A Variable Action Set Cellular Learning Automata-based Algorithm for Link Prediction in Online Social Networks

Mozhdeh KhaksarManshad[1]
Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
st_m_khaksarmenshad@azad.ac.ir
Mohammad Reza Meybodi
Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
mmeybodi@aut.ac.ir
Afshin Salajegheh
Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
a_salajegheh@azad.ac.ir

Abstract

Link prediction (LP) is a crucial issue in the online social network (OSN) evolution analysis. Since OSNs are growing in size on a daily basis, a growing need for scalable LP algorithms is being felt. OSNs are innately evolutionary, such that the characteristics, behavior, and activities of their components (including nodes and links) change over time. In analyzing social networks which are based on the time evolution model, LP helps us realize the logic of social network growth. Deriving time patterns of evolutionary changes according to the communities and neighbors of nodes in a network can be aptly used for LP. This article introduces a new algorithm based on irregular cellular learning automata (ICLAs) for LP in the near future in OSNs. The algorithm we propose here models the network as an ICLA. The ICLA weighs the real links in the network according to entities' participation in forming communities over consecutive time periods. This method lies in the premise that social networks include communities. Based on the communities formed over successive time periods, the presented method calculates the probability of link formation between every pair of nodes which are unconnected at the present time, estimating the chances of their connection in the near future. Experiments performed on real social networks show that the proposed algorithm produces good results in predicting link formation in OSNs.

Keywords: link prediction, cellular learning automata, communities, online social networks

1. Introduction

Because social networks' entities are growing day by day, they have provided useful information for issues related to social network analysis, such as link prediction (LP) problem. The problem of LP can be defined as evaluating the probability of the formation of new connections between entities in the near future [1-4]. Link prediction is important in many ways; on the one hand, it helps to understand the logic of social networks evolution, and on the other hand, it is used in various fields such as recommendation systems in e-commerce [5,6], protein-protein interaction prediction in bioinformatics [7], and identify hidden groups of criminals in security-related systems [8].

Existing methods for the LP problem are categorized into two different categories. The first category is similarity-based methods that use different approaches to calculate the similarity between nodes. For example, in node-based approaches [9-12], the similarities between nodes are calculated based on their common actions and interests. In topology-based methods [13], structural features of graphs are used to check the similarity between nodes. And finally, in social network criteria-based methods [14-18], concepts of social networks such as clustering coefficients or communities to assess the similarity between nodes is used. With the use of internal features, similarity-based criteria, and external information, the second category of LP, which is called learning-based methods, has been introduced. It includes different subgroups such as feature-based classification algorithms [19-22], learning

[1] Corresponding Author

automata (LA)-based algorithms [23-27], probabilistic -based algorithms [28,29], matrix factorization algorithm[30], and cellular LA-based methods [31].

The model introduced in this paper uses social network criteria-based similarity methods and ICLA, and we call it ICLA-LP.

The cellular learning automaton (CLA) [32] is an appropriate computation tool for dynamic and decentralized problems. They are formed by combining cellular automata (CAs)[33] and LAs [34]. The cellular automaton (CA) model is a dynamic and non-linear model which is discrete in space and time. Each cell in the CA can have finite information within other cells and the entire system. Since the cells have to decide over changing status based on limited information, certainty will be lacking in the decisions made. Adding learning to the cells' decision-making will help overcome this lack of certainty. As a decision-making machine, an LA is a good tool for learning the best action from the allowable actions by dealing with a random environment. The CLA outdoes a CA since it endeavors to learn the best action. It is nevertheless much more efficient than an LA because it improves the ability to learn through a set of interacting automata. The irregular CLA (ICLA) [35] is an extension of the CLA model in which the assumption of having a regular structure has been eliminated.

The model proposed here is founded upon two assumptions. The first is that entities in a network incline towards joining communities. The second is that in a social network, entities incline towards joining the communities of their friends. Thus, in the proposed model, measuring similarity between two nodes is based upon joint communities, where the degree of participation of a link in forming different communities will be decided through ICLAs.

This article proposes a new LP algorithm for social networks based on ICLA shortened as (ICLA-LP). In our proposed method, the social network has been modeled as an ICLA so that each vertex is equipped with a cell of ICLA and that there is an LA in each cell. The LAs which are resident in cells of ICLA evolve through interaction with the local environment, and the desired weight for connected links in the network will be calculated based on the network LAs' probability vector. The proposed algorithm indicates the level of dependency between pairs of nodes, estimating their tendency to forming a community together. The proposed algorithm's main structure is a weighted graph built upon the ICLA's action probability vector, specifying the level of interactions among the node's entire neighborhoods. As for LP, a math formula has been proposed in this algorithm to calculate the probability of link formation between unconnected pairs of nodes. This formula is described based on the weight of the number of paths among unconnected pairs of nodes. Generally speaking, the more paths with smaller lengths and higher weights among unconnected pairs of nodes, the greater the likelihood of a link between them in the future [36].

Using ICLAs, in this article, we have introduced a method that draws information from graph structure to offer a novel meaning of observing the quality of connections among pairs of vertices in networks. A new similarity criterion is introduced upon which connection quality. Since OSNs come with dynamic environments, and the nodes and links are added or omitted as time passes, LP's problem in OSNs calls for algorithms that can adapt to social network changes. According to past research, the CLA has demonstrated excellent performance in dynamic environments [37]; therefore, it offers a useful tool for OSNs.

