mathematics

MDPI

Article

Graph Embedding Method Based on Biased Walking for Link Prediction

Mingshuo Nie, Dongming Chen * and Dongqi Wang

Software College, Northeastern University, Shenyang 110169, China
* Correspondence: chendm@mail.neu.edu.cn

Abstract: Link prediction is an essential and challenging problem in research on complex networks, which can provide research tools and theoretical supports for the formation and evolutionary mechanisms of networks. Existing graph representation learning methods based on random walks usually ignore the influence of local network topology on the transition probability of walking nodes when predicting the existence of links, and the sampling strategy of walking nodes during random walks is uncontrolled, which leads to the inability of these methods to effectively learn high-quality node vectors to solve the link prediction problem. To address the above challenges, we propose a novel graph embedding method for link prediction. Specifically, we analyze the evolution mechanism of links based on triadic closure theory and use the network clustering coefficient to represent the aggregation ability of the network's local structure, and this adaptive definition of the aggregation ability of the local structure enables control of the walking strategy of nodes in the random walking process. Finally, node embedding generated based on biased walking paths is employed to solve the link prediction problem. Extensive experiments and analyses show that the TCW algorithm provides high accuracy across a diverse set of datasets.

Keywords: link prediction; biased walking; triadic closure theory; network clustering coefficient

MSC: 05C81; 91D30

check for updates

Citation: Nie, M.; Chen, D.; Wang, D. Graph Embedding Method Based on Biased Walking for Link Prediction. *Mathematics* **2022**, *10*, 3778. https://doi.org/10.3390/math10203778

Academic Editors: Jun Lin, Jing Shi and Fengjie Xie

Received: 22 August 2022
Accepted: 9 October 2022
Published: 13 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Complex network analysis is an important means of interdisciplinary research at present, attracting more and more researchers' attention. Link prediction is an important part of complex network analysis [1], which aims at predicting missing, spurious, or new links in the current structure of the network by using the structure information and attribute information of a given network [2]. Link prediction plays an important role in social network analysis [3,4], network reconstruction [5], and network evolution mechanisms [6–8]. In addition, link prediction in the theoretical analysis assists in comprehending the mechanism of propagation and diffusion of information [9]. According to the in-depth study of link prediction by researchers, we conclude that the graph embedding methods based on the random walk are the main methods to deal with the link prediction problem. A large number of researchers employ these methods to work out link prediction efficiently and accurately. Lv et al. [10] proposed that the degrees of the current node and its neighbor nodes jointly determine the transition probability in the process of walking. Zhou et al. [11] also employed the local features of nodes to define the transition probability of walking nodes and realized an improved random walk method combined with the graph embedding method. Nasiri et al. [12] proposed a random walk-based solution for the problem of link prediction in multiplexed networks. The multiplex biased local random walk performs better than the state-of-the-art methods of link prediction and corresponding unbiased case and improves prediction accuracy.

The methods based on random walks to deal with the problem of link prediction are highly interpretable. When solving link prediction with random walk-based graph

embedding methods, researchers commonly predefine the walking strategy of the walking nodes on the network structure. The walking nodes aggregate information on the network structure according to the fixed walking strategy to implement the embedding of the network structure into the low-dimensional vector space. However, most of these only consider the influence of the degree of the nodes on the transition probability, while ignoring the influence of the network's local structural features on the walking strategy, which leads to the random walk-based graph embedding methods failing to achieve high accuracy when solving link prediction problems. A large number of existing research shows that mining and analyzing the local topology of the network, i.e., subgraph pattern mining, can optimize the performance of most tasks on network information analysis [13–16].

To address the above challenges, we propose a graph embedding method based on biased walking for link prediction (**T**triadic **C**losure for **W**alk, TCW). The proposed algorithm can dynamically define walking policies based on the local clustering ability to walk nodes and solve the problem of link prediction with biased walking. The triadic closure theory proposes that the creation of new links in a network often tends to build new open triangle structures or closed triangle structures in the network [17–19]. Whereas the clustering coefficient of a network is based on the triangularity of nodes and measures the degree of aggregation of nodes in the network as a whole or locally, real-world entities tend to build relatively tight network clusters, and links are more likely to be established between nodes in a tightly connected cluster than randomly. According to the analysis of triadic closure theory and the definition of network clustering coefficients, the sampling range of the walking nodes during walking can be determined by the current local topology of the walking node.

The proposed algorithm can capture paths with higher similarity between nodes for the problem of link prediction, enabling the definition of biased walking policies for the specific task. Moreover, the automatic definition of control parameters can reduce the search space for manual specification of controllable parameters and effectively improve the prediction accuracy of the algorithm. Based on this controlled and biased walking process, we can explain how to determine the depth and breadth of the walking range based on the network's local structural features. Experimental results on real-world networks show that the TCW algorithm enables adaptive capture of the network local structure features of walking nodes and applies these features to the link prediction problem. However, the TCW algorithm has higher complexity, and we believe that it has excellent performance in the scenario of balancing efficiency and accuracy. In addition, this biased walking embedding converts the structure of the original network into tighter vectors, and network visualization experiments are conducted with the TCW algorithm and existing network representation learning algorithms.

Overall, our paper makes the following contributions:

- We propose a graph embedding method based on biased walking for link prediction. The algorithm can dynamically define the walking policies of nodes in the network based on the local clustering coefficients of the walking nodes, and solve the problem of link prediction by biased walking.
- We propose a method for defining control parameters based on triadic closure theory and network clustering coefficient definition to automate the setting of control parameters.
- The application of ternary closure theory to the link prediction task is implemented, and the proposed algorithm demonstrates that the network local structure information can satisfy the need for network structure information for the link prediction task.
- The experimental results on five real-world network datasets show that the TCW algorithm can significantly improve the accuracy of link prediction in most cases, while the node vectors generated by the algorithm better match the original network structure characteristics.

