Appl Intell (2017) 47:837–849
DOI 10.1007/s10489-017-0933-0

# Link prediction in fuzzy social networks using distributed learning automata

Behnaz Moradabadi[1] · Mohammad Reza Meybodi[1]

Published online: 27 April 2017


**Abstract** Link prediction is an area of social network research that tries to predict future links using a social network structure. This paper proposes a novel link prediction method (FLP-DLA) that is based on fuzzy social networks and distributed learning automata (DLA). Distributed learning automata are reinforcement-based optimization tools which try to learn and converge to the optimal behavior from environmental feedback using graph navigation. In the preprocessing phase of the FLP-DLA, the proposed method tries to calculate a fuzzy strength for each link based on the information of the network, such as event time. In the main phase of the FLP-DLA, it uses these fuzzy strengths in addition to DLA to determine the strength of test links. In each iteration of the proposed method, the DLA tries to find a path between the endpoints of a random test link; following this, the FLP-DLA calculates the fuzzy strength of the obtained path using the fuzzy strengths of the links through the path, and rewards or penalizes the DLA based on the path strength. The main phase is repeated until the LAs converge to an action. Finally, we use the strength of the test links as the output of the link prediction. The results reported in this paper have proven satisfactory, indicating the usefulness of the proposed method for some social network datasets.

✉ Mohammad Reza Meybodi
mmeybodi@aut.ac.ir

Behnaz Moradabadi
moradabadi@aut.ac.ir

1 Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

**Keywords** Social network · Fuzzy graph · Learning automata

## 1 Introduction

Today, social networks help people and organizations to exchange information more efficiently. A social network can be considered as a graph in which the nodes are individuals and the edges display some type of communication between the corresponding individuals, such as collaboration, friendship, like or dislike [1, 2] Since this kind of network is generally complex and highly dynamic, it is very important to understand its behavior in time [1, 3]. An area of research called social network analysis (SNA) aims to understand the dynamics of social networks. Link prediction is one of the main tasks undertaken by SNA [1, 4] which focuses on predicting communications that will be generated in the future. Link prediction is applicable to a wide variety of application areas. For example, in the area of the Internet and web science, it can be used in tasks such as automatic web hyperlink creation and website hyperlink prediction [1]. In e-commerce, one of the most prominent usages of link prediction is to build recommendation systems [5]. It also has various applications in other scientific disciplines. For instance, in bibliography and library science, it can be used for deduplication and record linkage [6]; in bioinformatics, it has been used in protein-protein interaction (PPI) prediction [7]. In security-related applications, it can be used to identify hidden groups of terrorists and criminals [2].

All link prediction methods address the following question: "Given a pair of nodes u and v in the current social network, how likely is it that u will interact with v in the future?". To solve the link prediction problem, there are several techniques proposed in the literature [8]: the most common approach is based on applying topological similarity metrics to non-existent links at time t to determine whether a link will appear at a time t' (t' > t). These methods generate scores for each link, and the scores are used to perform prediction tasks either using an unsupervised or a supervised technique. There are many similarity metrics, and these are briefly explained in Section II. The major challenge of standard link prediction methods is that they do not consider network evolution or repeated links; they only use the current network structure without considering the frequency and creation time of the links. Our work tries to overcome these problems by using both the fuzzy concept to model the strength of links and DLA to predict future links. The novelty of the proposed method is that it uses the fuzzy concept to present the link strength.

A learning automaton is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of allowed actions, through repeated interactions with a random environment [9]. Distributed learning automata (DLA) are networks of interconnected learning automata which collectively cooperate to solve a particular problem. In each iteration, an automaton randomly chooses one of its outgoing edges (actions) according to its action probabilities and activates the learning automaton at the other end of the selected edge. The activated automaton in turn randomly selects an action, which results in the activation of another automaton. The process of choosing the actions and activating automata is continued until a criterion is reached. The chosen actions, along with the path, are applied to the random environment. The environment evaluates the applied actions and generates a reinforcement signal to the LAs. The LAs along the chosen path update their action probability vectors on the basis of the reinforcement signal using a learning algorithm. These steps are repeated until the LAs converge to some action.

In this paper, to predict the future links a new link prediction is proposed which uses both fuzzy social networks and DLA (FLP-DLA). The proposed method has two steps: in the initialization phase, it models the strength of each link as a fuzzy variable by using the time and frequency information of the link. Thus, we have a fuzzy social network where each link has a fuzzy strength and we use this network as the input of the next step. The main phase of FLP-DLA uses distributed learning automata to predict the future links based on graph navigation. To do this in FLP-DLA, there is one LA for each node in the network. In each iteration, the LAs try to form a path in the network such that the link between the starting ($s_p$) and ending ($e_p$) nodes of the path belong to some test link ($s_p$, $e_p$). After a path is created, the strength of the path is calculated based on the fuzzy links in the path. FLP-DLA then assigns the strength of the path to the strength of test link ($s_p$, $e_p$), and based on this strength it rewards or penalizes each LA in the path. The main phase is repeated until the LAs converge to some paths. Finally, it uses the strength of the test links as the output of the link prediction. In order to examine the results of the proposed method, several evaluations are conducted, and the results of the proposed method are compared with those achieved by other link prediction techniques. In general, the experiments show that our approach performs better than other strategies.

The rest of this paper is organized as follows. Section 2 reviews the literature regarding the link prediction problem. Learning automata and distributed learning automata are described in Section 3. Section 4 introduces the proposed DLA-based link prediction method. Section 5 presents the experimental study for some social network datasets. Section 6 summarizes the main conclusion of our research.

## 2 Link prediction

A classic definition of the link prediction problem is expressed by: "Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time *t* to a given future time *t+1*" [2]. The most widespread approach to the problem is to explore the topological/structural patterns of the social network of interest [10]. Various metrics to describe node pairs have already been adopted in previous studies [11]; these explore the structural patterns of the network and commonly provide a degree of proximity/similarity between the nodes. There are many similarity metrics, including [2]: 1) local similarity metrics that only use the local information of a link to calculate the similarity metric, such as Common Neighbors, Salton Index, Jaccard Index, Hub Depressed Index, Hub Promoted Index, Leicht-Holme Newman Index (LHN1), Preferential Attachment Index, Adamic-Adar Index and Resource Allocation Index; 2) global similarity metrics that can use all the information in the network to calculate the similarity metric between two nodes, such as the Katz Index, Leicht-Holme-Newman Index (LHN2) and Matrix Forest Index (MFI); and 3) quasi-local metrics that do not require global topological information but use more information than local indices, such as Local Path Index, Local Random Walk, Superposed Random Walk, Average Commute Time, Cos+, random walk with restart, SimRank, Resource Allocation Index and Local Path Index. As previously mentioned, the starting point in these approaches is to extract the values/scores of different metrics that represent the proximity of pairs of nodes. Then, pairs of non-connected nodes are ranked according to a chosen metric (for instance the number of common neighbors) [2]. After that, the top L ranked pairs are assigned as predicted links. To put it another way, it is always assumed that the links which have the highest scores are most likely to occur. In the following, the seven most commonly used similarity measures that are used in the experiments are briefly presented.

1) Common Neighborhood: In this measure, two nodes x and y are more likely to have a link if they have many common neighbors. This score is defined as

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

where $\Gamma(x)$ denotes the neighbors of node x.

