



# Link prediction based on temporal similarity metrics using continuous action set learning automata



Behnaz Moradabadi\*, Mohammad Reza Meybodi

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

## ARTICLE INFO

### Article history:

Received 30 March 2015

Received in revised form 1 February 2016

Available online 13 April 2016

### Keywords:

Social network

Link prediction

Similarity metric

Temporal data

Learning automata

## ABSTRACT

Link prediction is a social network research area that tries to predict future links using network structure. The main approaches in this area are based on predicting future links using network structure at a specific period, without considering the links behavior through different periods. For example, a common traditional approach in link prediction calculates a chosen similarity metric for each non-connected link and outputs the links with higher similarity scores as the prediction result. In this paper, we propose a new link prediction method based on temporal similarity metrics and Continuous Action set Learning Automata (CALA). The proposed method takes advantage of using different similarity metrics as well as different time periods. In the proposed algorithm, we try to model the link prediction problem as a noisy optimization problem and use a team of CALAs to solve the noisy optimization problem. CALA is a reinforcement based optimization tool which tries to learn the optimal behavior from the environment feedbacks. To determine the importance of different periods and similarity metrics on the prediction result, we define a coefficient for each of different periods and similarity metrics and use a CALA for each coefficient. Each CALA tries to learn the true value of the corresponding coefficient. Final link prediction is obtained from a combination of different similarity metrics in different times based on the obtained coefficients. The link prediction results reported here show satisfactory of the proposed method for some social network data sets.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Nowadays people and organizations could communicate more efficiently using social networks. Social network can be visualized as graphs, where the nodes correspond to persons and the edges represent some form of communications between the corresponding persons [1,2]. Since this kind of network is generally complex and highly dynamic, it is really important to understand its behavior along time [1,3]. Social Network Analysis (SNA) is a research area that tries to study the dynamics of the network structure [3]. In this paper, our goal is to use dynamics of the network to predict what communications are most likely to be formed in the future. This problem is a well-known problem in SNA area that is called link prediction problem [1].

To deal with link prediction problem, there are several approaches [4]: the most common approach is based on applying topological similarity metrics to non existence links at time  $t$  to determine if a link will appear at a time  $t'$  ( $t' > t$ ). Such methods generate scores for each link and use the scores to perform prediction task either by an unsupervised or a supervised technique. There are many similarity metrics including local similarity metrics: Common Neighbors [5], Salton

\* Corresponding author.

E-mail addresses: [moradabadi@aut.edu](mailto:moradabadi@aut.edu) (B. Moradabadi), [mmeybodi@aut.edu](mailto:mmeybodi@aut.edu) (M.R. Meybodi).

Index [6], Jaccard Index [7], Hub Depressed Index [8], Hub Promoted Index [9], Leicht–Holme–Newman Index (LHN1) [9], Preferential Attachment Index [10], Adamic–Adar Index [11] and Resource Allocation Index [12], global similarity metrics: Katz Index [13], Leicht–Holme–Newman Index (LHN2) [9], Matrix Forest Index (MFI) [14] and Quasi local metrics that do not require global topological information but use more information than local indices: Local Path Index [15], Local Random Walk [16], Superposed Random Walk [16], Average Commute Time [17], Cos+ [18], random walk with restart [19], SimRank [20], Resource Allocation index and Local Path index [15].

The main problem of traditional methods for link prediction is that they cannot explore the network evolution, because they only consider the current network structure without considering the links occurrence time. On the other hand the different similarity metrics have shown different performance on different social networks. Our work tries to overcome these limitations by using CALA to determine the coefficients of different times and similarity metrics. The intuition of the proposed approach is that the final prediction depends on the past similarity scores but this dependency is not uniform between different times and different similarity metrics and it can be differently influenced by difference coefficients. In other words we do link prediction by exploring how different topological metrics in different periods influence the final similarity score.

Continuous Action set Learning Automata (CALA) is an adaptive decision making unit with real-valued actions that tries to learn the optimal action from a set of allowable actions by interacting with a random environment [21]. In each step, it selects an action from its action-set. The action selection in CALA is based on a probability distribution over the action set. The selected action is applied to the environment and then a reinforcement signal is produced by the environment. CALA updates the probability distribution of its actions according to both reinforcement signal and a learning algorithm and again chooses an action. These steps are repeated until CALA converges to some action.

In the proposed method, in order to accomplish link prediction in time  $t + 1$ , the link prediction problem is modeled as a noisy optimization problem by considering one coefficient for each of similarity metrics and time periods. The goal of optimization problem is to determine the coefficients of the different similarity metrics as well as the coefficients of different past times such that the overall error of the prediction task is minimized. This noisy problem is solved by a team of CALA: one CALA for each coefficient. Each CALA tries to learn the true value of the corresponding coefficient. Final link prediction is done using the scores that is calculated from the past similarity scores with considering times and similarity metrics coefficients. In order to evaluate the result of the proposed approach, we conducted some experiments and compared our results with the ones achieved by other link prediction approaches. In general, the experiments showed that our approach performs better than other strategies.

The rest of the paper is organized as follows. Section 2 introduces the time series link prediction problem and its literature review. The learning automata and continuous action set learning automata are described in Section 3. Section 4 introduces the proposed link prediction method based on CALA. Section 5 presents the experimental study for some social network data sets. Section 6 summarizes the main conclusion and discusses the future directions of our research.

## 2. Time series link prediction methods

In this section we first introduce the time series link prediction problem and then review the related works in this area.

### 2.1. Problem formulation

As it is defined in Ref. [22], the time-series link prediction problem is formally introduced as follows: Let  $V$  be the list of nodes,  $V = \{1, 2, \dots, N\}$ . A graph series is a list of graphs  $\{G_1, G_2, \dots, G_T\}$  corresponding to a list of adjacency matrices  $(M_1, M_2, \dots, M_T)$ . Each  $M_t$  is an  $N \times N$  matrix with each element  $M_{t(i,j)}$  corresponds to edge  $E_{t(i,j)}$  in  $G_t$ . The value of  $M_{t(i,j)}$  is from the set  $\{0, 1\}$  and it is the indicator of existence or not existence of the edge  $(i, j)$  during the period  $t$ . Then in the time series link prediction, we try to predict the occurrence or not occurrence of the links in time  $T + 1$  using previous times  $M_1, M_2, \dots, M_T$ .

### 2.2. Related works

In the following subsection we review the recent time series link prediction methods:

Ref. [23] has proposed a method for time-aware link prediction. They proposed an extension of the local probabilistic model that is introduced in Ref. [24] by using temporal information. Their proposed method used the link existence and the time of existence to generate a probabilistic model for link prediction. An empirical evaluation of technique was performed over two collaboration networks and they showed that link time occurrence can be considered as a main feature in the prediction result.

