SCIENTIFIC REPORTS
nature research

OPEN

Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm

Iftikhar Ahmad[1*], Muhammad Usman Akhtar[1], Salma Noor[2] & Ambreen Shahnaz[2]

Real world complex networks are indirect representation of complex systems. They grow over time. These networks are fragmented and raucous in practice. An important concern about complex network is link prediction. Link prediction aims to determine the possibility of probable edges. The link prediction demand is often spotted in social networks for recommending new friends, and, in recommender systems for recommending new items (movies, gadgets etc) based on earlier shopping history. In this work, we propose a new link prediction algorithm namely "Common Neighbor and Centrality based Parameterized Algorithm" (CCPA) to suggest the formation of new links in complex networks. Using *AUC* (Area Under the receiver operating characteristic Curve) as evaluation criterion, we perform an extensive experimental evaluation of our proposed algorithm on eight real world data sets, and against eight benchmark algorithms. The results validate the improved performance of our proposed algorithm.

Complex networks are effective descriptions of real world networks, where real world problems can be modeled in the form of complex network graphs[1]. Complex networks describe the interaction among the elements of complex systems such as computer, neural, chemical and online social networks[2]. In such networks, entities (such as computer, neurons etc.) are represented by nodes (also called vertices), whereas edges between pair of nodes depict interactions/associations between the nodes[3]. Complex networks have application in many divisions of applied science[4]. It has been applied in health care to predict the spread of epidemic diseases[5], and in the development of strategies to vaccinate the potential affectees to limit the spread of epidemic. Furthermore, the complex network analysis can be applied in legislative drives to influence maximum number of citizens[6], and in the development of road networks to improve routes[7]. Considerable efforts are made to understand the network evolution[8,9], and the fundamental topological structure of complex real world networks[10].

Due to ever evolving nature of complex networks, one crucial scientific issue related to complex network analysis is missing link prediction[11]. Networks are very agile in nature; fresh vertices and edges are added over the passage of time[12]. The basic idea of link prediction is to approximate the possibility of the existence of a link between pair of nodes, derived from the current topological structural attributes of the nodes[13]. For example, in online connected community networks, future associations can be suggested as likely-looking friendships, which can assist the system in recommending new friends and thus strengthen their dependability to the service[8]. In other words, link prediction provides a measure of social propinquity between pair of nodes. The only available information is the topological structure of the network[14].

Applications of the phenomenon include suggestion of new followers/friends on social websites such as Google Plus, Facebook, Foursquare, LinkedIn, and Twitter etc. In addition, it can also be used to suggest interests that are most likely collective. For example, recommendation of products on Amazon and Alibaba, recommendation of movies on Netflix, and ads display to users on Google AdWords and Facebook[15].

In this work, we present a novel algorithm for link prediction using the existing topological structure of the network. Our proposed algorithm named *Common Neighbor and Centrality based Parameterized Algorithm* (CCPA) identifies potential future edges/connections between nodes using common neighbors and centrality. The proposed algorithm is parameterized, i.e., it has the flexibility to let the user/system set the importance of common neighbor and centrality. The proposed algorithm is evaluated against eight commonly used standard algorithms for link prediction on eight data sets. Experimental evaluation suggests the better predictability of our proposed algorithm.

[1]Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, Pakistan. [2]Department of Computer Science, Shaheed Benazir Bhutto Woman University, Peshawar, Pakistan. *email: ia@uetpeshawar.edu.pk

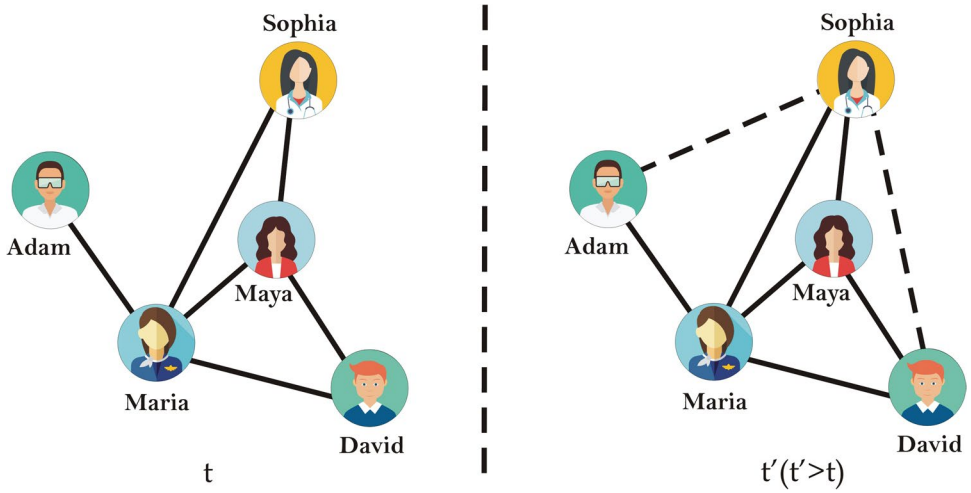

Figure 1. Graphical representation of missing link prediction; dashed lines depict possible edges.