2) Salton: This score is defined as

$$Salton(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \times |\Gamma(y)|}}$$

3) Jaccard Index [12]: This index was proposed by Jaccard, and is defined as:

$$Jaccard(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

4) Preferential Attachment (PA) [13]: The preferential attachment (PA) algorithm is based on the preferential attachment phenomena rule [14] that is discovered in a variety of social networks. In this method, the link score is set to be the product of the degrees of the involved nodes and it is defined as follows:

$$PA(x, y) = |\Gamma(x)| \times |\Gamma(y)|$$

5) Adamic-Adar Index (AA) [15]: This index is an extension of the common neighborhood method such that the less-connected neighbors have more weight. It is defined as:

$$AA(x, y) = \sum\nolimits_{Z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\Gamma(Z))}$$

6) Katz Index: In this metric, a similarity is defined as the sum of the number of paths with different lengths, such that shorter paths have more weights. It is defined by the following equation:

$$Katz(x, y, ) = \sum\nolimits_{l=1}^{\infty} \beta^l . |Path(x, y)^{<l>}|$$

where $|Path(x, y)^{<l>}|$ is the number of paths between x and y with length $l$. It is also shown that the Katz metric can be calculated based on the following equation:

$$Katz = (I - \beta A)^{-1} - I$$

7) LP Index: This index is a restricted version of the Katz metric such that only paths of length 1 and 2 are considered. This metric has a lower computational complexity in comparison to Katz and is defined as the following:

$$LP\ Index(x, y) = A^2 + \epsilon A^3$$

For more information about other similarity metrics, please refer to Liben-Nowell and Kleinberg [2]

The node-wise similarity-based approach searches appropriate measures of similarity between two nodes according to the content and/or semantics they present [2]. Each node on the network can be represented as a vector of features. The more similar two nodes are in terms of their particular attributes, the likelier they are to relate. The cosine coefficient, mutual information, and Dice coefficient [2] are examples of techniques used in this approach.

The approaches that are based on probabilistic models try to learn the best probabilistic model that abstracts the network information. The basic idea here is to create the model through a set of parameters $\theta$, given the observed social network G=(V,E) [2]. The existence of a link between the pair of nodes x and y is estimated by the conditional probability P(e(x,y)|$\theta$) [2]. This approach examines the elements of the network with the help of relational data models and enables us to encapsulate relevant information from node relationships. Relational Markov networks and relational Bayesian networks are two examples of models which use this approach.

In the rest of this section, we go on to review more recently proposed link prediction methods.

The authors of [16] proposed a data mining process called interaction prediction, which addresses a particular formulation of the link prediction problem for dynamic networks. Their approach predicts future interactions by combining dynamic social networks analysis, time series forecasts, feature selection such as similarity metrics, and network community structures. The proposed method focuses on the links within communities. Experiments on real-world interaction networks show that the proposed approach achieves encouraging results in cases of both balanced and unbalanced class distributions.

The authors of [17] have provided an approach for predicting future links by applying the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to optimize the weights used in a linear combination of sixteen neighborhood and node similarity indices. They examine a large dynamic social network with over $10^6$ nodes. Their method exhibits fast convergence and high levels of precision for the top twenty predicted links.

The authors of [18] proposed a new link prediction using information theory and mutual information of network structure (MI-LP). They compared the model with six typical prediction methods on ten networks. Their results show two good features, improving both the link prediction accuracy and the reasonable computational complexity.

In [19], a novel method called Multivariate Time Series Link Prediction is proposed for link prediction in evolving networks. This method integrates (1) the temporal evolution of the network; (1) node similarities, and (3) node connectivity information. This method uses existing connections and a calculated similarity metric for each time.

Finally, the authors compare different similarity metrics in their experiments.

In [20], a weighted approach for modeling the occurrence of time is used to generate a different similarity metric. Finally, in our previous research, a new link prediction method based on temporal similarity metrics and Continuous Action Set Learning Automata (CALA) is proposed [21]. The proposed method takes advantage of the use of different similarity metrics as well as different time periods. In the proposed algorithm, the link prediction problem is modeled as a noisy optimization problem, and a team of CALAs is used to solve the noisy optimization problem. The obtained link prediction results show a satisfactory application of the proposed method for some social network datasets.

Traditional social network analysis is based on deterministic models, calculating the value of some measures to analyze the relationship that exists between them. However, a simply determined value is not sufficient to describe these relations accurately. Since the fuzzy concept was proposed [22], several studies have shown that the introduction of the fuzzy system, using a fuzzy state to replace the original value, can appropriately solve the problem of uncertainty in social networks. The research in [23] used fuzzy logic to modify the original binary relations into multiple relations among individuals and to achieve the same efficiency in social networks. Reference [24] presented a method using fuzzy logic to explain social relations. These methods can improve the flexibility of the relationship between social networks, and thus reduce the conflict between individuals. The authors of [25] applied a fuzzy model for link prediction based on network characteristics, and achieved better results than those achieved by the traditional method. The introduction of a fuzzy model through fuzzy clustering to predict social network links also shows good performance [26]. Recently, some researchers have used the ordered weighted averaging (OWA) operator to obtain the fuzzy relationship between nodes [27]. This method needs some attributes of the network to calculate the relationship. Similarity indices such as common neighbors (CN), Katz, Salton, or Adamic-Adar (AA) can be considered attributes here. Although the method has the disadvantage of a large time complexity, it has the advantages of higher prediction accuracy and higher stability. The proposed method uses the fuzzy concept to model the strength of the links in social networks, and achieves better results in predicting future links in social networks.

## 3 Learning automata

A learning automaton [28] is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of allowed actions through repeated interactions with a random environment. The action is chosen at random, based on a probability distribution over the action set, and at each instant the given action serves as the input to the random environment. The environment responds in turn to the action taken with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action set so that the average penalty received from the environment is minimized. The environment can be described by a triple $E \equiv \{\alpha, \beta, c\}$, where $\alpha \equiv \{\alpha_1 \alpha_2, ..., \alpha_r\}$ represents the finite set of the inputs, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ denotes the set of the values that can be taken by the reinforcement signal, and $c \equiv \{c_1, c_2, \ldots, c_m\}$ denotes the set of the penalty probabilities, where the element $c_i$ is associated with the given action $\alpha_i$. If the penalty probabilities are constant, the random environment is said to be a stationary random environment, and if they vary with time, the environment is called a non-stationary environment. The environments depending on the nature of the reinforcement signal $\beta$ can be classified into P-models, Q-models and S-models. The environments in which the reinforcement signal can only take two binary values, 0 and 1, are referred to as P-model environments. Another class of the environment which allows a finite number of values in the interval [0, 1] can be taken by the reinforcement signal; such an environment is referred to as a Q-model environment. In S-model environments, the reinforcement signal lies in the interval $[a, b]$. The relationship between the learning automaton and its random environment is shown in Fig. 1 [28]. Learning automata can be classified into two main families [28]: fixed structure learning automata and variable structure learning automata. Variable structure learning automata are represented by a triple $<\beta, \alpha, T>$, where $\beta$ is the set of inputs, $\alpha$ is the set of actions, and $T$ is the learning algorithm. The learning algorithm is a recurrence relation which is used to modify the action probability vector. Various learning algorithms have been proposed. The proposed method uses the following S-model learning: Let $\alpha$ be the action chosen at step n from the probability distribution $p$. The linear reward-inaction