The authors of Ref. [25] have formulated the link prediction problem as a periodic temporal link prediction and studied that if the data have underlying periodic. In the proposed method they have introduced two matrix and tensor-based methods for predicting future links and summarized the data of multi time periods into a single matrix using a weight-based method. Then they have used a CANDECOMP/PARAFAC tensor decomposition to illustrate the usefulness of using natural three-dimensional structure of temporal link data and showed the superiority of their method on some bibliometric social networks.

Authors [26] have proposed a method called cross temporal link prediction which tries to predict the links in different time frames. Their method is an extension of the dimension reduction method that is proposed in Ref. [27]. It uses node attributes to predict the like occurrence of two nodes by including time stamp in prediction task. They first try to study the repeated link prediction problem. Second to predict unobserved links, the authors have proposed a cross-temporal locality preserving projection (CT-LPP) method in which data in different time frames is modeled by using low-dimensional latent feature space. Finally they have evaluated the cross temporal link prediction to show the accuracy improvement of the proposed method.

In Ref. [28] to propose a new time series link prediction, the topological matrices in the different times are modeled as a time series data and used by a forecasting model to predict the future links. So, for each link that must be predicted, the past similarity metrics is calculated as a time series data and the next value of the similarity metric is predicted using some forecasting models. In this method only one similarity metric is modeled by a prediction model.

In Ref. [22], a new link prediction approach using both the time-series patterns and similarity methods has been proposed. They build a time series data from the link occurrences in different periods and use autoregressive integrated moving average (ARIMA) model to predict the next value of time series data. The final prediction is calculated based on a combination of forecasting result in addition to a chosen similarity method. Their method only uses the link occurrence of different time to make a prediction. Their result showed that the proposed methods achieved a good performance compared to methods that use time-series models or similarity methods alone.

The authors of Ref. [29] have proposed a data mining process which addresses a particular formulation of the link prediction problem for dynamic networks, called Interaction Prediction. Their approach predicts future interactions by combining dynamic social networks analysis, time series forecast, feature selection such as similarity metrics and network community structure. The proposed method focuses on the links within the communities. Their experiments on real world interaction networks show that the proposed approach achieves encouraging results both in case of balanced and unbalanced class distributions. The authors of Ref. [30] have provided an approach for predicting future links by applying the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to optimize weights which are used in a linear combination of sixteen neighborhood and node similarity indices. They have examined 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.

In Ref. [31], a novel method called Multivariate Time Series Link Prediction, for link prediction in evolving networks that integrates (1) temporal evolution of the network; (2) node similarities, and (3) node connectivity information is proposed. This method uses existing connections and a calculated similarity metric for each time. Finally they compare different similarity metrics in their experiments.

In Ref. [32], a weighted approach for modeling the occurrence of time is used to generate a different similarity metric. Finally, in Ref. [33] a fuzzy concept is introduced to model the uncertainty, and a similarity metric is used to build a fuzzy relation model among individuals in the social network.

Based on our research in the time series link prediction methods, the link prediction approaches that consider the time of link occurrence are classified in two groups:

1. Using link occurrence of different times to model the link prediction problem.
2. Using similarity metrics in different times to model the link prediction problem.

Also, based on used methodology to forecast future links, the time-based link prediction algorithms can be categorized in three categories:

1. Using mathematical prediction models such as ARIM, VAR, . . . to model the link prediction problem.
2. Using graph summarization methods such as using decay factor.
3. Using Probabilistic models to generate a time-based probabilistic model such as Dynamic Bayesian Network, Maximum Entropy, . . . .

In most of mentioned methods, only one similarity metric or only the links occurrence is used to generate prediction model over time. In Refs. [30,34] there are two approaches that consider different similarity metrics: in Ref. [30] for each different similarity metric a weight is calculated by CMA-ES algorithms without considering different times; also in Ref. [34] a dynamic Bayesian network is used to model the dependence tree between different similarity metrics over time. Based on our research there is no recent link prediction algorithm that considers different similarity metrics in addition to links occurrence over time concurrently. In this paper we try to model the time series link prediction problem as a stochastic problem by considering different similarity metrics and link occurrence over time, concurrently.

### 3. Learning automata

#### 3.1. Continuous action set learning automata

Learning automata (LA) are a type of decision making units that try to learn the optimal action from the set of possible actions by interaction with an unknown stochastic environment [35,36]. In each iteration, the LA selects an action from its action probability distribution and sends it to the random environment [37–39]. The random environment evaluates the selected action and generates a stochastic response to the LA. This stochastic response is called reinforcement signal.

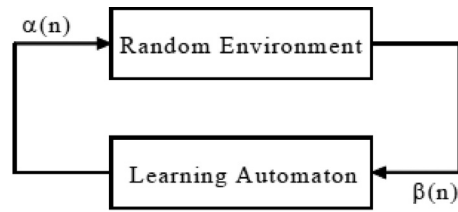


Fig. 1. The relationship between a learning automaton and its random environment.

Then the LA updates its action probability distribution using the reinforcement signal and a learning algorithm. This tool has many applications in different areas such as optimization tasks [21,40], capacity assignment problems [41–43], graph problems [44–46], artificial intelligence [47–49], social area [50–52] and other applications [53–56].

In learning automata the environment can be described by a triple  $E = \{\alpha, \beta, c\}$  where  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  denotes the finite set of possible actions for each learning automata,  $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$  represents the set of values that can be taken by the reinforcement signal, and  $c = \{c_1, c_2, \dots, c_r\}$  represents the set of penalty probabilities, where  $c_i$  is associated with the given action  $\alpha_i$ . Based on the penalty probabilities of the environment are constant or varied with time, the random environment is called stationary environment and non-stationary environment, respectively.

Also the random environment based on its possible values could be classified into three classes: P-, Q-, and S-model environments. The reinforcement signal in P-model environments has only two values  $\{0, 1\}$  while in Q-model is bounded in the interval  $[0, 1]$  and in S-model environments it is generated from a continuous random variable.

