embedding
Deep neural network

*Corresponding Author's Email Address:
Sahar.kianian@sru.ac.ir

Abstract

Background: The link prediction issue is one of the most widely used problems in complex network analysis. Link prediction requires knowing the background of previous link connections and combining them with available information. The link prediction local approaches with node structure objectives are fast in case of speed but are not accurate enough. On the other hand, the global link prediction methods identify all path structures in a network and can determine the similarity degree between graph-extracted entities with high accuracy but are time-consuming instead. Most existing algorithms are only using one type of feature (global or local) to represent data, which not well described due to the large scale and heterogeneity of complex networks.

Methods: In this paper, a new method presented for Link Prediction using node embedding due to the high dimensions of real-world networks. The proposed method extracts a smaller model of the input network by getting help from the deep neural network and combining global and local nodes in a way to preserve the network's information and features to the desired extent. First, the feature vector is being extracted by an encoder-decoder for each node, which is a suitable tool for modeling complex nonlinear phenomena. Secondly, both global and local information concurrently used to improve the loss function. More obvious, the clustering similarity threshold considered as the local criterion and the transitive node similarity measure used to exploit the global features. To the end, the accuracy of the link prediction algorithm increased by designing the optimization operation accurately.

Results: The proposed method applied to 4 datasets named Cora, Wikipedia, Blog catalog, Drug-drug-interaction, and the results are compared with laplacian, Node2vec, and GAE methods. Experimental results show an average accuracy achievement of 0.620, 0.723, 0.875, and 0.845 on the mentioned datasets, and confirm that the link prediction can effectively improve the prediction performance using network embedding based on global similarity.

Introduction

Graphs are one of the most widely used data structures in computer science and related fields. Social networks, protein-protein interactions, and recommender networks are the data structures modeled on the graph. In recent years, due to the widespread networks used in the real world, the analysis of graphs has attracted more consideration. Node classification [1], link prediction [2],

community detection [3], and recommender system [4] are widely used in the field of graph analysis [5] which link prediction has been considered in this paper. For more insight, your next connection on Facebook can be distinguished, using link prediction. The application of link predictions is not limited to social networks; For example, in the bioinformatics field, link predictions are used to detect protein-protein interactions [6]. In the field of e-commerce, link prediction is applicable to create a suggestion system [7]; and in security, it can help to find hidden terrorist groups [8]. As shown in Fig. 1, link predictions can be used to identify improper links and remove them from the network and also the investigation of the links to predict potential relationships between users and a social network [9].

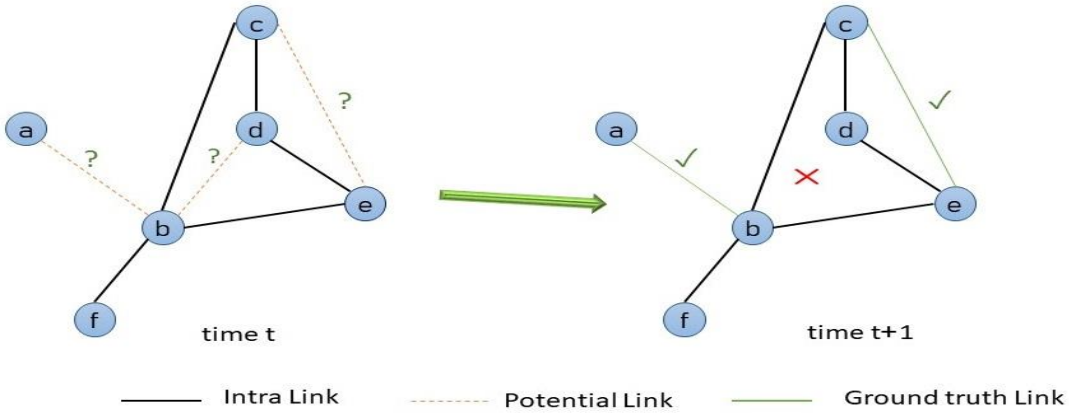


Fig. 1: An example of a bond prediction on a 6-node test graph.

So far, a variety of methods have been proposed for Link Prediction which includes, similarity-based methods [10,11], the maximum likelihood estimations (local and global) [12,13], and probabilistic methods [14,15]. Also, in recent years, we have seen an increment of approaches that automatically learn to encrypt the graph structure in nonlinear and low dimensional vectors. The idea of such methods is learning a data conversion function that attributes nodes to points in a vector space with low dimensions associated as embedding a node. The goal is to achieve a map of nodes in the whole network that represents all their structural features in the main graph. The node embedding techniques have led to advanced developments in network science [16]. As shown in Fig. 2, a node embedded on a small graph with 6 vertices; Every node, like u, is automatically converted to a numerical representative vector in the d-dimensional space, which d << n and n is the number of vertices in the graph.

Finally, the initial graph converts to n vectors which, has shown on the right side of Fig. 2.