Formal Problem Setting

Assume $G(V, E)$ to be an undirected graph, representing a complex network at a time t; V and E are building blocks of graph representing set of nodes and edges respectively. Loops and multiple-connections between nodes are not allowed[15].

Let, U represents the set of all possible edges between nodes in the graph, then $|U| = \frac{|V|(|V|-1)}{2}$. Let, $L = U - E$ be the set of missing links in the graph. Normally we are not aware which links may occur in the future, otherwise we do not need link prediction[16]. Link prediction aims to predict the possibility of link formation between two nodes at time t' $(t' > t)$[13]. The primary goal is to forecast new links among nodes that may take place in the near future.

The problem is clarified with a simple network of 5 persons (nodes) as shown in Fig. 1. The total number of possible links in a 5 node network is $\frac{5(5-1)}{2} = 10$. For the missing links $L = U - E$, the prediction task is to know the fundamental mechanism of link formation in particular complex network and using the current topological structural properties to estimate the non-existing links probability. In Fig. 1, solid lines represent links in the network at time t, and dashed lines represent the link that may occur in the future (for the sake of clarity only two dotted lines are shown in Fig. 1). Maria and Adam are friends, Maria and Sophia are also friends at time t. Possibly Maria introduces Sophia with Adam, and they become friends as well. Similarly, Sophia and David may become friend at time t'. A link prediction algorithm awards a similarity score S_{xy} to all links $l_{xy} \in L$ based on some pre-defined criterion. Note that l_{xy} represents link between nodes x and y. If S_{xy} is greater than or equal to a threshold, then a link is predicted between nodes x and y.

Literature Review

Considerable body of literature is devoted to the study of link prediction in complex networks (see[7] and the references therein). The problem is addressed both from graph theory, and machine learning perspective. Due to space limitation, we cannot elaborate on the research work in the domain of machine learning. The reader is referred to Nickle *et al.*[17] and Wang *et al.*[18]. In the following, we summarize the important works based on graph theoretic approach.

Wang *et al.*[19] presented a popularity based structural perturbation algorithm that made use of current popularity of node based on the assumption that an active node has more affinity to attract future nodes. The algorithm is based on similarity based approach that measures the possibility of links through knowing collective aspects, i.e., common friends, age differences, professions, and tracing locations which the two end points have in common. The proposed algorithm is evaluated on six data sets and against six algorithms. However, no statistical tests were performed to evaluate the significance of the proposed approach. Yang and Zhang[20], introduced an algorithm based on the common neighbors and distance metric to predict link in a variety of real world networks from the available topological structure of the network. The algorithm aims to find missing link probability between nodes who do not have common neighbors. The proposed algorithm is tested on eight data sets against standard benchmark algorithms using Areas Under the receiver operating characteristic Curve (AUC) as criterion. The algorithms are executed only once, thus increasing the possibility of data snooping bias. Pan *et al.*[21] critiqued the real life network to be incomplete and noisy, which makes link prediction algorithms hard to apply. The authors presented an algorithmic framework for missing link prediction by accounting for predefined Hamiltonian structures. Using AUC as evaluation criterion, the proposed framework is evaluated on seven data sets. However, like Wang *et al.*[19], Yang and Zhang[20], and Pan *et al.*[21] did not employ any statistical tests to evaluate the significance of the results.

Liao *et al.*[3] proposed two algorithms to address missing link prediction problem. The first algorithm is based on Pearson correlation coefficient. In the second algorithm, the correlation based method is integrated with resource allocation algorithm. The second algorithm is found to outperform the existing methods. The proposed

second scheme is a parametrized algorithm. However, the control parameter can only influence the correlation factor. Similarly, no statistical tests were performed. Ibrahim and Chen[15] criticized the existing approaches for missing link predictions based on static graph representation. Authors used temporal information, community structure and centrality to predict the formation of new links. Using AUC as evaluating criterion, authors analyzed the performance of their proposed algorithm using real world data sets. One of the significant drawback of the proposed scheme is the high computational cost.

Zhou *et al.*[2] empirically investigated missing link prediction of nine well known algorithms on six data sets. The results indicated that common neighbor is the best performing algorithm. The authors further proposed a novel algorithm based on resource allocation process, which achieved superior experimental performance than common neighbor algorithm. Murata and Moriyasu[22] presented an algorithm based on the proximity measures and weights of existing links in a weighted graph to predict possible future interactions in online social networks. The proposed algorithm was evaluated using Yahoo! Chiebukuro. For a detailed survey of link prediction techniques, the readers are referred to Lou and Zhou[7].

