A new similarity-based link prediction algorithm based on combination of network topological features

Hasan Saeidinezhad[1], Elham Parvinnia[1], Reza Boostani[2]
[1]Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
[2]Department of Computer Science Engineering and Information Technology, Electrical and Computer Engineering Faculty, Shiraz University, Shiraz, Iran

Article Info

Article history:

Received Mar 20, 2021
Revised Jan 8, 2022
Accepted Jan 19, 2022

Keywords:

Friend prediction
Graph network
Link prediction
Similarity criteria
Social networks

ABSTRACT

In recent years, the study of social networks and the analysis of these networks in various fields have grown significantly. One of the most widely used fields in the study of social networks is the issue of link prediction, which has recently been very popular among researchers. A link in a social network means communication between members of the network, which can include friendships, cooperation, writing a joint article or even membership in a common place such as a company or club. The main purpose of link prediction is to investigate the possibility of creating or deleting links between members in the future state of the network using the analysis of its current state. In this paper, three new similarities, degree neighbor similarity (DNS), path neighbor similarity (PNS) and degree path neighbor Similarity (DPNS) criteria are introduced using neighbor-based and path-based similarity criteria, both of which use graph structures. The results have been tested based on area under curve (AUC) and precision criteria on datasets and it shows well the superiority of the work over the criteria that only use the neighbor or the path.

Corresponding Author:

Elham Parvinnia
Department of Computer Engineering, Shiraz Branch, Islamic Azad University
Shiraz, Iran
Email: parvinnia@iaushiraz.ac.ir

1. INTRODUCTION

Networks are a platform for analyzing social structures that have focused all their attention on the relationships between the members that make up such a social structure. These members are called nodes or vertices that these nodes have properties that distinguish them [1]. For example, in the network of people in an organization, nodes characteristics can be considered as male or female or the position of each person in that organization. Therefore, when we talk about social networks, we mean social context and it consists of vertices (individual or organizational) that are connected by one or more specific types of relationships such as friendship, kinship, disease transmission, which are called edges. Network in its simplest form is a mapping that connects vertices by related edges; outlines, nodes are actors within the network and edges are the relationships between these actors [2].

Social networks have been studied in different fields. One of the most widely used fields of study in social networks is link prediction. The problem that link prediction seeks to address is: given a snapshot of the current state of the network, is it possible to predict which network members might be in a relationship in the next snapshot of the network? [3].

The importance of this issue becomes apparent when the recommender systems or online sales recommend people to find products of interest [4], or help them make new friends [5], social academias that allow people to find a co-author or expert [6], or large-scale communication networks that predict a specific person's contacts on a mobile phone [7]. It is also possible to use this field of research to complete a network by using incomplete and partial information of that network [8], [9] and to better understand the evolution of a network [10]. In addition, link recognition methods is widely used in the sciences of bioinformatics and biology, for example in predicting properties that are more likely to find their way into the future, or in identifying interactions between proteins, as well as in gene expression networks [11] can be used to identify links.

Existing techniques in the field of link prediction are divided into 3 categories: based on nodes, based on structure and based on social theory [12]. Node-based techniques estimate the similarity between nodes based on the unique characteristics of each network node. Characteristics such as gender, age and goods purchased can be considered and based on these characteristics, new friends will be suggested, and new products will be offered. Structure-based techniques use graph structures and are divided into three categories: neighbor-based, path-based, and random-based [13]. The most important neighbor-based algorithms such as common neighbor (CN), preferential attachment (PA) and Adamic Adar (AA) use direct nodes neighbors to estimate the similarity of 2 nodes. For example, the common neighbor algorithm, based on the higher the number of common neighbors between 2 nodes, the more likely it is to create a link between those 2 nodes, provides a measure of similarity between the two nodes *x* and *y* [14]:

$$CN(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

where $\Gamma(x)$ is the set of node *x* neighbors and $|\Gamma(x) \cap \Gamma(y)|$ indicates the number of common neighbors of the nodes *x* and *y*. The preferential attachment algorithm considers the product of the degrees of two nodes as a measure of the similarity of these two nodes, assuming that the higher the degree the 2 nodes have, the probability of creating a link between the two nodes increases. This algorithm provides a measure of similarity between the two nodes *x* and *y* [10]:

$$PA\ (x,y) = |\Gamma(x)|\,.\,|\Gamma(y)| \tag{2}$$

where $|\Gamma(x)|$ and $|\Gamma(y)|$ are degree of nodes *x* and *y*, respectively.

Also, the Adamic Adar algorithm, which has been widely used, is based on the fact that the fewer of the common neighbors the two nodes x and y have, the probability of creating a link between x and y increases. This algorithm is calculated [13]:

$$AA\ (x,y) = \sum_{Z\ \epsilon \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|} \tag{3}$$

where z is a node in the set of common neighbors of x and y, |Γ(z)| is degree of node z.