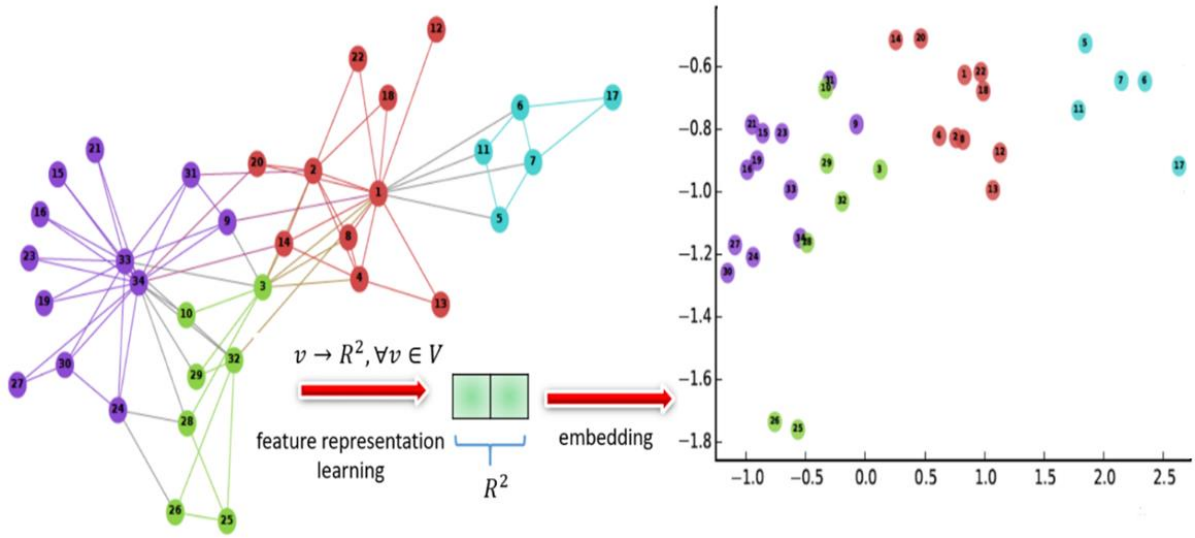


Fig. 2: left is, Graph structure of the Zachary Karate Club social network, where nodes are connected if the corresponding individuals are friends. The nodes are colored according to the different communities that exist in the network. right is, two dimensional visualization of node embedding generated from this graph using the DeepWalk method. The distances between nodes in the embedding space reflect similarity in the original graph, and the node embedding are spatially clustered according to the different color-coded communities [17].

The main challenge of using graphs in machine learning and link prediction is finding a way to extract information about the interaction between nodes, and integrate them into a machine learning model. To extract this information from networks, older approaches often use limited statistical information (such as vertices degree or clustering coefficients) or computational features to measure local neighbors. These classic approaches have limitations because these features are not flexible. They

often do not generalize to networks derived from other organisms, tissues, and experimental technologies, and only operate in a set of data with low experimental coverage [17].

In the following paper, a new node embedding approach presented for link prediction in the network to solve the existing challenge. Because depthless or superficial embedding methods are not able to use node features when encrypting. The proposed method designs a cryptographic-decoder that uses local and global attributes to identify information about the structure and characteristics of the graph. Therefore, the attribute vector extracted for any nodes, using a deep neural network, which is a suitable tool for modeling nonlinear-complex phenomena. Also, the link prediction accuracy has improved simultaneously, using global and structural information. In related work section, we will review the related works in the link prediction field. The proposed method to solve the problem of link prediction and the experiment results and their analysis have been presented in proposed method section and evaluation section, respectively. Finally, in conclusion section, we conclude and consider this issue related research path.

Related Work

Graph embedding methods can be divide into three groups: 1-Matrix factorization-based methods, 2-Random walk-based methods, 3-Neural network-based methods [18]. The characteristics of each above categories briefly provided in Table 1.

A. Matrix factorization-based methods

Matrix factorization-based methods represent the relationship between nodes in the form of a matrix. The purpose is to convert the data matrix into lower-dimensional matrices in a way that the main matrix's topological specification, structure, and characteristics be preserved [18]. To show the relationship among nodes, the node adjacency matrix, laplacian matrix, or Node transition probability matrix are applicable. Approaches to determining the matrix factor vary based on the matrix characteristics.

If the resulting matrix includes zeros and ones, as instance, For the Laplace matrix, eigenvalue decomposition, and for the unstructured matrices, gradient descent methods can be picking to obtain linear time embedding [16].

Table 1: List of graphs embedding methods [16]

Type	Similarity	Method	Year
Based on matrix factoring	1st degree neighbors	Laplacian Eigen maps [32]	**2001**
Based on matrix factoring	1st degree neighbors	Graph Factorization[19]	**2013**
Based on matrix factoring	Neighbors up to K distance	GraRep [21]	**2015**
Based on matrix factoring	Neighbors up to K distance	HOPE [22]	**2016**
Random walk-based	Neighbors up to K distance	DeepWalk [23]	**2014**
Random walk-based	Neighbors up to K distance	Node2vec [24]	**2016**
Based on the neural network	1st degree neighbors & 2nd degree neighbors	SDNE [20]	**2016**
Based on the neural network	Neighbors up to K distance	GCN [28]	**2017**
Based on the neural network	1st degree neighbors & 2nd degree neighbors	LINE [26]	**2015**

GraRep and HOPE algorithms are working based on matrix factorization. Due to the review of all pairs of nodes, time complexity in these methods is O (V^2), so high computational costs are one of the challenges in these algorithms. Also, there is a possibility of happening errors due to manual similarity measurement.

■ In GraRep [21], a method introduced for learning node indicator vectors in the weighted graph. Unlike the previous methods, this one has used the global information of graph structure by applying logarithmic conversion log-transformed, and node transition probability matrices to calculate latent vector matrices. This matrix constructed with neighbors at different distances, then the resulting matrices are added together.

■ Another way to measure multi-step similarity is to calculate the overlap rate of a node's neighbors. This method is known as HOPE, and the score of the k-step similarity is calculated using the neighbor overlap rate. The Katz Index function and Adamic-Adar scores are applicable for the overlap calculation [22].

B. Random walk-based methods

Random walk-based methods first select one of the

node's neighbors randomly and then move toward that neighbor, and repeating this process to obtain the node sequence. Then the word2vec model is used to learn the nodes sequence embedding, which the local similarities and structural information can be maintained. The Node2vec and DeepWalk algorithms are two cases of Random walk-based methods algorithms. This method employs a direct neighborhood method or second neighborhood relations ultimately and not able to reflect all the structural and global information of the graph, so a complete representation will not be provided for all graph nodes. Inefficiency for sparse graphs is another disadvantage of this method [18].

■ Bryan Peruzzi and his colleagues proposed the DeepWalk algorithm graph embedding [23]. In this method, the random walking algorithm is executed firstly on the input graph and produces several series of node sequences; this is repeated for all graph nodes so that a set of consecutive sequences is obtained for each node. Then, the Skip-gram algorithm is executed, using the sequences. This model is used to learn random walking on the graph, and a vector is generated for each node. The resulting vectors are used as feature vectors and guide the classifier, and finally evaluates.