In the results and experiments section, the proposed method has been compared with local similarity algorithms such as Jaccard (JC) [38], preferential attachment (PA) [39], Adamic-Adar(AA) [40], and common neighbors (CN)[3], a semi-local algorithm such as local path [41], a global algorithm such as Katz [42], and some recently-proposed methods such as ant colony LP (ACO-LP) [43], mutual information of network structure LP (MI-LP) [21], irregular CLA-based evolutionary computation (ICLA-EC-TSLAP) [31], interaction prediction (IP) [22], LA-based LP (LA-TSLP) [23], and fuzzy LP based on distributed LA (FLP-DLA) [25], continuous action set LA for link prediction (CALA-LP) [24], and covariance matrix adaptive evolution strategy (CMA-ES) [19].

We have summarized the contributions of this paper in the following:

- The ICLA is compatible with the social network structure, and it can adapt itself to the dynamic structure of the social network. For weighting the links of social network and considering the link and neighborhood dynamics, we introduced a model based on ICLA. For modeling the social network based on ICLA, we have assigned an LA to each vertex. The main role of each LA is to learn the weight of connections associated with the corresponding vertex.
- Each LA in a vertex considers local communities of consecutive time *1* to *T* in the dynamic social network to learn the weight of connections which are related to the corresponding vertex. So we consider the evolution of the temporal network by using ICLA.
- A new formula has been proposed for calculating the score of unconnected links by using the weighted network.

The upcoming sections of this paper have been sorted as follows: In Section 2, the problem's formulization and the related works have been presented. In Section 3, a brief explanation is given about the CLA and ICLA theories. Section 4 provides full descriptions of the proposed algorithm. In Section 5, the experiment results and their comparison have been elaborated. In Section 6, the main findings of the research and future related work have been briefly discussed.

2. Related backgrounds

2.1. Problem Formulation

In this paper, we concentrate specifically on undirected unipartite graphs. In our experiments, we consider deterministic networks and OSNs. OSNs are formally presented as taking after [44]. To begin with, we divide the temporal dataset of the network into snapshots sliced at times which go from time *1* to time *T*. Each one of these snapshots is representative of the state of the network at different time periods. Let V be the set of vertices, $V = (1,2,\dots,N)$. A set of graph sequence $(G_1, G_2, \dots, G_T)$ which relevant to a set of series of symmetric adjacent matrices $(M_1, M_2, \dots, M_T)$, where the network structure at time *t* is demonstrated by G_t with a vertex collection V_t and a link collection E_t. The graphs at times *t* and *t+1* have identical group of nodes, but have different group of links. Each M_t is a *N*×*N* matrix with values $M_t(i,j)$ which correspond with links $E_t(i,j)$. The value of $M_t(i,j)$ is picked out of set *{0, 1}*, which is representing the presence or absence of edge *(i, j)* at time *t*. With a graph series like this, our LP algorithm aims to predict the probabilities of the formation of the edge at time *T+1* by utilizing adjacency matrices $M = (M_1, M_2, \dots, M_T)$.

2.2. Related Work

Since the proposed method uses ICLA to consider edge quality in community formation as a vertex similarity criterion in time-series networks, in the present part of the paper, we will have a look at a number of algorithms for structural and time-series LP (TSLP). In the methods used so far for the problem of predicting connections in social networks based on LAs, only a set of LAs and similarity-based criteria have been used [23,25]. In [31], a new evolutionary algorithm called ICLA based-evolutionary computation (ICLA-EC) for link prediction problem has been proposed. In the proposed algorithm, an ICLA-EC has been assigned to each vertex in the social network, and there is a collection of LAs and a genome in each node. The genome in each node represents the predicted links for the corresponding node. Each genome in a vertex is developed based on local information as well as its previous experiences besides experiences of neighbor vertices in consecutive time *1* to *T*.

But in the way we have covered in this article, we have assigned a cell of ICLA to each vertex of the social network, and there is an LA in each cell. The irregular structure of ICLA is compatible with the social network structure, and the intuition of this algorithm is that the ICLA can adapt to changes in social networks. The actions of each LA in each vertex determine the weight of corresponding links which are related to the corresponding vertex. The importance of links is estimated based on their participation in constructing local communities over successive time *1* to *T*.

Reference [45] introduces a supervised structural LP algorithm that offers a new method to depict social networks' dynamicity for the LP problem. In this method, the network graphs, including triads of nodes, are specified, and their transition throughout the evolution of the network is measured. A Triad Transition Matrix (TTM) has been defined to store the probability of transition among the triads found in the network. Then they show how this matrix can measure the dynamicity of network evolution patterns. They also demonstrate the application of TTM in the LP problem, finding that the proposed method offers good performance for sparse networks analyzed over short time periods.

Reference [46] offers a structurally supervised LP algorithm which finds structures called vertex collection profile (VCP) in social networks. $VCP_{s,t}^{n,r}$ is a substructure which includes *s* and *t* nodes and has a total of *n* nodes, where there is a maximum of *r* relations between each pair of existing nodes. There is not any relationship between the two *s* and *t* nodes in a VCP. VCPs provide quite comprehensive information about the local structure surrounding pairs of nodes. If *n* and *r* are greater than 4, the number of VCPs will grow beyond imagination. In this algorithm, if a learning stage is added, it will suit the LP problem. Among the weak points of this algorithm is that it has a considerable execution time, which is impractical in the case of big networks where VCPs come with higher *n* values than 4.