One of the most important path-based algorithms is Katz algorithm. For calculating Katz algorithm, all the paths between the two nodes x and y are counted and included in the algorithm. Base on this algorithm, similarity between the nodes *x* and *y* is calculated:

$$Katz\ (x,y) = \sum_{l=1}^{\infty} \beta^{l}.\left|path_{x,y}^{l}\right| = \beta A + \beta^{2} A^{2} + \tag{4}$$

$\beta>0$ is a parameter that determines the effectiveness of long paths. The smaller the β value, the Katz algorithm performs similarly to the common neighbor because the effect of long paths on the similarity calculation is greatly reduced [14].

In other criteria instead of just using the first-order common neighbors second-order common neighbors are also used. This similarity which is called common neighbors degree penalization (CNDP) is calculated [15]:

$$CNDP\ (x,y) = \sum_{Z\ \epsilon \Gamma(x) \cap \Gamma(y)} \frac{|C_{(Z)}|}{|\Gamma(z)|^{\beta C}} \tag{5}$$

where z is a node in the set of common neighbors x and y, $\left|C_{(Z)}\right|$ the number of common neighbors that nodes z, x and y have, $|\Gamma(z)|$ degree of node z, C is the average clustering coefficient in the graph and β is a constant value, the optimal value of which is obtained by experimenting on different datasets by regression algorithms. the authors in [16] also used second-order neighborhood and they called it latent relationship. They also claimed that calculating similarity based on first-order neighborhood results low accuracy but benefiting the second-order neighborhood compensates that drawback.

Adaptive dynamic programming (ADP) [16] is one of the outstanding algorithms in this field. In this algorithm, the degree of common neighbors is penalized based on the amount of clustering coefficient in desired network. The purpose of this work is to present a similarity algorithm that can have an acceptable accuracy in predicting links for all networks with different structures. Accordingly, in order to take advantages of the unique structure of desired network in calculating the similarity, the following algorithm is presented, which parameter c represents the value of average clustering coefficient of the desired network and parameter β is a constant value that 2.5 is intended for it.

$$S\ (x,y) = \ \sum_{Z\ \epsilon\Gamma(x)\cap\ \Gamma(y)} |\Gamma(z)|^{-\beta C} \tag{6}$$

Another algorithm called node-coupling clustering approach based on common neighbors (NCCCN) benefits the clustered network information and combines it with the information of the common neighbors between the two nodes and obtains a measure of similarity between these two nodes based on these two sources of information. Also, in [18] a neighbor-based similarity criterion called triadic measure was introduced, which uses units called motifs. Motifs are various forms of small grids that include 3 nodes in a directional grid. In this article, 13 different types of motifs are introduced. The algorithm introduced in this paper calculates the similarity between 2 nodes *x* and *y* in a way that for each of the common neighbors such as *z* between these two nodes, the number of motifs consisting of *3* nodes *x*, *y*, and *z* is counted and the result is divided by thirteen this process is calculated for all common neighbors and finally added together, the resulting number is then divided by the total number of common neighbors between two nodes *x* and *y*. The result shows the similarity between 2 nodes *x* and *y*. Network clustering information is very useful information in the link prediction process [19]. To this end, the authors clustered the study network in [20] and observed that there is a large relationship between these clusters, which is derived from the graph structure, and the accuracy of the algorithm in identifying the link. Also, Bastami *et al.* [21] proposed a method for identifying gravity-based links that also used cluster information in the network. They applied their algorithm in parallel to the detected clusters to increase the execution speed. They also used the Adamic Adar similarity criterion to calculate the similarity between the nodes.

Many studies have worked on the combination of structural features of the network and it has been observed that node-based features can be very useful in calculating similarity [22]. In [23] authors designed a framework that used both the graph structure and the unique features of nodes to identify links. They evaluated their work on co-authorship and co-starring datasets and found that combining graph information yields much better results than when using a single feature. They used the γ parameter to adjust the degree of interference of each feature.

In another study [24], authors acknowledged that the similarity between the activities of two nodes was proportional to the distance between the two nodes, meaning that the smaller the distance between the two nodes, the greater their similarity. They combined the unique features of the node, such as its activity and trajectory, with the structural features of the network, and designed a supervised classifier that significantly increased the accuracy of the prediction. As mentioned before, none of the above studies investigate the effect of path and common neighbor and node topological attributes (such as degree) combined, in this regard in this paper three similarity-based link prediction algorithms are proposed which cover the different combinations of these topological features and compare them with the pre-mentioned algorithm like ADP, NCCCN and CNDP which only use one of these attributes [25]. Also the proposed algorithm uses the information of direct and indirect neighbors of second degree (the same as CDNP) which has been proved that has better performance rather than only using direct neighbors. The first proposed algorithm called degree neighbor similarity (DNS) uses the information of nodes degree and neighbors of two nodes. The second one called path neighbor similarity (PNS) calculate the similarity based on path and neighbors information, and the last proposed algorithm called degree path neighbor similarity (DPNS) is a combination of two other algorithms and uses the information of degree, path and neighbors combined. The results reported in section 4 of the paper show the superiority of the proposed algorithm over the CNDP algorithm and other corresponding algorithms. The continuation of the article is as follows: in section 2 the presented algorithm will be explained in detail, in section 3 the results will be reviewed and analyzed and in the last part of the article conclusions and future works will be stated.