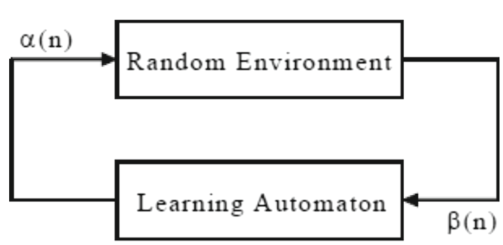


**Fig. 1** The relationship between a learning automata and its random environment

algorithm is one of the learning schemas, and its recurrence equation for updating the action probability vector $p$ is defined as the following equation:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a.(1-\beta(n)).\\ &\quad (1-p_i(n)) - b.\beta(n)\, p_i.(n)\\ p_j(n+1) &= p_j(n) + a.(1-\beta(n)).p_j(n)\\ &\quad + \frac{b.\beta(n)}{r-1} - b.\beta(n).p_i(n)\; j \neq i \end{aligned} \tag{1}$$

where $a$ and $b$ denote the reward and penalty parameters and determine the amount of increase and decrease of the action probabilities, respectively.

Learning automata have been found to be useful in systems where incomplete information exists about the environment wherein the system operates and recently, several learning automata-based approaches have been presented for improving the performance of many applications [29–31].

### 3.1 Distributed learning automata

A distributed learning automata (DLA) [32, 33], as shown in Fig. 2, is a network of interconnected learning automata which collectively cooperate to solve a particular problem. Formally, a DLA can be defined by a triple $<A, E, T>$, where $A = \{A_1, A_2, ..., A_n\}$ is the set of learning automata, $E \subset A \times A$ is the set of the edges in which edge $e(i, j)$ corresponds to the action $\alpha_{ij}$ of the automaton $A_i$, and $T$ is the set of learning schemes with which the learning automata update their action probability vectors. The operation of a DLA can be described as follows. At first, a random node is chosen as the root automaton, and it randomly chooses one of its outgoing edges (actions) according to its action probabilities and activates the learning automaton at the other end of the selected edge. The activated automaton also randomly selects an action which results in the activation of another automaton. The process of choosing the actions and activating the automata is continued until a leaf automaton (an automaton which interacts with the environment) is reached. The chosen actions, along the path induced by the activated automata between the root and leaf, are applied to the random environment. The environment evaluates the applied actions and emits a reinforcement signal to the DLA. The activated learning automata along the chosen path update their action probability vectors on the basis of the reinforcement signal using the learning schemes. This procedure repeats until the probability of the chosen paths is close enough to unity. For example, in Fig. 2, every automaton has two actions. If automaton $A_1$ selects $\alpha_3$ from its action set, then it activates automaton $A_3$. Afterwards, the automaton $A_3$ will choose one of its possible actions and so on.

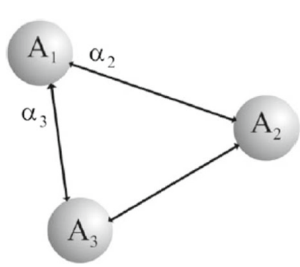


**Fig. 2** Distributed Learning Automata

## 4 Proposed link prediction method in fuzzy social networks

This section describes the proposed link prediction method for fuzzy social networks. To do this, we first review the fuzzy concepts that are used in the proposed method. Then, we describe the preprocessing phase to show how the proposed method models one social network into a fuzzy social network. After that, we propose a link prediction algorithm based on the fuzzy social network.

### 4.1 Fuzzy concepts

The fuzzy set theory, introduced by Zadeh [22], is suitable for dealing with the uncertainty and imprecision associated with information concerning various parameters. In this section, we briefly review the main concepts and definitions of fuzzy sets, fuzzy variables and the fuzzy social networks.

**Definition 1** A fuzzy set $A$ in $R$ (real line) is defined to be a set of ordered pairs $A = \{(x, \mu_A(x)) \mid x \in R\}$, where $\mu_A(x)$ is called the membership function for the fuzzy sets.

**Definition 2** A fuzzy set $A$ is called **normal** if there is at least one point $x \in R$ with $\mu_A(x) = 1$.

**Definition 3** A fuzzy set $A$ on $R$ is **convex** if for any $x, y \in R$ and any $\lambda \in [0,1]$, we have $\mu_A(\lambda x + (1-\lambda)y) \geq \min(\mu_A(x)\, \mu_A(y))$.

**Definition 4** A fuzzy **number** is a fuzzy set on the real line that satisfies the conditions of **normality** and **convexity**.

**Definition 5** A **L-R** fuzzy number denoted by $M = (m, \alpha, \lambda)$, has the following membership function:

$$\mu_M(x) = \begin{cases} 0 & x \leq m-\alpha \\ 1 - \dfrac{m-x}{\alpha} & m-\alpha < \mathrm{x} < \mathrm{m} \\ 1 & x = m \\ 1 - \dfrac{x-m}{\lambda} & m < x < m+\lambda \\ 0 & m+\lambda \leq \mathrm{x} \end{cases}$$

**Definition 6** ([34]) A fuzzy social network is defined as a fuzzy relational structure $\tilde{G} = \left(V, \tilde{E}\right)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a non-empty set of actors or nodes and $\tilde{E} = \begin{bmatrix} \widetilde{e_{11}} & \cdots & \widetilde{e_{1n}} \\ \vdots & \ddots & \vdots \\ \widetilde{e_{n1}} & \cdots & \widetilde{e_{nn}} \end{bmatrix}$ is a fuzzy relation on V. Also $\widetilde{\mu_E} = \begin{bmatrix} \mu(\widetilde{e_{11}}) & \cdots & \mu(\widetilde{e_{1n}}) \\ \vdots & \ddots & \vdots \\ \mu(\widetilde{e_{n1}}) & \cdots & \mu(\widetilde{e_{nn}}) \end{bmatrix}$ is a membership function. In other words, each $\widetilde{e_{ij}}$ is a fuzzy variable that shows a relationship between nodes i and j. The proposed method uses the concept of the strength of the links as a membership function to transform a social network into a fuzzy social network, where each link has a fuzzy number that shows the strength of the link.

The procedure of creating a fuzzy social network is described in detail in the next section.

### 4.2 Preprocessing phase

This section describes how the FLP-DLA models a static social network into a fuzzy one. The construction of a fuzzy social network based on activity records is done by capturing the activities and interactions among the nodes of the current network. An activity record is defined as a pair: <Date, Co-No>. Date represents the date of the link occurrence and Co-No represents the number of nodes that are evolving with the corresponding activity. For example, in co-authorship social networks, the Co-No is the number of authors that have collaborated in publishing the corresponding paper, and in email networks the Co-No is the number of recipients of an email. An example of an activity record looks like this:

$< 2002/02/13, 3 >$

This paper tries to calculate a fuzzy strength for each link based on the Date and Co-No variables. To do this, it first converts the Date of each activity to a fuzzy one by defining a fuzzy variable for the Date variable. Different experts define fuzzy sets in various ways. Here, Date is defined as a fuzzy variable like the following fuzzy sets: if the social network is used in the time interval $[t\, t']$, then this time interval is split into three sections: {very-old, old, recent}. For example, if the time interval of the used social network is [1994–2002], the Date fuzzy variable is represented by the fuzzy sets in Fig. 3.