The LA based on its possible actions can be partitioned into two main classes: FALA and CALA [35]. In FALA, the action set is finite and the action probability distribution of a FALA with  $r$  actions is defined by an  $r$ -dimensional probability distribution. In contrast, in CALA the possible actions are a set of real values and CALA uses a probability distribution function to display its actions probability. The relationship between the learning automaton and its random environment has been shown in Fig. 1. In the following part of this section, we introduce the CALA that is used in this paper:

In Ref. [21], a CALA is given, in which the action probability distribution at instant  $n$  is a normal distribution with mean  $\mu_n$  and standard deviation  $\sigma_n$ . At each instant, the CALA chooses an action according to its probability distribution function and updates its action probability distribution using the reinforcement signal that is generated from the environment by updating  $\mu_n$  and  $\sigma_n$ . Since CALA has no knowledge of the reinforcement signal  $\beta(\cdot)$ , the objective of automaton is to identify the optimal action, which results in the minimum value of  $\beta(\cdot)$ . This is achieved by the learning algorithm that updates the action probability distribution using the reinforcement signal that is sent from the random environment. The general learning algorithm of CALA is described as follows:

Consider a function  $f : \mathfrak{X} \rightarrow \mathfrak{X}$ , such that only noisy values of  $f(\alpha)$  are available for measurement of any  $\alpha$ . Noisy values also correspond to  $\beta(\alpha)$  and we have  $f(\alpha) = [\beta(\alpha) | \alpha]$ . Hence the optimal action of the CALA is an  $\alpha$  that minimizes  $f(\alpha)$  and  $[\beta(\alpha) | \alpha]$ . To do this at instance  $k$ , the CALA chooses action  $\alpha_k$  according to the normal distribution  $N(\mu_k, \sigma_k)$ . Then the CALA obtains two responses from the environment for two actions  $\mu_k$  and  $\alpha_k$ . Let these reinforcement signals be  $\beta_{\mu_k}, \beta_{\alpha_k}$ , respectively. Then  $\mu_k$  and  $\sigma_k$  are updated according to the following equations:

$$\mu_{k+1} = \mu_k + \lambda \frac{\beta_{\alpha_k} - \beta_{\mu_k}}{\phi(\sigma_k)} \frac{\alpha_k - \mu_k}{\phi(\sigma_k)} \tag{1}$$

$$\sigma_{k+1} = \sigma_k + \lambda \frac{\beta_{\alpha_k} - \beta_{\mu_k}}{\phi(\sigma_k)} \left[ \left( \frac{\alpha_k - \mu_k}{\phi(\sigma_k)} \right)^2 - 1 \right] - \lambda K(\sigma_k - \sigma_L) \tag{2}$$

where

$$\phi(\sigma) = \begin{cases} \sigma_L & \text{for } \sigma \leq \sigma_L \\ \sigma & \text{for } \sigma > \sigma_L > 0 \end{cases} \tag{3}$$

and  $\lambda$  is the learning parameter for controlling step size  $0 < \lambda < 1$ ,  $K$  is a large positive constant and  $\sigma_L$  is the lower bound on  $\sigma$ . The iteration continues until  $\mu_k$  does not change appreciably and  $\sigma_k$  is close to  $\sigma_L$ .

The idea behind the updating rule given by (1) and (2) is as follows: If  $\sigma_k$  get ‘better’ response from the environment, then  $\mu_k$  move towards  $\alpha_k$ . Otherwise it is moved away from  $\alpha_k$ . For updating  $\sigma_k$ , whenever an action  $\alpha_k$  is away from  $\mu_k$  by more than one standard deviation and it improves the reinforcement signal or when the action selection within one standard deviation from the current mean is worse, we increase the variance; otherwise we decrease it.

#### 4. The proposed time series link prediction approach

This section proposes a new time series link prediction based on CALA (CALA-LP) which uses temporal data of similarity metrics for prediction task. The proposed algorithm formulates the link prediction problem as a noisy optimization problem. The goal of optimization problem is to define a new score that is a combination of different similarity metrics in different

past times such that the overall error of prediction task is minimized. To solve the noisy optimization problem we use the optimization tool CALA. In the following sections we first formulate our problem as a noisy optimization problem. Second we introduce the proposed CALA to solve the optimization problem in detail and study its convergence. Third we review the similarity metrics that we use in the proposed method. Finally we describe the procedure to predict links in time  $T + 1$ .

#### 4.1. The problem formulation

In the proposed algorithm, matrix  $X_t (n_t \times n_s)$  is the similarity matrix of time  $t$ , where  $n_t$  and  $n_s$  are the number of links and the number of used similarity metrics, respectively. Each row of  $X_t, x_t(i) = \{x_{t1}, x_{t2}, \dots, x_{tn_s}\}$ , displays the feature vector of link  $i$ , where each feature corresponds to the score of some similarity metric. Now, we formulate the link prediction problem as the following noisy optimization problem:

$$\text{maximize} \left( \sum_{t=1}^T \gamma^{T-t+1} |A_t(X_t W - \theta_t)|_2 \right) + \delta |A_{T+1}(X_{T+1} W - \theta_{T+1})|_2 \quad 0 < \gamma \leq 1. \tag{4}$$

Or equivalently

$$\text{minimize} - \left( \left( \sum_{t=1}^T \gamma^{T-t+1} \|A_t(X_t W - \theta_t)\|_2 \right) + \lambda \|A_{T+1}(X_{T+1} W - \theta_{T+1})\|_2 \right) \quad 0 < \gamma \leq 1 \tag{5}$$

where  $A_t$  is an indicator matrix of size  $(n_t \times 1)$  that is calculated based on the following equation:

$$A_t(i) = \begin{cases} 1 & \text{The link } i \text{ exists in time } t \\ -1 & \text{otherwise} \end{cases} \tag{6}$$

and  $A_{T+1}(n_t \times 1)$  is the indicator matrix of time  $T + 1$ . Because we do not know the true real value of matrix  $A_{T+1}$ , this matrix is generated as a random matrix consisting of  $\{-1, 1\}$  and this matrix can be viewed as the noise parameter in Eqs. (4) and (5).

The goal of the proposed method is to find the following parameters:

- (1) A weight matrix  $W$  in size  $(1 \times n_s)$  with one weight for each similarity metric to determine the importance of the corresponding similarity metric.
- (2) A time decay factor  $\gamma$  such that maximizing the equation  $\|A_t(X_t W - \theta_t)\|_2$  in the times that are near to  $T + 1$  can be more important than to the times that are far from  $T + 1$ .
- (3) A threshold  $\theta_t$  for each time  $t$  such that the higher values than  $\theta_t$  are considered as connected pairs and the values lower than  $\theta_t$  are considered as non-connected pairs.
- (4) A regularization term  $\delta$  to balance the two main parts of the problem.