Proposed Algorithm

Our proposed algorithm is based on two vital properties of nodes, namely the number of common neighbors and their centrality. Common neighbor refers to the common nodes between two nodes. Centrality refers to the prestige that a node enjoys in a network. Since the seminal work of Freeman[23], centrality is based on two key factors in an undirected graph, namely closeness and betweenness. Intuitively, closeness centrality refers to the average shortest distance between any given two nodes, whereas betweenness centrality is the measure of control a node has, to influence the flow of information/communication among the nodes of a network. A node will have high betweenness centrality, if the shortest path between the various nodes passes through it. In this work, we consider closeness centrality as parameter for missing link prediction. Formally, we define closeness centrality C_{xy} between two nodes x and y in a network with N nodes as follows;

$$C_{xy} = \frac{N}{d_{xy}}$$

Note that d_{xy} is the shortest distance between the nodes x and y. Using common neighbor, and closeness centrality, we propose a new algorithm for missing link prediction. The algorithm calculates similarity score S_{xy} as follows;

$$S_{xy} = \alpha \cdot (|\Gamma(x) \bigcap \Gamma(y)|) + (1 - \alpha) \cdot \frac{N}{d_{xy}} \tag{1}$$

Parameter $\alpha \in [0, 1]$ is a user defined value that controls the weight/importance of common neighbor and centrality. $\Gamma(x)$ represents the neighbors of a node x. Note that the value of associated with common neighbor and centrality constitutes a zero sum condition, i.e., increasing the importance of one factor will result in lowering the importance (weight) of other factor.

Experimental Setting

Methodology. For investigating the performance of our proposed algorithm, we evaluate the algorithm on eight data sets, and against eight different algorithms. The adopted methodology is as following;

Each data set is divided into two distinct and non-overlapping graphs namely training (G^T) and probe (G^P) graphs. Training graph (G^T) is obtained by randomly sampling over the original graph G. The remaining edges, those not included in G^T, forms G^P. Analogously, the set of edges included in G^T are referred to as E^T, and those included in G^P are referred to as E^P, i.e., $E = E^T + E^P$. Note that E^T and E^P are mutually exclusive. However, the nodes in G^T and G^P may overlap. For our experiments, we have included 80% of edges in E^T, and the remaining 20% in E^P. Figure 2 is a graphical depiction of the process.

Graph G^T (analogously E^T) is the input to a link prediction algorithm, which considers the existing topological properties of the G^T, and predicts future links in the form of new graph G'. In order to measure the performance of algorithm, we compute the number of true positive (TP) edges and false positive (FP) edges predicted by an algorithm. An edge e is said to be TP, if $e \in G'$, and $e \in G^P$, i.e., the link is present both in the predicted graph and the probe graph. Recall that edges in training graph and probe graph are mutually exclusive, so it is not possible for an edge to be present simultaneously in G^T and G^P. FP is defined as $e \in G'$, and $e \notin G$, i.e., FP refers to a wrongly predicted edge which should not exist.

As graph G^T (and hence G^P) is obtained randomly, we performed the experiments 15 times to ensure that results obtained are not by chance. For each run, we produced G^T (and hence G^P) randomly, G^T was then used as input to the algorithm, which produced the resultant graph G'. G' was compared with G^P and G to obtain TP and FP. Our results are based on the average values of the 15 runs. The value of parameter α can range from 0 to 1 (both inclusive). For our proposed algorithm we report the average results obtained for $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

Algorithms. We compare the performance of our proposed algorithm with the following set of algorithms.

1. Common Neighbor and Distance (*CND*): The algorithm is based on two key structural properties of a complex network, i.e., common neighbor and distance. For any two non-connected nodes x and y, a score S_{xy} is calculated as shown in Eq. 2 to reflect the likelihood of link formation between the nodes x and y[20]. Recall that $\Gamma(x)$ refers to the neighbors of node x, CN_{xy} is the number of common nodes between node x and y, and d_{xy} is the distance between and x and y.