From Fig. 3, it can be observed that three fuzzy sets have been defined: very-old, old and recent. Each of these fuzzy sets can be represented using a triangular fuzzy number.

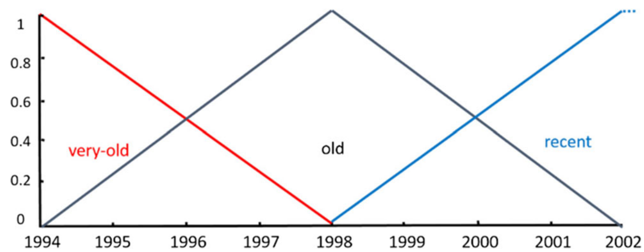


**Fig. 3** Fuzzy Sets for Date Fuzzy Variable

Table 1 shows the value of the fuzzy sets representing the date variable. The FLP-DLA uses the L-R fuzzy number notation, where a fuzzy number is defined as a tuple (m, $\alpha$, $\lambda$), where m is the mean value, $\alpha$ is the left spread, and $\lambda$ is the right spread [22].

Now that we have defined a fuzzy number for each of the possible values assigned to the date variable, we can redefine our social activity records as follows:

$< 2002/02/13, 3 > \rightarrow < \text{recent}, 3 > \rightarrow < (2002, 4, 0), 3 >$

We have thus introduced an approach to convert the Date variable into an L-R fuzzy number. Now we want to use this fuzzy value and the Co-No information together to calculate the link strength. We define the link strength as an L-R fuzzy variable, S(n, $\tau$, $\upsilon$), where n is the mean value, $\tau$ is the left spread and $\upsilon$ is the right spread, and we propose the following equation to obtain the link strength:

$$n = x_m \times e^{-\frac{Co-No}{2}}, \tau = x_\alpha, \upsilon = x_\beta \tag{2}$$

where x is the Date fuzzy number, $x_m$ is the mean value of the x, $x_\alpha$ is the left spread, $x_\beta$ is the right spread and Co-No represents the cooperation number of the link. This function is designed such that the newer papers with lower cooperation number have a higher strength. For example, assume we have a social activity in the form of $< (2002, 4, 0), 3 >$. The strength of the link based on (2) is the L-R fuzzy number S(0.2231, 4, 0). Thus, using (2), we finally have a fuzzy social network in which each link has an L-R fuzzy strength. Figure 4 shows an example of converting a social network to a fuzzy social network. In the next step, we use this fuzzy social network to predict future links.

**Table 1** L-R fuzzy numbers for the Date variable

| Fuzzy Set | L-R Fuzzy Number |
|---|---|
| Very-old | (1994, 0, 4) |
| Old | (1998, 4, 4) |
| Recent | (2002, 4, 0) |

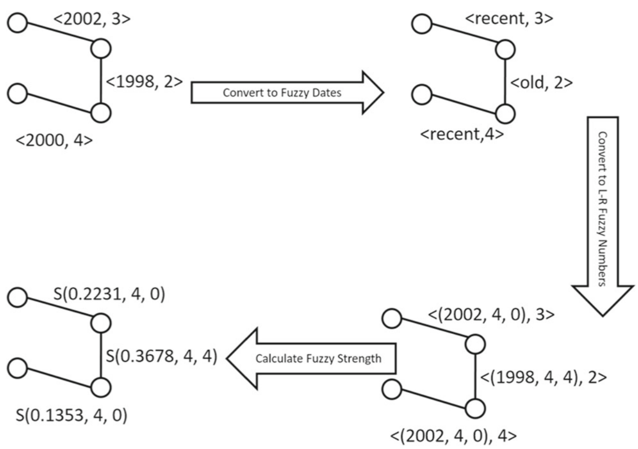


**Fig. 4** Example of Converting a Social Network to a Fuzzy Social Network

### 4.3 Proposed fuzzy link prediction

This section describes the proposed link prediction based on the fuzzy social network that was introduced and generated in the two previous sub-sections. The proposed method takes advantage of DLA to predict future links according to the following procedure: in the beginning, each node of the graph is equipped with a learning automaton, and as a result, a network of learning automata isomorphic to the graph is initially constructed. The set of distributed learning automata is defined by the tuple $< A\alpha >$, where $A \equiv \{A_1 A_2, \ldots, A_n\}$ is the set of learning automata corresponding to the set of nodes, and $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ denotes the set of actions in which $\alpha_i \equiv \left\{\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{r_i}\right\}$ is the set of actions that can be taken by the learning automaton $A_i$. In the proposed algorithm, the action set of each learning automaton (e.g. $A_i$) is to choose one of the neighbors $j$ based on probability $p_{ij}$ that is initialized as:

$$p_{ij} = \frac{1}{d(v_i)} \tag{3}$$

where $d(v_i)$ is the degree of vertex $v_i$.

At the beginning of the algorithm, all learning automata are considered inactive. The FLP-DLA first chooses a node $A_i$ as the start node from the set of nodes $N_{test}$. The $N_{test}$ is the set of end-point nodes of links that must be predicted. Automaton $A_i$ is selected and activated. The activated automaton selects one of its neighbors ($A_j$) according to the probability vector $A_i$. Then, $A_j$ selects one of its neighbors $A_k$ according to the probability vector of $A_j$. This procedure is continued until node $A_k$ belongs to the $N_{test}$ set. Thus, there is a fuzzy path $P_r$ between $A_i$ and $A_k$ such that the pair $(A_i A_k)$ belongs to our test set. Now, the FLP-DLA calculates the strength of path $P_r$, $S(P_r)$ based on the following equation:

$$S(P_r) = \frac{\sum_{l \in links in P_r} EV(S_l)}{k} \tag{4}$$

where $EV(S_l)$ is the expected value of the strength of link $l$and $k$ is the length of $P_r$. Since the strength of each link is an L-R fuzzy number, the expected value of the strength of link $l$, $EV(S_l)$ is calculated based on the following equation:

$$EV(x) = n - \frac{\upsilon - \tau}{3} \tag{5}$$

In other words, the strength of path $P_r$ is equal to the average of the expected strength values of the links in the path $P_r$ Now, FLP-DLA normalizes this strength to a number value in the interval [0, 1] to generate a $\beta$ signal. Then, it rewards or penalizes all the learning automata in path $P_r$ based on the $\beta$ value using (1). It also assigns the obtained