2. MATERIALS AND METHOD

The basic framework of all similarity-based link prediction algorithms is the same, and the only difference is how the similarity criterion is calculated. Accordingly, this paper presents a similarity criterion that works better on common datasets than other similarity-based criteria such as ADP, NCCCN, and triadic measure, as well as CNDP. In this criterion, in addition to using the information of neighbors, the paths between two nodes and the degrees of the source and destination node are also contributed as shown in Figure 1.

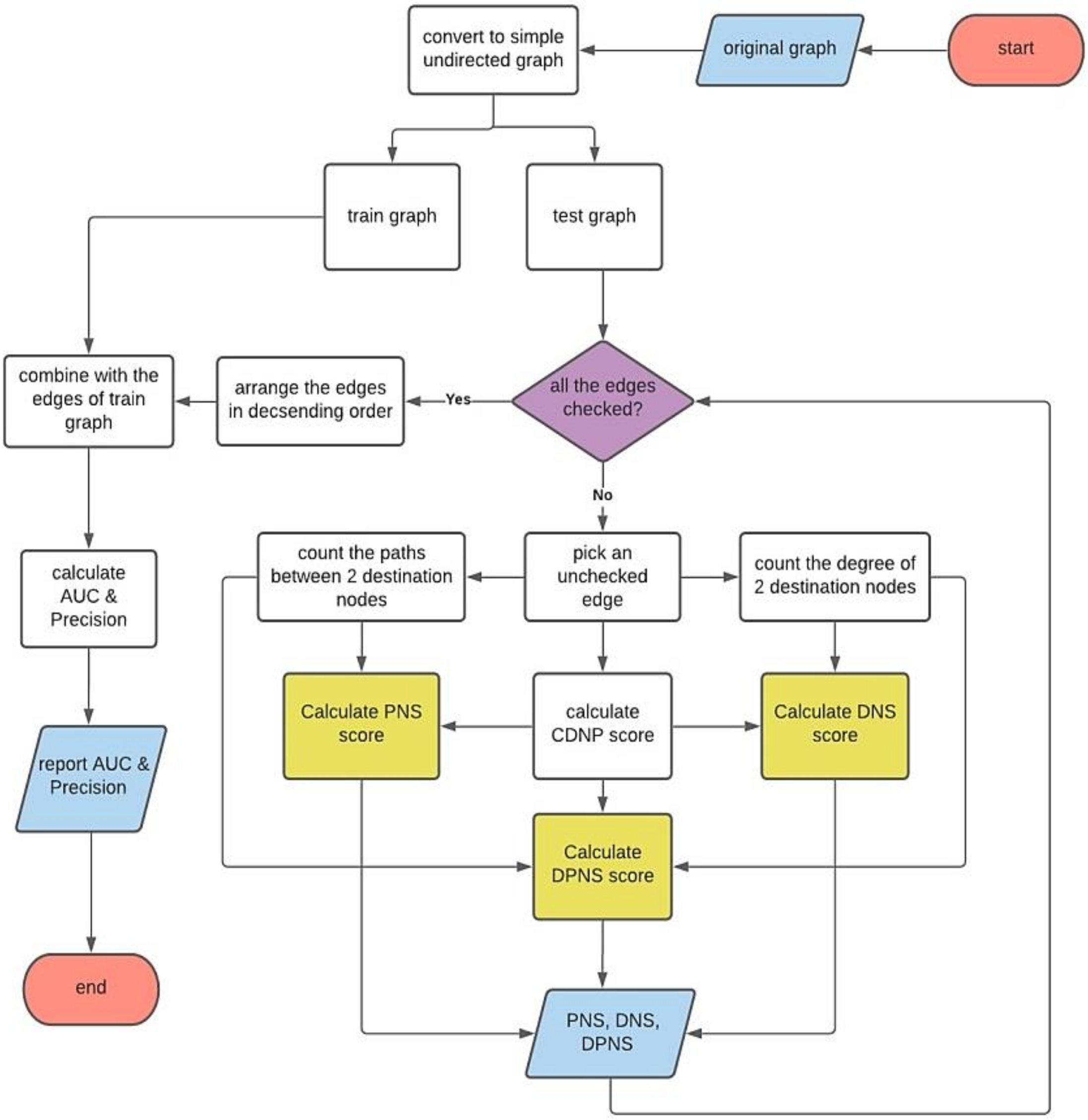
start
original graph
convert to simple undirected graph
train graph
test graph
all the edges checked?
Yes
No
arrange the edges in decsending order
combine with the edges of train graph
calculate AUC & Precision
report AUC & Precision
end
count the paths between 2 destination nodes
pick an unchecked edge
count the degree of 2 destination nodes
Calculate PNS score
calculate CDNP score
Calculate DNS score
Calculate DPNS score
PNS, DNS, DPNS