In other words, the proposed method tries to find weight  $W$ , a decay factor  $\gamma$ , and threshold  $\theta_t$  such that  $\gamma^{T+1-t}(\theta_t - x_t(i)W)$  be maximized for each non-connected pair and  $\gamma^{T+1-t}(x_t(i)W - \theta_t)$  be maximized for each connected pair.

Thus we modeled the link prediction problem as a noisy optimization problem and this optimization problem can be solved using CALA optimization tool that is described in the next subsection.

#### 4.2. The proposed continuous action set learning automata

This section first describes the proposed learning automata and then studies its convergence:

##### 4.2.1. Proposed learning automata

This subsection presents the proposed learning automata to optimize the link prediction problem that is modeled in the previous phase. In the proposed method there is a CALA for each of variables that must be determined. So, we have a set of  $n_{LA} = (n_\theta + n_w + 1)$  CALA that is defined as a set  $\langle LA_\theta, LA_W, LA_\gamma, LA_\delta \rangle$  where  $LA_\theta$  is the set of  $n_\theta$  CALAs, one CALA for each threshold  $\theta_t$ ,  $LA_W$  is the set of  $n_w$  CALAs, one CALA for determining the weight of each similarity metric, and  $LA_\gamma$  and  $LA_\delta$  are two CALAs to determine the value of parameters  $\gamma$  and  $\delta$ , respectively. Also to generate a reinforcement signal based on the CALAs' chosen actions, we define a common reinforcement function  $\beta(\theta, W, \gamma, \delta)$  as the following equation:

$$\beta(\theta, W, \gamma, \delta) = - \left( \left( \sum_{t=1}^T \gamma^{T-t+1} \|A_t(X_t W - \theta_t)\|_2 \right) + \delta \|A_{T+1}(X_{T+1} W - \theta_{T+1})\|_2 \right). \tag{7}$$

As we saw in the above equation, the reinforcement signal  $\beta(\theta, W, \gamma, \delta)$  is the same optimization problem in Eq. (4) for parameters  $(\theta, W, \gamma, \delta)$  such that the lower values for  $\beta(\theta, W, \gamma, \delta)$  are more valuable than the higher values of  $\beta(\theta, W, \gamma, \delta)$ . Now, to solve the mentioned optimization problem, we follow the following procedure:

At each instance  $k$  in the proposed method, each  $LA_j$  ( $1 \leq j \leq n_{LA}$ ) chooses an action  $\alpha_k(j)$  according to its normal distribution,  $N(\mu_k(j), \sigma_k(j))$ , and we generate two common reinforcement signals  $\beta_{\mu_k}, \beta_{\alpha_k}$  as the following equations:

$$\beta_{\mu_k} = \beta(\mu_{\theta_k}, \mu_{W_k}, \mu_{\gamma_k}, \mu_{\delta_k}), \quad \beta_{\alpha_k} = \beta(\alpha_{\theta_k}, \alpha_{W_k}, \alpha_{\gamma_k}, \alpha_{\delta_k}). \quad (8)$$

Then each  $LA_j$  updates its  $\mu_k(j)$  and  $\sigma_k(j)$  based on the  $\beta_{\mu_k}$  and  $\beta_{\alpha_k}$  according to Eqs. (1) and (2). In other words, in each iteration, all CALAs choose their actions; these actions are evaluated to generate reinforcement signals; then the normal distribution parameters of each CALA are updated based on the generated reinforcement signals. This procedure repeats until the value of each CALA converges to some value.

#### 4.2.2. Analysis

This subsection analyzes the proposed CALA using some game theory concepts. In game theory a game is defined by a triple set  $\langle I, A, F \rangle$  where  $I$  is the number of players, and  $A = \{A_1, A_2, \dots, A_I\}$  is the information and actions available to each player at each decision point, and the function  $F = \{F^1(\alpha), F^2(\alpha), \dots, F^I(\alpha)\}$  is the set of payoff functions for each outcome  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_I\}$  where  $\alpha_i$  is the selected action of player  $i$ . The players only receive the payoff signals and they have no knowledge about the payoff functions.

**Definition 1.** We say that profile  $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$  is an optimal point of the game if for each  $i$ ,  $1 \leq i \leq N$  and each profile  $\alpha'$  we have

$$F^i(\alpha^*) \geq F^i(\alpha'). \quad (9)$$

In other words  $\alpha^*$  is a local maximum of  $F^i$  and also is the Nash equilibrium of the game [57].

**Definition 2.** The common payoff game is a special case of the game such that  $F^i(\cdot) = F(\cdot)$  for all  $i$ . In other words in common payoff games all players get the same payoff. Now  $\alpha^*$  will be an optimal point if

$$F(\alpha^*) \geq F(\alpha') \quad \forall (\alpha') \in N(\alpha^*), \quad (10)$$

where

$$N(\alpha^*) = \{(\alpha) \in D : \alpha, \alpha^* \text{ differs in only one component}\}, \quad (11)$$

and  $D \cong \prod_{i=1}^I A_i$  such that each element of  $D$  is a tuple of action choices by all  $I$  players [57].

Now we use these definitions to analyze the proposed CALAs:

**Lemma 1.** *The proposed team of CALAs in this paper can be considered as a common payoff game.*

**Proof.** It can be seen easily that the proposed team of CALAs is a common payoff game where each CALA is a player, and the action space of each player is the set of possible actions for the corresponding CALA. Also, like in common payoff games after choosing actions by all CALAs, all of them get a common payoff and so a common reinforcement signal.

**Theorem 1.** *Consider the common payoff game of CALA where all the automata use the updating rule given by Eqs. (1) and (2). Based on [57], if the learning step-size ( $\lambda$ ) is sufficiently small then the CALA team converges to a local maximum point and hence the Nash equilibrium of the corresponding game.*

#### 4.2.3. Similarity metrics

This subsection briefly presents the seven commonly similarity measures that we use as the feature values in proposed method:

(1) Common Neighborhood [58]: 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

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \quad (12)$$

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

(2) Salton: This score is defined as

$$Salton(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \times |\Gamma(y)|}}. \quad (13)$$

(3) Jaccard Index [6]: This index was proposed by Jaccard and it is defined as:

$$Jaccard(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \quad (14)$$

- (4) Preferential Attachment (PA) [10,59]: The preferential attachment (PA) algorithm is based on the preferential attachment phenomena rule [5] 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:

$$PA(x, y) = |\Gamma(x)| \times |\Gamma(y)|. \tag{15}$$

- (5) Adamic-Adar Index (AA) [11]: This index is an extension of common neighborhood method such that the less-connected neighbors have more weight and it is defined as:

$$AA(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\Gamma(z))}. \tag{16}$$

- (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 as the following equation:

$$Katz(x, y) = \sum_{l=1}^{\infty} \beta^l \cdot |Path(x, y)^{(l)}| \tag{17}$$

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:

$$Katz = (I - \beta A)^{-1} - I. \tag{18}$$

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

$$LP\ Index(x, y) = A^2 + \epsilon A^3. \tag{19}$$

For more information about other similarity metrics please refer to Liben-Nowell and Kleinberg [4].