■ The Node2Walk algorithm is based on the Skip-gram algorithm and works similarly to DeepWalk with the difference that the Biased Random Walk algorithm has replaced with a simple Random Walk algorithm. This algorithm has a deviation parameter and behaves more flexibly to collect node information. This method has high scalability because it uses the first surface and first depth search and also uses the direct and second-degree neighborhood relations of the node. Therefore, it attains two local and global views of nodes and adjusts the search space by defining different parameters, and makes the node sampling operation more varied than previous algorithms [24].

C. Neural network-based methods

The main issue in the network embedding approach is learning a function to map network space to one space with a tinier size. Some methods, such as matrix factoring, assume mapping performance is linear. However, the process of network formation is complex and nonlinear, so a linear function may not be sufficient to map the main network in the embedded space [25]. Deep neural networks have been very successful in modeling complex nonlinear phenomena in various fields, such as speech recognition and computer vision. Therefore, the use of deep neural networks is an efficient solution in cases where complex information is available. However, in the field of network representation learning, a small number of users have used deep neural networks.

■ In 2015, a method was proposed that is mainly applicable for learning features in large-scale network embedding information graphs [26]. The Line operation algorithm is performed in two steps; primarily, the first-order and second-order proximities are calculated, and in the second step, minimizing the first and second step Closeness. The proximity of the first-order is calculated similarly to the graph factoring that keeps the proximity matrix and multiplies a point close to each other is their both goals, except that the graph factoring does so by minimizing the difference between the proximity matrix and the point multiplication, but This algorithm uses two common probability distributions for each pair of vertices, one uses the proximity matrix and the other uses node embedding. The difference between this algorithm and the Node2Walk and DeepWalk algorithm is in extracting sample nodes. Other methods of embedding the node based on the deep neural network consider the global neighbors of each node as the node input; so, for scattered large-scale sparse graph g, calculations are costly and Non-optimal.

■ The Convolution Algorithm solves this problem by defining a Convolution operator in the graph. This model collects the Neighborhoods embedding of a node and uses the embedding function and the previous embed to achieve the new embedment. Embedding aggregation with the help of a local neighbors causes beneficial effects in scalability and provides the description of neighboring neighbors for multiple iterations. In this method, embedding is achievable without supervision by setting up unique tags for each node. The filter creation approaches like spatial and spectral filters are completely different in such categories. Spatial filters work directly on the main graph and the adjacent matrix; While spectral filters work in the Laplacian Graph spectrum [27].

Proposed Method

In the present paper, a decoder-encoder is designed based on deep neural networks to recognize the information about the structure and the graph characteristics.

In this method, a powerful threshold presented to evaluate the loss rate of the encoder-decoder by combining local and global characteristics. Therefore, the efficiency of the link prediction algorithm has increased with the loss function improvement in the encoder-decoder.

Fig. 3 shows the steps of the proposed algorithm. The whole process consists of three main steps: 1- Feature extraction 2- loss function, 3- Optimization. Before introducing the proposed method.

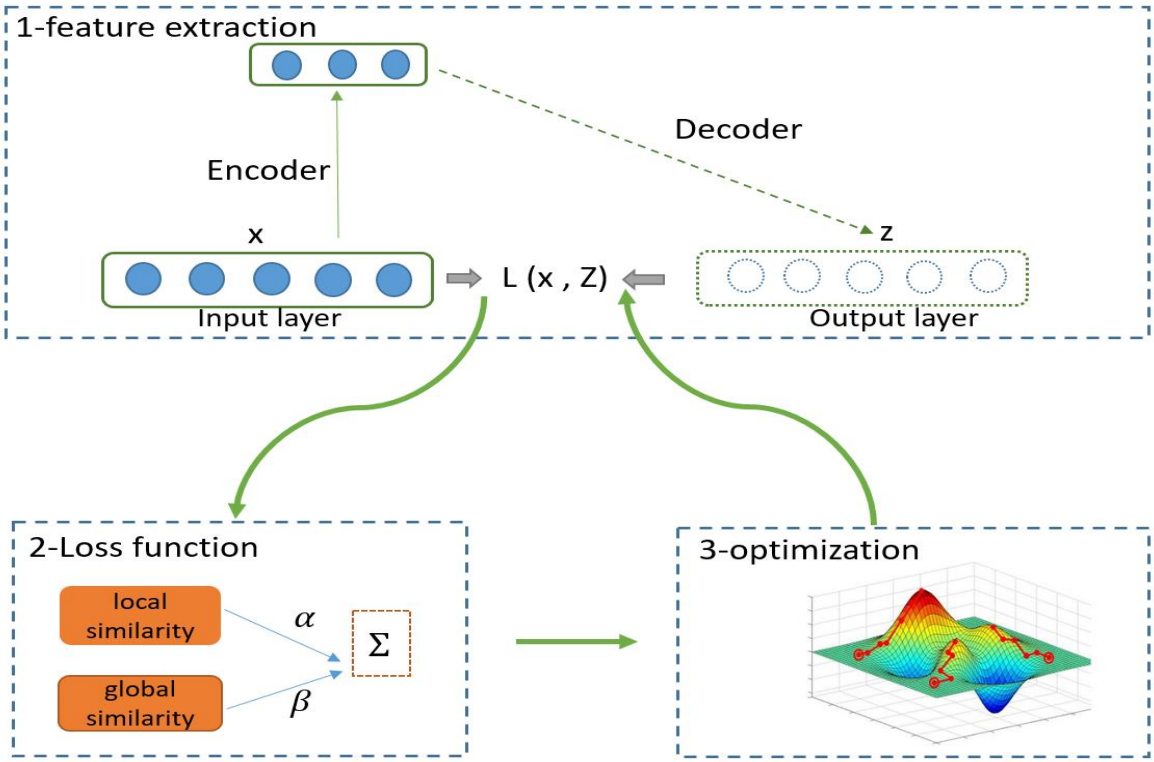


Fig. 3: The steps of performing the proposed method.