In [43], a structural LP algorithm is introduced based on the ACO-LP ant colony algorithm. In this approach, firstly, particular triangular-triad subgraphs have been recognized in the network. Next, their evolvement is examined to predict new links in the social network. The ant colony algorithm has discovered triangular subgraphs. The ACO-LP improves the execution time, and the achieved results for some datasets are much better than other unsupervised LP methods.

Reference [47] introduces a new probabilistic LP algorithm using the concept of clustering coefficient. The clustering coefficient shows the inclination towards forming clusters in a graph. In other words, the clustering coefficient is achieved by dividing the number of cycles of the length of *k* by the number of the paths of the length of *k* in the Graph. Therefore, the likelihood of length 3 and 4 paths turning into length 3 and 4 cycles will determine the probability of link formation between two nodes. Using a graph's clustering coefficient, the parameters of this probabilistic model will be obtained.

In Reference [48], special subgraphs named microscopic are studied in directed graphs. In this study, it has been observed that some of the sub-graphs are found in social networks more often. These subgraphs are called bi-fan and include four nodes and four directed links. In subgraphs which have one link less than the Bi-fan structure, the link under consideration will very probably be formed in the near future. This structure makes up the cornerstone for an LP algorithm that is introduced in Ref. [48].

In [49], a new TSLP algorithm has been proposed, and the number of repetitions of the edges between the nodes is calculated in different time intervals. An autoregressive integrated moving average model (ARIMA) has been used to predict links in the near future. Huang et al. [50] have proposed a new TSLP algorithm that used time-series patterns and similarity-based methods together, and the subsequent value has been predicted by using time-series data and the ARIMA model. The final predictions have been computed according to a combination of anticipated results and some similarity-based algorithms. Also, an interaction prediction (IP) algorithm has been proposed in [22] for networks which are used feature selection, besides time-series prediction for predicting new connections in the future. In [51], a new similarity criterion for predicting links in a time-evolving social network based on node ranking has been proposed. The authors have also used a new adaptive model to predict each relation's score in the network. The achieving results in [51] show that the proposed model can be used to predict links in both scale-free networks and dynamic social networks. Mallek et al. [52] have introduced a novel link prediction method using common community and neighborhood information. In this method, a new model for considering uncertainties in the connections structure has been proposed. Then a new approach for link prediction by using belief function tools has been introduced.

In [24], continuous-action learning automata (CALA), besides temporal similarity-based methods, has been considered, and a new learning-based algorithm called CALA-LP has been introduced. The CALA-LP considers the LP problem as an optimization one, and a set of CALA is used to discover the appropriate solutions to the problem. The achieved results have shown that the CALA-LP algorithm is suitable for some social network datasets. Likewise, in [23], a new method based on LAs has been introduced, which has utilized LAs to predict the presence or absence of links at time *T+1* by using different temporal similarity-based metrics through time *1 to T*. Similarly, in [25], an algorithm has been proposed based on fuzzy concepts and distributed LAs (DLAs) which is called FLP-DLA. It has estimated the strength of connections by using the information of the social network. The connections' power has been considered as the outcome of the LP algorithm. Moradabadi et al. [27] have introduced an LP method based on LAs for stochastic networks. Some of the similarity criteria have been redefined for stochastic networks. Then an LA-based method has used the proposed similarity criteria for predicting connections in the stochastic social networks. Also, in [26], a new LP method based on LAs for weighted social networks has been introduced. In the proposed method, an LA has been assigned to each test link that must be predicted in the near future, and each LA tries to learn the actual weight for the corresponding test link by using the importance of real connections in the weighted social network. In [31], a novel evolutionary TSLP method based on ICLA and evolutionary computing (ICLA-EC-TSLP) has been proposed. Local information between vertices in the successive time *1* to *T* has been used for predicting links in time *T+1*.

Another TSLP algorithm for predicting links in evolving networks has been introduced in [53] called Multivariate TSLP. It has combined similarities of nodes and information of nod connectivity at different times. The authors in [54] have proposed an LP method based on evolutionary strategy and covariance matrix called CMA-ES. They have utilized CMA-ES to enhance the weights used in a linear combination of some similarity-based algorithms. They have employed some social networks with more than 10^6 nodes. The CMA-ES has demonstrated rapid convergence with high accuracy for the first twenty predicted links in their experiments.

The authors of [55] have introduced a novel method based on mutual information of network structure and information theory, and it is called MI-LP. For analyzing the MI-LP, they have used ten networks and compared them with six link prediction algorithms. The obtained results have shown that MI-LP has a logical computational complexity, and it improves Precision.

3. Background

This section briefly describes learning automata (LAs) and cellular learning automata (CLAs). Then, we will review the ICLA as an extension of the basic form of the CLA.

3.1. LA

A learning automaton (LA) [34,56] can be a decision-making tool that adaptively learns the optimal action via recurrent interactions with a random unknown environment. This process is performed through cooperation between the automaton and the random environment on the other side. At each stage, the LA, upon the action-sets' probability distribution, chooses an action out of a limited collection of actions and applies that to the environment.

After that, the selected action is evaluated by the environment, and feedback of the environment is received by the LA, which is used to update the action probabilities [56]. Fig. 1 indicates how an LA interacts with its environment. LAs can be categorized into two significant families [56], i.e., fixed LAs and variable structure LAs.