#### 4.2.4. Prediction phase

After all CALAs converge to some value, to generate the final prediction of the proposed algorithm we calculate the score of each test link  $i$  based on the following equation:

$$Score(i) = \sum_{t=1}^{T+1} \gamma^{T-t+1} (x_t(i)W_t(i) - \theta_t), \tag{20}$$

and use this score to predict links in time  $T + 1$ .

### 5. Experimental results

In order to evaluate the performance of the proposed algorithm, some computer experiments have been conducted and the performance of the proposed algorithm has been compared in terms of performance and accuracy. In these experiments, we use the quality of solutions, computation time, and convergence rate of the proposed algorithm as the performance criteria. In the rest of this section, we will first give the data set and evaluation metrics we used in our experiment and then give a set of two experiments. In the first experiment, the performance of the proposed algorithm is compared with the performance of some link prediction algorithms and in the second experiment, the computation time and convergence rate of the CALA-LP are computed and reported respectively.

#### 5.1. Data set

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

- (1) Co-authorship Networks: A type of social network where the nodes represent the authors and two authors are connected if they have collaborated in a paper. Collaboration network is widely used to understand the topology and dynamics of complex networks. In this paper we have adopted three co-authorship networks from three sections of Arxiv<sup>1</sup> and extracted data from the years 1993 to 2003 for all these data sets. The first network is composed by authors that collaborated in theoretical high energy physics<sup>2</sup> (hep-th). The second one is formed by authors who published papers

<sup>1</sup> <http://www.arxiv.org>.

<sup>2</sup> <http://arxiv.org/archive/hep-th>.

**Table 1**  
Network size in terms of nodes and edges.

Data set	Nodes	Edges	Description
Hep-th	9877	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

in the high energy physics<sup>3</sup> (hep-ph) and the third one is sampled from collaboration in the Astro Physics<sup>4</sup> (Astro-ph). In these data sets if an 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 by  $k$  authors this generates a completely connected (sub) graph on  $k$  nodes.

- (2) Email Communication Networks: A type of social network where the nodes of the network are email addresses and if an address  $i$  sent at least one email to address  $j$ , the graph contains an undirected edge from  $i$  to  $j$ . In our experiment we use two email communication data sets: Enron email communication network<sup>5</sup> and Eu-All email communication network<sup>6</sup>. The Enron email communication network includes a data set around half million emails that is public by the Federal Energy Regulatory Commission and we extract data from May 1999 through May 2002 (36 months). Also the Eu-All email communication network was extracted using email data from a large European research institution and we extracted data from October 2003 to May 2005 (18 months). The network specification of each data set is presented in Table 1. Since these networks are highly sparse, 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.

## 5.2. Evaluation metrics

This section presents the two common evaluation metrics that we use in our experiments:

- (1) AUC Metric [1]: If we rank all of 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 time we usually pick a missing link and a nonexistent link randomly and compare their scores. If among  $n$  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:

$$\text{AUC} = \frac{n' + 0.5n''}{n}. \quad (21)$$

If the AUC value has a value more than 0.5 it is better than the random link prediction algorithm and the more far from 0.5 means the more accurate algorithm.

- (2) Precision [1]: If we predict  $L$  links to be connected and  $Lr$  links from  $L$  links are right, the Precision is defined as:

$$\text{Precision} = \frac{Lr}{L}. \quad (22)$$

Clearly, higher precision means higher prediction accuracy.

## 5.3. Link prediction comparison

This section evaluates the proposed CALA-LP accuracy. To do this, for collaboration networks (Hep-th, Hep-ph and Astro-ph), we consider the data from 1993 to 2002 as the training data (each year as a time period) and year 2003 as test data. Also for email networks (Enron and EuAll), we consider the first 70% available months as the training data (each month as a time period) and the 30% remaining months as the test data. For all conducted experiments, the initial parameters of each CALA-LP  $j$ ,  $(\mu_0(j), \sigma_0(j))$  are chosen randomly, the parameter  $\lambda$  is set to 0.005 and  $\sigma_L$  is set to  $10^{-3}$ . In order to improve the comparison of the proposed algorithm, we choose a set of algorithms in three categories:

- Similarity based algorithms: this group of algorithms contains a set of common similarity metrics such as CN, Salton, Jaccard, PA, AA, Katz and LP [1].
- Supervised algorithms: in this group of algorithms we choose Interaction Prediction (IP) [29] and CMA-ES [30] algorithms that try to predict future links using a linear combination of similarity metrics.
- Time-aware algorithms: in this group of algorithms two recent time aware algorithms LP-ARIMA [22] and TW-LP [32] are chosen for comparison with the proposed algorithm.

<sup>3</sup> <http://arxiv.org/archive/hep-ph>.

<sup>4</sup> <http://arxiv.org/archive/Astro-ph>.

<sup>5</sup> <http://www.cs.cmu.edu/~enron/>.

<sup>6</sup> <http://snap.stanford.edu/data/email-EuAll.html>.



**Table 2**  
AUC measures of proposed CALA-LP and other link prediction methods.

Method	Data set					Average
	Hep-th	Hep-ph	Astro-ph	Enron	EuAll	
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	0.779
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
LP-ARIMA	0.9132	0.8802	0.7737	0.8932	0.7621	0.8444
TW-LP	0.9043	0.8699	0.7512	0.8901	0.7526	0.8336
CALA-LP	<b>0.9321</b>	<b>0.8823</b>	<b>0.7999</b>	<b>0.9059</b>	<b>0.7732</b>	<b>0.8586</b>

**Table 3**  
Precision measures of proposed CALA-LP and other link prediction methods.

Method	Data set					Average
	Hep-th	Hep-ph	Astro-ph	Enron	EuAll	
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	0.6360	0.4821	0.5663
LP-ARIMA	0.6462	0.6383	0.5399	0.6400	0.5001	0.5851
TW-LP	0.6235	0.6354	0.5332	0.6239	0.4932	0.5818
CALA-LP	<b>0.6525</b>	<b>0.6425</b>	<b>0.5621</b>	<b>0.6689</b>	<b>0.5134</b>	<b>0.6078</b>

The parameters of the used algorithms are borrowed from their references. Tables 2 and 3 present the average AUC and precision scores based on the 10-fold cross validation method, respectively. Also, in order to have a better comparison, a set of one-tailed  $t$ -tests were taken on the 10-fold cross validation results. In this experiment, we consider the result of CALA-LP as algorithm A2. Hence, if the result of  $t$ -test between the CALA-LP and another algorithm is a negative value then the CALA-LP statistically outperforms another algorithm. The statistical differences between the experiments obtained with the CALA-LP, and other algorithm for the test data sets are listed in Table 4. Also, it should be mentioned that for the following tables given in this section, the best results are highlighted.