D. Feature extraction

Extracting the feature is the most important part of the proposed model, which is done by an encryption-decoder. Encryptions-decoders play an essential role in unsupervised learning and deep networks. The purpose of using encoders is to represent data in a way that to be used in classification. The best advantage of password-decoders is the automatic selection of features [28].

As can be seen in Fig. 4, the encoder is a neural network that receives a set of data without labels and encrypts them and tries to re-represent the inputs at the output so that they have the least possible difference with the input value. In encoding, the input data is mapped to the attribute space, and in decoding, space is converted from the attribute space back to its original state. The main part of an encryption-decoder is the intermediate hidden layer that is used as the extracted property for categorization.

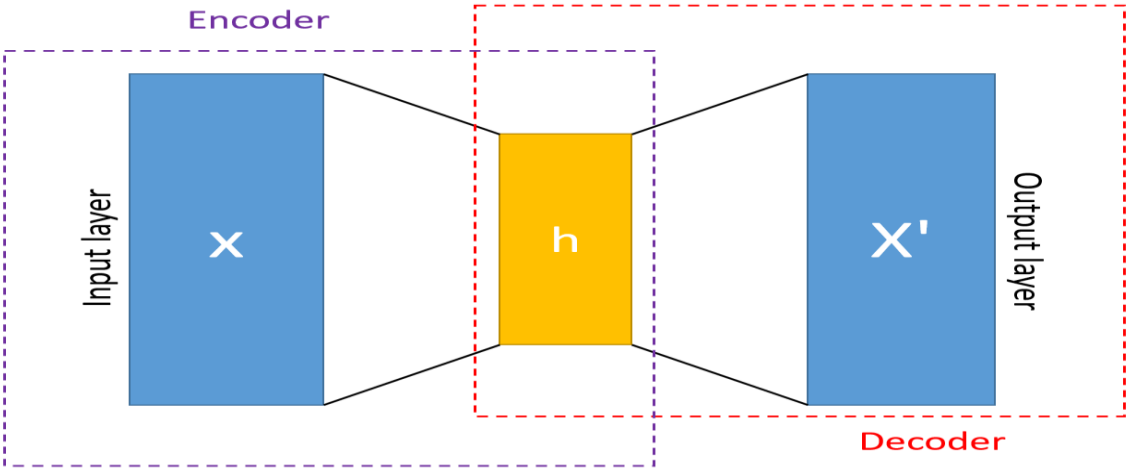


Fig. 4: General autoencoders view.

The encryption section is a feature extraction function that calculates the feature vector considering the inputs.

Therefore, if we display the feature vector with h, the encoder with f_θ and the data set with X_i, the equation (1) is established.

$$h^t = f_\theta(x^t). \qquad x^t = \{x^1. \ldots . x^T\} \tag{1}$$

The h vector is the calculated property of x. The decryption section is a function that displays it with $g_\theta$and according to the equation (2) mapping maps of the space characteristic of latent features or The latent space makes its way into the entrance space.

$$r = g_\theta(h) \tag{2}$$

Auto encoders attempt to minimize the reconstruction loss $L(x.r)$ between the original data and the reconstructed data. This is done by reconstructing r from x with pre-training and measuring the difference between x and r.

E. Loss Function

As we know, using global information in loss calculation leads to high accuracy, but it takes a lot of time to be calculated. On the other hand, actions based on local information are generally faster but provide less accuracy. Because the network type is large-scale, irregular, and heterogeneous real-world networks, both local and global structures are important [29]. In the proposed method, by combining local and global features, the accuracy of the loss function has improved.

i. local similarity

The proposed method considers the Integrated degree-related clustering similarity as a local index. Studies have shown that the combination of some structural features can lead to a strong threshold to show the nodes similarity level. Integrated degree-related clustering is one of the combinational thresholds on link prediction scope with high performance which is defined as below [30]:

$$DC_{ij} = \sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} dc(V_z) \tag{3}$$

$\Gamma(v_i) \cap \Gamma(v_j)$ indicates the number of common neighbors between the nodes v_i and v_j, and $dc(v_z)$represents the clustering related to the degree and is defined as the following equation :

$$dc = \frac{1}{N}\sum_{i=1}^{N} C(k_v)(k_v)^r \tag{4}$$

In equation (4) N indicates the number of nodes in the network. $C(k_v)$ represents the clustering coefficient of node v with a degree of k and r indicates the set coefficient of the network. Therefore, in order to achieve local characteristics, the loss function is defined as equation (5).

People who share their connections on social media tend to form associations or clusters. The tendency of nodes to form clusters in a graph is called the clustering coefficient, which is the ratio of existing edges between neighbors, to the maximum possible edges between neighbors of a vertex. The network clustering mechanism plays a critical role in edge formation, but in most cases, the networks are incomplete due to data loss, so the clustering coefficient cannot be accurate enough. Hence, by applying the corrections, the Integrated degree-related clustering criterion obtained. Using the average clustering coefficient $\frac{1}{N}\sum_{i=1}^{N} C(k_v)$ instead of the clustering coefficient $C(k_v)$ increases the scalability of the Integrated degree-related clustering threshold. Besides, according to the preferential attachment mechanism, the probability of forming a new connection from another node to node i is proportional to the degree of node i; this means that neighbors with a higher degree may link with each other with higher possibility. Therefore, the mean degree of neighbor's node $(k_v)^r$ applied to merge the clustering coefficient into the degree related clustering threshold. Finally, the combination of neighbors mean degree and the two nodes common neighbors threshold $\sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} dc(V_z)$ has been used to increase the threshold accuracy. According to the improvements, using three indicators of node degree, common neighbors of the two nodes, and the average clustering coefficient results in realizing the clustering threshold related to the first degree as a fused criterion to calculate the similarity between two nodes.