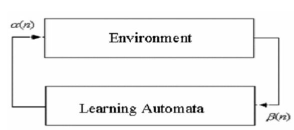


Figure1. The interaction between LAs and their environment

Subsequent to the automaton choosing the action i, the reinforcement signal will be received from the environment. When an LA gets a positive response from the environment ($\beta = 0$), the action probabilities are updated through Equation (1):

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (1)$$
$$p_{j \neq i}(n+1) = (1-a)p_j(n)$$

When an LA gets a negative response from the environment ($\beta = 1$), action probabilities are updated following Equation (2):

$$p_i(n+1) = (1-b)p_i(n) \qquad (2)$$
$$p_{j \neq i}(n+1) = \left({}^{b}/_{r-1}\right) + (1-b)p_j(n)$$

In which r represents the number of actions that may be selected by the automata, while a and b are the reward and punishment parameters. If $b = 0$, the above learning Equation is labeled as linear reward-Inaction L_{R-I}; and provided that $0 < b << a < 1$, it is called the linear reward-ε penalty $L_{R-\varepsilon P}$. When $a = b$, the presented Equation called linear reward–penalty L_{R-P}.

It has been shown in many existing studies that LAs are an appropriate tool for operating in dynamic environments with incomplete information. Also, in many studies, it has been demonstrated that LAs can solve many NP-hard problems [57-59]. Recently, a number of algorithms based on LAs have been proposed in [23,25,24,27,26]for improving LP in social networks.

3.2. Variable Action-set LA

The LA is called a variable action-set if the number of actions which can be adopted by the LA at every instant can change with time. Authors in [60] have shown that if the reinforcement scheme is L_{R-I}, an LA that has a variable number of actions is completely expedient and is also ε-optimal. An automaton as this has a finite set of n actions, $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$. $A = \{A_1, A_2, \dots, A_m\}$ signifies the group of action subsets, and $A(k) \subseteq \alpha$ is a subset to the entire actions that may be selected by the LA at each instant k. The selection of specific action subsets is performed randomly by an external agency based on the probability distribution $q(k) = \{q_1(k), q_2(k), \dots, q_m(k)\}$ which is defined over the actions' possible subsets, in which $q_i(k) = prob[A(k) = A_i | A_i \in A, 1 \leq i \leq 2^n - 1]$.

$\hat{p}_i(k) = prob[\alpha(k) = \alpha_i | A(k), \alpha_i \in A(k)]$ is the probability of choosing action α_i, with the condition that action subset $A(k)$ has previously been selected, and that $\alpha_i \in A(k)$. The scaled probability $\hat{p}(k)$ is defined as:

$$\hat{p}_i(k) = p_i(k)/K(k) \qquad (3)$$

Where $K(k) = \sum_{\alpha_i \in A(k)} = p_i(k)$ is the sum of the probabilities of the actions in subset $A(k)$, and $p_i(k) = prob[\alpha(k) = \alpha_i]$.

The process of selecting an action and also updating the probabilities of action in a variable action-set LA can be described as the following. Let $A(k)$ denote the action subset chosen at instant k. Before picking up action, the probabilities for the selected subset's entire actions are scaled as defined in Equation (3). After that, the automaton will randomly choose one of its probable actions based on the scaled action probability vector $\hat{p}(k)$. Judging from the response that the environment has given, the LA updates its scaled action probability vector. Please take into account that the probability of the available actions is only updated. In the end, the actions of the chosen subsets' probability vector are rescaled as:

$$p_i(k+1) = \hat{p}_i(k+1). K(k) \qquad (4)$$

for all $\alpha_i \in A(k)$, absolute expediency, as well as ε-optimality of the above-described method, are proved in [60]. The pseudo-code of the variable action-set LA has been illustrated in Fig. 2.

Algorithm 1: Variable action-set LA

Input: Action-set α
Output: Action probability vector p
Assumptions:
Initialize r-dimensional action-set $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ with r actions
Initialize r-dimensional action probability vector $p = \{p_1, p_2, \dots, p_r\} = \{\frac{1}{r}, \frac{1}{r}, \dots, \frac{1}{r}\}$ at time k
Let j denotes the current checking action
Let i denotes the selected action by the automaton
Begin
  While (LA converge to an action)
    Calculate the available action of the LA
    Calculate the sum of the probability of available actions
    For each action $j \in \{1,2,\dots,r\}$ do
      If (α_j is available action)
        Scale action probability vector $\hat{p}_i(k)$ according to Equation (3)
      End if
    End for
    The LA selects an action based on the probability vector of available actions $\hat{p}(k)$
    The environment evaluates the selected action and gives the reinforcement signal $\beta \in \{0,1\}$ to the LA
    For each available action $j \in \{1,\dots,m\}$ do
      If ($\beta = 0$) //favorable action
        The selected action by LA is rewarded according to Equation (1)
      Else if ($\beta = 1$) //unfavorable action
        The selected action by LA is punished according to Equation (2)
      End if
    End for
    For each action $j \in \{1,\dots,r\}$ do
      If (α_j is available action)
        Rescale the probability vector of selected available action by Equation (4)
      End
    End for
  End while
End Algorithm

Figure2. The pseudo-code of the variable action-set LA

3.3. CLA and Irregular CLA

Cellular learning automaton (CLA) [32], which is created by merging cellular automaton (CA) [33] and LA, is a capable model for some decentralized issues. The potentials of LA can fully demonstrate themselves when a multiplicity of automata interact with each other. A CLA is defined as a CA, and there is an LA in each cell. The LA, a cell resident, uses its action probability to specify the cell's state. Like in the CA, a rule governs the operation of the CLA. The rule for the CLA and the actions chosen by the neighboring LAs regulate the reinforcement signal that is to go to the LA that resides in a cell.