The results reported here (Tables 2 through 4) demonstrate that the proposed CALA-LP is able to achieve an AUC and Precision measures that is significantly better than algorithms of Group 1 (Similarity based algorithms): CN, Salton, Jaccard, PA, AA, Katz and LP. So, we can conclude that the proposed algorithm is better than all considered static methods. Also in comparison with the IP and CMA-ES we see that the proposed algorithm achieves a better result in AUC and precision measures; so we also conclude that the CALA-LP is superior to recent link prediction algorithms that consider different similarity metrics but without considering the time of link occurrence. Finally, from the result that is reported here we can see that the proposed algorithm is a little better in AUC and precision measures in comparison to the LP-ARIMA and TW-LP that use the time information of the social graph. So, the results of the CALA-LP suggest that the prediction is better with considering different similarity metrics and different times with different coefficients. Also, to show the importance of each similarity metric, we report the obtained weights from the proposed algorithm for each data set in Table 5. From the result reported here we can conclude that the importance pattern of local similarity metrics in co-authorship networks is similar to each other and the LP and Katz have more weights in comparison to the local similarity metrics. So in these networks the local information has little influence in link prediction in comparison to the global or quasi local similarity metrics. In other hand from Table 5 we can see that in Email networks the global similarity metrics and local similarity metrics have equivalent influence on the prediction result. So in this network the global information does not give an extra better result in comparison to the local similarity metrics by considering link occurrence time.

#### 5.4. Performance of the CALA-LP

In this experiment the computation time and the speed of convergence of the CALA-LP are demonstrated for all five data sets. These algorithms are implemented in MATLAB R2009a on a PC, which has a single CPU of Intel(R) Core(TM)2 Duo 3.33 GHz and 8 GB of memory. Table 6 shows the computation time for proposed algorithm over 2000 iterations. It

**Table 4**

The result of statistical *t*-test between the CALA-LP and other link prediction methods.

Method	Metric	
	AUC	Precision
CN	−38.2432	−38.9437
Salton	−36.4000	−35.928
Jaccard	−89.4058	−71.2311
PA	−97.0617	−68.535
AA	−40.6023	−38.283
Katz	−9.8070	−6.9514
LP	−19.9132	−17.3435
IP	−8.525	−7.4271
CMA-ES	−8.1021	−7.0101
LP-ARIMA	−5.2421	−3.2626
TW-LP	−6.6268	−3.5481

**Table 5**

Weight of each similarity metric in CALA-LP for used data sets.

Method	Data set				
	Hep-th	Hep-ph	Astro-ph	Enron	EuAll
CN	0.402	0.3742	0.4372	0.3537	0.3840
Salton	0.1530	0.1793	0.1800	0.1294	0.1765
Jaccard	0.0735	0.0601	0.0582	0.2042	0.2940
PA	0.0901	0.1294	0.1040	0.3025	0.3420
AA	0.2410	0.1374	0.2703	0.4184	0.4529
Katz	0.5294	0.4193	0.3602	0.3692	0.3602
LP	0.6526	0.5050	0.6104	0.4021	0.4520

**Table 6**

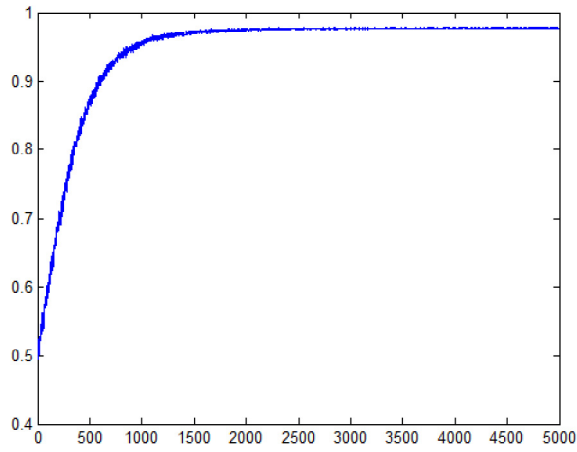
Computation time comparison for the proposed algorithm (in seconds).

Method	Data set				
	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
CALA-LP	1150.53	1522.43	1439.42	5190.43	6210.45

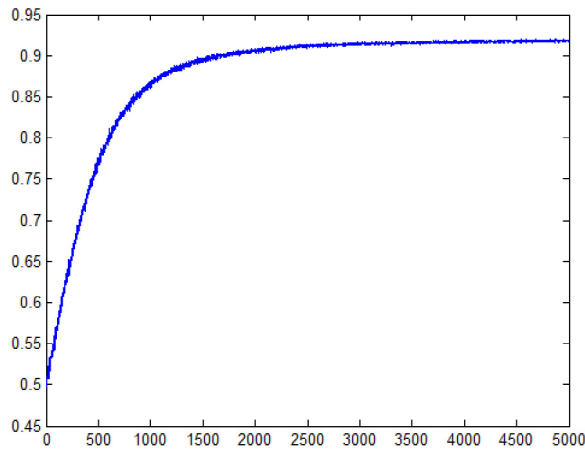
can be seen that the CALA-LP requires a rational computation time in order to consider different similarity metrics and different times. Because Hep-th has lower size than Hep-ph and Hep-ph has lower size than Astro-ph, Table 6 shows that the proposed algorithm has lower dependency to the size of the test network in terms of computation time. It must be noted that the computation time reported here is only the running time of the proposed algorithm, not the running time of computing the considered similarity features. In the next experiment of this section, to show the convergence rate of the proposed algorithm, the convergence rate based on true prediction for data sets Hep-th and Hep-ph, Astro-ph are presented in Figs. 2 through 4, respectively. Based on represented figures, we can see that the stochastic process of choosing action for each CALA in addition to applying the reinforcement algorithm leads to the improvement of prediction in early iterations and continue to make the prediction more precise in the further iterations.