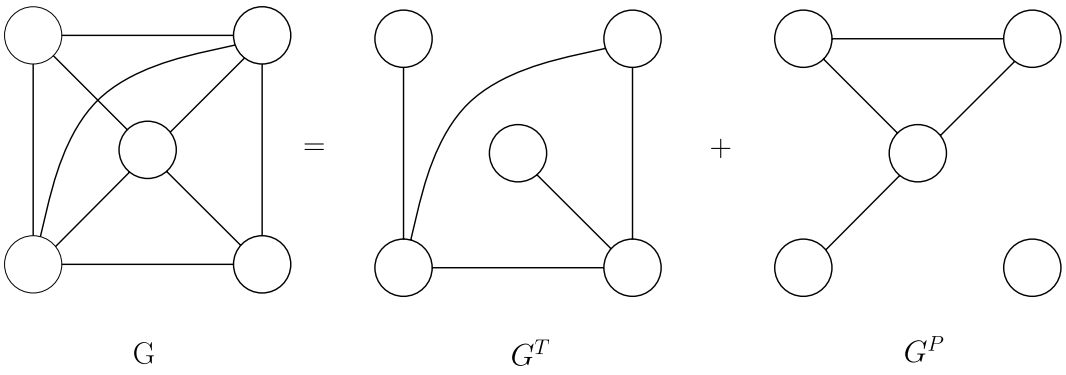

Figure 2. Dividing the original graph in training and probe set.

$$S_{xy} = \begin{cases} \dfrac{CN_{xy} + 1}{2} & \Gamma(x) \bigcap \Gamma(y) \neq \varnothing \\ \dfrac{1}{d_{xy}} & \text{otherwise} \end{cases} \tag{2}$$

2. Preferential Attachment (*PA*): In Preferential Attachment algorithm, the score S_{xy} depends on the degree of node x and y respectively, and is calculated as show in Eq. 3 [24]. Note that k_x represents the degree of a node x.

$$S_{xy} = k_x \cdot k_y \tag{3}$$

3. Adamic Adar (*AA*): Adamic Adar is based on the hypothesis that it is more likely that two nodes x and y are introduced by common neighbors who are more likely to be unpopular in the network. In other words, it is more likely that nodes x and y will be introduced by a node i than node j, if the degree of i is lower than the degree of j. The formula for S_{xy} is given in Eq. 4. Note that $\Gamma(x)$ refers to the neighbors of node x.

$$S_{xy} = \sum_{z \in \Gamma(x) \bigcap \Gamma(y)} \frac{1}{logK_z} \tag{4}$$

4. Common Neighbor (*CN*): In Common Neighbor algorithm the score for link prediction is computed by finding the number of common neighbors between two distinct nodes[24]. The formula for S_{xy} calculation of is given in Eq. 5.

$$S_{xy} = |\Gamma(x) \bigcap \Gamma(y)| \tag{5}$$

5. Sorensen Index (*SI*): In Sorensen algorithm, twice of common nodes is divided by the product of degrees of two distinct nodes for calculation of S_{xy} [24].

$$S_{xy} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k_x + k_y} \tag{6}$$

6. Jaccard Index (*JI*): Jaccard Index considers only the common neighbors between the nodes to calculate S_{xy} as following[25];

$$S_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{7}$$

7. Resource Allocation (*RA*): Resource Allocation (*RA*) calculates S_{xy} on the basis of intermittent nodes connecting node x and y. The similarity index is defined as the amount of resource node x receives from node y through indirect links. Each intermediate link contributes a unit of resource. *RA* is symmetric, i.e., $RA(x, y) = RA(y, x)$[26].

$$S_{xy} = \sum_{z \in \Gamma(x) \bigcap \Gamma(y)} \frac{1}{K_z} \tag{8}$$

8. Hub Promoted Index (*HPI*): Hub Promoted Index (*HPI*) is a measure defined as the ratio of common neighbors of nodes x and y to the minimum of degrees of the nodes[27]. HPI is computed as:

Network	N	M	$<d>$	$<k>$
Karate	34	78	2.408	4.588
USAir	332	2126	2.7381	12.807
Dolphins	62	159	3.357	5.129
Polbook	105	441	3.079	8.400
Word	112	425	2.536	7.589
Neural	306	2147	2.455	14.0327
Circuit	512	819	6.858	3.199
E-mail	1133	5451	3.606	9.622

Table 1. Illustration of properties of eight real world networks. N: number of nodes in graph (G), M: number of edges in G, $<d>$: average distance, $<k>$: average degree.

$$S_{xy} = \frac{\Gamma(x) \cap \Gamma(y)}{\min\{k_x, k_y\}} \tag{9}$$

Data sets. We are using real-world complex network data sets for evaluation of our proposed algorithm against the selected set of algorithms. Gathering a valid data set is time-consuming and labor-intensive process, as most of the data sets are not available publicly. We selected eight popular real-world data sets for our experiments. A brief description of each data set is as following;