The rest of the paper is structured as follows. In Section 2, we briefly review related work proposed by researchers to solve the problem of link prediction. We formulaically define the algorithm execution process for solving the problem of link prediction using a graph embedding method based on random walking in Section 3. In Section 4, we present the motivation for the TCW algorithm proposal, the definition of network clustering coefficients, and the technical details of the TCW algorithm, and evaluate the performance of the TCW algorithm in link prediction experiments and network visualization experiments. Finally, we conclude with a discussion of the TCW algorithm and present some promising future work.

2. Related Work

Network theory is an important tool for describing and analyzing complex systems throughout a variety of disciplines [20]. Link prediction is an important area of research in the field of network data mining, which aims to mine the missing links from a given network or predict the links that may be generated in the future [2]. These similarity-based methods are one of the most basic and simple link prediction algorithms, and most of these heuristics have been shown to work well for specific types of networks, while not being effective for other types of networks [21]. This similarity-based method typically employs the global, quasi-local, and global topology of the network for calculating the similarity of nodes, e.g., common neighbors (CN) [22] directly calculates the number of common neighbors, Adamic–Adar (AA) [23] assigns weight values to improve the CN based on the degree of the nodes, and CAR-based common neighbors (CAR) [24] considers not only the common neighbors of a node pair but also the local clustering coefficients between node pairs.

More definitions of similarity metrics based on the local structure of the network are shown in Table 1. Given a network $G = (V, E)$, $\Gamma(i)$ and $\Gamma(j)$ denote the number of neighbors of node i and node j, respectively, k_i denotes the degree of node i, and $C(i)$ denotes the local clustering coefficient of node i.

Table 1. The definitions of similarity metrics based on network local structure.

Similarity Metrics	Definitions
Salton [25]	$S(i,j)_{Salton} = \frac{\lvert\Gamma(i)\cap\Gamma(j)\rvert}{\sqrt{k_i \times k_j}}$
Sørensen [23]	$S(i,j)_{Srensen} = \frac{2\lvert\Gamma(i)\cap\Gamma(j)\rvert}{k_i+k_j}$
Hub Promoted [26]	$S(i,j)_{HPI} = \frac{\lvert\Gamma(i)\cap\Gamma(j)\rvert}{min\{k_i,k_j\}}$
Hub Depressed [26]	$S(i,j)_{HDI} = \frac{\lvert\Gamma(i)\cap\Gamma(j)\rvert}{max\{k_i,k_j\}}$
Leicht–Holme–Newman Local [27]	$S(i,j)_{LHNL} = \frac{\lvert\Gamma(i)\cap\Gamma(j)\rvert}{k_i \times k_j}$
Node Clustering Coefficient [24]	$S(i,j)_{CCLP} = \sum_{z\in\Gamma(i)\cap\Gamma(j)} C(z)$

In addition, embedding-based methods are used to solve the problem of high-dimensional data in graph structures. Graph embedding is considered a dimensionality reduction technique that maps the graph structure data into a low-dimensional vector space to preserve the neighborhood structure of nodes. Several graph embedding techniques have been proposed and successfully applied to link prediction problems [28,29]. Guo et al. [30] proposed the multiscale variational graph autoencoder, which is an algorithm that learns multiple sets of low-dimensional vectors of different dimensions to represent the mixed probability distribution of the original graph data by a graph encoder with multiple sampling in each dimension. Zhou et al. [31] proposed a novel method of symmetric neighborhood convolution to perform the representation learning task in bipartite networks, and this algorithm achieves high prediction performance in the link prediction problem.

3. Preliminaries

Given a network $G = (V, E)$, where $|V|$ denotes the number of nodes and $|E|$ denotes the number of links in network G, the TCW algorithm that we execute in a given network G to solve the problem of link prediction refers to predicting the set of missing links in network G and the set of links that have not yet been generated in network G based on information such as the set of nodes V and the structural features in network G.

This paper proposes a vector representation of nodes in a network based on a biased walking sampling strategy, and the process of obtaining a vector representation of a link can be formally represented as follows. In the given network $G = (V, E)$, the initialized n nodes start walking in parallel to obtain a truncated walking path of the specified-length $Walkpath_i = (v_1, v_2, v_3, \cdots, v_m)$, $i \in (0, n)$, $m \leq n$. We consider the sequence of walking nodes obtained after the execution of the walking process is completed as a sequence of words in the natural language-processing domain, and use the word embedding model to process the node walking sequence to obtain a vector representation of the nodes in a given network. The process for link prediction based on the vector representation of nodes can be defined as follows. According to the vector representations $f(u)$ and $f(v)$ of node u and node v in network G, we calculate the vector representation $g : V \times V \to \mathbb{R}^d$ of the link with the binary operator O, where d is the dimension of the vector, and finally, according to the obtained vector representation of the link, we classify the link with the classification model of the vector, which results in the prediction of the existence of the link.

4. Methodology

The main idea of the proposed algorithm is to define control parameters based on the network clustering coefficients, which automate the biased walking and employ the results of biased walking in a network embedding model to obtain a representation of link vectors for link prediction tasks. The conceptual framework of TCW is shown in Figure 1.