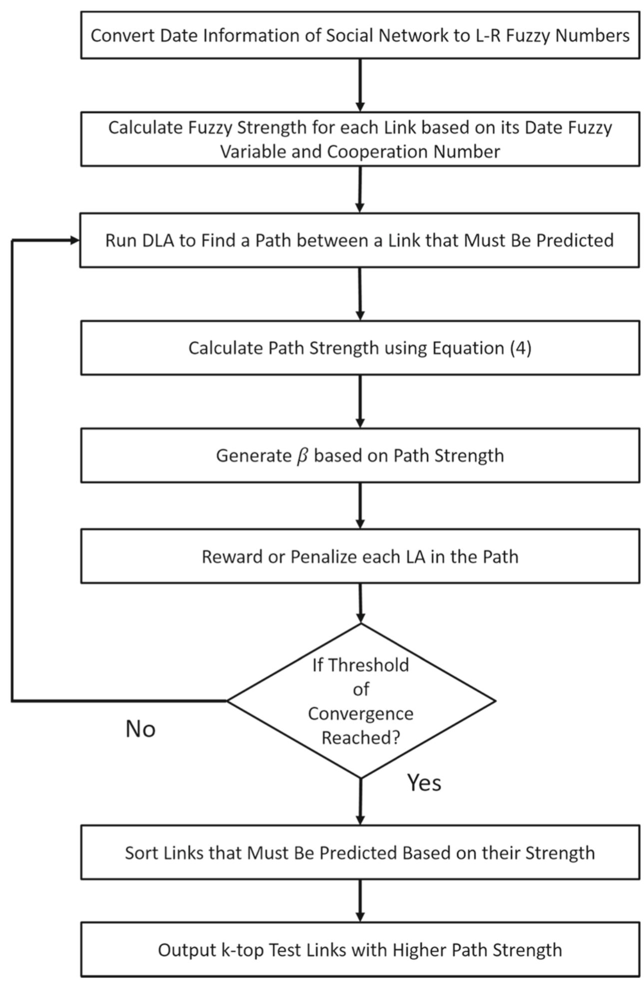


**Fig. 5** Pseudocode of the Proposed Link Prediction in Fuzzy Social Network based on Distributed Learning Automata

path strength to the link strength. Then it again chooses a node from set $N_{test}$ and creates a fuzzy path. This process is iteratively repeated until a predefined number of iterations are carried out or the process exceeds a predefined threshold of convergence. The threshold of convergence is defined by the product of the action probability vector of the learning automata, as follows:

$$\tau = \prod_{i \in T} \arg\max\{p_i\ (t)\} \tag{6}$$

where $t$ is the iteration number, and $T$ is the set of training nodes in the graph. In addition, $p_i\ (t)$ shows the probability vector of automaton $i$ in iteration $t$.

At the end of this process, k-top test links with higher strength will be taken as the future links. The pseudocode of the proposed algorithm based on the distributed learning automaton is shown in Figs. 5 and 6. In Fig. 5, a set of learning automata isomorphic to the nodes of the graph is first considered, and the Date information of the social network used is converted to a set of L-R fuzzy numbers. Then for each link, the fuzzy strength of the link is calculated based on (1). In each iteration of the FLP-DLA, a random start node is chosen for the DLA, and the DLA tries to find a path between the endpoints of a test link. When this path is created, the DLA stops and calculates the strength of the obtained path using (4) and (5). This strength is used as the reinforcement signal to reward or penalize each LA in the path. Finally, the convergence metric of the DLA is calculated based on (6). Until the convergence metric reaches a predefined threshold, a random start node is selected again and the procedure of finding the path and rewarding or penalizing the DLA is repeated.

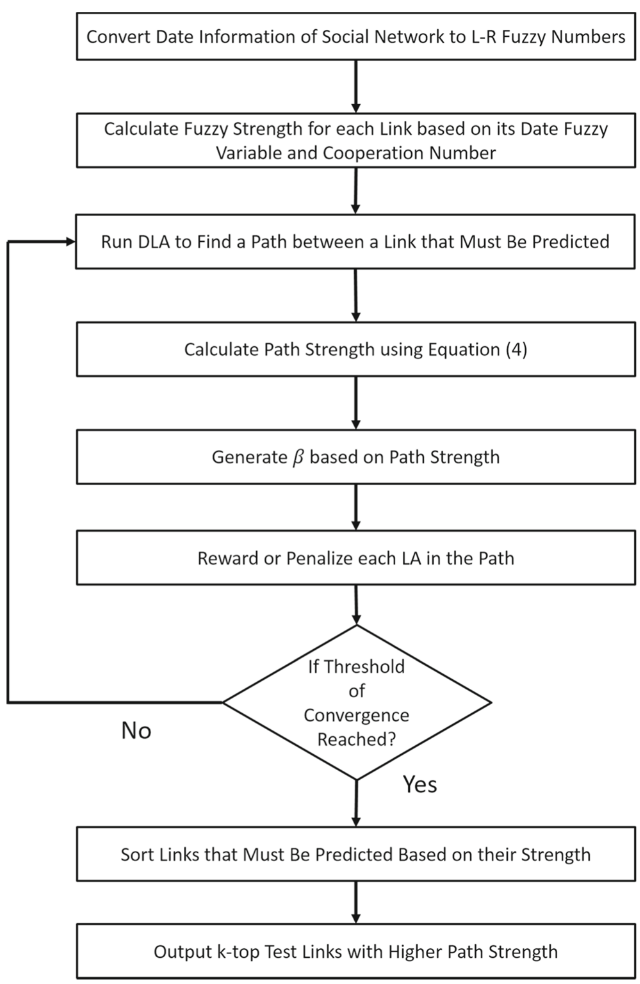


**Fig. 6** Block Diagram of the Proposed Link Prediction in Fuzzy Social Networks based on Distributed Learning Automata

## 5 Experiment results

In this section, in order to evaluate the performance of the proposed algorithm, several computer experiments have been conducted and the performance of the proposed algorithm is compared in terms of performance and accuracy. The rest of this section first gives the dataset and evaluation metrics that are used in the experiments and then gives a set of experiments.

### 5.1 Data set and evaluation metrics

In this section, the data of the social network and the evaluation metrics used in our experiments are described. For the experiments developed in this work, we consider the following two groups of networks:

- Co-Authorship Networks: A type of social network where the nodes represent the authors and two authors are connected if they have collaborated on a paper. The collaboration network is widely used to understand the topology and dynamics of complex networks. This paper uses three co-authorship networks from three sections of Arxiv[1] and extracted data from the years 1993 to 2003 for all the datasets. The first network is composed of authors that collaborated in theoretical high energy physics[2] (hep-th). The second one is formed of authors who published papers in high energy physics[3] (hep-ph) and the third is sampled from collaborations in astrophysics[4] (Astro-ph). In these datasets, if author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored

[1] http://www.arxiv.org
[2] http://arxiv.org/archive/hep-th
[3] http://arxiv.org/archive/hep-ph
[4] http://arxiv.org/archive/Astro-ph

**Table 2** Network Size in Terms of Nodes and Edges

| Data Set | Nodes | Edges | Description |
|---|---|---|---|
| Hep-th | 9,877 | 51,971 | Collaboration network of Arxiv High Energy Physics Theory |
| Hep-ph | 12,008 | 237,010 | Collaboration network of Arxiv High Energy Physics |
| Astro-ph | 18,772 | 396,160 | Collaboration network of Arxiv Astro Physics |
| Email-Enron | 36,692 | 367,662 | Email communication network from Enron |
| Email-EuAll | 265,214 | 420,045 | Email network from a EU research institution |

by k authors, this generates a completely connected (sub) graph on k nodes.

- Email Communication Networks: This is a type of social network where the nodes of the network are email addresses. If an address i has sent at least one email to address j, the graph contains an undirected edge from i to j. In our experiments, we use two email communication data sets: the Enron email communication network[5] and Eu-All email communication network.[6] The Enron email communication network includes a dataset of around half a million emails that were made public by the Federal Energy Regulatory Commission, and we extract data from May 1999 through May 2002 (36 months). In addition, the Eu-All email communication network was extracted using email data from a large European research institution, and data were extracted from October 2003 to May 2005 (18 months). The network specification of each dataset is presented in Table 2. Since these networks are highly sparse, in order to make computation feasible we reduce the number of candidate pairs by choosing only the ones that have at least two connections on the network.

