

# Chapter 5

## Link Prediction by Network Analysis

Salim Afra, Alper Aksaç, Tansel Özyer, and Reda Alhajj

### 1 Introduction

In the recent surge and evolution of the world wide web, many opportunities arose for analyzing user-generated data, where the term big data became a buzz word and is now used almost everywhere such that high volume of data is available at all times in the Internet. Content of the available web-based data is mostly generated from on-line social networks and e-commerce web applications, among others. Domain specific data encapsulates either homogeneous or heterogeneous actors and the links connecting them leading to an  $n$ -mode network, where  $n$  is number of heterogeneous groups of actors. For instance, data generated from social networks relates mostly to interactions between users/visitors of the networks, where people are modeled as nodes and a friendship relationship is reflected as links connecting people. On the other hand, data generated from e-commerce websites models items (clothes, food, electronics, etc.) and people as nodes to reflect items viewed and bought by people. Accomplished purchase may suggest linking people to items. This behavior of interaction, whether between people or people and items, may be modeled as a social network. A social network can be viewed as a graph where a vertex represents a person or an item, and an edge corresponds to the underlying relationship between vertexes, e.g., friendship, collaboration, among others.

One of the attractive areas for network analysis is collaborations in research where researchers mostly coauthor papers reporting their findings. Collaboration between authors may last short or long leading to a number of coauthored papers over a period of time. Thus, collaboration may be modeled as a social network, and hence can be represented as a graph  $G(V, E)$  where  $V$  is set of nodes or

---

S. Afra • A. Aksaç • T. Özyer (✉) • R. Alhajj  
Department of Computer Science, University of Calgary, Calgary, AB, Canada T2N 1N4  
e-mail: [salim.afra@ucalgary.ca](mailto:salim.afra@ucalgary.ca); [aaksa@ucalgary.ca](mailto:aaksa@ucalgary.ca); [ozyer@etu.edu.tr](mailto:ozyer@etu.edu.tr); [alhajj@ucalgary.ca](mailto:alhajj@ucalgary.ca)

vertexes representing authors and  $E$  is set of edges or links that exist only between researchers who have coauthored at least one paper. Building such a network or the similar, e.g., whether representing scholars or friends on Facebook or other networking sites, provides the possibility of analyzing and may be predicting or uncovering hidden links in the graph. The latter predictions may highlight a possible fruitful collaboration between potential researchers and hence would lead to a recommendation system (RS) which may bring to the attention of target researchers the importance of initiating a collaboration.

A recommendation system is an important mechanism which assists people in exploring items of their interest by guiding them into the specific set of available items in a system's directory. This kind of systems do their recommendations and predictions based on user's preferences and behavior. A separate profile is built for each user and items previously searched for or preferred by a user would help a recommender system in deciding what similar products to recommend. Recommendation systems are used in different domains and are very common in websites such as Google, Amazon, and other e-commerce websites in order to recommend to their users some suggested searches or guide them in buying new items. For example, Amazon recommendation system works by getting a list of items "user A" searched for and viewed, then uses this historical information, and checks what other users examined and purchased while also looking at the same set of items. After this step, the recommendation system will use the set of similar items for recommendations to the selected user. Recommendation systems are also used nowadays in social networking platforms such as Facebook and others to help in suggesting friends. This is done by predicting hidden links between actors and using some common features between users of social media. Such information may lead to new friendships between individuals in a social networking platforms such as Facebook and Twitter. In a similar settings, scientists are in need of different collaboration partners, i.e., experts in a specific topic similar to their research field. Indeed, research interests, co-citation, and bibliographic coupling have constituted some key metric and measure in searching for potential collaboration within a network.

Link prediction is also extensively used and important in the security domain. Since criminal activities occur in groups, finding a criminal may lead to identifying his/her whole criminal partners. Such that we can build a criminal network where nodes represent criminals and relationships represent an involvement of two criminals in an act. Performing link prediction in this kind of network will help governments, intelligence agencies, and other security companies to identify criminals and unveil possible actors involved.

Motivated by the above description, the work described in this paper tackles the issue of relating nodes in a general social network and then making appropriate suggestions by finding hidden links in the analyzed network. Completing this work will help in:

- Uncovering hidden relationships between nodes
- Categorizing and filtering the network
- Predicting links between nodes.

The method described in this paper has been tested on some benchmark networks. The reported results demonstrate the applicability and effectiveness of the proposed approach.

The rest of the paper is organized as follows: Section 2 reviews the most popular previous works. The proposed method is described in Sect. 3. Section 5 presents the conducted experiments, the evaluation process, and the results. Section 6 is conclusions and future research directions.

## 2 Related Work

A considerable amount of research work cover recommendation systems and link prediction, and how they may be applied in different fields. For example, in [1] the authors worked on rating prediction and recommendation of items for users. They carry out the ratings prediction by treating individual user-item ratings as predictors of missing ratings. The final rating is estimated by fusing predictions from the following sources: predictions based on ratings of same item by other users, predictions based on different item ratings made by same user, and ratings predicted based on data from similar users' ratings of similar items. Also in [2], the authors built an algorithm FolkRank ranking scheme which generates personalized rankings of items in a folksonomy and recommends users, tags, and resources. The basic idea is to extract communities of interest from folksonomy, which are represented by their top tags and most influential persons and resources. Once these communities are identified, interested users can find them and participate in. This way community members can more easily get to know each other by using link prediction. Furthermore, [3] proposed two new improved approaches for link prediction: (1) CNGF algorithm based on local information network and (2) KatzGF algorithm based on global information network.