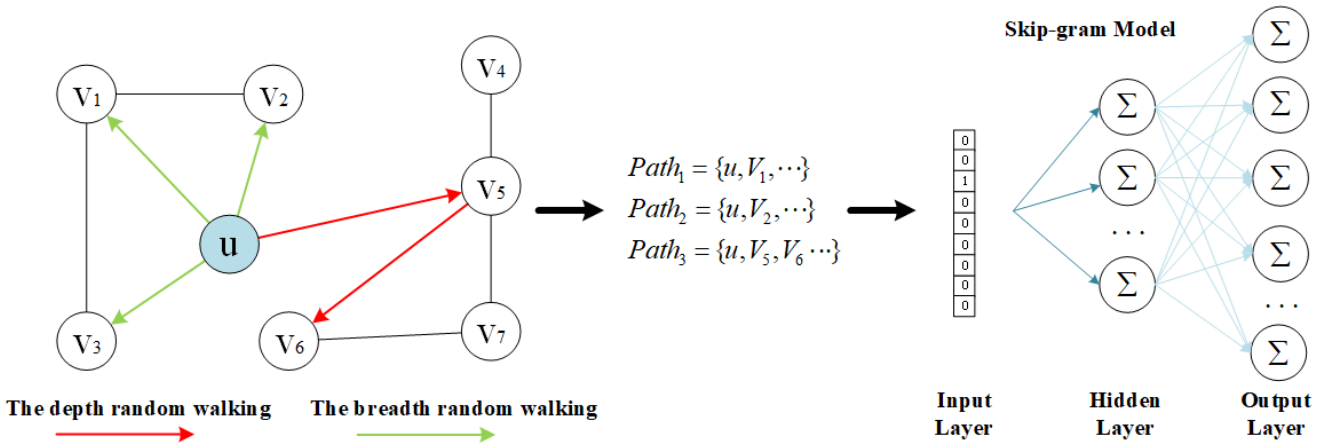


Figure 1. Conceptual framework of TCW.

4.1. Triadic Closure Theory and Network Clustering Coefficients

The clustering coefficient is defined as a measure of neighborhood connectivity. This metric can denote the degree of completeness of a node's connectivity in its neighborhood. Triadic closure theory is a key concept in network clustering coefficients. Luce and Perry [32] first proposed the concept of global clustering coefficients, which are defined on the basis of triadic closure theory in network analysis, employing the number of closed triangular structures and the overall number of triangular structures in the network. Watts et al. employed local clustering coefficients for determining whether a given graph can be described as a small-world network [33]. The local clustering coefficient of a node in a given network reflects the ability to construct a complete graph by aggregating the

neighborhoods of that node. The average clustering coefficient is the average of the local clustering coefficients of all nodes in the network.

Given a network $G = (V, E)$ with $|V|$ node, for any node $u \in V$, deg_u denotes the degree of node u, Γ_c denotes the number of closed triangles in the network, Γ denotes the number of all triangles in the network, and $deg_u(deg_u - 1)/2$ denotes the total number of neighboring node-pairs of node u.

Definition 1. *Global clustering coefficients. The global clustering coefficient c for a given network can be defined as Equation (1).*

$$C = \frac{\Gamma_c}{\Gamma}. \tag{1}$$

Definition 2. *Local clustering coefficients. The local clustering coefficient $C(u)$ for node u in a given network $G = (V, E)$ is defined as Equation (2).*

$$C(u) = \frac{2\Gamma(u)}{deg_u(deg_u - 1)}. \tag{2}$$

Definition 3. *Average clustering coefficient. The average clustering coefficient C for a given network $G = (V, E)$ is defined as Equation (3) [33].*

$$C = \frac{1}{|V|} \sum_{u \in V} C(u). \tag{3}$$

The time complexity of computing the clustering coefficients depends on the number of nodes and links of the given network, as well as the second-order moments of the degree distribution of that network. As the size of the network grows, the second-order moments of the degree distribution of the network gradually converge to a finite constant value. In order to efficiently obtain a representation of the clustering ability of the network, the approximate representation for the clustering ability of the network can be obtained by employing the approximate method of computing the clustering coefficients, thus reducing the complexity of computing the clustering coefficients.

4.2. Graph Embedding Method Based on Biased Walking for Link Prediction

Many networks in the real world have higher neighbor connectivity than random networks of the same scale. According to the formula and characteristics of the network clustering coefficient, the network clustering coefficient can indicate the clustering ability of a network. In triadic closure theory, the second-order neighbors of a node are more likely to establish a link with the current node. Taking the network local clustering coefficients as the parameter to control random walking allows the random walking process to be controlled in the set of nodes that are likely to establish a link, and therefore we can leverage these random walking paths to obtain the vector representation. The mechanism for generation of vector representations of nodes will transform from being generated based on a random process to a greater tendency to generate vectors that apply the link prediction task, leading to more accurate link prediction results. The way to control the random walk process based on the local clustering coefficients of the walking nodes also improves the controllability of the model. Different random walking policies have different features. Neighbor sampling based on breadth random walking will focus more on random walking within strongly connected neighbors of the given node, and using this sampling method will provide a more tightly connected walking path to the given node, the obtained information about the neighbors will reflect the local close structure of the network, and the nodes obtained by this walking method have some recurrence relationship with each other; therefore, these nodes are more likely to obtain similar vector representation. Neighbor sampling based on

deep random walking will focus more on random walking outside the strongly connected neighbors of a given node, thereby further extending the sampling scope of the nodes and obtaining neighbor information that better reflects the relatedness of the nodes, which reveals the structural relationship between the nodes and their neighbors.