Figure 1. Diagram of the proposed method

In this regard, first the studied network is implemented as a graph $G=(V, E)$ where V is a set of vertices and E is a set of edges between the vertices. Graph G is then pre-processed to make the edges undirected if the edges are directional, and also converted to a simple graph if the graph is multi-edge during pre-processing. Then, on a simple and undirected graph G, the link recognition algorithm is applied as follows. First, the edges of the graph are divided into 5 parts by five-folds cross validation method, one part is considered as test data and 4 parts as train data. Thus, the train data contains 0.8 of the edges of the graph, and the test data contains the rest of the edges (0.2 of the edges of the graph) plus the edges that did not exist in the original graph at all. For every two nodes in the test data, the edge between them are calculated with similarity to the presented criterion, and then the edges are arranged in descending order based on the similarity value. After calculating the similarity score for all edges, a sample of test data containing m edges is picked, which m is equal to 20% of the original graph edges and is added to the train data these edges are predicted edges. Then true positive and false negative are calculated and the area under curve (AUC) and precision are estimated based on that. To calculate the similarity between two nodes based on the combination of path and neighbor information PNS, we follow the similarity criterion:

$$path\_neghibor\ Similarity\ (x,y) = \sum_{l=3}^{5} \alpha^{l}.\left|path_{x,y}^{l}\right| \times \sum_{Z\ \epsilon\Gamma(x)\cap\Gamma(y)} \frac{|C_{(Z)}|}{|\Gamma(z)|^{\beta C}} \quad (7)$$

In this regard, α is the coefficient that determines the effect of long distances, small values α reduce the effect of long distances. $\left|path_{x,y}^{l}\right|$ Indicates the number of paths of length l between two nodes *x* and *y*. Due to the computational complexity of finding long paths, only paths between 3 to 5 have been investigated. Also $\left|C_{(Z)}\right|$, is the number of common neighbors of the nodes *z*, *x* and y, $|\Gamma(z)|$ degree of node *z*, *C* the average clustering coefficient in the graph and β is a constant value that its optimal value is obtained by experimenting on different datasets by regression method.

There is also another criterion based on neighbors and the degree of source and destination nodes DNS, which is:

$$degree\_neghibor\ Similarity\ (x,y) = |\Gamma(x)| \times |\Gamma(y)| \times \sum_{Z\ \epsilon\Gamma(x)\cap\Gamma(y)} \frac{|C_{(Z)}|}{|\Gamma(z)|^{\beta C}} \quad (8)$$

where $|\Gamma(x)|$ is degree of node *x*, $|\Gamma(y)|$ is degree of node y, and the rest of the parameters are the same as in formula 6. Also, in order to compare these similarity criterias, another criterion is used which is a combination of these two criterias to check if the degree and path information of source and destination nodes and also direct and indirect neighbors of second degree are contributed in the similarity calculation. (DPNS) Then how accurately can the similarity between the two nodes be estimated.

$$degree\_path\_neghibor\ Similarity\ (x,y) = |\Gamma(x)| \times |\Gamma(y)| \times \sum_{l=3}^{5} \alpha^{l}.\left|path_{x,y}^{l}\right| \times \sum_{Z\ \epsilon\Gamma(x)\cap\Gamma(y)} \frac{|C_{(Z)}|}{|\Gamma(z)|^{\beta C}} \quad (9)$$

where $|\Gamma(x)|$ degree of node x, $|\Gamma(y)|$ degree of node y and α is a coefficient that determines the effect of long paths. $\left|path_{x,y}^{l}\right|$ indicates the number of paths of length *l* between two nodes *x* and *y*. $\left|C_{(Z)}\right|$ is the number of common neighbors of the node *z*, *x* and *y*, $|\Gamma(z)|$ is degree of node z, C is the average clustering coefficient in the graph and β is a constant value, the optimal value that its optimal value is obtained by experimenting on different datasets by regression method. For example, in Figure 2 to calculate the similarity between the two nodes x and y the values of the parameters are written in Table 1.