1. Karate: Data set of Zachary Karate club network, which shows the correlation of 34 members of a university Karate club. The data set was first studied by Wayne W. Zachary for over three years from 1970 to 1972 to study the clash arose between instructor and administrator[28].
2. Dolphins: It is a network investigated by Lusseau *et al.*[29]. The network consists of 62 bottlenose dolphins who lived in Doubtful Sound of New Zealand between 1994 and 2001.
3. Polbook: Books about US politics, compiled by Valdis Krebs. Nodes represent books sold online by amazon.com. The edges represent frequent co-purchasing of books by the same buyer. The unpublished network is available online (Social network analysis software & services for organizations, communities, and their consultants. Retrieved from www.orgnet.com).
4. Word: This is the undirected network of common noun and adjective adjacencies for the novel "David Copperfield". A node denotes either a noun or an adjective. An edge ties two words that occur in adjacent positions. The network is not bipartite, i.e., there are edges connecting adjectives with adjectives, nouns with nouns and adjectives with nouns[30].
5. Circuit: Electronic circuits can be seen as system where links are wires, and nodes are electronic parts (like capacitors, transistors, etc.). Circuit data is retrieved from www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks).
6. Email: This is a network of e-mail exchanges between members of the Universitat Rovira i Virgili (Tarragona). Nodes represent users, and a link is formed between nodes if there is email communication between them. The data is available at http://deim.urv.cat/alexandre.arenas/data/welcome.htm.
7. USAir: The network of the US air transportation system, which contains 332 airports and 2126 airlines which connects the US around the globe[31].
8. Neural: This data symbolizes the C. Elegans neural network. Graph is being processed in order to remove repeated edges. (See data set at http://wormwiring.org/).

Table 1 Summarizes key properties of the selected data sets.

Evaluation criterion. A link prediction algorithm assigns a score S_{xy} to every missing link (i.e., $U - E^T$). The score S_{xy} quantifies the likelihood of a missing link to be existent in the near future. If S_{xy} equals or surpasses a threshold value, then link is confirmed and considered to occur in the next temporal unit. *AUC* (Area Under the receiver operating characteristic Curve) is used as an evaluation criterion to judge the performance of our selected set of algorithms on the considered data sets. AUC value reflects the probability that a randomly chosen existing link is given a higher similarity score S_{xy} than a randomly chosen non-existent link. AUC is calculated by picking an existing (TP) and a non-existing (FP) link and scores are compared. Among n independent observations/comparisons, let n_1 observations/comparisons resulted in a higher score for existing links, and n_2 observations have resulted in same score, then *AUC* is calculated as following[32];

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

A good link prediction algorithm should have an *AUC* value close to 1.

	Karate	USAir	Dolphins	Polbook	Word	Neural	Circuit	Email
RA	0.651 (0.09)	0.685 (0.08)	0.713 (0.08)	0.835 (0.09)	0.638 (0.08)	0.819 (0.06)	0.534 (0.02)	0.793 (0.06)
AA	0.645 (0.07)	0.657 (0.12)	0.701 (0.07)	0.826 (0.08)	0.642 (0.04)	0.803 (0.08)	0.537 (0.03)	0.799 (0.07)
Jaccard	0.542 (0.11)	0.849 (0.08)	0.725 (0.08)	0.792 (0.05)	0.583 (0.08)	0.755 (0.05)	0.525 (0.02)	0.807 (0.09)
CN	0.647 (0.06)	0.895 (0.06)	0.731 (0.08)	0.842 (0.07)	0.639 (0.11)	0.817 (0.07)	0.536 (0.03)	0.816 (0.07)
CND	0.66 (0.11)	0.906 (0.05)	0.746 (0.05)	0.874 (0.05)	0.651 (0.08)	0.821 (0.07)	0.629 (0.09)	0.862 (0.04)
PA	0.593 (0.1)	0.789 (0.06)	0.582 (0.11)	0.606 (0.11)	0.664 (0.12)	0.704 (0.09)	0.527 (0.09)	0.717 (0.1)
SI	0.567 (0.09)	0.844 (0.07)	0.732 (0.09)	0.823 (0.07)	0.582 (0.1)	0.749 (0.08)	0.54 (0.01)	0.825 (0.06)
HPI	0.657 (0.12)	0.88 (0.06)	0.721 (0.06)	0.837 (0.07)	0.594 (0.07)	0.761 (0.12)	0.528 (0.04)	0.797 (0.09)
CCPA	0.646 (0.09)	0.91 (0.06)	0.753 (0.09)	0.864 (0.07)	0.657 (0.09)	0.839 (0.08)	0.631 (0.11)	0.875 (0.05)

Table 2. Average *AUC* and standard deviation of the algorithms on selected data sets.