There are also several efforts that investigate expert recommendation for business institutions, e.g., [2, 4]. Petry et al. [5] developed an expert recommendation system called ICARE, which recommends in an organization experts to work with. The focus of the work does not include relations between authors from their publications and citations, it rather considers organizational level of people, their availability, and their reputation. Reichling and Wulf [6] investigated effectiveness of a recommender system for a European industrial association in supporting their knowledge management, foregone a field study, and interviews with the employees. Experts were defined according to their collection of written documents which were automatically analyzed. Additionally, a post-integrated user profile with information about their background and job is used. Using bookmarking services of individual users in building user profiles provides further information about users' interests and confirms their recommendations.

Research on link prediction can also be found in [7], where authors proposed a supervised machine learning framework for co-offence prediction. The authors build a network of criminals and offenders first, then they started to find hidden links

between known criminals and potential ones by relating offenders to socially related, geographically related, or experience related criminal cooperation opportunities. Additionally, Benchettara et al. [8] proposed a new link prediction algorithm to predict links in large-scale two-mode social networks. Based on topological attributes introduced in the paper, the score (or likelihood) of a link between two nodes can be measured, and they defined link prediction as a two class discrimination problem. Thus, a supervised machine learning approach is applied using these attributes to learn a prediction model. Finally, they validated their results on real datasets which are DBLP bibliographical database and bipartite transaction graph of an on-line music e-commerce site. Hasan et al. [9] developed another successful work using supervised learning for prediction; BIOBASE and DBLP networks are used to validate the model.

Another domain of link prediction in the research domain is to recommend possible future partnership to authors who never worked together before. Using this link prediction, the system will suggest people from other domains to work on similar projects, and this may lead to a fruitful partnership to the benefit of the community. However, the focus of the research is mostly focused on homogeneous networks of authors. Brandão et al. [10] modeled a social network of authors for recommending collaborations in academic networks. They presented two new metrics for their social network, namely institutional affiliation aspect and the geographic localization information. They analyzed how these metrics influence the resulting recommendations. Chen et al. [11] proposed a new way for scholar recommendations based on community detection. They used SCHOLST dataset in order to build a network of authors who are clustered into communities based on their research fields. Then they calculated friendship scores for each community in order to do coauthor recommendation based on communities. Davis et al. [12] introduced two approaches for link prediction in heterogeneous networks. In the first algorithm called unsupervised multi-relational link predictor (MRLP), they extended the well-known Adamic/Adar approach. Secondly, they used their previous research based on homogeneous networks in this study by extending for heterogeneous networks. A supervised framework for high performance link prediction (HPLP) shows that a supervised approach is superior to others, including MRLP. Tang et al. [13] proposed a methodology based on modularity analysis of heterogeneous YouTube dataset. Finally, Radivojac et al. [14] analyzed disease-gene networks.

Heck [15] focused on author link prediction, where authors are also modeled as nodes in a social network. What makes this work interesting is the selection of links between authors where bookmarking services are included in edge identification along with author co-citation and bibliographic coupling measurements. They argued how it is important to consider bookmarking along with the other metrics for better link prediction. Sun et al. [16] developed a methodology to predict coauthor relationship among authors in heterogeneous bibliographic network. They tested their algorithm on DBLP bibliographic network and according to their results prediction can be improved using logistic regression-based coauthorship prediction model based on meta path-based topological features. These are the combination of different meta paths and measures.

Discovering new hidden links in a social network is not a trivial task. In [17], when recommending new friendships in a traditional social network, the number of friends in common can be used to estimate the social proximity between users' ground model to smooth the rating predictions. Lichtnwalter and Chawla [18] showed how to evaluate developed methodologies in order to select the best technique. For more detailed information on this topic, the reader may refer to the review reported in [19, 20] (Table 5.1).

**Table 5.1** The 11 similarity metrics used in link prediction; set  $\Gamma(u)$  represents neighbors of node  $u$  in the network and  $|\Gamma(u)|$  shows degree of node  $u$

	Algorithm	Description
Adamic/Adar	$\sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log  \Gamma(z) }$	This index measures similarity with counting of common neighbors $z$ between nodes $u$ and $v$ by weighing the less-connected or rare neighbors more.
Jaccard	$\frac{ \Gamma(u) \cap \Gamma(v) }{ \Gamma(u) \cup \Gamma(v) }$	Common neighbors are divided by total number of neighbors of $u$ and $v$ . It looks for uniqueness in shared neighborhood.
Dice	$\frac{2 \Gamma(u) \cap \Gamma(v) }{ \Gamma(u)  +  \Gamma(v) }$	Common neighbors are divided by their arithmetic mean. It is a semimetric version of Jaccard.
Katz	$\sum_{\ell=1}^{\infty} \beta^{\ell} \cdot  \text{paths}_{u,v}^{\ell} $	This index looks for path lengths and counts by weighting shorter paths between nodes more heavily. Parameter $\beta \in [0, 1]$ controls the contribution of paths and $\ell$ represents the length between nodes. Smaller values for $\beta$ will decrease the contribution of higher values for $\ell$ .
Common neighbors	$ \Gamma(u) \cap \Gamma(v) $	This index measures the number of shared neighbors.
Preferential attachment	$ \Gamma(u)  \cdot  \Gamma(v) $	New connections are directly correlated with high degree of neighbors.
Salton	$\frac{ \Gamma(u) \cap \Gamma(v) }{\sqrt{ \Gamma(u)  \cdot  \Gamma(v) }}$	Common neighbors are divided by their geometric mean.
Resource allocation	$\sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{ \Gamma(z) }$	It is so similar to Adamic/Adar. While this index takes linear form, Adamic/Adar takes $\log$ form. But, this index is inversely more proportional to higher common neighbors.
Hub promoted	$\frac{ \Gamma(u) \cap \Gamma(v) }{\min\{ \Gamma(u) ,  \Gamma(v) \}}$	Common neighbors are divided by minimum degree of neighborhood.
Hub depressed	$\frac{ \Gamma(u) \cap \Gamma(v) }{\max\{ \Gamma(u) ,  \Gamma(v) \}}$	Common neighbors are divided by maximum degree of neighborhood.
Leicht–Holme–Newman	$\frac{ \Gamma(u) \cap \Gamma(v) }{ \Gamma(u)  \cdot  \Gamma(v) }$	Common neighbors are divided by square of their geometric mean.