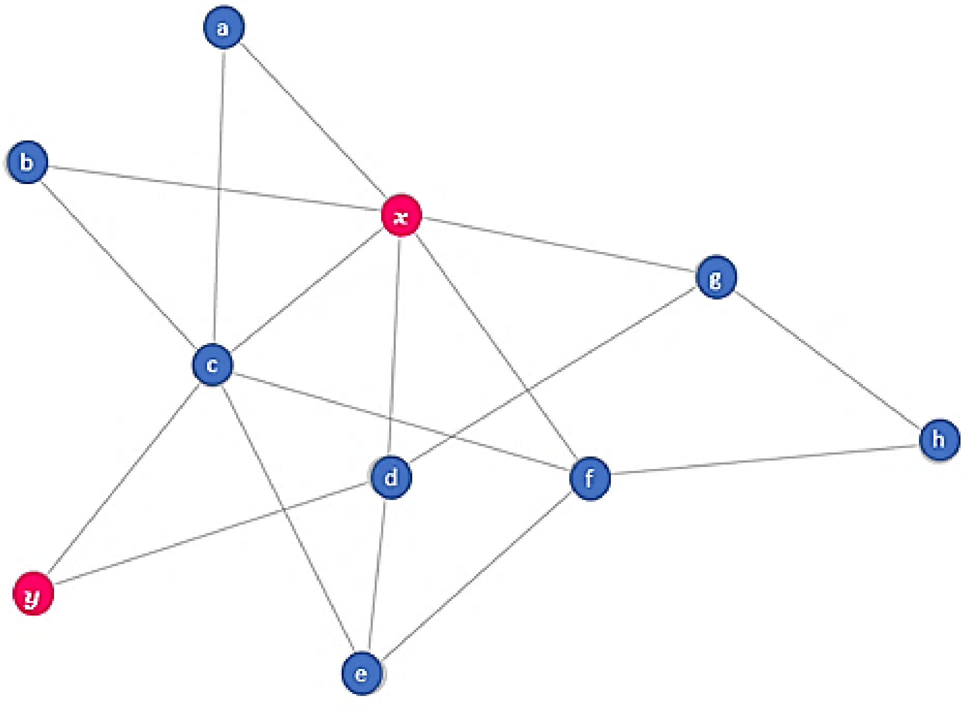

Figure 2. Network graph

Table 1. The parameter values of formulas 6, 7, and 8 according to Figure 2

Parameters	Value	Set of nodes/paths
x_degree	6	
y_degree	2	
$path_{x,y}^{1}$	0	
$path_{x,y}^{2}$	2	{xcy, xdy}
$path_{x,y}^{3}$	3	{xacy, xbcy, xgdy}
$path_{x,y}^{4}$	3	{xfecy, xdecy, xcedy}
$path_{x,y}^{5}$		
$\Gamma(x)$		{a, b, c, d, f, g}
$\Gamma(y)$		{c, d}
Z		{c, d}
$C_{(z)}\ if\ z : c$	$\emptyset$	
$C_{(z)}\ if\ z : d$	$\emptyset$	
$\Gamma(z)\ if\ z : c$		{a, b, f, x, y}
$\Gamma(z)\ if\ z : d$		{e, g, x, y}

3. RESULTS AND DISCUSSION

The desired dataset attributes are given in detail in Table 2, which the first column, includes the name of the database, |V| the number of nodes, |E| the number of edges, K the average degree, C the average clustering coefficient, average shortest path length (ASPL) the average length of the shortest path, D the diameter and H is the heterogeneity in the graph. All of these datasets are real-world datasets. Also, the value of average clustering coefficient for each of the datasets is mentioned in column 5 of the mentioned table.

Table 2. The statistical information of experimental datasets

Network	|V|	|E|	K	C	ASPL	D	H
BUP	105	441	8.4	0.49	3.08	7	1.42
CEG	297	2148	14.46	0.29	2.46	5	1.80
UAL	332	2126	12.81	0.63	2.74	6	3.46
INF	410	2765	13.49	0.46	3.63	9	1.39
SMG	1024	4916	9.6	0.31	2.98	6	3.95
EML	1133	5451	9.62	0.22	3.61	8	1.94

CEG is a biological network, SMG is a co-authorship network, UAL is an airport traffic network, EML is a network of people who send e-mails to each other, BUP is a network of political blogs and INF is a network of contacts in an exhibition. The above datasets are available at https://noesis.ikor.org/datasets/link-prediction. The experiments were performed on a system with 32 RAM and an Intel core i5 processor at 3.1 and 3.4 GHz. The value for the parameters is as follows:

$\alpha : 0.1\ , 0.05$
$l : 5$
$\beta\ in\ AUC\ metric : 1.76$
$\beta\ in\ precision\ metric : 1.84$

In order to evaluate the performance of the proposed algorithm and other algorithms, we use two criteria: AUC and precision. The method of calculating each of the two criteria is as follows: [15] in order to calculate the AUC value, a link from the test data set (including non-existent and non-observation links) and a link from the non-existent set (including 0.2 edges removed from the main graph as non-existent set) are picked and the AUC values of these two links will be obtained from (10):

$$AUC = \frac{n_1+0.5n_2}{n} \tag{10}$$

where n is the number of comparisons, $n1$ is the number of times that score of the link chosen from the test set is more than the other link, and $n2$ is the number of times that both links have the same score.

In this paper, n is equal to the product of the number of edges of both sets. In other word, all the edges of both sets are compared. The precision criterion is also obtained from the (11):

$$precision = \frac{A}{T} \tag{11}$$

where *A* is the number of correctly predicted links and *T* is the total number of predicted links.

The results of Table 3 show that the DNS algorithm, which is a combination of neighbor information between two nodes and two-node degrees, performs better on BUP and CEG datasets than other algorithms, but failed to outperform its competitors on other datasets. On INF and EML data, CNDP algorithm, which uses only neighbor information, and on SMG data, CNDP algorithm, which uses only neighbor information, had better results. It can be seen that the degree of nodes is not very effective and it cannot always be claimed that the higher the degree of nodes, the more likely it is to form a link.