The transfer probabilities of depth random walking and breadth random walking can be controlled in a balanced way to obtain walking paths from the local close structure of the nodes and the more extensive nonlocal close structure, thereby improving the accuracy of neighbor sampling and obtaining more efficient walking paths. The depth random walking and breadth random walking processes of the nodes are shown in Figure 2.

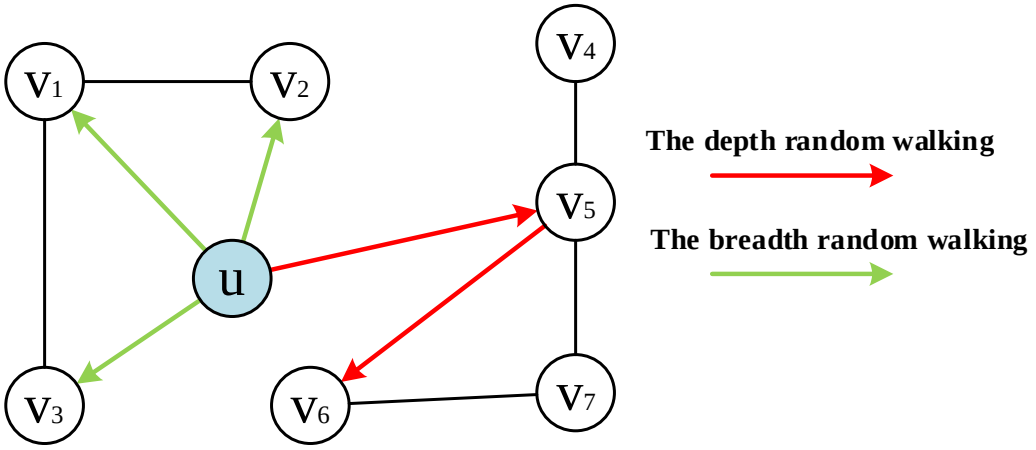


Figure 2. Depth random walking and breadth random walking of nodes.

This paper proposes the utilization of triadic closure theory to control the random walking process. Specifically, the selection of a deep random walking policy or a breadth random walking policy is controlled according to the local clustering coefficients of the network to automate the determination of nodes carrying out a breadth or a depth walking. This automatic setting of control parameters allows for more efficient exploration of the network structure and the obtaining of node vectors for application to link prediction tasks. The walk control parameters, the transfer probability of the nodes and the nonnormalized transfer probability of the algorithms in this paper are defined in Equations (4)–(7).

Definition 4. *Walking control parameters. Given a network $G = (V, E)$ of $|V|$ nodes, where nodes $u \in V$, α_r denote the probability that a walking node will revisit the previous node immediately in the process of taking a random walking tour. α is the parameter that controls the trade-off between the breadth of walking and the depth of walking.*

$$\alpha_r = 1 + C(u) \tag{4}$$

$$\alpha = \begin{cases} C(u) & if\ C(u) \neq 0 \\ 1 + \frac{1}{|V|} \sum\limits_{u \in V} C(u) & if\ C(u) = 0 \end{cases} \tag{5}$$

According to the definition of the control parameter, the transfer probability from walking node u to node v during the walking process can be defined as $P(u \in i | v \in j)$.

Definition 5. *Transfer probability. Given a network $G = (V, E)$ with $|V|$ nodes, the transfer probability $P(u \in i | v \in j)$ of a node is defined as:*

$$P(u \in i | v \in j) = \begin{cases} \frac{p_{i,j}}{Z} & if\ edge(i,j) \in E \\ 0 & if\ edge(i,j) \notin E \end{cases} \tag{6}$$

where Z denotes the normalization constant and $p_{i,j}$ denotes the nonnormalized transfer probability of node i and node j, which is determined by the given control parameters α and α_r.

Definition 6. *Nonnormalized transfer probability. Given a network $G = (V, E)$ with $|V|$ nodes, the nonnormalized transfer probability is defined as Equation (7).*

$$p_{i,j} = \begin{cases} \frac{w_{i,j}}{\alpha_r} & if\ d(i,j) = 0 \\ w_{i,j} & if\ d(i,j) = 1 \\ \frac{w_{i,j}}{\alpha} & if\ d(i,j) = 2 \end{cases} \tag{7}$$

where $w_{i,j}$ denotes the weight of the link between node $i \in u$ and node $j \in v$, and $d(i,j)$ denotes the shortest path distance between node b and node v. When $d(i,j) = 0$, it means that the next walking node of a given node is itself; when $d(i,j) = 1$, it means that the next walking node of a given node is an adjacent node of that node, i.e., a first-order neighbor node; when $d(i,j) = 2$, it means that the next walking node of a given node is a disconnected node of that node, i.e., a non-first-order neighbor node.

According to Equations (4)–(7), the algorithm employs the random walking method to obtain the walking paths if both α and α_r are set to 1. The transfer probability of the walking nodes is only correlated with the link weights $w_{i,j}$ between the nodes. In the algorithm proposed in this paper, first, α_r is set to a value greater than 1 initially to ensure that the possibility of the walking node visiting the node that has just been visited during the walking process is small, while the difference between α_r and 1 is determined according to the local clustering ability of the neighboring structure of the network where the walking node is located. This method of setting the parameter value dynamically enables the walking node not to visit the node that has been visited during the sampling process as much as possible. The more aggregation capability of a node's neighbor structure denotes that the walking node is more controlled in selecting the next walking node, and converting this controllability into transferability of the node to unvisited nodes will serve to capture the larger scope of sampling.