In order to carry out experiments for collaboration networks (Hep-th, Hep-ph and Astro-ph), the proposed method considers the data from 1993 to 2002 as the training data (taking each year as a time period) and the year 2003 as test data. Additionally for email networks (Enron and EuAll), it considers the first 70% of the available months as training data (each month as a time period) and the remaining 30% as the test data. Because the FLP-DLA is a stochastic method, all the results reported in the next sections are calculated based on the mean of 30 random runs of the FLP-DLA.

[5] http://www.cs.cmu.edu/~enron/

[6] http://snap.stanford.edu/data/email-EuAll.html

To evaluate the proposed method using other link prediction methods, we use the two common evaluation metrics as follows:

- AUC Metric [35]: If we rank all of the non-existent links based on their scores, the AUC metric can be interpreted as the probability that a random missing link has a higher score than a random non-existent link. In the algorithmic implementation, at each point in time it randomly picks a missing link and a nonexistent link and compares their scores. If from n independent comparisons, there are $n'$ times that missing links have a higher score and $n''$ times that they have the same score, the AUC value is:

$$AUC = \frac{n' + 0.5n''}{n} \tag{7}$$

  If the AUC value has a value of more than 0.5, it is better than the random link prediction algorithm; the farther the value from 0.5, the more accurate the algorithm.

- Precision [35]: If we predict L links to be connected and $L_r$ links from L links are right, the precision is defined as:

$$Precision = \frac{L_r}{L} \tag{8}$$

Clearly, higher precision means higher prediction accuracy.

### 5.2 Experiments

This section presents three experiments to evaluate the performance and accuracy of the proposed method. The first experiment compares the accuracy of the FLP-DLA with the other link prediction methods. The second experiment compares the metrics of the predicted social network with the original one. Finally, the third experiment compares the computational complexity of the proposed method.

#### *5.2.1 Experiment 1: FLP-DLA comparison with other link prediction methods*

This section compares the proposed FLP-DLA with some classical and recent link prediction methods. In order to improve the comparison, a set of algorithms in three categories is chosen:

- Common Similarity-Based Link Predictions: in this group of methods, the following common similarity-based link prediction methods are chosen for comparison with the proposed method: CN, Salton, Jaccard, PA, AA as local similarity metrics, Katz as the global

**Table 3** AUC Measures of Proposed FLP-DLA and Other Link Prediction Methods

| Method/Data Set | Hep-th | Hep-ph | Astro-ph | Enron | EuAll | Average |
|---|---|---|---|---|---|---|
| CN | 0.7945 | 0.7025 | 0.6791 | 0.8123 | 0.6643 | 0.7305 |
| Salton | 0.7850 | 0.6854 | 0.6441 | 0.8087 | 0.6285 | 0.7103 |
| Jaccard | 0.6438 | 0.6026 | 0.5719 | 0.7010 | 0.6259 | 0.6290 |
| PA | 0.6400 | 0.6101 | 0.5574 | 0.6743 | 0.6049 | 0.6173 |
| AA | 0.7562 | 0.7109 | 0.6840 | 0.8045 | 0.6097 | 0.7130 |
| Katz | 0.8487 | 0.8611 | 0.7486 | 0.8896 | 0.7008 | 0.8097 |
| LP | 0.8305 | 0.8128 | 0.7115 | 0.8542 | 0.6905 | 0.7790 |
| IP | 0.8525 | 0.8548 | 0.7328 | 0.8836 | 0.7376 | 0.8112 |
| CMA-ES | 0.8462 | 0.8501 | 0.7241 | 0.8601 | 0.7243 | 0.8009 |
| MI-LP | 0.8950 | 0.8791 | 0.7510 | 0.8547 | 0.7165 | 0.8192 |
| FLP | 0.8654 | 0.8752 | 0.7456 | 0.8727 | 0.7452 | 0.8208 |
| FLP-DLA | **0.9004** | **0.8895** | **0.7560** | **0.9014** | **0.7720** | **0.8438** |

similarity metric and LP as the quasi-local similarity metric [2].

- Supervised Link Predictions: in this group of algorithms, three recent link prediction algorithms are chosen, Interaction Prediction (IP) [36], CMA-ES [37] and MI-LP [18], that try to predict future links.
- Fuzzy Link Predictions: in this group of algorithms the recent fuzzy link prediction FLP [25] is chosen for comparison with the proposed algorithm.

It should be noted that the parameters of the used algorithms are borrowed from their references. Tables 3 and 4 present the average AUC and precision scores based on 30 random runs of the FLP-DLA, respectively. In addition, in order to give a better comparison, a set of one-tailed t-tests (with $\alpha$ =0.05 and degree of freedom=98) were carried out on the results such that if the result of the t-test between the FLP-DLA and another algorithm is a negative value, then the FLP-DLA statistically outperforms that algorithm. The statistical differences between the experiments obtained with the FLP-DLA and the other methods for the test data sets are listed in Table 5. It should also be mentioned that for the following tables given in this section, the best results are highlighted.

Tables 3 and 4 demonstrate that the proposed FLP-DLA is able to achieve AUC (0.8438) and precision (0.5965) measures which are significantly better than local similarity-based algorithms: CN (AUC=0.7305, precision=0.4812), Salton (AUC=0.7103, precision=0.4603), Jaccard (AUC=0.6290, precision=0.4098), PA (AUC=0.6173, precision=0.3889), AA (AUC=0.7130, precision=0.4635). Given that the proposed fuzzy link prediction uses the global information of the social network to predict future links, it is not surprising that the FLP-DLA performs better than local similarity metrics. The proposed method is also superior to the Katz (AUC=0.8097, precision=0.5691) and LP (AUC=0.7790, precision=0.5329) link predictions. Since Katz uses the path lengths between two nodes to predict future links, this shows that the strength of the path between two nodes results in a better prediction than using

**Table 4** Precision Measures of the Proposed FLP-DLA and Other Link Prediction Methods

| Method/Dataset | Hep-th | Hep-ph | Astro-ph | Enron | EuAll | Average |
|---|---|---|---|---|---|---|
| CN | 0.5421 | 0.4532 | 0.4291 | 0.5620 | 0.4196 | 0.4812 |
| Salton | 0.5395 | 0.4260 | 0.4021 | 0.5582 | 0.3758 | 0.4603 |
| Jaccard | 0.4856 | 0.3690 | 0.3620 | 0.4529 | 0.3795 | 0.4098 |
| PA | 0.4821 | 0.3699 | 0.3052 | 0.4283 | 0.3593 | 0.3889 |
| AA | 0.5027 | 0.4549 | 0.4503 | 0.5598 | 0.3500 | 0.4635 |
| Katz | 0.6135 | 0.6298 | 0.5049 | 0.6473 | 0.4503 | 0.5691 |
| LP | 0.5703 | 0.5831 | 0.4611 | 0.6184 | 0.4319 | 0.5329 |
| IP | 0.5853 | 0.6086 | 0.5025 | 0.6204 | 0.4702 | 0.5574 |
| CMA-ES | 0.6032 | 0.6140 | 0.5000 | 0.6158 | 0.4637 | 0.5481 |
| MI-LP | 0.6273 | 0.6382 | 0.5458 | 0.5861 | 0.4578 | 0.5710 |
| FLP | 0.6154 | 0.6259 | 0.5249 | 0.6184 | 0.4899 | 0.5749 |
| FLP-DLA | **0.6452** | **0.6401** | **0.5499** | **0.6420** | **0.5057** | **0.5965** |