### 3 The Methodology

#### 3.1 The Algorithm

In our algorithm, we used centrality measures and path information to predict new links between nodes. Eigenvector centrality points popular nodes in a network. However, unpopular nodes (with not many connected links) may be more informative and discover strong links due to rarity in real networks. Betweenness centrality shows whether a certain part of a network is centralized or not. Centralized networks have a higher betweenness value since they have controller nodes to which everyone is connected. This situation will lead to less interaction between nodes because their connections will be over central nodes. Decentralized networks can have more shortest paths and can be more flexible. Also, we are not only looking for common neighbors while predicting new links to get information. The path-passed approach can provide more information compared to locally dealing with nodes in a network. Hence, close nodes will serve more possible connections, we are only considering shortest paths between nodes, meanwhile gaining from the complexity.

$$\sum_{z \in s.paths_{u,v}} \sum_{x \in V(z)} \frac{\exp(-c_{eigen}(x)) \cdot c_{betw}(x)^{-1}}{length(z)} \quad (5.1)$$

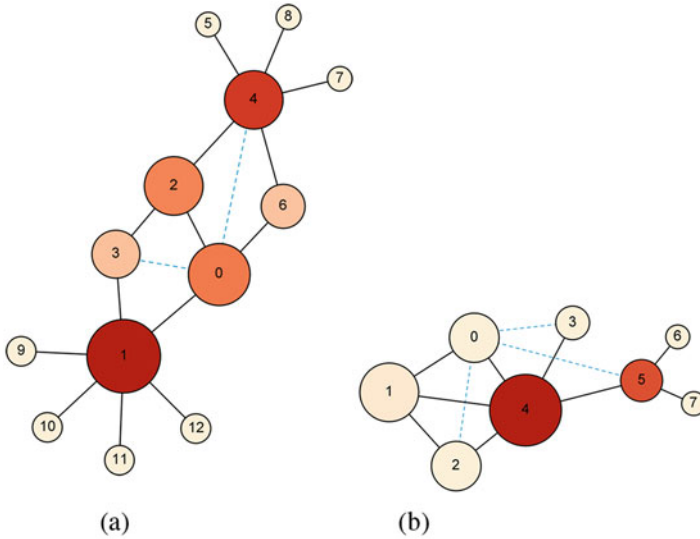
where nodes  $u$  and  $v$  satisfy  $\{u, v \in V | e_{u,v} \in E\}$  in a given network  $G(V, E)$ . ( $V$ ) and ( $E$ ) are sets of vertices and edges, respectively.  $z$  is a shortest path, denoted  $s.paths$ , between  $u$  and  $v$ ;  $x \in V(z)$ .  $c_{eigen}(x)$  is eigenvector centrality of node  $x$ , and  $c_{betw}(x)$  represents betweenness centrality of node  $x$ .  $length(z)$  shows number of hops in path  $z$  between nodes  $u$  and  $v$ .

**Case 1:** In Fig. 5.1a, shortest paths between 0 and 4 pass via nodes 2 and 6, while shortest paths from 0 to 3 pass via nodes 1 and 2. Node 2 is common for both cases. In this situation, the edge from 0 to 4 is more probable than the edge from 0 to 3 since node 6 has lower eigenvector and betweenness centralities compared to node 1. In our algorithm, we used  $exp(x)$  function to avoid negative values and higher weight for rare neighbors. Also, the difference using  $exp$  is insignificant when the value is small, while it is larger when the value is large. The number of hops is 1 in this case since there is no shorter path of length 2.

**Case 2:** In Fig. 5.1b, the edge from 0 to 2 is more probable than the edges from 0 to 3 and from 0 to 5 since it has many more shortest paths than the others.

#### 3.2 Graph Database

In the implementation of our algorithm, we have used a graph database to store the graph network data. Traditionally storing data in a relational database dominated due to its high performance. But recently data types have changed in the Internet



**Fig. 5.1** Illustrative example of our prediction algorithm on a simple network. Size of nodes represents eigenvector centrality (larger size means higher value), while color of nodes shows betweenness centrality (*more reddish* means higher value). **(a)** Case 1. **(b)** Case 2

more towards social networking and big data domains; this involves complex interconnected information. Thus, storing and manipulating complex data has become an issue using traditional relational databases. This motivated for the development of several database structures like graph databases. A graph database provides a method or a tool to model and store a graph related data by focusing on the relationship between entities and attributes of the nodes as basic constructs of a data model [21].

We have used Neo4j tool which is an open source graph database based on Java combining graph storage services and a high performance scalable architecture for graph-based operations. In our work, we used Java libraries provided by Neo4j to create and store the datasets. We also used the graph-based methods in the library in order to get the shortest paths between nodes, eigenvalue, and betweenness centralities of each node.