## 6. Conclusion

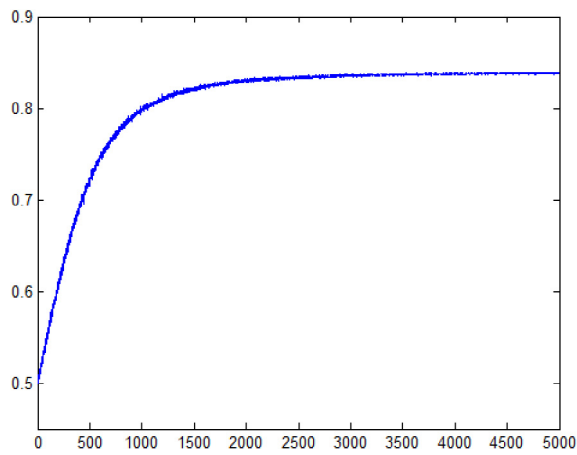
This paper presents a new time series link prediction method which uses Continuous Action set Learning Automata to predict the occurrence or not occurrence of each link in time  $T + 1$ . In the proposed method, the link prediction problem is formulated as a noisy optimization problem by considering coefficients for each similarity metric and each time period. The intuition of the proposed approach is that the final prediction depends on the past score metrics but this dependency is not uniform between different similarity metrics and different times; although it can be differently influenced by different periods and different similarity metrics. Then we use a team of CALA, one CALA for each parameter that must be determined in the optimization problem. Each CALA tries to learn the true value of the corresponding parameter. Final link occurrence is predicted using similarity scores with considering time and similarity metrics coefficients. The experimental results reported here show that the proposed algorithm is superior to other link prediction algorithms. The better result can be due to the



**Fig. 2.** The convergence rate of the CALA-LP for Hep-th data set.



**Fig. 3.** The convergence rate of the CALA-LP for Hep-ph data set.



**Fig. 4.** The convergence rate of the CALA-LP for Astro-ph data set.

learning capability of continuous action set learning automata that shows the link occurrence in the network depends on different similarity metrics and different periods with different coefficients.