The control parameter α is used for the trade-off between depth and breadth walking for walking nodes. The probability of a walking node visiting a node far from the current node is determined according to the local clustering coefficient of the current node when the local clustering coefficient of the node is not 0. According to features of local clustering coefficients with a range between 0 and 1, the larger the local clustering coefficient of the current walking node, the more clustering power of the neighbor structure of the current node, then the larger the value of a, the less probability that the walking node would sample nodes far from itself, and the less controllable the walking range would be far from the current node. However, the smaller the local clustering coefficient, the weaker the local clustering ability of the neighbor structure of the current node, and the smaller the value of α, the more the node added to the walking paths will tend to be more distant from the distal neighbor node of the current walking node. When the local clustering coefficient of the node is set to 0, it means that the neighbors' structure of the current node lacks clustering ability. Therefore, this paper proposes to set α to a value greater than 1 to control the sampling scope of random walking in the neighbor of the current walking node, and the walking policy of the walking node is set to breadth walking, which ensures that the obtained walking paths can obtain more complete information of the local structure of the network. This setting of parameters prevents the problem of missing information due to the nodes choosing to walk away from the nodes several times for lack of clustering ability, leading to insufficient sampling of local information for the network.

The algorithm proposed in this paper selects initial nodes in the network for biased random walking in parallel and randomly to obtain biased walking paths for the link prediction tasks, and the obtained walking paths are used as input data for vectorized representation employing the word embedding algorithm in the field of natural language

processing, i.e., the sequences of nodes in the walking paths are treated as sequences of words in a sentence, and the network structure is mapped into the low-dimensional vector space employing the word embedding model. This parallel, asynchronous representation of node vectors improves the computational efficiency and scalability of the TCW algorithm effectively.

In this paper, alias sampling is also employed to improve the computational efficiency of the vector representation [34]. This sampling method allows the sampling of nodes in the network according to a specified probability distribution within the time complexity $O(1)$. Stochastic gradient descent (SGD) is employed as the optimization algorithm for the training process of the algorithm. The pseudocode of the TCW algorithm proposed in this paper is shown in Algorithms 1 and 2.

Algorithm 1. **Triadic Closure for Walk**

Input: Network $G = (V, E)$, Dimension of vectors d, Walking number of nodes N, Length of walking paths L, Window size in the word embedding model l.
Output: Mapping function f_{emb}.
Initialize $Walks = []$
for $i = 1$ **to** N **do**
  for all nodes $u \in V$ **do**
    $walk = TCPWalk(G, u, L)$
    Append $walk$ to $Walks$
  end for
end for
$f_{emb} = StochasticGradientDescent(l, d, Walks)$
return f_{emb}

Algorithm 2. **TCPWalk**

Input: Network $G = (V, E)$, Start nodes of the walking paths $u \in V$, Length of walking paths L.
Output: Walking paths $walk$
Append u to $walk$
for $j = 1$ **to** L **do**
  Initialize $node$ as the last node in $walk$
  Initialize $Nodes = []$
  $\alpha_r = 1 + LocalClusteringCoefficient(node)$
  if $LocalClusteringCoefficient(node) = 0$ **then**
    $\alpha = 1 + AverageClusteringCoefficient(G)$
  else
    $\alpha = LocalClusteringCoefficient(node)$
  end if
  Append $Neighbours(node, G)$ to $Nodes$
  Choose v from $Nodes$ according to $AliasSample(Nodes, \alpha, \alpha_r)$
  Append v to $walk$
end for
return $walk$

The loss function of the vector representation learning model of nodes in this paper is defined according to the loss function in the Node2Vec [34]. It is assumed that the given node and the neighbor nodes are independent of each other and there is symmetry in the feature space between the neighbor nodes and the given node when calculating the node similarity. The loss function is defined in Equations (8) and (9).

$$\max_f \sum_{u \in V} \left[-logZ_u + \sum_{n_i \in N(u)} f(n_i) \cdot f(u) \right], \tag{8}$$

$$Z_u = \sum_{v \in V} exp(f(u) \cdot f(v)). \tag{9}$$

5. Experiments

The experiments in this paper will use four traditional heuristic link prediction algorithms (AA [22], Jaccard [35], PA [36], and RA [37]), two network representation learning algorithms that perform direct matrix manipulation (LAP [38] and HOPE [39]), and two network representation learning algorithms based on random walking (DeepWalk (DW) [40] and Node2Vec (N2V) [34]), as algorithms to be compared with the proposed TCW algorithm. Furthermore, the visual representation of vector clustering is performed with the LAP, HOPE, and TCW algorithms to show the vector visualization results of the algorithms for node embedding on different networks. The experiments will employ the AUC metric [22] as an evaluation metric for the experimental results. Generally, the larger the scale of the links in the training set, the higher the AUC value when evaluating the link prediction algorithm.

5.1. Datasets

In this paper, the effectiveness of the TCW algorithm on the link prediction task is evaluated by employing five categories of real-world network datasets. Given a network $G = (V, E)$, the set E of known links in network G is divided into a training set E_T and a test set E_P based on the evaluation metrics of the link prediction algorithm, where $E_T \cup E_P = E$ and $E_T \cap E_P = \varnothing$. The same number of nonexistent links from the network are randomly selected as negative samples to participate in the training and testing process of the algorithm. The following are the experimental datasets: CITESEER [41], CORA [41], PUBMED [41], POWER [33], and YEAST [42]. The statistics of the experimental dataset are shown in Table 2, where $|V|$ denotes the number of nodes, $|E|$ denotes the number of links, $< k >$ denotes the average degree of the network, $< t >$ denotes the transmissibility of the network, and $< C >$ denotes the average clustering coefficient of the network.

Table 2. Statistics of network datasets.