$$L_{local} = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} DC_{ij}\left\|h_i^k - h_j^k\right\|_2^2 = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} DC_{ij}\left\|h_i - h_j\right\|_2^2 \tag{5}$$

As h_i indicates the $v_i$node feature vector and DC represents the local Similarity network, theL_{local} can also be displayed as follows:

$$L_{local} = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} DC_{ij}\left\|h_i - h_j\right\|_2^2 = Tr(H^T L_{DC} H) \tag{6}$$

In equation (6) $H \in R^{|v| \times d_i}$ indicates matrix network embedding representation. L is a loss function such as the Euclidean distance and Tr represents the trace of a matrix. Besides, $L_{DC} = D - DC_{ij}$ is the graph regularization matrix (Laplacian matrix) of the similarity matrix and $D = [d_{ij}] \in R^{|v| \times |v|}$ |is a diagonal matrix, thus $D_{i.i} = \sum_j DC_{ij}$is calculated.

ii. global similarity

In the proposed method, the Transitive Node Similarity Measure used to exploit the global feature [31]. Global methods, identify all path structures that are very difficult to be calculated for large social networks, while this model follows the shortest path between two nodes in the network, so it takes less time and complexity compared to global algorithms.

In the Transitive Node Similarity Measure, the length of the shortest path between the nodes and the nodes similarity in the neighborhood that make up that path are considered to calculate the similarity of the two

nodes in the network. The equation (7) and re-equation (8) shows the transfer node's similarity index method of calculation:

$$\mathrm{TNS_{ij}} = \begin{cases} 0. & \text{if there is no path between } \mathrm{v_i}.\mathrm{v_j} \\ \mathrm{sim}(\mathrm{v_i}.\mathrm{v_j}). & \text{if } \mathrm{v_i}.\mathrm{v_j} \text{ are nighborse} \\ \prod_{\mathrm{h=1}}^{\mathrm{k}} \mathrm{sim}(\mathrm{v_{p_h}}.\mathrm{v_{p_{h+1}}}). & \text{otherwise} \end{cases} \quad (7)$$

The global threshold between the two nodes is achievable through Equation (7); in such a way that, if there is no path between the nodes i and j, the threshold value is zero.

If the two nodes i and j are also direct neighbors, the global threshold value is calculated through the equation (8) and equation (9). Finally, if there exists a path between two nodes i and j but they are not directly in the neighborhood, the nodes located in the path between i and j are determined primarily, then the similarity of the two neighboring nodes in the path is calculated using the equation (8). Finally, the global threshold has resulted from the product of the similarities between the neighboring nodes that construct the path.

$$Sim(v_i.v_j) = \begin{cases} 0. if (v_i.v_j) \notin \varepsilon \wedge (v_i.v_j) \notin \varepsilon \\ \frac{1}{\deg(v_i)+\deg(v_j)-1}. otherwise \end{cases} \quad (8)$$

The path between two nodes in the network affects the information about their connection. Also, the shorter path results in the greater probability of a link creation between the two nodes. Therefore, the shortest path between two nodes is a good threshold for describing two nodes similarity. On the other hand, most pairs of nodes are separated by a small number of nodes in the network due to the theory of small world (both arbitrary persons on the planet are connected by 6 or fewer intermediaries). For this reason, sometimes the shortest path between two nodes does not perform well. Therefore, the transitive node similarity threshold was used. This criterion calculates the similarity of the node $\frac{1}{\deg(v_i)+\deg(v_j)-1}$ in our shortest path between the nodes by considering the degree of the node $Sim(v_i.v_j)$, so it has higher accuracy. According to equation (8), if two nodes are neighbors, the sum of nodes degree is calculated and reversed, and if they are not, the similarity degree is zero. Therefore, in order to achieve the global characteristics, the loss function is defined as equation (9).

$$L_{global} = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} TNS_{ij}\left\|h_i^k - h_j^k\right\|_2^2 = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} TNS_{ij}\left\|h_i - h_j\right\|_2^2 \quad (9)$$

As h_i indicates the $v_i$node feature vector and TNS represents Transitive Node Similarity Measure, the L_{global} can also be displayed as follows:

$$L_{global} = \sum_{i=1}^{|v|}\sum_{j=1}^{|v|} TNS_{ij}\left\|h_i - h_j\right\|_2^2 = Tr(H^T L_{TNS} H) \quad (10)$$

In equation (10) $H \in R^{|v|\times d_i}$ indicates Adjacency matrix of embedded network. L is a loss function such as the Euclidean distance and Tr represents the trace of matrix(the elements positioned on matrices main diagonal). Besides, $L_{DC} = D - TNS_{ij}$ is the graph regularization matrix (Laplacian matrix) of the similarity matrix and $D = [d_{ij}] \in R^{|v|\times|v|}$ |is a diagonal matrix, thus $D_{i.i} = \sum_j TNS_{ij}$ is calculated. Finally, both optimization goals are combined, and the overall optimization goal is defined as expressed in (11).

$$L_{mix} = (\alpha L_{local} + \beta L_{global}) \quad (11)$$

The α and β are hyper parameters that are responsible to control the balance between the two global similarity and local similarity thresholds. In fact, the α hyper parameter determines the effectiveness of the local criterion.

If α = 0, the predictive function is performed only through the global threshold. The β hyper parameter indicates the effect of the global standard effect on the prediction operation.