## References

- [1] Linyuan Lü, Tao Zhou, Link prediction in complex networks: A survey, *Physica A* 390 (6) (2011) 1150–1170.
- [2] Mohammad Al Hasan, Mohammed J. Zaki, A survey of link prediction in social networks, in: *Social Network Data Analytics*, Springer, US, 2011, pp. 243–275.
- [3] Amr Ahmed, Eric P. Xing, Recovering time-varying networks of dependencies in social and biological studies, *Proc. Natl. Acad. Sci.* 106 (29) (2009) 11878–11883.
- [4] Liben Nowell, David, Jon Kleinberg, The link prediction problem for social networks, *J. Amer. Soc. Inf. Sci. Technol.* 58 (7) (2007) 1019–1031.
- [5] Newman, E.J. Mark, Clustering and preferential attachment in growing networks, *Phys. Rev. E* 64 (2) (2001) 025102.
- [6] Gerard Salton, Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.
- [7] P. Jaccard, Etude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bulletin de la Societe Vaudoise des Science Naturelles* 37 (1901) (1901) 547.
- [8] Erzsébet Ravasz, et al., Hierarchical organization of modularity in metabolic networks, *Science* 297 (5586) (2002) 1551–1555.
- [9] E.A. Leicht, Petter Holme, Mark E.J. Newman, Vertex similarity in networks, *Phys. Rev. E* 73 (2) (2006) 026120.
- [10] Albert-László Barabási, Réka Albert, Emergence of scaling in random networks, *Science* 286 (5439) (1999) 509–512.
- [11] Lada A. Adamic, Eytan Adar, Friends and neighbors on the web, *Soc. Netw.* 25 (3) (2003) 211–230.
- [12] Tao Zhou, Linyuan Lü, Yi-Cheng Zhang, Predicting missing links via local information, *Eur. Phys. J. B* 71 (4) (2009) 623–630.
- [13] Leo Katz, A new status index derived from sociometric analysis, *Psychometrika* 18 (1) (1953) 39–43.
- [14] Pavel Chebotarev, Elena Shamis, *The matrix-forest theorem and measuring relations in small social groups*, 2006. ArXiv Preprint math/0602070.
- [15] Linyuan Lü, Ci-Hang Jin, Tao Zhou, Similarity index based on local paths for link prediction of complex networks, *Phys. Rev. E* 80 (4) (2009) 046122.
- [16] Weiping Liu, Linyuan Lü, Link prediction based on local random walk, *Europhys. Lett.* 89 (5) (2010) 58007.
- [17] Douglas J. Klein, M. Randić, Resistance distance, *J. Math. Chem.* 12 (1) (1993) 81–95.
- [18] Francois Fouss, et al., Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, *IEEE Trans. Knowl. Data Eng.* 19 (3) (2007) 355–369.
- [19] Sergey Brin, Lawrence Page, The anatomy of a large-scale hypertextual Web search engine, *Comput. Netw. ISDN Syst.* 30 (1) (1998) 107–117.
- [20] Glen Jeh, Jennifer Widom, SimRank: a measure of structural-context similarity, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2002.
- [21] G. Santharam, P.S. Sastry, M.A.L. Thathachar, Continuous action set learning automata for stochastic optimization, *J. Franklin Inst. B* 331 (5) (1994) 607–628.
- [22] Zan Huang, K.J. Lin Dennis, The time-series link prediction problem with applications in communication surveillance, *INFORMS J. Comput.* 21 (2) (2009) 286–303.
- [23] Tomasz Tylenda, Ralitsa Angelova, Srikanta Bedathur, Towards time-aware link prediction in evolving social networks, in: *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, ACM, 2009.
- [24] Chao Wang, Venu Satuluri, Srinivasan Parthasarathy, Local probabilistic models for link prediction, in: *Seventh IEEE International Conference on Data Mining*, 2007. ICDM 2007, IEEE, 2007.
- [25] Daniel M. Dunlavy, Tamara G. Kolda, Evrim Acar, Temporal link prediction using matrix and tensor factorizations, *ACM Trans. Knowl. Discov. Data* 5 (2) (2011) 10.
- [26] Satoshi Oyama, Kohei Hayashi, Hisashi Kashima, Cross-temporal link prediction, in: *2011 IEEE 11th International Conference on Data Mining (ICDM)*, IEEE, 2011.
- [27] Jean-Philippe Vert, Yoshihiro Yamanishi, Supervised graph inference, in: *Advances in Neural Information Processing Systems*, 2004.
- [28] da Silva Soares, Ricardo Paulo, R. Bastos Cavalcante Prudencio, Time series based link prediction, in: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2012.
- [29] G. Rossetti, R. Guidotti, D. Pennacchioli, D. Pedreschi, F. Giannotti, 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, Vol. 2015, pp. 553–558.
- [30] C.A. Bliss, M.R. Frank, C.M. Danforth, P.S. Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, *J. Comput. Sci.* 5 (5) (2014) 750–764.
- [31] A. Ozcan, S.G. Oguducu, Multivariate temporal Link Prediction in evolving social networks, in: *2015 IEEE/ACIS 14th International Conference on Computer and Information Science ICIS*, 2015, pp. 185–190.
- [32] S. Huang, Y. Tang, F. Tang, J. Li, Link 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*, 2014, pp. 706–709.
- [33] L. Yang, W. Zhang, Y. Chen, Time-series prediction based on global fuzzy measure in social networks, *Front. (Boulder)* 1 (1) (2015) 350–365.
- [34] A. Potgieter, K.A. April, R.J.E. Cooke, I.O. Osunmakinde, Temporality in link prediction: Understanding social complexity, *Emerg. Complex. Organ.* 11 (1) (2009) 69–83.
- [35] M. Thathachar, P. Shanti Sastry, Varieties of learning automata: an overview, *IEEE Trans. Syst. Man Cybern. B* 32 (6) (2002) 711–722.
- [36] Kumpati S. Narendra, Mandayam A.L. Thathachar, *Learning Automata: An Introduction*, Courier Dover Publications, 2012.
- [37] M.A.L. Thathachar, Bhaskar R. Harita, Learning automata with changing number of actions, *IEEE Trans. Syst. Man Cybern.* 17 (6) (1987) 1095–1100.
- [38] M.A.L. Thathachar, P. Shanti Sastry, A hierarchical system of learning automata that can learn the globally optimal path, *Inf. Sci.* 42 (2) (1987) 143–166.
- [39] M. Thathachar, Vijay V. Phansalkar, Convergence of teams and hierarchies of learning automata in connectionist systems, *IEEE Trans. Syst. Man Cybern.* 25 (11) (1995) 1459–1469.
- [40] Hamid Beigy, M.R. Meybodi, A new continuous action-set learning automaton for function optimization, *J. Franklin Inst. B* 343 (1) (2006) 27–47.
- [41] Oommen, B. John, T. Dale Roberts, Continuous learning automata solutions to the capacity assignment problem, *IEEE Trans. Comput.* 49 (6) (2000) 608–620.
- [42] Mohsen Jahanshahi, Mehdi Dehghan, Mohammad Reza Meybodi, On channel assignment and multicast routing in multi-channel multi-radio wireless mesh networks, *Int. J. Ad Hoc Ubiquitous Comput.* 12 (4) (2013) 225–244.
- [43] Mohsen Jahanshahi, Mehdi Dehghan, Mohammad Reza Meybodi, A mathematical formulation for joint channel assignment and multicast routing in multi-channel multi-radio wireless mesh networks, *J. Netw. Comput. Appl.* 34 (6) (2011) 18.
- [44] Akbari Torkestani, Javad, Mohammad Reza Meybodi, Finding minimum weight connected dominating set in stochastic graph based on learning automata, *Inform. Sci.* 200 (2012) 57–77.
- [45] Akbari Torkestani, Javad, Mohammad Reza Meybodi, A learning automata-based heuristic algorithm for solving the minimum spanning tree problem in stochastic graphs, *J. Supercomput.* 59 (2) (2012) 1035–1054.
- [46] Akbari Torkestani, Javad, Mohammad Reza Meybodi, Learning automata-based algorithms for finding minimum weakly connected dominating set in stochastic graphs, *Int. J. Uncertain. Fuzziness Knowl.-based Syst.* 18 (06) (2010).
- [47] J. Kazemi Kordestani, A. Ahmadi, M.R. Meybodi, An improved differential evolution algorithm using learning automata and population topologies, *Appl. Intell.* (2014) <http://dx.doi.org/10.1007/s10489-014-0577-2>.
- [48] D. Yazdani, B. Nasiri, A. Sepas-Moghadam, M.R. Meybodi, M.R. Akbarzadeh-Totonchi, mNAFSA: A novel approach for optimization in dynamic environments with global changes, in: *Swarm and Evolutionary Computation*, Elsevier, 2014, <http://dx.doi.org/10.1016/j.swevo.2014.05.002>.
- [49] A. Alizadegan, M.R. Meybodi, B. Asady, A novel hybrid artificial bee colony algorithm and differential evolution for unconstrained optimization problems, *Adv. Comput. Sci. Eng.* 8 (1) (2012).
- [50] Mohammad Soleimani-Pouri, Alireza Rezvanian, Mohammad Reza Meybodi, An ant based particle swarm optimization algorithm for maximum clique problem in social networks, in: *State of the Art Applications of Social Network Analysis*, Springer. International Publishing, 2014, pp. 295–304.

- [51] Mohammad Soleimani-Pouri, Alireza Rezvanian, Mohammad Reza Meybodi, Finding a maximum clique using ant colony optimization and particle swarm optimization in social networks, in: Proceedings of the 2012 International Conference on Advances in Social. Network Analysis and Mining (ASONAM 2012), IEEE Computer Society, 2012.
- [52] Alireza Rezvanian, Mohammad Rahmati, Mohammad Reza Meybodi, Sampling from complex networks using distributed learning automata, *Physica A* 396 (2014) 224–234.
- [53] Mohammad S. Obaidat, et al., Learning automata-based bus arbitration for shared-medium ATM switches, *IEEE Trans. Syst. Man Cybern. B* 32 (6) (2002) 815–820.
- [54] H. Mostafaei, M. Esnaashari, M.R. Meybodi, A coverage monitoring algorithm based on learning automata for wireless sensor networks, *Int. J. Appl. Math. Inf. Sci.* (2014).
- [55] S.M. Safavi, M.R. Meybodi, M. Esnaashari, Learning automata based face-aware mobicast, in: *Wireless Personal Communications*, Springer, 2014.
- [56] A. Safari Mamaghani, K. Asghari, M.R. Meybodi, Designing a new structure based on learning automata to improve evolutionary algorithms (with considering some case study problems), *J. Adv. Comput. Res.* 4 (3) (2013) 1–24.
- [57] Thathachar, A.L. Mandayam, Pidaparty S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*, Springer, 2003.
- [58] Jilin Chen, et al., Make new friends, but keep the old: recommending people on social networking sites, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2009.
- [59] Yan-Bo Xie, Tao Zhou, Bing-Hong Wang, Scale-free networks without growth, *Physica A* 387 (7) (2008) 1683–1688.