Datasets	|*V*|	|*E*|	<*k*>	<*t*>	<*C*>	Categories
YEAST	2375	11693	9.85	0.47	0.31	Biological Network
CORA	2708	5278	3.90	0.09	0.24	Citation Network
CITESEER	3327	4676	2.81	0.13	0.14	Citation Network
POWER	4941	6594	2.67	0.10	0.08	Electricity Network
PUBMED	19,717	44,327	4.50	0.05	0.06	Citation Network

5.2. Experimental Results

To evaluate the accuracy of the TCW algorithm for link prediction on the data of different network categories and whether the prediction performance of the algorithm will be affected by the number of training samples. In the experiments of this paper, the link prediction performance of the proposed algorithm is evaluated with 70–90% of the total data as training data. The experimental results are shown in Tables 3–5. The maximum values of the experimental results on different datasets are marked in bold and the second-highest values are underlined.

Table 3. The experimental results of link prediction using 70% of the training links (AUC, %).

Datasets	AA	Jaccard	PA	RA	LAP	HOPE	DW	N2V	*TCW*
YEAST	87.99	87.81	86.35	88.01	<u>94.30</u>	84.44	93.78	93.71	**95.66**
CORA	69.83	69.67	72.15	69.83	80.09	77.21	87.99	<u>88.85</u>	**89.25**
CITESEER	66.39	66.37	80.63	66.39	79.37	72.26	90.93	<u>91.06</u>	**91.66**
POWER	60.44	60.44	60.64	60.44	79.32	65.72	**86.49**	<u>86.39</u>	85.83
PUBMED	63.10	63.08	91.26	63.10	92.04	84.46	91.79	<u>92.05</u>	**92.59**

Table 4. The experimental results of link prediction using 80% of the training links (AUC, %).

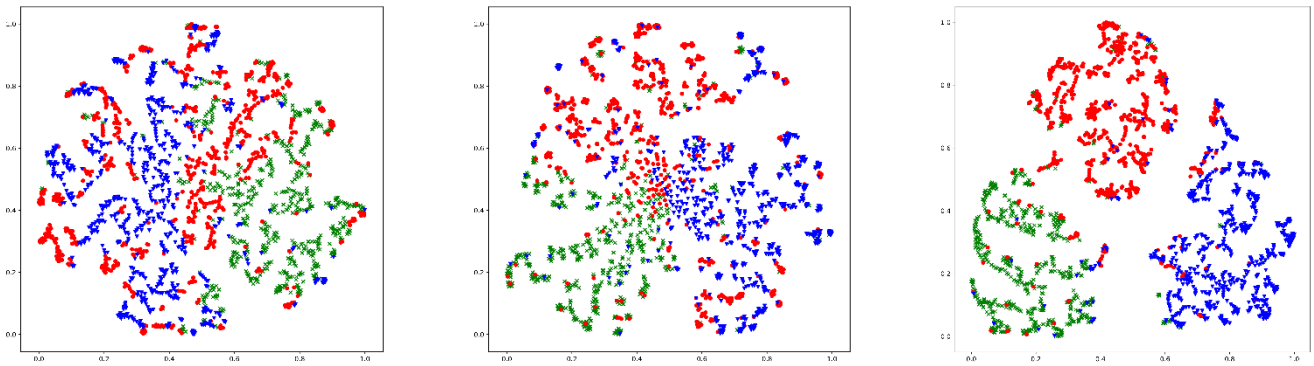
Datasets	AA	Jaccard	PA	RA	LAP	HOPE	DW	N2V	*TCW*
YEAST	88.92	88.68	85.57	88.97	94.67	90.09	94.56	95.16	**95.89**
CORA	74.19	74.07	71.46	74.19	85.79	81.01	**90.34**	88.96	89.83
CITESEER	72.06	71.97	77.18	72.06	84.14	79.69	93.91	93.64	**94.34**
POWER	60.32	60.32	57.03	60.32	81.78	74.95	87.16	86.54	**87.45**
PUBMED	66.77	66.75	90.94	66.77	94.26	86.03	91.31	**94.84**	94.04

Table 5. The experimental results of link prediction using 90% of the training links (AUC, %).

Datasets	AA	Jaccard	PA	RA	LAP	HOPE	DW	N2V	*TCW*
YEAST	90.10	89.84	85.86	90.15	95.50	92.62	94.50	93.30	**96.58**
CORA	76.90	76.70	69.12	76.90	86.79	83.08	91.52	91.55	**91.59**
CITESEER	75.92	75.86	77.97	75.92	87.61	83.55	94.01	94.05	**95.72**
POWER	63.35	63.35	56.76	63.35	89.43	81.81	92.43	92.04	**93.21**
PUBMED	69.82	69.79	91.13	69.81	**94.09**	87.34	93.21	93.51	93.86

According to Tables 3–5, the prediction performances of the TCW algorithm show superior performance in most of the experiments that divide the training data with different division ratios. In which, the TCW algorithm achieves superior results on all real data sets when compared to the traditional heuristic link prediction algorithm based on the local topology of the network. Most of the experiments achieve advantages when compared to the network representation learning algorithm based on direct matrix manipulation and the classical network embedding algorithm based on a random walking sampling policy. Experiments demonstrate that the proposed algorithm achieves superior prediction performance in solving problems of link prediction and the insensitivity to the number of training samples.

To evaluate whether the TCW algorithm can produce node vector representations that match the original network structure, this paper employs a manual synthetic dataset for node vector visualization experiments. The manual dataset employed in the experiments is an artificial network produced with the LFR benchmark program, which contains 2000 nodes and 4725 links with 4.725 average degree, and includes three communities with a size ranging from 500 to 800 nodes [43]. The node vectors learned by the LAP, HOPE, and TCW algorithms are visualized in two dimensions by employing the t-SNE algorithm in this paper, and the experimental results are shown in Figure 3.