## 4 Datasets

The best way to test our algorithm is to apply it on real-world networks to check if we can successfully predict links between real entities. Accordingly, we have applied our algorithm on six well-known real-world network datasets where the number of nodes and edges is shown in Table 5.2:

**Table 5.2** Dataset networks size

	# nodes	# edges
Zachary Karate club	34	78
Dolphin social network	62	159
Les Misérables	77	254
Books about us politics	105	441
Word adjacencies	112	425
American college football	115	613

- Zachary Karate Club [22]: is one of the most popular networks in terms of community structure. This network corresponds to members of a karate club at a US university in the 1970s; members are friends. This club has induced fights between its members such that members were split into half. This makes it a perfect real scenario network for link prediction on a member to check to which group he belongs.
- Dolphin Social Network [23]: is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound.
- Les Misérables [24]: is a network corresponding for co-appearance of characters in the novel Les Misérables. It is interesting to test on this network as there are several communities to apply link prediction on them.
- Books About US Politics ([orgnet.com](http://orgnet.com)): is a network of books about US politics sold by Amazon where edges between books represent frequent co-purchasing of books by same buyers.
- Word Adjacencies [25]: is an adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens.
- American College Football [26]: is a network of American football games between Division IA colleges during regular season in Fall 2000.

## 5 Experiments and Results

After collecting the datasets related to the various networks, the following steps are applied to run our algorithm which will output the confusion matrix for the evaluation code:

1. Randomly remove  $\delta$  percentage of edges
2. Run the algorithms presented in Table 5.1 on the new network and get the corresponding confusion matrix
3. Calculate eigenvalue and betweenness centralities for all nodes in the network
4. Run our proposed algorithm
5. Select value  $\alpha$ , which serves as a threshold for the algorithms predicted results.

The results shown in Tables 5.3, 5.4, 5.5, 5.6, 5.7, and 5.8 are average results where for each network we perform Step 1 of removing edges randomly ten times.



**Table 5.3** Karate

	Accuracy			Sensitivity			Specificity			Precision			Miss rate			FIScore		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Ours	0.99	0.99	0.97	0.11	0.10	0.18	0.99	0.99	0.98	0.04	0.11	0.16	0.89	0.90	0.82	0.06	0.09	0.11
Adamic	0.98	0.96	0.95	0.32	0.32	0.39	0.98	0.97	0.96	0.05	0.09	0.12	0.68	0.68	0.61	0.09	0.13	0.18
Jaccard	0.92	0.94	0.96	0.53	0.37	0.35	0.92	0.94	0.96	0.02	0.05	0.11	0.48	0.63	0.65	0.03	0.08	0.17
Dice	0.90	0.93	0.95	0.81	0.53	0.40	0.90	0.93	0.96	0.02	0.06	0.12	0.19	0.47	0.60	0.04	0.10	0.18
Katz1	0.85	0.87	0.89	0.77	0.69	0.66	0.85	0.88	0.89	0.01	0.04	0.08	0.23	0.31	0.34	0.03	0.08	0.14
Katz2	0.97	0.97	0.97	0.15	0.08	0.01	0.97	0.98	0.99	0.01	0.03	0.01	0.85	0.92	0.99	0.03	0.04	0.01
Katz3	0.97	0.97	0.97	0.00	0.00	0.00	0.98	0.98	0.99	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
Common neighbors	0.91	0.93	0.95	0.74	0.53	0.40	0.91	0.93	0.96	0.02	0.06	0.12	0.26	0.47	0.60	0.04	0.10	0.18
Preferential attachment	0.96	0.95	0.95	0.39	0.37	0.43	0.96	0.96	0.96	0.03	0.06	0.12	0.61	0.63	0.57	0.05	0.11	0.19
Salton index	0.90	0.93	0.95	0.81	0.53	0.40	0.90	0.93	0.96	0.02	0.06	0.12	0.19	0.47	0.60	0.04	0.10	0.18
Resource allocation	0.99	0.97	0.96	0.25	0.26	0.33	0.99	0.98	0.97	0.05	0.10	0.13	0.75	0.74	0.67	0.09	0.14	0.18
Hub promoted index	0.90	0.93	0.95	0.81	0.53	0.40	0.90	0.93	0.96	0.02	0.06	0.12	0.19	0.47	0.60	0.04	0.10	0.18
Hub depressed index	0.91	0.94	0.95	0.56	0.38	0.37	0.91	0.94	0.96	0.02	0.05	0.12	0.44	0.62	0.63	0.03	0.08	0.18
Leicht-Holme-Newman index	0.93	0.95	0.96	0.56	0.26	0.25	0.93	0.96	0.97	0.02	0.04	0.10	0.44	0.74	0.75	0.04	0.07	0.15

**Table 5.4** AdjNoun

	Accuracy			Sensitivity			Specificity			Precision			Miss Rate			FIScore			
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	
Ours	1.00	1.00	0.99	0.05	0.09	0.09	1.00	1.00	1.00	0.01	0.04	0.05	0.95	0.91	0.91	0.01	0.01	0.06	0.06
Adamic	0.99	0.99	0.99	0.16	0.38	0.19	1.00	0.99	0.99	0.02	0.04	0.08	0.84	0.62	0.81	0.03	0.03	0.08	0.11
Jaccard	0.99	0.99	0.99	0.23	0.17	0.15	0.99	0.99	0.99	0.01	0.03	0.04	0.77	0.83	0.85	0.02	0.02	0.04	0.07
Dice	0.97	0.98	0.99	0.58	0.40	0.32	0.97	0.98	0.99	0.01	0.03	0.06	0.42	0.60	0.68	0.02	0.02	0.06	0.10
Katz1	0.98	0.98	0.98	0.60	0.66	0.54	0.98	0.98	0.98	0.01	0.04	0.06	0.40	0.34	0.46	0.02	0.02	0.08	0.10
Katz2	0.99	0.99	0.99	0.21	0.21	0.05	0.99	0.99	1.00	0.01	0.05	0.04	0.79	0.79	0.95	0.03	0.03	0.08	0.05
Katz3	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
Common neighbors	0.99	0.98	0.98	0.44	0.52	0.38	0.99	0.98	0.99	0.02	0.04	0.06	0.56	0.48	0.62	0.04	0.04	0.07	0.10
Preferential attachment	1.00	0.99	1.00	0.26	0.38	0.17	1.00	0.99	1.00	0.03	0.09	0.14	0.74	0.63	0.83	0.05	0.05	0.15	0.15
Salton index	0.97	0.98	0.99	0.70	0.41	0.36	0.97	0.98	0.99	0.01	0.03	0.06	0.30	0.59	0.64	0.02	0.02	0.06	0.10
Resource allocation	1.00	0.99	0.99	0.12	0.31	0.15	1.00	0.99	1.00	0.02	0.05	0.09	0.88	0.69	0.85	0.03	0.03	0.09	0.11
Hub promoted index	0.97	0.98	0.98	0.70	0.49	0.38	0.97	0.98	0.99	0.01	0.03	0.06	0.30	0.51	0.62	0.02	0.02	0.07	0.10
Hub depressed index	0.98	0.99	0.99	0.40	0.27	0.23	0.98	0.99	0.99	0.01	0.03	0.05	0.60	0.73	0.77	0.02	0.02	0.05	0.08
Leicht-Holme-Newman index	1.00	1.00	0.99	0.00	0.01	0.05	1.00	1.00	1.00	0.00	0.01	0.03	1.00	0.99	0.95	0.00	0.01	0.01	0.04