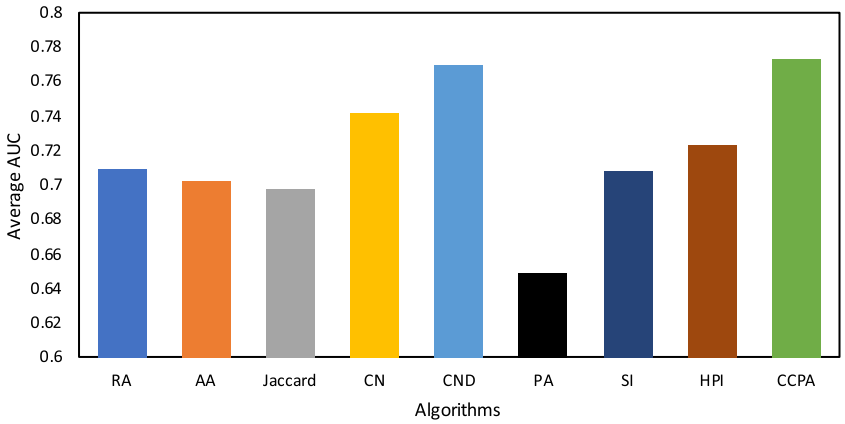

Figure 3. Average AUC of algorithms.

Results and Discussions

Table 2 summarizes the results based on *AUC* value obtained for each algorithm on various data sets. Standard deviation of *AUC* is also given in the parenthesis. Recall that these results are the average values of 15 runs. Note that for *CCPA*, we consider $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and report the average *AUC* over all values of α for each data set. We observe that *CCPA*'s average *AUC* values are the highest among the set of algorithms, and thus outperforms all the competing algorithms. The average *AUC* value of *CCPA* is 0.77, which is 7.7% better than the average of *AUC* values of other algorithms. We found that the *AUC* of *CND* (0.76) is very close to that of *CCPA*. However, the performance of *CND* is not consistent. We observed that *CCPA* is the best performing algorithm on 5 data sets (namely USAir, Dolphins, Neural, Circuit, and Email) whereas *CND* is the best performing algorithm on two data sets (Karate and Polbook) only. The worst performing algorithm is *PA* which achieves an average *AUC* value of 0.64, which is 16% less than that of *CCPA*. Figure 3, depicts the average *AUC* of the considered algorithms on all data sets. It is pertinent to mention that our reported *AUC* values are sightly different than those reported in the literature (for instance see Yang and Zhang[20]). There can be multiple reasons to describe the discrepancy. For instance, just like Yang and Zhang[20], we sampled G^T and G^P randomly. This might result in different training and probe sets which can ultimately results in different *AUC*. However, the differences are not significant.

Next, we present the overall performance of the algorithms on the selected data sets. The objective is to identify which data sets are hard to predict in comparison to others. In order to achieve the objective, we calculated the average *AUC* of all the selected set of algorithms on each data set. Results are summarized in Fig. 4. We observed that the worst average *AUC* is achieved by the algorithms on "Circuit" (0.55) and "Word" (0.62) data set, whereas the highest *AUC* is achieved on "US Air" data set (0.823). A circuit graph represents a connection between various parts (such as transistors, capacitors etc). The nodes are represented by capacitors etc, whereas edges are represented by wires. The "Word" data set is a network of adjectives and noun from Charles Dickens novel "David Copperfield". In the network, nodes represent the adjectives and nouns, whereas edges represent pair of words that occur in adjacent positions in the novel. The low value of *AUC* indicates that it is significantly difficult to predict natural language networks. The highest average *AUC* (0.823) is observed for USAir. US Air is a network of US air transportation, where nodes represent airports, and connection represents flights operated between these air ports by various airlines. As it is a human made network, where connections are not arbitrarily distributed, therefore, predicting the connections are comparatively easier than natural complex networks.

The effect of choosing α. In order to find the effect of α over the obtained values of *AUC*, we report the results obtained by executing the proposed algorithm on various value of $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,$

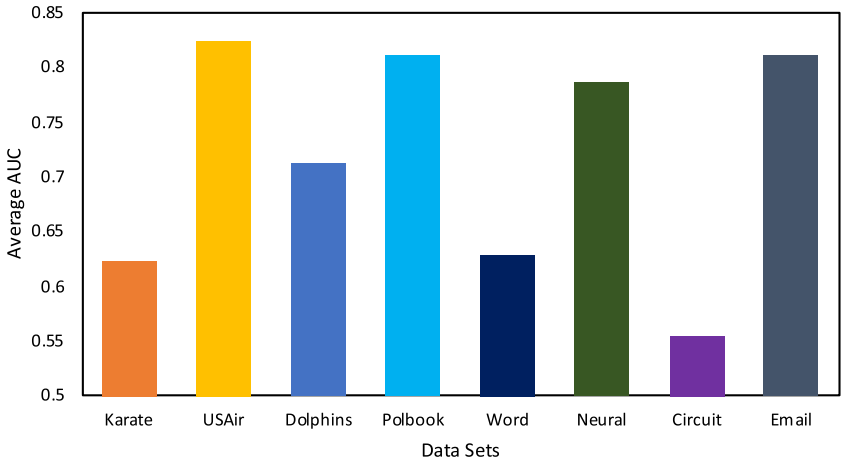

Figure 4. Average *AUC* of algorithms on each data set.