(**a**) (**b**) (**c**)

Figure 3. Visualization of vectors obtained by LAP (**a**), HOPE (**b**), and TCW (**c**).

According to the visualization results in Figure 2, the TCW algorithm allows nodes belonging to the same community to be mapped to their neighbors in the two-dimensional vector space. While the LAP and HOPE algorithms both employ direct operations on matrices that can represent network structure information to learn the network representation, such network representation learning methods normally have the characteristics of solid representation and nonscalability, whereas the TCW algorithm learns the vector representation of nodes by feeding the walking sequences into the word embedding model and employs stochastic gradient descent to continuously optimize the model parameters in the process of learning the vectors, which can obtain more comprehensive and complete network representation information than direct matrix operations, and therefore has improved ability to learn network representation.

Finally, we compare the time complexity of the proposed algorithm and the baseline. The time complexity is shown in Table 6, where d_m denotes the maximum degree of a network, r is the number of sampled neighbors per node, n is the number of nodes, D is input dimension, d is output dimension, N is the number of training data points, k is the number of nearest neighbors, L is the iteration number, and m is the number of nonzero elements in adjacency matrix. The proposed TCW algorithm has higher complexity, but better accuracy of link prediction. Therefore, we consider that the TCW algorithm has a significant advantage in the scenario of balancing accuracy and complexity.

Table 6. Time complexity of the methods.

Methods	Time Complexity
AA	$O(d_m^3 n)$
Jaccard	$O(d_m^2 n) \sim O(d_m^3 n)$
PA	$O(d_m^2 n) \sim O(d_m^3 n)$
RA	$O(d_m^3 n)$
LAP	$O(D\log(k)N\log(N)) + O(DNk^3) + O(dN^2)$
HOPE	$O(md^2L)$
DW	$O(dnlog(n))$
N2V	$O(drn)$
TCW	$O(drn) + O(n^3)$

6. Discussion and Conclusions

The DeepWalk algorithm employs a completely random walk policy, which is limited by the fact that the walking process is not controlled and does not allow for the complete exploration of the network topology, leading to missing information in the network representation learning process. The LINE algorithm mainly employs the breadth random walking policy and lacks flexibility in exploring more in-depth nodes. The Node2Vec algorithm trades off the probability of selecting the depth and breadth walking policies by defining the parameters; however, the parameters require manual specification and often require specialist domain knowledge to configure the different parameters for different datasets.

The algorithm proposed in this paper implements control for the selection of depth and breadth walking policies based on network clustering coefficients, which results in the automatic acquisition of controlled biased walking paths. The local clustering coefficients of the nodes can be easily obtained according to different network topology features, and the selection of the depth and breadth walking policies can be modified based on the different abilities of the local aggregation of the networks. The TCW algorithm provides more flexibility and usability when exploring networks in terms of network topology than traditional algorithms based on random walking sampling policies. The experimental results on several real-world network datasets prove the accuracy of the TCW algorithm for link prediction tasks, and show that the TCW algorithm can obtain vector representations that better reflect the structural features of the original network.

Author Contributions: Conceptualization, M.N., D.C. and D.W.; formal analysis, M.N.; funding acquisition, D.C. and D.W.; methodology, M.N. and D.C.; project administration, M.N., D.C. and D.W.; resources, M.N.; software, M.N.; supervision, D.C. and D.W.; validation, D.C.; visualization, M.N.; writing—original draft, M.N.; writing—review and editing, D.C. and D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Technologies Research and Development Program of Liaoning Province in China under grant 2021JH1/10400079 and the Fundamental Research Funds for the Central Universities under grant 2217002.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there is no conflicts of interest regarding the publication of this paper.