F. Optimization

After calculating the similarity between the two nodes, the loss function is optimized during a process to correct and update the neural network weights to achieve the minimum loss. Therefore, in this section, minimizing the L_{mix}function is the desire. The main step of calculating the partial derivative who's mathematically described in details, is shown in equation (12):

$$\frac{\partial L_{mix}}{\partial w^k} = \alpha\frac{\partial L_{local}}{\partial w^k} + \beta\frac{\partial L_{global}}{\partial w^k} \quad i = 1.2. \dots .K \quad (12)$$

In equation (12), the K parameter indicates the number of layers. Also w^k describe the K - layer's weight.

$$\frac{\partial L_{local}}{\partial \theta^k} = \frac{\partial L_{local}}{\partial H^k} \cdot \frac{\partial H^k}{\partial \theta^k} = 2[(L + L^T) \cdot H^k] \cdot \frac{\partial H^k}{\partial \theta^k} \quad (13)$$

Equation (13) represents the partial derivative of L_{local} so that $\theta^k \in (w^k.b^k)$ is defined as follows. The b^kalso represents the K-layer bias. H represents the adjacency matrix of the embedded network, and each layer's H according to the values of weight, bias, and H of the previous layer with the help of the back-propagation neural network is calculated as $H^k = (H^{k-1}W^k + b^k)$. Also, the partial derivative of L_{local} is also calculated in equation (14).

$$\frac{\partial L_{global}}{\partial \theta^k} = \frac{\partial L_{global}}{\partial H^k} \cdot \frac{\partial H^k}{\partial \theta^k}$$
$$= 2[(L + L^T) \cdot H^k] \cdot \frac{\partial H^k}{\partial \theta^k} \quad (14)$$

Evaluation

In evaluation part, 4 data sets have been tested using our proposed method to verify its validity. The result of the experiments has been compared with Laplacian, Node2vec, and GAE methods. Laplacian Method [32] is a non-linear dimensionality reduction algorithm, introduced based on spectral techniques and manifold learning.

Manifold learning is a powerful tool for reducing nonlinear dimension. The inherent parameters of the system, which are the main factor in distinguishing data from each other, are identified using this tool, and the entire set is placed on a manifold that represents the actual relationship of the parameters. In this way, the relationship between data is expressed in a space with a low-dimensional.

Node2vec [24] is a more advanced version of Deep Walk. The DeepWalk algorithm has limitations and cannot control the path. Instead of walking randomly, this algorithm searches for Breadth-first Sampling (BFS) and Depth-first Sampling (DFS). Adds to the description of the network structure in random paths.

GAE Method [33] is based on the Convolutional Neural Networks, it also uses a cryptographic and decoder to make nodes embedded. Its satisfactory results in reducing the dimensional reduction indicate the success of the convolutional neural network for obtaining graphical structural information.

A. Data Set

In this study, the proposed method was tested on 4 data sets. The general characteristics of this data set are summarized in Table 2.

Blog catalog [34]: A social network with 10312 bloggers and 333983 social relationships. Each node is represented by a blogger, and each edge indicates a relationship between two bloggers. The network has 39 different tags and shows the bloggers' interest in different topics. A blogger may have different tags.

Wikipedia Data Collection [35]: A common network of words. This network has 4,777 nodes, 184,812 edges and 40 different tags.

Cora Data Collection [36]: A subset of the entire citation data set. It consists of 7 sub-categories with 2708 scientific articles on machine learning and 5429 citation links between them.

Drug-drug interaction [37]: A comprehensive and accessible online database that contains accurate information about drugs and drug targets. This data set consists of 2191 drugs and includes 242027 drug-to-drug interactions.

Table 2: Specifications of the data set used in the experiment

Data collection name	Number of edges	Number of nodes
Blog catalog	333983	10312
Wikipedia	12765	2405
Cora	5278	2708
Drug-drug interaction	242027	2191

B. Performance evaluation

In order to evaluate the proposed method, the performance accuracy rate threshold and Fmeasure measurement rate and the area below the operational curve of Area under curve (AUC) receiver, have been used.

The Accuracy indicates the number of correctly categorized samples in relational form equation (15):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (15)$$

The Fmeasure Measurement Rate, which is designed to establish a balanced relationship between Precision validity and recall, is defined according to equation (16):

$$\text{Fmeasure} = 2\left(\frac{Precision \cdot Recall}{Precision+Recall}\right) \quad (16)$$

TP and TF are examples that have been correctly identified by the model during the evaluation phase; TP represents the interaction patterns and TF represents the non-interaction patterns.

FNs are non-interactive pairs that are mistakenly known as interactions, and FPs are data that are incorrectly known as non-interactions. The AUC threshold indicates the desperation of positive and negative values. A high value means that the model separates the positive and negative values of the negative sample well, and the low value means that the model has randomly worked.

C. Experimental tests

The proposed method is applied to Cora, Wikipedia, Blog catalog, Drug-drug interaction separately and its results are compared with Laplacian, Node2vec and GAE methods. The relevant tables show the threshold performance, ACC accuracy rate, F1 measurement rate, and AUC operational sub-area . Also, in order to make the specific assessment possible, the best results are displayed in bold. The results of applying the proposed method on the Cora data set are shown in Table 2.

As obviously shown Table 3, the ACC and AUC threshold values are higher than the other three proposed methods, while the F1 GAE method is higher than the other methods.

Table 3: Cora data evaluation results

Type	Method	F1	ACC	AUC
Based on matrix factoring	Laplacian	0.589	0.588	0.621
Random walk-based	Node2vec	0.555	0.557	0.584
Neural network	GAE	**0.611**	0.568	0.606
Neural network	Proposed method	0.562	**0.620**	**0.645**

Table 4 shows the evaluation results on the Wikipedia data set. The accuracy of the proposed method on the Wikipedia data set has increased by about 7% compared to the best method, the Node2vec method, and the highest values of the two F1 and AUC thresholds on this dataset belong to the proposed method.

Table 4: Wikipedia data evaluation results

Type	Method	F1	ACC	AUC
Based on matrix factoring	Laplacian	0.785	0.790	0.664
Random walk-based	Node2vec	0.857	0.860	0.932
Neural network	GAE	0.779	0.744	0.816
Neural network	Proposed method	**0.875**	**0.875**	**0.943**

The evaluation of the proposed method is reported on the Blog catalog data set according to Table 5. As can be seen, the results reported on the Blog catalog data collection shows an improvement in the performance of the proposed method compared to the other three methods, and as can be seen, all three performance criteria have increased significantly.