**Table 5** Results of Statistical T-Test between the FLP-DLA and Other Link Prediction Methods

| Method/Metric | AUC | Precision |
|---|---|---|
| CN | −36.8581 | −36.0312 |
| Salton | −34.7168 | −34.8645 |
| Jaccard | −87.2769 | −70.6142 |
| PA | −95.4301 | −65.9421 |
| AA | −38.5284 | −36.7381 |
| Katz | −7.5308 | −5.4301 |
| LP | −17.3185 | −15.5428 |
| IP | −6.5712 | −5.0010 |
| CMA-ES | −6.5782 | −5.6891 |
| MI-LP | −3.8194 | −2.8124 |

the path length between them. In comparison to the LP, since FLP-DLA is better than Katz, it is not surprising that it is also better than LP. In addition, from Tables 3 and 4 it can be seen that the FLP-DLA is able to achieve an AUC and precision which is much better than IP (AUC=0.8112, precision=0.5574). Considering that both IP and FLP-DLA use time series information, a better result can be obtained for two main reasons: 1) in contrast to IP, FLP-DLA uses a non-exact value for the date of the link occurrence and considers the number of collaborations in modelling the link strength; and 2) FLP-DLA estimates the path strength using many graph navigations, but in IP the next value of every feature in the network is estimated using a forecasting model. From the results, it can be seen that FLP-DLA is better than CMA-ES (AUC=0.8009, precision=0.5481) because FLP-DLA uses the Date information while CMA-ES does not; it only tries to combine the similarity metrics by considering a weight for each similarity metric. Thus, since FLP-DLA is very much better than similarity-based methods, it is not unexpected that FLP-DLA would be better than CMA-ES. FLP-DLA is also better than MI-LP (AUC=0.8192, precision=0.5710) since MLP only uses mutual information of the network structure while FLP-DLA uses the Date concept, and predicts links using iterative graph navigation. Finally, FLP-DLA is a little better than FLP (AUC=0.8208, precision=0.5749) because FLP does not use the Date concept and only uses fuzzy concepts in the clustering coefficient concept. FLP-DLA uses the fuzzy Date concept to include the Date of the link occurrence in the predicting task, because it is assumed here that very old or very weak links are not important for the prediction task. Table 5 also confirms the obtained result and shows that FLP-DLA outperforms several recent link prediction methods. From the results of this section, it can be concluded that the prediction is better when the event date is considered instead of the static structure, and the concept of fuzzy strength is used instead of the binary value of link existence.

### *5.2.2 Experiment 2: Comparison of the metrics of the obtained network with the original network*

This experiment compares the topological features of the predicted network which is obtained from the prediction results with the original social network. The predicted network is a network with $n$ random predicted edges, where $n$ is the number of edges in the original network. It therefore has the same number of edges as the original network, and the goal in this experiment is to compare the topological features of the original network such as the number of connected components, efficiency and clustering coefficient with the predicted network. If the values of these features in both the original and predicted network are similar, it can be concluded that the proposed algorithm works well. For each data set, we test this experiment on its largest connected component. Table 6 summarizes the topological features of the original and predicted network for the largest component of the datasets used. In Table 6, $NUM_{C_x}$ is the number of connected components in network X and the size of the largest one. For example, 1222/2 means that this network has two connected components and the largest one contains 1222 nodes. In this table, $e_x$ is the efficiency of network X, $C_x$ is the clustering coefficient and $K_x$ is the average degree of the network. From the obtained result, it can be seen that the topological features in the original network and the predicted network are approximately similar, which confirms the efficiency of the proposed algorithm. Likewise, from the results of the previous section and the reported topological features of the original network, it can be concluded that the proposed algorithm does better in networks with a higher cluster coefficient and higher average degree.

**Table 6** Results of Topological Features in the Original Network and Predicted Network

| Dataset | $NumC_{Original}$ | $NumC_{Predicted}$ | $E_{Original}$ | $E_{Predicted}$ | $C_{Original}$ | $C_{Predicted}$ | $K_{Original}$ | $K_{Predicted}$ |
|---|---|---|---|---|---|---|---|---|
| Hep-th | 4042/1 | 3465/1 | 0.185 | 0.145 | 0.764 | 0.652 | 4.247 | 3.128 |
| Hep-ph | 7031/1 | 6052/1 | 0.131 | 0.117 | 0.683 | 0.600 | 4.021 | 3.247 |
| Astro-ph | 9032/1 | 7518/1 | 0.115 | 0.094 | 0.581 | 0.504 | 2.538 | 2.065 |
| Email-Enron | 14704/1 | 10874/1 | 0.073 | 0.056 | 0.236 | 0.201 | 2.531 | 1.995 |
| Email-EuAll | 18046/1 | 11582/1 | 0.054 | 0.041 | 0.145 | 0.94 | 1.032 | 0.976 |

**Table 7** Comparison of Computation Time for the Proposed Algorithm (in Seconds)

| Method/Dataset | Hep-th | Hep-ph | Astro-ph | Enron | EuAll |
|---|---|---|---|---|---|
| CN | 115.42 | 124.58 | 250.24 | 2704.20 | 3187.98 |
| Salton | 130.38 | 132.64 | 270.42 | 2930.45 | 3260.76 |
| Jaccard | 125.34 | 128.59 | 257.52 | 2853.87 | 3305.65 |
| PA | 80.42 | 85.76 | 176.90 | 1604.05 | 1870.65 |
| AA | 120.35 | 124.74 | 234.51 | 2801.42 | 3705.32 |
| Katz | 149.87 | 157.76 | 310.83 | 4703.65 | 5204.61 |
| LP | 135.24 | 151.00 | 280.09 | 3487.61 | 4270.09 |
| FLP-DLA | 5121.21 | 10510.32 | 18451.87 | 8100.00 | 25495.10 |

#### *5.2.3 Experiment 3: computational complexity*

In this experiment, the computation time and the speed of convergence of the FLP-DLA are demonstrated for all five datasets. These algorithms are implemented in MATLAB R2009a on a PC, which has a single CPU of Intel(R) Core i7-960 3.2 GHz and 16 GB of memory. Table 7 shows the computation time for the proposed algorithm over 2000 iterations; it can be seen that the FLP-DLA requires a rational computation time based on its accuracy.

## 6 Conclusion

One of the challenges of the link prediction problem is that many of the suggested methods do not consider the date of link occurrence and use only the binary version of the social network. In this paper, a new link prediction for fuzzy social networks based on distributed learning automata (FLP-DLA) is proposed. The proposed method first models the social network as a fuzzy social network, where each link has a fuzzy strength. The fuzzy strength is defined using the date of link occurrence and the number of collaborations in the corresponding link, since it is assumed that very old links are not important in the prediction task. Then, the fuzzy links are used by distribution learning automata to find the strength of a path for any link that must be predicted. DLA tries to find the path strength using a reinforcement mechanism and graph navigation. In the proposed method, there is one LA for each node in the network, and in each iteration, the DLA tries to generate a path such that the start and end nodes of the path $(\mathbf{s_p}, \mathbf{e_p})$ belong to some test link. Then, it uses the strength of the generated path to find the strength of the link $(\mathbf{s_p}, \mathbf{e_p})$. It also uses this strength to reward or penalize the set of learning automata that exist along the path. Finally, the links with higher strength are selected as future links.