**Table 5.5** Dolphins

	Accuracy			Sensitivity			Specificity			Precision			Miss Rate			FIScore		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Ours	1.00	0.99	0.99	0.00	0.08	0.05	1.00	0.99	0.99	0.00	0.05	0.06	1.00	0.92	0.95	0.00	0.06	0.05
Adamic	0.96	0.97	0.98	0.63	0.58	0.31	0.96	0.97	0.98	0.02	0.07	0.10	0.38	0.42	0.69	0.04	0.13	0.15
Jaccard	0.97	0.97	0.98	0.50	0.50	0.29	0.97	0.98	0.99	0.02	0.08	0.11	0.50	0.50	0.71	0.04	0.13	0.16
Dice	0.96	0.97	0.98	0.63	0.58	0.31	0.96	0.97	0.98	0.02	0.07	0.10	0.38	0.42	0.69	0.04	0.13	0.15
Katz1	0.94	0.94	0.96	0.63	0.77	0.38	0.94	0.94	0.97	0.01	0.05	0.07	0.38	0.23	0.63	0.03	0.09	0.11
Katz2	0.98	0.99	0.99	0.13	0.10	0.03	0.99	0.99	0.99	0.01	0.04	0.02	0.88	0.90	0.98	0.02	0.06	0.02
Katz3	0.99	0.99	0.99	0.00	0.00	0.00	0.99	0.99	0.99	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
Common neighbors	0.96	0.97	0.98	0.63	0.58	0.31	0.96	0.97	0.98	0.02	0.07	0.10	0.38	0.42	0.69	0.04	0.13	0.15
Preferential attachment	0.91	0.93	0.96	0.75	0.69	0.34	0.91	0.93	0.96	0.01	0.04	0.05	0.25	0.31	0.66	0.02	0.07	0.09
Salton index	0.96	0.97	0.98	0.63	0.58	0.31	0.96	0.97	0.98	0.02	0.07	0.10	0.38	0.42	0.69	0.04	0.13	0.15
Resource allocation	0.97	0.97	0.98	0.56	0.48	0.31	0.97	0.97	0.98	0.03	0.07	0.10	0.44	0.52	0.69	0.05	0.12	0.15
Hub promoted index	0.96	0.97	0.98	0.63	0.58	0.31	0.96	0.97	0.98	0.02	0.07	0.10	0.38	0.42	0.69	0.04	0.13	0.15
Hub depressed index	0.96	0.97	0.98	0.56	0.58	0.31	0.96	0.97	0.98	0.02	0.08	0.10	0.44	0.42	0.69	0.04	0.14	0.15
Leicht-Holme-Newman index	0.99	0.99	0.99	0.06	0.23	0.13	0.99	0.99	0.99	0.01	0.10	0.11	0.94	0.77	0.88	0.02	0.14	0.12

**Table 5.6** Football

	Accuracy			Sensitivity			Specificity			Precision			Miss Rate			FIScore		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Ours	0.99	1.00	1.00	0.29	0.14	0.11	0.99	1.00	1.00	0.02	0.04	0.09	0.71	0.86	0.89	0.03	0.09	0.11
Adamic	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.13	0.18
Jaccard	0.99	0.99	0.99	0.72	0.74	0.57	0.99	0.99	0.99	0.04	0.09	0.14	0.28	0.26	0.43	0.08	0.08	0.17
Dice	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.10	0.18
Katz1	0.97	0.97	0.98	0.99	0.91	0.76	0.97	0.97	0.98	0.01	0.03	0.07	0.01	0.09	0.24	0.02	0.08	0.14
Katz2	1.00	1.00	1.00	0.67	0.41	0.11	1.00	1.00	1.00	0.04	0.11	0.08	0.33	0.59	0.89	0.08	0.04	0.01
Katz3	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
Common neighbors	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.10	0.18
Preferential attachment	0.96	0.97	0.97	1.00	0.99	0.92	0.96	0.97	0.97	0.01	0.03	0.04	0.00	0.01	0.08	0.02	0.11	0.19
Salton index	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.10	0.18
Resource allocation	0.99	0.99	0.99	0.81	0.73	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.19	0.27	0.43	0.04	0.14	0.18
Hub promoted index	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.10	0.18
Hub depressed index	0.99	0.99	0.99	0.82	0.75	0.57	0.99	0.99	0.99	0.02	0.07	0.13	0.18	0.25	0.43	0.04	0.08	0.18
Leicht–Holme–Newman index	0.99	0.99	0.99	0.82	0.74	0.53	0.99	0.99	1.00	0.02	0.08	0.18	0.18	0.26	0.47	0.05	0.07	0.15