Table 5: Blog catalog data evaluation results

Type	Method	F1	ACC	AUC
Based on matrix factoring	Laplacian	0.785	0.790	0.864
Random walk-based	Node2vec	0.857	0.860	0.932
Neural network	GAE	0.779	0.744	0.816
Neural network	Proposed method	**0.875**	**0.875**	**0.943**

The fourth experimental dataset is Drug-drug interaction, the results of which are given in Table 6. The values in Table 6 show an increase in all three performance measurement criteria. The proposed method has been able to increase the accuracy rate by about 6% compared to the best method on the Drug-drug interaction dataset.

Table 6: Drug-drug interaction data evaluation results

Type	Method	F1	ACC	AUC
Based on matrix factoring	Laplacian	0.727	0.718	0.797
Random walk-based	Node2vec	0.814	0.814	0.898
Neural network	GAE	0.784	0.740	0.835
Neural network	Proposed method	**0.847**	**0.845**	**0.923**

Results and Discussion

In this section, we further examine the superiority of the proposed method towards the other baselines on experimental networks. From the differences among four algorithms summarized in Table 7, we can see that the proposed method has used all three techniques of local similarity, global similarity, and deep neural network to describe the nodes similarity. Therefore, it presents better performance than other compared algorithms.

Table 7: A summary of the differences among four algorithms

Type	Laplacian	Node2vec	GAE	Proposed method
Local similarity	✓	✓		✓
global similarity		✓	✓	✓
Deep neural network			✓	✓

■ Local similarities are using a small part of the network, so they have high speed and scalability. In the proposed method, the degree-related clustering used as a local threshold, which is a compound criterion of several structural features and able to improve the low accuracy weakness that exists in the local criteria.

■ Global similarities exploit the entire network structure, thus provides a more comprehensive and accurate description of network nodes. The similarity index of the transfer node, used as the global similarity

in the proposed method, which is calculated based on the shortest path between the two nodes. So this threshold could improve the weakness of the high computational complexity in the global similarity.

■ The depthless or superficial embedding methods often have a linear function for mapping the main network in the embedding functional space and have limitations. The proposed method uses an encoder-decoder, so it uses neural networks for data representation. Generally, deep neural networks are fitting in modeling complex nonlinear phenomena.

Also, the implementation results of the proposed method on Drug-drug interaction, Blogcatalog, Cora, and Wikipedia datasets indicate better performance on more solid or complete datasets. The highest accuracy resulted in the Blogcatalog dataset, and the lowest accuracy belonged to the Cora dataset.

Conclusion

A dataset contains tens to hundreds of features. However, not all features are meaningful for predictive linking algorithms. To this end, this paper presents a method of embedding a node based on a deep learning model to automatically extract useful information about the structure and characteristics of the graph. So far, similar solutions have been proposed, but most of them use only direct neighbors and second-class neighbors as related nodes. In the real world, however, networks are large-scale, irregular, and heterogeneous, so both the local structure and their global structure are important, as is the case with the proposed method. The proposed method could successfully provide a strong threshold for evaluating the similarity of nodes, by coupling local and global network characteristics that resulted in the accuracy improvement of the algorithm. The combination of global and local similarity thresholds makes the proposed algorithm applicable to a large dataset. Also, choosing the right global standard has made it possible for the algorithm to remain acceptable while increasing the accuracy, speed, and complexity, and ultimately preventing the loss of important information due to the use of a deep neural network approach.

Author Contributions

This paper is the result of F. Mirmousavi's MSc thesis supervised by S. kianian.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Conflict of Interest

There is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy have been completely observed by the authors.

Abbreviations

$\boldsymbol{x^t}$	the input data
$\boldsymbol{h^t}$	reconstructed data
$\boldsymbol{K}$	number of layers
$\boldsymbol{w^k}$	the k-th layer weight matrix
$\boldsymbol{\theta^k}$	the overall parameter
$\boldsymbol{b^k}$	the k-th layer biases
$\boldsymbol{L_{local}}$	local loss function
$\boldsymbol{DC_{ij}}$	Integrated degree-related clustering
$\boldsymbol{dc}$	clustering related to the degree
N	number of vertexes
$\boldsymbol{C(k_v)}$	clustering coefficient of node v with a degree of k
$\boldsymbol{k_v}$	degree of node v's
$\boldsymbol{r}$	assortative coefficient
$\boldsymbol{Tr}$	represents the trace of a matrix
$\boldsymbol{L_{DC}}$	Laplacian matrix DC
D	diagonal matrix
$\boldsymbol{L_{global}}$	global loss function
$\boldsymbol{TNS}$	Transitive Node Similarity
$\boldsymbol{deg(v_i)}$	degree of node v's
$\boldsymbol{H}$	network embedding representation matrix
$\boldsymbol{\alpha}$	tradeoff factors among the objective functionL_{local}
$\boldsymbol{\beta}$	tradeoff factors among the objective L_{global}
ACC	accuracy rate
$F1$	Assessment rate
AUC	operational sub-are.

References

[1] G. Tsoumakas, I. Katakis, "Multi-label classification: An overview," International Journal of Data Warehousing and Mining (IJDWM), 3(3): 1-13, 2007.

[2] L. Lü, T. Zhou, "Link prediction in complex networks: A survey," Physica A: statistical mechanics and its applications, 390(6), 2011.

[3] S. Fortunato, Community detection in graphs. Physics reports, 2010.

[4] L. Lü, M. Medo, C.H. Yeung, Y.C. Zhang, Z.K. Zhang, T. Zhou, Recommender systems. Physics reports, 519(1): 1-49, 2012.

[5] S. Wang, Q.Wang, M. Gong. "Multi-Task Learning Based Network Embedding," Frontiers in Neuroscience, 2020.

[6] E.Airoldi "Mixed membership block models for relational data with application to protein-protein interactions in Proc. International Biometric Society-ENAR Annual Meetings, 2006.

[7] Z. Huang, X. Li, H. Chen, "Link prediction approach to collaborative filtering," in Proc.5th ACM/IEEE-CS Joint Conference on Digital Libraries, NK,: 141–142.