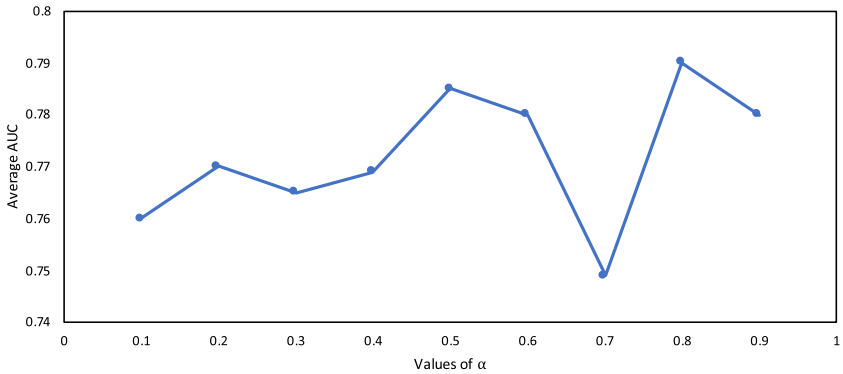

Figure 5. Average *AUC* of *CCPA* for various values of α.

	Karate	USAir	Dolphins	Polbook	Word	Neural	Circuit	Email
AUC	0.7	0.94	0.82	0.9	0.74	0.88	0.68	0.91
α	0.8	0.8	0.6	0.5	0.6	0.7	0.9	0.6

Table 3. Best *AUC* values and the corresponding values of α.

0.9}. We report the average values of *AUC* for a single value of α over all the data sets. Figure 5 represents a graphical view of the trend observed in *AUC* for various values of α.

It is interesting to note that there is no significant change in the average *AUC* value of *CCPA* based on the change in value of α. The minimum average value (0.749) is obtained for $\alpha = 0.7$, whereas the maximum value (0.79) is obtained for $\alpha = 0.8$. The standard deviation of the averaged *AUC* value is 0.013 which is insignificant as well. Even for different data sets, we could not find a trend to identify the value of α for which the proposed algorithm *CCPA* will produce optimum results. The optimum value (the value of α producing the highest average *AUC* value) varies from data set to data set. For instance, on *Karate* data set the highest *AUC* (0.7) is obtained for $\alpha = 0.8$, whereas for *Dolphins* data set the corresponding α value is 0.6. For other data sets, the resultant α values are summarized in Table 3. One observation that stands out is that for all data sets the highest average *AUC* value is obtained for $\alpha \geq 0.5$. For the *Circuit* data set the value is as high as 0.9.

In order to improve the applicability of our proposed algorithm in the real world, we attempt to find a statistical property that can identify optimal value of α. We analyse the results to find a correlation between the optimal value of α for a particular dataset (Table 3) and its key properties (Table 1). For instance, we investigated if there is any correlation between the optimal value of α and $<k>$ (average degree). We consider various statistical properties (such as <d>, <k>, the ratio of *M* and *N*, clustering coefficient etc), but could not find any correlation that can hold true for all datasets. We then divided the data sets in two classes. *Class 1* includes *Karate*, *Dolphins*, and *Neural* data sets, whereas the remaining 5 data sets are included in *Class 2*. Note that *Class 1* are mainly natural data sets where the relationship between nodes is dependent on a natural phenomenon with little to no human intervention. *Class 2* contains man made networks. For *Class 1* data sets, we identified a correlation between $<d>$ (average distance) and optimal value of α. We found that smaller values of $<d>$ resulted in higher value of α. For example, *Karate* data set has the smallest value of $<d> = 2.408$ and the highest optimal value of $\alpha = 0.8$. As the

value of $<d>$ increases, the optimal value of α decreases. Rather surprisingly, we could not establish any correlation between various properties of data sets and optimal value of α for *Class 2*. It will be interesting to perform a thorough analysis of the proposed *CCPA* algorithm on a multitude of data sets to obtain a general inference for choosing the optimal value of α based on the key statistical properties of a network.

Conclusion

Motivated by the challenging nature of missing link prediction in complex networks, we present a novel algorithm based on the two key properties of a network, namely common neighbor, and centrality. Unlike previous algorithms, the proposed algorithm is parametrized where user/system has the ability to control the importance of factors under consideration. We compare our proposed algorithm on eight real life data sets and against eight standard algorithms. Results based on *AUC* (Area Under the receiver operating characteristic Curve) shows the superior performance of our proposed algorithm. Further, the performance of algorithm is reviewed with respect to change in the value of α.