Table 3. Comparing the results of DNS, ADP, NCCCN, triadic measure, and CNDP algorithms on 6 datasets

Algorithm	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
ADP	precision	0.254	0.153	0.515	0.419	**0.159**	0.197
	AUC	0.749	0.800	**0.931**	0..869	**0.833**	0.797
NCCCN	precision	0.242	0.160	0.502	0.293	0.148	0.211
	AUC	0.753	0.781	0.919	0.879	0.812	0.802
Triadic Measure	precision	0.229	0.147	0.483	0.364	0.136	0.173
	AUC	0.733	0.734	0.893	0.483	0.782	0.785
CNDP	precision	0.263	0.150	0.478	**0.434**	0.143	**0.203**
	AUC	0.853	0.817	0.928	**0.920**	0.812	**0.820**
DNS	precision	**0.265**	**0.167**	**0.595**	0.368	0.142	0.151
	AUC	**0.855**	**0.821**	0.930	0.911	0.820	0.810

The evaluation results of the PNS algorithm on the mentioned datasets are given in Tables 4 and 5. These results are obtained with the value of α=0.1 in Table 4 and α=0.05 in Table 5. As can be seen, the PNS algorithm, which in addition to the information of the common neighbors of the two nodes, also uses the path information between the two nodes, has better results than the DNS. In Table 4, the PNS algorithm outperformed competing algorithms on all data by at least one AUC or precision criterion. It can be seen that it is superior to other algorithms in terms of AUC criterion on BUP, CEG, INF and EML datasets and has better performance on precision criteria on CEG, UAL and SMG datasets. By reducing the value of α from 0.1 to 0.05, which reduces the effect of longer paths in Table 4, it can be seen that the proposed algorithm performed better on all datasets except BUP in terms of both precision and AUC criteria and on the BUP dataset, it is superior to other algorithms in terms of AUC.

Table 4. Comparing the results PNS, ADP, NCCCN, triadic measure, and CNDP algorithms on 6 datasets $(\alpha: 0.1)$

Algorithm	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
ADP	precision	0.254	0.153	0.515	0.419	0.159	0.197
	AUC	0.749	0.800	**0.931**	0..869	**0.833**	0.797
NCCCN	precision	0.242	0.160	0.502	0.293	0.148	**0.211**
	AUC	0.753	0.781	0.919	0.879	0.812	0.802
Triadic Measure	precision	0.229	0.147	0.483	0.364	0.136	0.173
	AUC	0.733	0.734	0.893	0.483	0.782	0.785
CNDP	precision	0.263	0.150	0.478	**0.434**	0.143	0.203
	AUC	0.853	0.817	0.928	0.920	0.812	0.820
PNS	precision	0.258	**0.183**	**0.589**	0.401	**0.159**	0.194
	AUC	**0.867**	**0.825**	0.924	**0.920**	0.820	**0.822**

Table 5. Comparing the results of PNS algorithms, ADP, NCCCN, triadic measure and CNDP algorithms on 6 datasets $(\alpha: 0.05)$

Algorithm	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
ADP	precision	0.254	0.153	0.515	0.419	0.159	0.197
	AUC	0.749	0.800	0.931	0..869	0.833	0.797
NCCCN	precision	0.242	0.160	0.502	0.293	0.148	0.211
	AUC	0.753	0.781	0.919	0.879	0.812	0.802
Triadic Measure	precision	0.229	0.147	0.483	0.364	0.136	0.173
	AUC	0.733	0.734	0.893	0.483	0.782	0.785
CNDP	precision	0.263	0.150	0.478	0.434	0.143	0.203
	AUC	0.853	0.817	0.928	0.920	0.812	0.820
PNS	precision	0.256	**0.175**	**0.596**	**0.437**	**0.161**	**0.214**
	AUC	**0.870**	**0.837**	**0.931**	**0.924**	**0.836**	**0.824**

From the results, it is obvious that the effect of path information on improving the accuracy of prediction is very effective and selecting the appropriate value of α and adjusting the effect of long distances can greatly increase the accuracy of prediction. In order to more accurately investigate the effect of path information and node degree information, using similarity criterion introduced in formula 8 which uses a combination of both information, this link prediction algorithm was performed on the above datasets and the results are reported in Tables 6 and 7.

As can be seen from the results in Tables 6 and 7, the combination of path information and node degrees reduces the accuracy of the PNS and makes the results worse than when only path information is used. However, it performs better than other algorithms and is found to be superior to all datasets except INF in at least one of the criteria. Table 6 evaluates the DPNS results with α=0.1 and Table 7 considers α=0.05 for the DPNS criterion. It can be seen that a lower value of α improves the results in DPNS as it did in the PNS algorithm, indicating that the less the effect of long distances, the greater the accuracy of the prediction.

In order to make a better compare the criteria presented in this article, in Table 8, the results of each of the criteria are reported on the above datasets. The superiority of the PNS criterion with α=0.05 is easily visible. As can be seen from the results in Table 8, the PNS criterion with α=0.05 had better performance and higher prediction accuracy for the 3 INF, SMG and EML datasets. For other data, it can be seen that in terms of AUC value, PNS criterion is superior to other criteria with α=0.05.