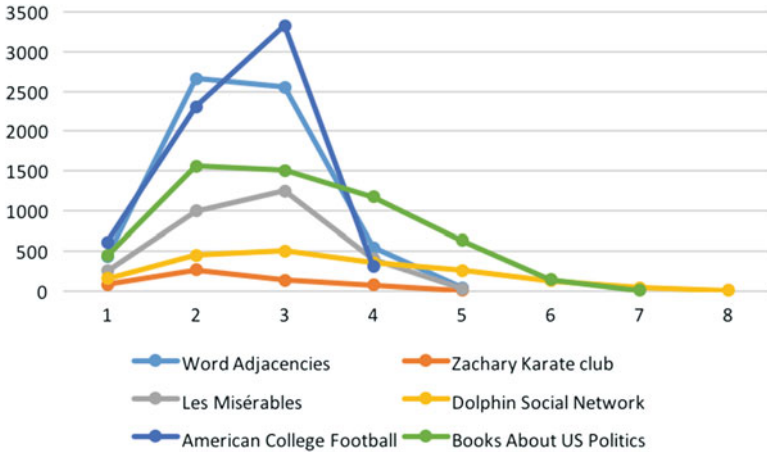
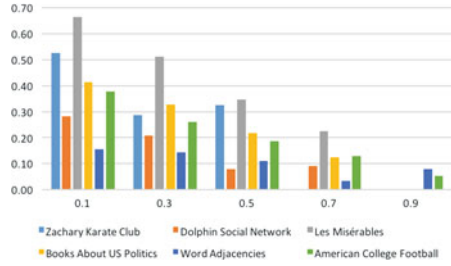
**Table 5.7** Les Misérables

	Accuracy			Sensitivity			Specificity			Precision			Miss Rate			FIScore		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Ours	1.00	1.00	0.99	0.44	0.05	0.27	1.00	1.00	1.00	0.18	0.44	0.20	0.56	0.95	0.73	0.26	0.09	0.23
Adamic	0.99	0.99	0.99	0.80	0.61	0.60	0.99	0.99	0.99	0.05	0.13	0.20	0.20	0.39	0.40	0.09	0.21	0.30
Jaccard	0.98	0.99	0.99	0.76	0.64	0.47	0.98	0.99	0.99	0.03	0.10	0.15	0.24	0.36	0.53	0.06	0.17	0.23
Dice	0.97	0.98	0.98	0.80	0.72	0.58	0.97	0.98	0.99	0.02	0.07	0.14	0.20	0.28	0.42	0.04	0.13	0.23
Katz1	0.97	0.98	0.98	0.60	0.38	0.50	0.97	0.98	0.99	0.02	0.05	0.12	0.40	0.62	0.50	0.03	0.09	0.19
Katz2	0.99	0.99	0.99	0.60	0.39	0.24	0.99	0.99	0.99	0.04	0.10	0.16	0.40	0.61	0.76	0.08	0.16	0.19
Katz3	0.99	0.99	0.99	0.00	0.00	0.00	0.99	0.99	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
Common neighbors	0.99	0.99	0.98	0.76	0.55	0.61	0.99	0.99	0.99	0.05	0.13	0.14	0.24	0.45	0.39	0.09	0.21	0.23
Preferential attachment	0.98	0.99	0.98	0.52	0.34	0.49	0.98	0.99	0.98	0.02	0.06	0.11	0.48	0.66	0.51	0.05	0.10	0.18
Salton index	0.97	0.98	0.98	0.84	0.78	0.60	0.97	0.98	0.99	0.02	0.08	0.14	0.16	0.22	0.40	0.04	0.14	0.23
Resource allocation	0.99	0.99	0.99	0.72	0.58	0.54	0.99	0.99	0.99	0.06	0.15	0.22	0.28	0.42	0.46	0.11	0.24	0.32
Hub promoted index	0.97	0.98	0.98	0.88	0.79	0.60	0.97	0.98	0.99	0.02	0.08	0.14	0.12	0.21	0.40	0.04	0.14	0.23
Hub depressed index	0.98	0.98	0.98	0.76	0.64	0.56	0.98	0.98	0.99	0.03	0.08	0.14	0.24	0.36	0.44	0.06	0.15	0.23
Leicht-Holme-Newman index	0.99	0.99	0.99	0.20	0.28	0.20	0.99	1.00	0.99	0.03	0.12	0.13	0.80	0.72	0.80	0.05	0.17	0.16