The CLA mechanism can be defined according to the following. Firstly, each internal state of cells is specified. The vector for action probability of each LA resides in the cells is initialized based on past experiences or otherwise at random. After that, the LA in each cell determines its state according to the action probability vector, receiving a local environment response. In the end, every LAs' action probability vector is updated upon the response of the environment. This procedure is done over and over until the optimal state of each cell has been reached.

CLAs can be divided into two groups: synchronous and asynchronous CLAs. In the first group, the LAs, cell residents, in the entire lattice of cells are activated simultaneously. For the second group, only some of the LAs are activated independently of each other at each given point in time.

An irregular CLA (ICLA) [35] is a generalization of the traditional CLA that overcomes a rectangular grid structure's limitations. An ICLA can be defined as an undirected graph where each node is, in fact, a cell which is equipped with an LA, and in which the neighbors of any specific cell create the local environment of the cell. Notwithstanding its irregular structure, ICLA behaves equivalently with CLA.

ICLA can be defined formally by a structure $A = (G, \emptyset, L, f)$, in which:

- $G = (V, E)$ represents an undirected graph, in which V is the set of vertices while E is the set of edges.
- $\emptyset$ indicates the state set for the ICLA.
- L denotes the set of LAs, each of which is given to a cell of the ICLA.
- $f: \emptyset^{N(i)} \rightarrow \beta$ represents the local rule of the ICLA for each node, in which $N(i)$ is the collection of the neighbor vertices of vertex i in graph G, $\emptyset^{N(i)}$ denotes the state of LAs in the neighbor vertices, while β is the value set for the response. It calculates every LA's response according to the current states of the LAs that reside in the neighbor vertices.

4. The Proposed ICLA based algorithm for LP (ICLA-LP)

As we have mentioned in earlier sections, the basis for the proposed algorithm in this article is to weight social network links according to their participation in forming communities in different time periods. If an entity has a connection with an individual in a community, it has shared favorites and goals with the individuals in that community; in the near future, it will very probably form connections with other individuals in that community. As it has been shown in Fig. 3(a), nodes v_B, v_C and v_D form a triad community, and there is a connection between a vertex v_A as the source vertex and vertex v_B as the destination vertex in the community, so it is likely that in the near future, there will be a link between vertex $v_A$and other vertices in the community, vertices v_C and v_D. For predicting the probability of the formation of a new link in the future between two vertices, no less than one path should exist between the two vertices. Therefore, LP problem can be defined as the problem to predict connectins between a source and a destination node within a community (Fig. 2(a)). The smallest community which can exist for LP is one with two vertices; while the sum of vertices in a community rises, the probability of the formation of new links among source vertices and destination vertices in the community increases.

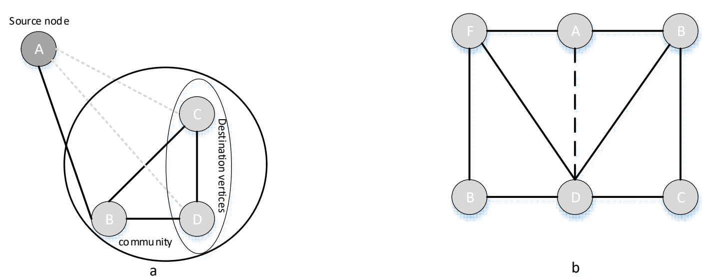


Figure 3. (a) v_B, v_C, v_D represent the community of destination vertices and link E_{AB} is a connection between a source vertex and a destination vertex, so in the near future, there may be two links E_{AD} and E_{AC}between source vertex and destination vertices shown in the figure with the dotted lines. (b) Link AD gets higher weight because it is overlapped between two communities v_A, v_F, v_D and v_A, v_B, v_D.

As social networks are dynamic and branching, changes occur in social networks in different time periods. Communities may appear, merge, disappear, divide, or get smaller or larger, or even go unchanged in different time periods. Understanding the network's evolutionary patterns will help us realize the network's evolution, which will finally enable us to offer approaches to the LP problem.

In this article, we use ICLA to analyze network evolution through community dynamicity. The intuition behind this algorithm, which is suitable for social networks' requirements, is the irregular structure. This article's community dynamicity is analyzed based on the participation of entities in building local communities in time-series networks using ICLA. Also, the dynamicity of communities will consider temporary neighborhood changes.

Tapping local information, in this paper, we have used ICLA to weight links which have participated in forming triad communities as triangles. It was also possible to analyze more complicated communities with greater numbers of nodes, but that would entail a more complex algorithm, which is not the aim of this paper. The links that form triangular communities are given more scores compared to other edges. As demonstrated in Fig. 3(b), if two triangular communities share an edge (link E_{AD} is shared between triangular communities v_A, v_F, v_D and v_A, v_B, v_D), it will be given higher weight throughout the algorithm.