Table 6. Comparing the results of DPNS algorithms, ADP, NCCCN, triadic measure, and CNDP algorithms on 6 datasets (α:0.1)

Algorithm	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
ADP	precision	0.254	0.153	0.515	0.419	**0.159**	0.197
	AUC	0.749	0.800	**0.931**	0..869	**0.833**	0.797
NCCCN	precision	0.242	0.160	0.502	0.293	0.148	**0.211**
	AUC	0.753	0.781	0.919	0.879	0.812	0.802
Triadic Measure	precision	0.229	0.147	0.483	0.364	0.136	0.173
	AUC	0.733	0.734	0.893	0.483	0.782	0.785
CNDP	precision	0.263	0.150	0.478	**0.434**	0.143	0.203
	AUC	0.853	0.817	0.928	**0.920**	0.812	0.820
DPNS	precision	**0.267**	**0.166**	**0.602**	0.384	0.142	0.161
	AUC	**0.858**	**0.819**	0.921	0.913	0.818	**0.820**

Table 7. Comparing the results of DPNS algorithms, ADP, NCCCN, triadic measure, and CNDP algorithms on 6 datasets (α:0.05)

Algorithm	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
ADP	precision	.254	0.153	0.515	0.419	0.159	0.197
	AUC	0.749	0.800	**0.931**	0..869	0.833	0.797
NCCCN	precision	0.242	0.160	0.502	0.293	0.148	0.211
	AUC	0.753	0.781	0.919	0.879	0.812	0.802
Triadic Measure	precision	0.229	0.147	0.483	0.364	0.136	0.173
	AUC	0.733	0.734	0.893	0.483	0.782	0.785
CNDP	precision	0.263	0.150	0.478	0.434	0.143	**0.203**
	AUC	0.853	0.817	0.928	**0.920**	0.812	0.820
DPNS	precision	**0.272**	**0.172**	**0.606**	0.397	0.155	0.177
	AUC	**0.860**	**0.826**	0.924	0.916	**0.834**	**0.821**

Table 8. Comparison of DNS, PNS, and DPNS algorithms for 0.5 and 0.1 for α on 6 datasets

Algorithm	α	Evaluation metric	BUP	CEG	UAL	INF	SMG	EML
DNS	__	precision	0.265	0.167	0.595	0.368	0.142	0.151
		AUC	0.855	0.821	0.930	0.911	0.820	0.810
PNS	0.1	precision	0.258	**0.183**	0.589	0.401	0.159	0.194
		AUC	0.867	0.825	0.924	0.920	0.820	0.822
DPNS	0.1	precision	0.267	0.166	0.602	0.384	0.142	0.161
		AUC	0.858	0.819	0.921	0.913	0.818	0.820
PNS	0.05	precision	0.256	0.175	0.596	**0.437**	**0.161**	**0.214**
		AUC	**0.870**	**0.837**	**0.931**	**0.924**	**0.836**	**0.824**
DPNS	0.05	precision	**0.272**	0.172	**0.606**	0.397	0.155	0.177
		AUC	0.860	0.826	0.924	0.916	0.834	0.821

4. CONCLUSION

In this paper, we tried to introduce a new similarity-based algorithm by combining structural information of graph such as nodes neighbors and path information between nodes, as well as using the degree of nodes in the current snapshot of the network. For this purpose, three different criteria were presented by combining different information from the network in order to more accurately examine the impact of each factor on the accuracy of the link prediction. The first DNS criterion is obtained by combining the information of two desired nodes which is degree and direct and indirect neighbors of the two nodes. The second criterion called PNS is obtained by combining the information of 3 to 5 length paths between two nodes and direct and indirect neighbors of two nodes. The third criterion, DPNS, is obtained by combining the information of the desired nodes which is paths with the lengths of 3 to 5 between the two nodes, direct and indirect neighbors, and also the degree of nodes. The results show that DNS performed better on some data such as BUP and CEG than other algorithms but did not succeed on other data. The results of PNS and DPNS algorithms with different values of α show that this algorithm works better than DPS and for α=0.05 on all datasets has better results than competitors. It can also be seen that PNS performs better than DPNS, which indicates that it is very important to use path information between two nodes. As mentioned in the review of previous work, using clusters in the network can greatly improve the result of the work, so in later work, in addition to the above information, cluster information can also be used. Also, in networks where user information is available, this information can be used to improve forecast accuracy.

REFERENCES

[1] S. P. Borgatti, M. G. Everett, and J. C. Johnson, *Analyzing social networks*. Sage, 2018.

[2] S. Wasserman and K. Faust, *Social network analysis*, vol. 22, no. 1954. Cambridge University Press, 1994.

[3] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, May 2007, doi: 10.1002/asi.20591.

[4] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Transactions on the Web*, vol. 6, no. 2, pp. 1–33, May 2012, doi: 10.1145/2180861.2180866.