**Table 5.8** Polbooks

	Accuracy			Sensitivity			Specificity			Precision			Miss Rate			F1Score		
	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Ours	1.00	1.00	1.00	0.14	0.05	0.07	1.00	1.00	1.00	0.09	0.19	0.11	0.86	0.95	0.93	0.11	0.08	0.09
Adamic	0.99	0.99	0.99	0.77	0.58	0.55	0.99	0.99	0.99	0.03	0.11	0.13	0.23	0.42	0.45	0.06	0.19	0.20
Jaccard	0.98	0.99	0.99	0.84	0.62	0.33	0.98	0.99	0.99	0.02	0.09	0.11	0.16	0.38	0.67	0.05	0.15	0.17
Dice	0.98	0.99	0.99	0.98	0.80	0.55	0.98	0.99	0.99	0.02	0.08	0.13	0.02	0.20	0.45	0.05	0.14	0.20
Katz1	0.98	0.98	0.99	0.77	0.72	0.62	0.98	0.98	0.99	0.02	0.06	0.10	0.23	0.28	0.38	0.04	0.11	0.17
Katz2	0.99	0.99	1.00	0.55	0.36	0.10	0.99	0.99	1.00	0.03	0.09	0.07	0.45	0.64	0.90	0.06	0.14	0.09
Katz3	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
Common neighbors	0.99	0.99	0.99	0.75	0.84	0.55	0.99	0.99	0.99	0.03	0.08	0.13	0.25	0.16	0.45	0.06	0.14	0.20
Preferential attachment	0.98	0.98	0.98	0.59	0.62	0.54	0.98	0.98	0.98	0.01	0.04	0.07	0.41	0.38	0.46	0.03	0.08	0.12
Salton index	0.98	0.99	0.99	0.98	0.83	0.55	0.98	0.99	0.99	0.02	0.08	0.13	0.02	0.17	0.45	0.05	0.14	0.20
Resource allocation	0.99	1.00	0.99	0.73	0.47	0.42	0.99	1.00	0.99	0.04	0.14	0.15	0.27	0.53	0.58	0.08	0.21	0.22
Hub promoted index	0.98	0.99	0.99	0.98	0.84	0.55	0.98	0.99	0.99	0.02	0.08	0.13	0.02	0.16	0.45	0.05	0.14	0.20
Hub depressed index	0.98	0.99	0.99	0.84	0.71	0.40	0.98	0.99	0.99	0.02	0.08	0.11	0.16	0.29	0.60	0.04	0.14	0.18
Leicht-Holme-Newman index	0.99	0.99	1.00	0.86	0.31	0.09	0.99	1.00	1.00	0.03	0.08	0.10	0.14	0.69	0.91	0.06	0.13	0.09

**Fig. 5.2** Clustering coefficient analysis of datasets



**Fig. 5.3** Path distance distribution of datasets

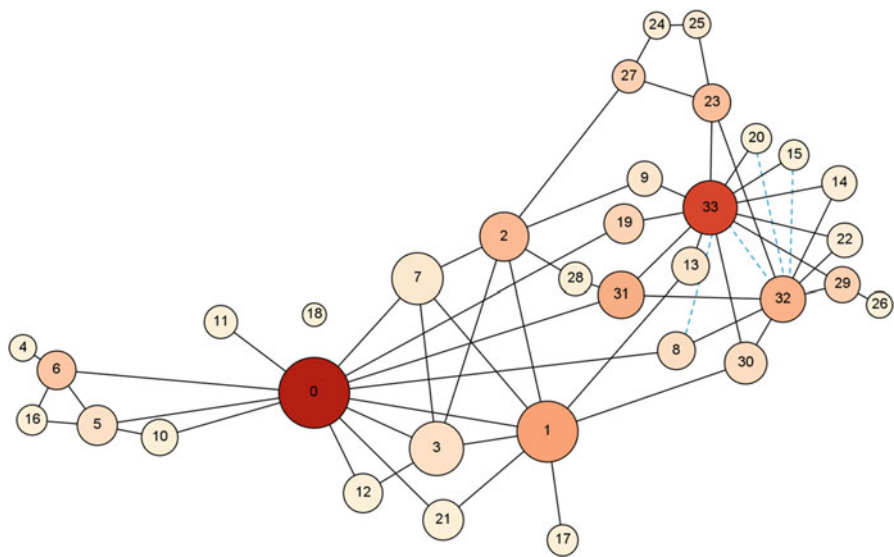
Moreover, we chose three different values of  $\delta$  for removing 10, 30, and 50% of the edges. We decided on these values based on clustering coefficient analysis. This analysis may help in finding missing links between nodes, called structural holes. High degree nodes will have lower local clustering values in this analysis which means more structural holes will exist in the network so that central nodes will collect all the flow of information and reduce alternative paths. In Fig. 5.2, we can see that we got more less local clustering values for 0.7 and 0.9 to the edge removal percentage. In addition, we experimentally used  $\alpha = 0.1$  in all tests as threshold value for accepting an edge as predicted. We did this testing for the networks by choosing  $\ell = 5$ , which is the maximum depth used in all algorithms to search for a shortest paths between two nodes. In Fig. 5.3, we figured the path distance distribution of the datasets to show why we chose 5 as maximum depth. As shown for all networks, the performance of the algorithms are close to each other but our algorithm most of the times reports better values for the evaluation metrics than the other algorithms. In the various tables, *I* means  $\delta = 0.1$ , *II* means  $\delta = 0.3$ , and *III* means  $\delta = 0.5$ .

According to these tables, our precision and F1-score values are better than others in many cases by considerable margins. However, it is hard to decide which

algorithm is better than others from accuracy results since the results are close to each other by small fraction. Also, we can clearly see from specificity results that we are not predicting non-existing links since our results are the higher compared to others, and our dataset is imbalanced which means that the number of negative examples ( $FP + TN$ , connections not to be predicted) is not close to the number of positive examples ( $TP + FN$ , connections to be predicted). Because of this, our sensitivity values are low since we have many false negatives by having more positive examples than negative examples.

To further check into how our algorithm is functioning and the advantage of using social network analysis in investigating the results of link prediction, we show in Fig. 5.4 a sample run of our algorithm on the Karate network with 30% of its edges removed. After removing edges, we ran the evaluation metrics we presented above to investigate the behavior of different algorithms compared to ours. In order to explain how our algorithm is predicting different than others, we used the color of the nodes to represent betweenness values on a white–red scale where white corresponds to low betweenness while red represents high betweenness. We also used size of the nodes to represent eigenvalues where the size of a node is directly proportional to its eigenvalue.