References

1. Wang, Z.; Liang, J.; Li, R. A fusion probability matrix factorization framework for link prediction. *Knowl.-Based Syst.* **2018**, *159*, 72–85. [CrossRef]
2. Martínez, V.; Berzal, F.; Cubero, J.-C. A survey of link prediction in complex networks. *ACM Comput. Surv.* **2016**, *49*, 69. [CrossRef]
3. Lin, D.; Chen, J.; Wu, J.; Zheng, Z. Evolution of Ethereum Transaction Relationships: Toward Understanding Global Driving Factors from Microscopic Patterns. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 559–570. [CrossRef]
4. Wang, D.; Nie, M.; Chen, D.; Wan, L.; Huang, X. Node Similarity Index and Community Identification in Bipartite Networks. *J. Internet Technol.* **2021**, *22*, 673–684.
5. Zhang, C.-J.; Zeng, A. Prediction of missing links and reconstruction of complex networks. *Int. J. Mod. Phys. C* **2016**, *27*, 1650120. [CrossRef]
6. Singh, A.K.; Lakshmanan, K. PILHNB: Popularity, interests, location used hidden Naive Bayesian-based model for link prediction in dynamic social networks. *Neurocomputing* **2021**, *461*, 562–576. [CrossRef]
7. Chen, D.; Nie, M.; Wang, J.; Kong, Y.; Huang, X. Community Detection Based on Graph Representation Learning in Evolutionary Networks. *Appl. Sci.* **2021**, *11*, 4497. [CrossRef]
8. Huang, X.; Chen, D.; Ren, T. A Feasible Temporal Links Prediction Framework Combining with Improved Gravity Model. *Symmetry* **2020**, *12*, 100. [CrossRef]
9. Wu, J.; Shen, J.; Zhou, B.; Zhang, X.; Huang, B. General link prediction with influential node identification. *Phys. A Stat. Mech. Appl.* **2019**, *523*, 996–1007. [CrossRef]
10. Lv, L.; Yi, C.; Wu, B.; Hu, M. An Improved Link Prediction Algorithm Based on Comprehensive Consideration of Joint Influence of Adjacent Nodes for Random Walk with Restart. In Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, Tianjin, China, 23–26 April 2020; pp. 380–387.
11. Zhou, Y.; Wu, C.; Tan, L. Biased random walk with restart for link prediction with graph embedding method. *Phys. A Stat. Mech. Appl.* **2021**, *570*, 125783. [CrossRef]
12. Nasiri, E.; Berahmand, K.; Li, Y. A new link prediction in multiplex networks using topologically biased random walks. *Chaos Solitons Fractals* **2021**, *151*, 111230. [CrossRef]
13. Curado, M. Return random walks for link prediction. *Inf. Sci.* **2020**, *510*, 99–107. [CrossRef]
14. Zhang, M.; Chen, Y. Link prediction based on graph neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 5165–5175.
15. Cao, Z.; Wang, L.; De Melo, G. Link prediction via subgraph embedding-based convex matrix completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
16. Mingshuo, N.; Dongming, C.; Dongqi, W. Reinforcement learning on graphs: A survey. *arXiv* **2022**, arXiv:2204.06127.
17. Romero, D.; Kleinberg, J. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In Proceedings of the International AAAI Conference on Web and Social Media, Washington, DC, USA, 23–26 May 2010; pp. 138–145.
18. Fang, Z.; Tang, J. Uncovering the formation of triadic closure in social networks. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
19. Shin, J.; Kim, K.; Park, D.; Kim, S.; Kang, J. Bipartite Link Prediction by Intra-Class Connection Based Triadic Closure. *IEEE Access* **2020**, *8*, 140194–140204. [CrossRef]
20. Huang, X.; Chen, D.; Ren, T.; Wang, D. A survey of community detection methods in multilayer networks. *Data Min. Knowl. Discov.* **2021**, *35*, 1–45. [CrossRef]
21. Zhang, M.; Chen, Y. Weisfeiler-lehman neural machine for link prediction. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 575–583.
22. Adamic, L.A.; Adar, E. Friends and neighbors on the web. *Soc. Netw.* **2003**, *25*, 211–230. [CrossRef]
23. Sørensen, T.J. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*; Munksgaard Copenhagen: Copenhagen, Denmark, 1948; Volume 5.

24. Wu, Z.; Lin, Y.; Wang, J.; Gregory, S. Link prediction with node clustering coefficient. *Phys. A Stat. Mech. Appl.* **2016**, *452*, 1–8. [CrossRef]
25. Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; McGraw-Hill, Inc.: Columbus, OH, USA, 1986.
26. Ravasz, E.; Somera, A.L.; Mongru, D.A.; Oltvai, Z.N.; Barabási, A.-L. Hierarchical organization of modularity in metabolic networks. *Science* **2002**, *297*, 1551–1555. [CrossRef]
27. Leicht, E.A.; Holme, P.; Newman, M.E. Vertex similarity in networks. *Phys. Rev. E* **2006**, *73*, 026120. [CrossRef]
28. Toprak, M.; Boldrini, C.; Passarella, A.; Conti, M. Harnessing the Power of Ego Network Layers for Link Prediction in Online Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2022**, 1–13. [CrossRef]
29. Jin, D.; Wang, R.; Wang, T.; He, D.; Ding, W.; Huang, Y.; Wang, L.; Pedrycz, W. Amer: A New Attribute-Missing Network Embedding Approach. *IEEE Trans. Cybern.* **2022**, 1–14. [CrossRef] [PubMed]
30. Guo, Z.; Wang, F.; Yao, K.; Liang, J.; Wang, Z. Multi-Scale Variational Graph AutoEncoder for Link Prediction. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event/Tempe, AZ, USA, 21–25 February 2022; pp. 334–342.
31. Zhou, C.; Zhang, J.; Gao, K.; Li, Q.; Hu, D.; Sheng, V.S. Bipartite network embedding with Symmetric Neighborhood Convolution. *Expert Syst. Appl.* **2022**, *198*, 116757. [CrossRef]
32. Luce, R.D.; Perry, A.D. A method of matrix analysis of group structure. *Psychometrika* **1949**, *14*, 95–116. [CrossRef] [PubMed]
33. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. [CrossRef]
34. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
35. Jaccard, P. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bull. Soc. Vaud. Sci. Nat.* **1901**, *37*, 241–272.
36. Barabâsi, A.-L.; Jeong, H.; Néda, Z.; Ravasz, E.; Schubert, A.; Vicsek, T. Evolution of the social network of scientific collaborations. *Phys. A Stat. Mech. Appl.* **2002**, *311*, 590–614. [CrossRef]
37. Zhou, T.; Lü, L.; Zhang, Y.-C. Predicting missing links via local information. *Eur. Phys. J. B* **2009**, *71*, 623–630. [CrossRef]
38. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural Inf. Process. Syst.* **2001**, *14*.
39. Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; Zhu, W. Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1105–1114.
40. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
41. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective Classification in Network Data. *AI Mag.* **2008**, *29*, 93. [CrossRef]
42. von Mering, C.; Krause, R.; Snel, B.; Cornell, M.; Oliver, S.G.; Fields, S.; Bork, P. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* **2002**, *417*, 399–403. [CrossRef] [PubMed]
43. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef] [PubMed]