Also, a vertex has high social dynamicity if most of its neighbors play a part in triangular communities and if it at the same time preserves its neighborhoods in consecutive time periods. Also, if an entity changes its communities in successive time periods, its social dynamicity will decrease. The higher the vertex's social dynamicity, the more weight will go to the links related to that vertex, and vice versa. As it has been shown in Fig. 4, vertex v_{A_1} preserves its neighborhoods in successive time periods t_1 , t_2 , so, v_{A_1} has high social dynamicity, and the links which are connected to v_{A_1} can gain high weights during the ICLA-LP algorithm.

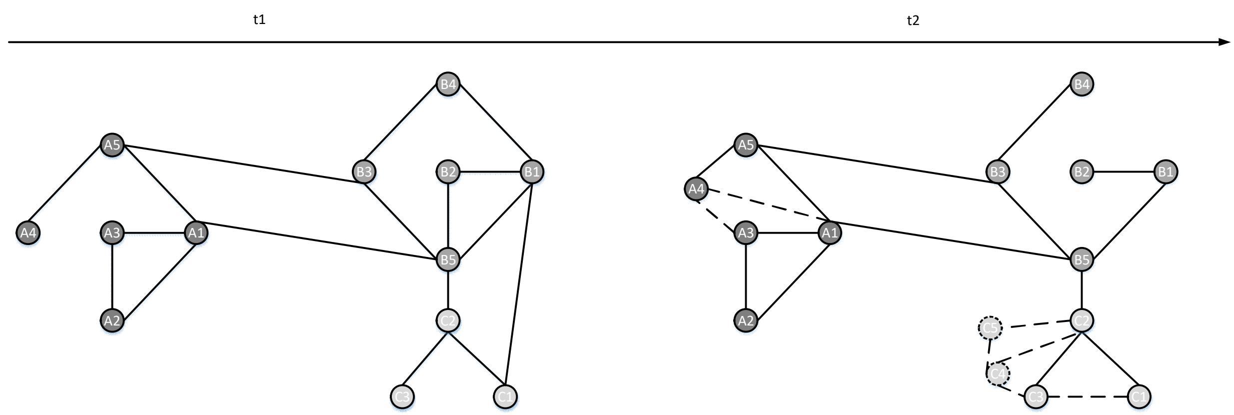


Figure 4. A demonstration of the inclination of entities to change the community in an online network in two different time-series t_1, t_2. The color of entities indicates their neighborhood and the communities they are a part of. Three entities of $v_{A_1}, v_{B_1}, v_{C_1}$ along with triangular communities they have formed in two time periods t_1, t_2 with the help of their neighborhoods have been illustrated. As time advanced from t_1 to t_2, entities v_{A_1} and v_{C_1} found new neighborhoods, and the number of their triangular communities increased. If they preserve these same neighborhoods in future time periods, their links will gain high weights.

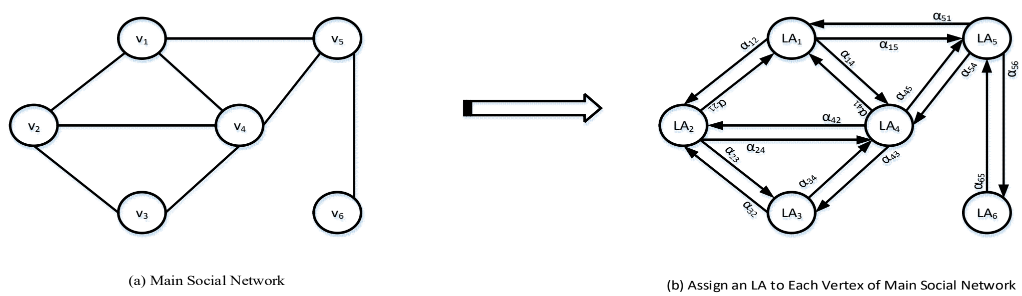
In the first step of the proposed algorithm, the links formed by a triangular triad community will be weighted by analyzing the ICLA. As we have shown the main steps of the ICLA-LP algorithm for weighting the connections in social network in Fig. 8, it consists of modeling step, running step, and change step, which is described as follows:

Modeling step:

For modeling a dynamic social network, we use variable action set CLA. In this way, the main social network is considered which consists of a set of vertices $V = (1,2, \dots, N)$, and a set of connections which are appeared in different timestamps $(1,2, \dots, T)$, in other words, as we have explained in section 2.1 the connections of the main social network is the collection of connections in the adjacent matrix $(M_1, M_2, \dots, M_T)$. In the ICLA-LP, the social network is modeled as an ICLA. ICLA is also defined as a connected graph in which each node indicates a cell, and each link shows an adjacency relation between two cells. There is an LA in each cell, so we have a set of *N* LAs,

$A = \{A_1, A_2, \dots, A_N\}$, with the asset of action sets $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ in which $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ir_i}\}$ defines the set of actions which can be taken by LA A_i for each $\alpha_i \in \alpha$ and $r_i$is the number of adjacent of a vertex. For example, if a vertex has *m* adjacent vertices, then r_i will be equal to m.

As it has been shown in Fig. 5, an LA has been assigned to each vertex of the main social network. In this paper, we consider undirected social networks. In Fig. 5 (b), the actions which are assigned to each cell are shown. Each action related to a cell shows the adjacent relationship between the corresponding vertex with another vertex. For example α_{12} is an action which is related to LA_1 and shown that there is a connection between a vertex $v_1$and vertex v_2. Likewise, α_{21} is an action which is related to LA_2 and shown that there is a connection between a vertex $v_2$and vertex v_1. In the proposed algorithm, we use the action probability of α_{12} and α_{21} for calculating the weight of the edge $E_{12} = E_{21}$.