Our model is compared with some common similarity methods which show that the proposed method is completely better than static similarity-based methods because of the use of the date concept and global navigation in the graph. Our method is an iteration-based method, and thus takes a higher computational time in comparison to most similarity-based methods; however, its computational time is acceptable considering its accuracy. We also compare the proposed algorithm with several recent link prediction algorithms, and here the proposed FLP-DLA is again superior to recent methods in terms of accuracy. Finally, we compare the topological features of the obtained network with the original network. The experiment shows that the features of predicted network are similar to the original network that confirms the FLP-DLA. From the result, we can conclude that the date of the link occurrence in terms of fuzzy variables (the non-exact value) and the number of collaborations are good metrics to model the strength of links; we can also conclude that using the strength of links to estimate the strength of non-existent links is a good approach towards predicting future links.

## References

1. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. Phys A: Stat Mech Its Appl 390(6):1150–1170
2. Al Hasan M, Zaki MJ (2011) A survey of link prediction in social networks. In: Social network data analytics. Springer, pp 243–275
3. Garcia G, Dario etal. (2014) Evaluating link prediction on large graphs. In: Artificial intelligence research and development: Proceedings of the 18th international conference of the Catalan association for artificial intelligence. IOS Press, vol 277
4. Armengol E (2015) Evaluating link prediction on large graphs. In: Artificial intelligence research and development: proceedings of the 18th international conference of the Catalan association for artificial intelligence. vol 277. IOS Press
5. Huang Z, Li X, Chen H (2005) Link prediction approach to collaborative filtering. In: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries. ACM
6. Elmagarmid AK, Ipeirotis PG, Verykios VS (2007) Duplicate record detection: A survey. IEEE Trans Knowl Data Eng 19.1:1–16
7. Valerio F (2009) A graph-based semi-supervised algorithm for protein function prediction from interaction maps. In: International conference on learning and intelligent optimization. Springer Berlin Heidelberg, Heidelberg

8. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58.7:1019–1031
9. Narendra KS, Thathachar MAL (2012) Learning automata: an introduction. Courier Corporation
10. Murata T, Moriyasu S (2008) Link prediction based on structural properties of online social networks. N Gener Comput 26(3):245–257
11. Al Hasan M, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: SDM'06: Workshop on link analysis, counter-terrorism and security
12. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill, Inc., New York
13. Albert-László B, Albert R (1999) Emergence of scaling in random networks. Science 286.5439:509–512
14. Newman MEJ (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64.2:025102
15. Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Netw 25.3:211–230
16. Rossetti G, Guidotti R, Pennacchioli D, Pedreschi D, Giannotti F (2015) Interaction Prediction in dynamic networks exploiting community discovery. In: Proceedings of the 2015 IEEE/ACM international conference on advances in social networks ana. s and mining, 2015, pp 553–558 : s.n
17. Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. J Comput Sci 5(5):750–764
18. Fei T, Xia Y, Zhu B (2014) Link prediction in complex networks: a mutual information perspective. PloS One 9.9:e107056
19. Ozcan A, Oguducu SG (2015) Multivariate temporal Link Prediction in evolving social networks. In: 2015 IEEE/ACIS 14th international conference on computer and information science (ICIS), pp 185–190
20. Huang S, Tang Y, Tang F, Li J (2014) ink prediction based on time-varied weight in co-authorship network. In: Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD), pp 706–709
21. Moradabadi B, Meybodi MR (2016) Link prediction based on temporal similarity metrics using continuous action set learning automata. Phys A: Stat Mech Its Appl 460:361–373
22. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
23. Nair PS, Sarasamma ST (2007) Data mining through fuzzy social network analysis. In: Fuzzy information processing society, 2007. NAFIPS'07. Annual Meeting of the North American, pp 251–255
24. Brunelli M, Fedrizzi M (2009) A fuzzy approach to social network analysis. In: International conference on advances in social network analysis and mining 2009. ASONAM'09, pp 225–230
25. Bastani S, Jafarabad AK, Zarandi MHF (2013) Fuzzy models for link prediction in social networks. Int J Intell Syst 28(8):768–786
26. Yang L, Zhang W, Chen Y (2015) Time-series prediction based on global fuzzy measure in social networks. Frontiers, 1
27. He Y, Liu JNK, Hu Y, Wang X (2015) OWA operator based link prediction ensemble for social network. Expert Syst Appl 42(1):21–50
28. Thathachar MAL, Sastry PS (2003) Networks of learning automata: Techniques for online stochastic optimization. Springer
29. Rezvanian A, Meybodi MR (2010) An adaptive mutation operator for artificial immune network using learning automata in dynamic environments. In: 2010 2nd world congress on nature and biologically inspired computing (NaBIC), pp 479–483
30. Rezvanian A, Meybodi MR (2010) LACAIS: learning automata based cooperative artificial immune system for function optimization. In: Contemporary computing. Springer, pp 64–75
31. Rezvanian A, Rahmati M, Meybodi MR (2014) Sampling from complex networks using distributed learning automata. Phys A: Stat Mech Its Appl 396:224–234
32. Beigy H, Meybodi MR (2006) Utilizing distributed learning automata to solve stochastic shortest path problems. Int J Uncertainty Fuzziness Knowledge Based Syst 14(5):591–615
33. Soleimani-Pouri M, Rezvanian A, Meybodi MR (2012) Solving maximum clique problem in stochastic graphs using learning automata. In: 2012 4th international conference on computational aspects of social networks (CASoN), pp 115–119
34. Hu R-J, Li Q, Zhang G-Y, Ma W-C (2015) Centrality measures in directed fuzzy social networks. Fuzzy Info Eng 7(1):115–128
35. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. Phys A: Stat Mech Its Appl 390.6:1150–1170
36. Rossetti G, Guidotti R, Pennacchioli D, Pedreschi D, Giannotti F (2015) Interaction prediction in dynamic networks exploiting community discovery. In: Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining, 2015, pp 553–558 : s.n
37. Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. J Comput Sci 5(5):750–764

**Behnaz Moradabadi** received the B.S. from Tabriz University (2009), Tehran, Iran, M.S. from Sharif University (2011), Tehran, Iran, and she is now Ph.D. student in computer engineering in the Amirkabir University of Technology, Tehran, Iran (from 2013). Her current research interests include social networks, learning systems, soft computing, and information retrieval.

**Mohammad Reza Meybodi** received the B.S. and M.S. degrees in economics from Shahid Beheshti University, Tehran, Iran, in 1973 and 1977, respectively, and the M.S. and Ph.D. degrees in computer science from Oklahoma University, Norman, OK, USA, in 1980 and 1983, respectively. He is currently a Full Professor with Computer Engineering Department, Amirkabir University of Technology, Tehran. Prior to his current position, he was an Assistant Professor with Western Michigan University, Kalamazoo, MI, USA, from 1983 to 1985, and an Associate Professor with Ohio University, Athens, OH, USA, from 1985 to 1991. His current research interests include channel management in cellular networks, learning systems, parallel algorithms, soft computing, and software development.