[5] J. Mori, Y. Kajikawa, H. Kashima, and I. Sakata, "Machine learning approach for finding business partners and building reciprocal relationships," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10402–10407, Sep. 2012, doi: 10.1016/j.eswa.2012.01.202.

[6] T. Wohlfarth and R. Ichise, "Semantic and event-based approach for link prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5345, Springer Berlin Heidelberg, 2008, pp. 50–61.

[7] T. Raeder, O. Lizardo, D. Hachen, and N. V. Chawla, "Predictors of short-term decay of cell phone contacts in a large scale communication network," *Social Networks*, vol. 33, no. 4, pp. 245–257, Oct. 2011, doi: 10.1016/j.socnet.2011.07.002.

[8] D. J. Marchette and C. E. Priebe, "Predicting unobserved links in incompletely observed networks," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1373–1386, Jan. 2008, doi: 10.1016/j.csda.2007.03.016.

[9] M. Kim and J. Leskovec, "The network completion problem: inferring missing nodes and edges in networks," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, Apr. 2011, pp. 47–58, doi: 10.1137/1.9781611972818.5.

[10] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, no. 3–4, pp. 590–614, Apr. 2001, doi: 10.1016/S0378-4371(02)00736-7.

[11] W. Almansoori *et al.*, "Link prediction and classification in social networks and its application in healthcare and systems biology," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 1, no. 1–2, pp. 27–36, Jun. 2012, doi: 10.1007/s13721-012-0005-7.

[12] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, Jan. 2015, doi: 10.1007/s11432-014-5237-y.

[13] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, Jul. 2003, doi: 10.1016/S0378-8733(03)00009-1.

[14] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953, doi: 10.1007/BF02289026.

[15] S. Rafiee, C. Salavati, and A. Abdollahpouri, "CNDP: link prediction based on common neighbors degree penalization," *Physica A: Statistical Mechanics and its Applications*, vol. 539, p. 122950, Feb. 2020, doi: 10.1016/j.physa.2019.122950.

[16] V. Martínez, F. Berzal, and J.-C. Cubero, "Adaptive degree penalization for link prediction," *Journal of Computational Science*, vol. 13, pp. 1–9, Mar. 2016, doi: 10.1016/j.jocs.2015.12.003.

[17] F. Li, J. He, G. Huang, Y. Zhang, Y. Shi, and R. Zhou, "Node-coupling clustering approaches for link prediction," *Knowledge-Based Systems*, vol. 89, pp. 669–680, Nov. 2015, doi: 10.1016/j.knosys.2015.09.014.

[18] F. Aghabozorgi and M. R. Khayyambashi, "A new similarity measure for link prediction based on local structures in social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 501, pp. 12–23, Jul. 2018, doi: 10.1016/j.physa.2018.02.010.

[19] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks: a local naïve bayes model," *EPL (Europhysics Letters)*, vol. 96, no. 4, Nov. 2011, doi: 10.1209/0295-5075/96/48007.

[20] X. Feng, J. C. Zhao, and K. Xu, "Link prediction in complex networks: a clustering perspective," *The European Physical Journal B*, vol. 85, no. 1, Jan. 2012, doi: 10.1140/epjb/e2011-20207-x.

[21] L. Jin, J. B. D. Joshi, and M. Anwar, "Mutual-friend based attacks in social network systems," *Computers & Security*, vol. 37, pp. 15–30, Sep. 2013, doi: 10.1016/j.cose.2013.04.003.

[22] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, Jul. 2001, doi: 10.1103/PhysRevE.64.025102.

[23] C. Yu, X. Zhao, L. An, and X. Lin, "Similarity-based link prediction in social networks: a path and node combined approach," *Journal of Information Science*, vol. 43, no. 5, pp. 683–695, Oct. 2017, doi: 10.1177/0165551516664039.

[24] Z. Yin, M. Gupta, T. Weninger, and J. Han, "LINKREC: a unified framework for link recommendation with user attributes and graph structure," 2010, doi: 10.1145/1772690.1772879.

[25] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, “Human mobility, social ties, and link prediction,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1100–1108, doi: 10.1145/2020408.2020581.

BIOGRAPHIES OF AUTHORS

Hassan Saeidinezhad Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran PhD student. Main directions of investigations: Data Mining, Recommender Systems. He can be contacted at email: saeidinezhad@gmail.com

Elham Parvinnia Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran Ph.D., Associate Professor. Main directions of investigations: Data Mining, Machine Learning, Algorithm Design, Recommender Systems. First degree Ph.D. student from Shiraz University. She can be contacted at email: parvinnia@iaushiraz.ac.ir.

Reza Boostani CSE & IT Dept., Electrical and Computer Engineering Faculty, Shiraz University, Shiraz, Iran. Ph.D., Full Professor. Main directions of investigations: medical hardware, fuzzy sets and fuzzy logic, BCI, Depth of Anesthesia, Pain Measurement include biomedical signal processing, statistical pattern recognition, and machine learning. He can be contacted at email: boostani@shirazu.ac.ir.