$\alpha=\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$

$\alpha_1 =\{\alpha_{12}, \alpha_{14}, \alpha_{15}\}$ $\alpha_2 =\{\alpha_{21}, \alpha_{24}, \alpha_{23}\}$ $\alpha_3 =\{\alpha_{32}, \alpha_{34}\}$ $\alpha_4 =\{\alpha_{41}, \alpha_{42}, \alpha_{43}, \alpha_{45}\}$

$\alpha_5 =\{\alpha_{51}, \alpha_{54}, \alpha_{56}\}$ $\alpha_6 =\{\alpha_{65}\}$

Available Action List=$\{\alpha_{12}, \alpha_{14}, \alpha_{15}, \alpha_{21}, \alpha_{24}, \alpha_{23}, \alpha_{32}, \alpha_{34}, \alpha_{41}, \alpha_{42}, \alpha_{43}, \alpha_{45}, \alpha_{51}, \alpha_{54}, \alpha_{56}, \alpha_{65}\}$

Figure 5. (a)A sample social network and (b) the structure of the network when an LA is assigned to each vertex (cell) of the network

Running step:

In this step, we consider time-series graphs to learn the links' weight in the main Graph. The connections that have formed a triangular community will be weighted by analyzing the ICLA at different times for weighing connections. ICLA-LP algorithm starts learning from Graph G_1. There are some available actions based on the links which are appeared in Graph G_1 and other actions are unavailable. The action probability vector of available actions is scaled based on Equation (3). Also, each LA may be in one of the two following modes: inactive or active. In the beginning, all LAs are in inactive mode. At time k the procedure of ICLA-LP based on available action for Graph G_t is described in the following stages:

Let us have an Active-LA list, which is empty at first. ICLA-LP selects the i^{th} inactive LA (LA_i) at random and add it to Active-LA. The LA_i select an action based on its action probability vector. If its selected action is different from its prior action, it activates the neighbors to choose new actions, and active LAs will be added to Active-LA. The local rule for LA_i gets the selected action by itself and its neighbors' actions and generates a reinforcement signal as the following: if the LA_i selects action α_{ij} and one of its adjacent LAs, for example, the l^{th} LA (LA_l) selects action α_{lj}then the chosen action of LA_i is rewarded; otherwise, it is penalized, the action probability vector of LA_i is rescaled upon Equation (4) and LA_i is removed from Active-LA. This procedure is repeated for every LA that is activated.

We have considered running step for Graph G_1 based on Fig. 6. It has been illustrated that the Available Action List consists of available actions in time t_1. The available actions are shown with black color, and unavailable actions are shown with the gray color in the Available Action List. For more details, in the beginning, Active-LA is empty and LA_1 is added to Active-LA randomly, so Active-LA={LA_1}. LA_1 has two actions $\alpha_1 = \{\alpha_{14}, \alpha_{15}\}$ and action probability vector of LA_1 is set to $p_1 = \{0.5, 0.5\}$ initially. Three iterations of the running step of ICLA-LP is described by numerical example as follows. The penalty and reward parameter in this example is set to 0.1, and the learning algorithm of Equations (1) and (2) is exploited to update the action probability vector.

Iteration i=1:

(a) An action has been selected by the active LA i.e. LA_1 We have assumed that the selected action is α_{15} and it is different from the prior action of LA_1.
(b) The adjacent LAs of LA_1 are activated and added to Active-LA, Active-LA={LA_1, LA_4, LA_5}.
(c) The adjacent LAs which are activated select actions based on their action probability vectors. The available action for LA_4 is $\alpha_4 = \{\alpha_{41}, \alpha_{42}, \alpha_{43}, \alpha_{45}\}$ and for LA_5 is $\alpha_5 = \{\alpha_{51}, \alpha_{54}, \alpha_{56}\}$. We have assumed that LA_4 selects action α_{45} and LA_5 selects action α_{56}. As we have mentioned in stage (a) LA_1 selects action α_{15} and one of its adjacent LAs LA_4 selects action α_{45}, so the selected action α_{15} is rewarded based on Equation (1). In other words, LA_1 which is related to v_1 selects v_5 as one of its neighbors and LA_4 which is residing in v_4 and is one of the v_1's neighbors select v_5 too, so the vertices v_1, v_4, v_5 make a triad triangular community and α_{15} related to link E_{15} is rewarded because it is an edge of a triangular community. After updating the stage, the action probability vector of LA_1 will be $p_1 = \{0.45, 0.55\}$. The calculating of each action probability vector is as follows:

$$p_{15} = p_{15} + a(1 - p_{15}) = 0.5 + 0.1(1 - 0.5) = 0.55$$
$$p_{14} = (1 - a)p_{15} = (1 - 0.1)0.5 = 0.45$$

(d) LA_1 is removed from Active-LA and Active-LA={LA_4, LA_5}.