[8] M. Al Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security 30: 798-805, 2006.

[9] Z. Huang, D.K. Lin, "The time-series link prediction problem with applications in communication surveillance," INFORMS Journal on Computing, 21(2): 286-303, 2009.

[10] L. A. Adamic, E. Adar, Friends and neighbors on the web, Social 1networks, 25(3):230 , 2003.

[11] P. Jaccard, Etude comparative de la distribution florale dans une portiondes Alpes et du Jura, Impr. Corbaz, 1901.

[12] A, Clauset, C. Moore, M.E. Newman, "Hierarchical structure and the prediction of missing links in networks," Nature, 453: 98-101, 2008.

[13] H. C. White, S. A. Boorman, R. L. Breiger, "Social structure from multiple networks," i. blockmodels of roles and positions, America journal of sociology780.

[14] N. Friedman, L. Getoor, D. Koller, A. Pfeffer, "Learning probabilistic relational models," In IJCAI, 99: 1300-1309, 1999.

[15] D. Heckerman, C. Meek, D. Koller, "Probabilistic entity-relationship models, PRMs, and plate models," Introduction to statistical relational learning, 201-238, 2007.

[16] P. Goyal, E. Ferrara, "Graph embedding techniques, applications, and performance: A survey, "Knowledge-Based Systems, 151:78-94, 2018.

[17] W. L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, "arXiv preprint," arXiv:1709.05584, 2017.

[18] X. Yue, et al., "Graph embedding on biomedical networks: methods, applications and evaluations," Bioinformatics, 36(4): 1241-1251, 2020.

[19] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, A. J. Smola, "Distributed large-scale natural graph factorization," in: Proce 22nd international conference on World Wide Web, ACM, :37–48, 2013.

[20] D. Wang, P. Cui, W. Zhu, "Structural deep network embedding," in Proce. 22nd International Conference on Knowledge Discovery and Data Mining, ACM,:1225–12, 2016.

[21] C. Hongyun, V.W. Zheng, K. Chen-Chuan Chang. "A comprehensive survey of graph embedding: Problems, techniques, and applications." IEEE Transactions on Knowledge and Data Engineering 3(2018):1616-1637, 2018.

[22] Ou, Mingdong, et al., "Asymmetric transitivity preserving graph embedding," in Proce. 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.

[23] B. Perozzi, R. Al-Rfou, S. Skiena, "August. Deepwalk: "Online Learning Of Social Representations," In Proce 20th ACM SIGKDD international conference on Knowledge discovery and data mining,: 701-710, 2014.

[24] A. Grover, J. Leskovec, "node2vec: Scalable feature learning for networks," in Proce 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016.

[25] C. Peng, et al., "A survey on network embedding," IEEE Transactions on Knowledge and Data Engineering, 31(5): 833-852, 2018.

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, "Line: Large-scale Information Network Embedding," in Proce. 24th International Conference on World Wide Web,: 1067-1077, 2015.

[27] T.N. Kipf, M. Welling, "Semi-supervised Classification With Graph Convolutional networks" arXiv,: 1-11, 2016.

[28] Y. Wang, Y. Hongxun, S. Zhao, "Auto-encoder based dimensionality reduction," Neurocomputing, 184: 232-242, 2016.

[29] A. Papadimitriou, "Panagiotis Symeonidis, and Yannis Manolopoulos," Friendlink: link prediction in social networks via bounded local path traversal," 2011 International Conference on Computational Aspects of Social Networks (CASoN). IEEE, 2011.

[30] C. Xing, et al., "The application of degree related clustering coefficient in estimating the link predictability and predicting missing links of networks," Chaos: An Interdisciplinary Journal of NonlinearScience, 29(5):053135, 2019.

[31] S. Panagiotis, E. Tiakas, Y. Manoulos, "Transitive node similarity for link prediction in social networks with positive and negative links," in Proce. The fourth ACM conference on Recommender systems, 2010.

[32] M. Belkin, P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," In Advances in neural information processing systems,: 585591, 2002.

[33] T.N. Kipf, M. Welling, "Variational graph auto-encoders," arXiv preprint arXiv:1611.07308, 2016.

[34] L. Tang, H. Liu, "Relational learning via latent social dimensions," In Proce 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, :817–826,Paris,France,Jun.2009.

[35] http://www.mattmahoney.net/dc/textdata

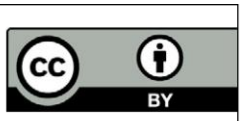
[36] T. Yang, R. Jin, Y. Chi, Z. Shenghuo, "Combining link and content for community detection: a discriminative approach," In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,: 927–936,Paris,France,Jun.2009.

[37] D.S. Wishart, Y.D. Feunang, A.C. Guo, et al., "Drugbank 5.0: a major update to the drugbank database for 2018," Nucleic acids research, 46(D1), D1074–D1082, 2017.

Biographies

Seyedeh Farzaneh Mirmousavi received her B.Sc. degree in computer Engineering (2015) from Arak University, she is a Master's student of Computer Engineering in Shahid Rajaee student Training University and her research interests are complex networks, machine learning, and artificial intelligence.

Sahar Kianian received her B.Sc. degree in computer Engineering (2007) from razi University, also M.Sc. and Ph.D. degrees in computer Engineering from Isfahan University (2010 and 2016, respectively). She is an assistant professor of computer engineering at Shahid Rajaee University. her research interests are the application of algorithms, machine learning and data science to complex networks, focuses on protein interactions, connections of neurons and relationships among people. Applications include disease prediction, drug discovery, event detection and tracking, recommendation system, web mining and social influence mining.



How to cite this paper:
S.F. Mirmousavi, S. Kianian "Link Prediction using network embedding Based on global similarity" Journal of Electrical and Computer Engineering Innovations, 8(1): 97-108, 2020.

DOI: 10.22061/JECEI.2020.7135.359

URL: http://jecei.sru.ac.ir/article_1430.html