After running the algorithms on this network, we show edges which were successfully predicted by our algorithm as blue dashed lines; these edges connect nodes 32–33, 8–33, 15–32, and 20–32. All other algorithms have predicted the link between nodes 32 and 32 except Leicht–Holme–Newman algorithm. This is because these nodes have a large number of common neighbors (5) facilitating the



**Fig. 5.4** Example of our prediction algorithm on the Karate Network with 30% of the edges removed



prediction of this edge. While none of the other algorithms predicted the existence of the other edges which were successfully predicted by our algorithm, except for Katz which predicted the edge between nodes 8 and 32. This reported result is due to the fact that there is no common neighbors between nodes 8–33, 15–32, and 20–32. Thus, the other algorithms failed to predict these links. While our algorithm successfully predicts the mentioned links because it does not only use common neighbors between two nodes but also considers the sum of all shortest paths between the two nodes.

## 6 Conclusions

In this paper, we tackled the problem of predicting the existence of links in a graph by using network analysis. Finding hidden relationships between actors in a network has various advantages in predicting different future partnership, collaboration, etc., that based on the actors properties can help and accomplish a new trend in the research domain. It also provides the ability to unveil already existing links between people. For example, detecting series of related criminals for security reasons. By performing link prediction using social network analysis, we are able to benefit from existing graph theory algorithms that provide good analytical solutions to the problem. In our paper, we used a combination of shortest path, betweenness, and eigenvalue centralities for the link prediction algorithm. We showed with examples how our algorithm can perform better on real-world datasets than other link prediction algorithms which mostly focus on common neighbors for prediction. We will continue to expand the algorithm by incorporating various other features from the network like closeness, connectedness, etc. We will also investigate the possibility of developing a classifier to help in the process.

## References

1. Wang J, De Vries AP, Reinders MJT (2006) Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 501–508
2. Hotho A, Jäschke R, Schmitz C, Stumme G (2006) Information retrieval in folksonomies: search and ranking. Springer, Berlin
3. Dong L, Li Y, Yin H, Le H, Rui M (2013) The algorithm of link prediction on social network. *Math Probl Eng* 2013:7 pp.
4. Heck T, Peters I, Stock WG (2011) Testing collaborative filtering against co-citation analysis and bibliographic coupling for academic author recommendation. In: Proceedings of the 3rd ACM RecSys' 11 workshop on recommender systems and the social web, pp 16–23
5. Petry H, Tedesco P, Vieira V, Salgado AC (2008) Icare. a context-sensitive expert recommendation system. In: ECAI'08, pp 53–58

6. Reichling T, Wulf V (2009) Expert recommender systems in practice: evaluating semi-automatic profile generation. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 59–68
7. Tayebi MA, Ester M, Glässer U, Brantingham PL (2014) Spatially embedded co-offence prediction using supervised learning. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 1789–1798
8. Benchettara N, Kanawati R, Rouveiol C (2010) Supervised machine learning applied to link prediction in bipartite social networks. In: 2010 international conference on advances in social networks analysis and mining (ASONAM). IEEE, New York, pp 326–330
9. Hasan MA, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: SDM'06: workshop on link analysis, counter-terrorism and security
10. Brandão MA, Moro MM, Lopes GR, Oliveira JPM (2013) Using link semantics to recommend collaborations in academic social networks. In: Proceedings of the 22nd international conference on World Wide Web companion. International World Wide Web conferences steering committee, pp 833–840
11. Chen J, Tang Y, Li J, Mao C, Xiao J (2014) Community-based scholar recommendation modeling in academic social network sites. In: Web information systems engineering–WISE 2013 workshops. Springer, Berlin, pp 325–334
12. Davis D, Lichtenwalter R, Chawla NV (2011) Multi-relational link prediction in heterogeneous information networks. In: 2011 international conference on advances in social networks analysis and mining (ASONAM). IEEE, New York, pp 281–288
13. Tang L, Wang X, Liu H (2009) Uncovering groups via heterogeneous interaction analysis. In Ninth IEEE international conference on data mining, 2009. ICDM'09. IEEE, New York, pp 503–512
14. Radivojac P, Peng K, Clark WT, Peters BJ, Mohan A, Boyle SM, Mooney SD (2008) An integrated approach to inferring gene–disease associations in humans. *Proteins Struct Funct Bioinf* 72(3):1030–1037
15. Heck T (2013) Combining social information for academic networking. In: Proceedings of the 2013 conference on computer supported cooperative work. ACM, New York, pp 1387–1398
16. Sun Y, Barber R, Gupta A, Aggarwal CC, Han J (2011) Co-author relationship prediction in heterogeneous bibliographic networks. In: 2011 international conference on advances in social networks analysis and mining (ASONAM). IEEE, New York, pp 121–128
17. Lopes GR, Moro MM, Wives LK, De Oliveira JPM (2010) Collaboration recommendation on academic social networks. In: Advances in conceptual modeling–applications and challenges. Springer, Berlin, pp 190–199
18. Lichtenwalter R, Chawla NV (2012) Link prediction: fair and effective evaluation. In: Proceedings of the 2012 international conference on advances in social networks analysis and mining (ASONAM 2012). IEEE Computer Society, New York, pp 376–383
19. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *J Am Soc Inf Sci Technol* 58(7):1019–1031
20. Lü L, Zhou T (2011) Link prediction in complex networks: a survey. *Physica A* 390(6):1150–1170
21. Silvescu A, Caragea D, Atramentov A (2002) Graph databases
22. Zachary WW (1977) An information flow model for conflict and fission in small groups. *J Anthropol Res* 33(4):452–473
23. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav Ecol Sociobiol* 54(4):396–405
24. Knuth DE (1993) *The Stanford GraphBase: a platform for combinatorial computing*, vol 37. Addison-Wesley, Reading, MA
25. Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys Rev E* 74(3):036104
26. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. *Proc Natl Acad Sci* 99(12):7821–7826