Received: 27 March 2019; Accepted: 28 December 2019;
Published online: 15 January 2020

References

1. Newman, M. E. The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003).
2. Zhou, T., Lü, L. & Zhang, Y.-C. Predicting missing links via local information. *Eur. Phys. J. B* **71**, 623–630 (2009).
3. Liao, H., Zeng, A. & Zhang, Y.-C. Predicting missing links via correlation between nodes. *Phys. A: Stat. Mech. its Appl.* **436**, 216–223 (2015).
4. Al Hasan, M., Chaoji, V., Salem, S. & Zaki, M. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security* (2006).
5. Kamath, P. S. *et al*. A model to predict survival in patients with end-stage liver disease. *Hepatol.* **33**, 464–470 (2001).
6. Tumasjan, A., Sprenger, T. O., Sandner, P. G. & Welpe, I. M. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Fourth international AAAI conference on weblogs and social media* (2010).
7. Lü, L. & Zhou, T. Link prediction in complex networks: A survey. *Phys. A: Stat. Mech. its Appl.* **390**, 1150–1170 (2011).
8. Dorogovtsev, S. N. & Mendes, J. F. Evolution of networks. *Adv. Phys.* **51**, 1079–1187 (2002).
9. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D. Complex networks: Structure and dynamics physics reports, vol. 424 (2006).
10. Getoor, L. & Diehl, C. P. Link mining: a survey. *Acm Sigkdd Explor. Newsl.* **7**, 3–12 (2005).
11. Gong, N. Z. *et al*. Joint link prediction and attribute inference using a social-attribute network. *ACM Trans. Intell. Syst. Technol. (TIST.)* **5**, 27 (2014).
12. Gupta, P. *et al*. Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, 505–514 (ACM, 2013).
13. He, Y.-l, Liu, J. N., Hu, Y.-x & Wang, X.-z Owa operator based link prediction ensemble for social network. *Expert. Syst. Appl.* **42**, 21–50 (2015).
14. Redner, S. Networks: teasing out the missing links. *Nat.* **453**, 47 (2008).
15. Ibrahim, N. M. A. & Chen, L. Link prediction in dynamic social networks by integrating different types of information. *Appl. Intell.* **42**, 738–750 (2015).
16. Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47 (2002).
17. Nickel, M., Murphy, K., Tresp, V. & Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **104**, 11–33, https://doi.org/10.1109/JPROC.2015.2483592 (2016).
18. Wang, P., Xu, B., Wu, Y. & Zhou, X. Link prediction in social networks: the state-of-the-art. *Sci. China Inf. Sci.* **58**, 1–38, https://doi.org/10.1007/s11432-014-5237-y (2015).
19. Wang, T., He, X.-S., Zhou, M.-Y. & Fu, Z.-Q. Link prediction in evolving networks based on popularity of nodes. *Sci. Rep.* **7**, 7147 (2017).
20. Yang, J. & Zhang, X.-D. Predicting missing links in complex networks based on common neighbors and distance. *Sci. Rep.* **6**, 38208 (2016).
21. Pan, L., Zhou, T., Lü, L. & Hu, C.-K. Predicting missing links and identifying spurious links via likelihood analysis. *Sci. Rep.* **6**, 22955 (2016).
22. Murata, T. & Moriyasu, S. Link prediction based on structural properties of online social networks. *N. Gener. Comput.* **26**, 245–257 (2008).
23. Freeman, L. C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1**, 215–239 (1978).
24. Lü, L., Jin, C.-H. & Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Phys. Rev. E* **80**, 046122 (2009).
25. Liben-Nowell, D. & Kleinberg, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **58**, 1019–1031 (2007).
26. Lü, L. & Zhou, T. Link prediction in weighted networks: The role of weak ties. *EPL (Europhys. Lett.* **89**, 18001 (2010).
27. Bliss, C. A., Frank, M. R., Danforth, C. M. & Dodds, P. S. An evolutionary algorithm approach to link prediction in dynamic social networks. *J. Comput. Sci.* **5**, 750–764 (2014).
28. Zachary, W. W. An information flow model for conflict and fission in small groups. *J. anthropological Res.* **33**, 452–473 (1977).
29. Lusseau, D. *et al*. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **54**, 396–405 (2003).
30. Newman, M. E. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104 (2006).
31. Xu, Z. & Harriss, R. Exploring the structure of the us intercity passenger air transportation network: a weighted complex network approach. *GeoJournal* **73**, 87 (2008).
32. Jiang, M., Chen, Y. & Chen, L. Link prediction in networks with nodes attributes by similarity propagation. *arXiv preprint arXiv:1502.04380* (2015).

Author contributions

I.A. lead and supervised the research, contributed in algorithm design, and write up. M.U.A. contributed in the design of algorithm and experiments. S.N. contributed in the data acquisition, and evaluation of algorithm. A.S. worked on the algorithm design, data acquisition, and write up. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to I.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.