Iteration i=2:

(a) The next LA from Active-LA i.e. LA_4 selects an action from Available Action List $\alpha_4 = \{\alpha_{41}, \alpha_{42}, \alpha_{43}, \alpha_{45}\}$ based on action probability vector $p_4 = \{0.25, 0.25, 0.25, 0.25\}$. We have assumed that the selected action is α_{43} and it is different from the prior action of LA_4.
(b) The adjacent LAs of LA_4 are activated and added to Active-LA, Active-LA={$LA_4, LA_5, LA_1, LA_2, LA_3, LA_5$}.
(c) The adjacent LAs which are activated select actions based on their action probability vectors. We have assumed that LA_1 selects action α_{15} , LA_2 selects action α_{24}, LA_3 selects action α_{34}, and LA_5 selects action α_{56}. So, the selected action α_{43} by LA_4 is penalized based on Equation (2). In other words, LA_4 which is related to v_4 selects v_3 and v_3 has not been selected by adjacent LAs. So, there is not a triad triangular community and α_{43} related to link E_{43} is penalized because it is not an edge of a triangular community. After updating the stage, the action probability vector of LA_4 will be $p_4 = \{0.258, 0.258, 0.225, 0.258\}$. The calculating of each action probability vector is as follows:

$$p_{43} = (1 - b)p_{43} = (1 - 0.1)0.25 = 0.225$$
$$p_{41} = \left({b}/{r - 1}\right) + (1 - b)p_{41} = \left({0.1}/{4 - 1}\right) + (1 - 0.1)0.25 = 0.258$$
$$p_{42} = \left({b}/{r - 1}\right) + (1 - b)p_{42} = \left({0.1}/{4 - 1}\right) + (1 - 0.1)0.25 = 0.258$$

$$p_{45} = \left({}^{b}/_{r\,-\,1}\right) + (1 - b)p_{45} = \left({}^{0.1}/_{4\,-\,1}\right) + (1 - 0.1)0.25 = 0.258$$

(e) LA_4 is removed from Active-LA and Active-LA=$\{LA_5, LA_1, LA_2, LA_3, LA_5\}$.

Iteration i=3:

(a) The next LA from Active-LA i.e. LA_5 selects an action from the available action list $\alpha_4 = \{\alpha_{51}, \alpha_{54}, \alpha_{56}\}$ based on action probability vector $p_4 = \{0.33, 0.33, 0.33\}$. We have assumed that the selected action is α_{56} and it is different from the prior action of LA_5.

(b) The adjacent LAs of LA_5 are activated and added to Active-LA, Active-LA=$\{LA_5, LA_1, LA_2, LA_3, LA_5, LA_1, LA_4, LA_6\}$.

(c) The adjacent LAs which are activated select actions based on their action probability vectors. We have assumed that LA_1 selects action α_{15} , LA_4 selects action α_{43}, and LA_6 selects action α_{65}. So, the selected action α_{56} by LA_5 is penalized based on Equation (2). In other words, LA_5 which is related to v_5 selects v_6 as one of its neighbors and v_6 has not been selected by adjacent LAs. So, α_{56} related to link E_{56} is penalized because it is not an edge of a triangular community. After updating step, the action probability vector of LA_4 will be $p_4 = \{0.347, 0.347, 0.297\}$. The calculating of each action probability vector is as follows:

$$p_{56} = (1 - b)p_{56} = (1 - 0.1)0.33 = 0.297$$
$$p_{51} = \left({}^{b}/_{r\,-\,1}\right) + (1 - b)p_{51} = \left({}^{0.1}/_{3\,-\,1}\right) + (1 - 0.1)0.33 = 0.347$$
$$p_{54} = \left({}^{b}/_{r\,-\,1}\right) + (1 - b)p_{54} = \left({}^{0.1}/_{3\,-\,1}\right) + (1 - 0.1)0.33 = 0.347$$

(f) LA_4 is removed from Active-LA and Active-LA=$\{LA_1, LA_2, LA_3, LA_5, LA_1, LA_4, LA_6\}$.

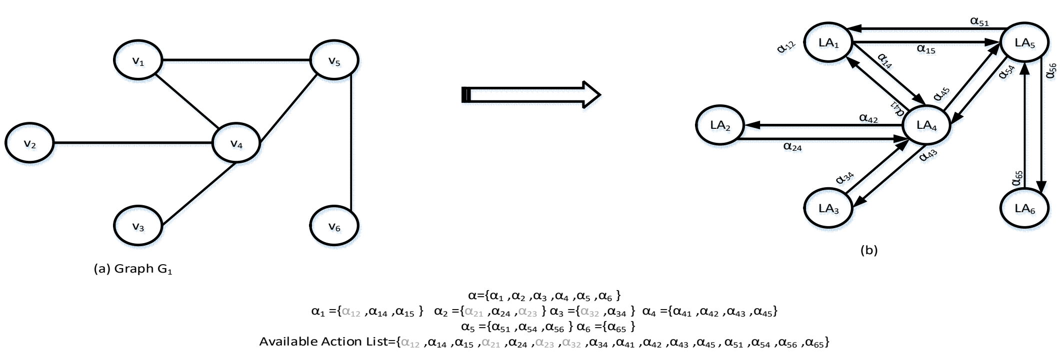


Figure 6. The illustration of Graph G_1 and available actions in time 1

Change step:

This step of ICLA-LP is proposed to adapt the ICLA-LP in online social networks. In online social networks, links can be appeared or disappeared during the time, and we must predict future relations again. In the proposed algorithm, after the running step is terminated, we have a weighted network that we can use for link prediction for the structure of the social network until time T. now for describing the capability of the ICLA-LP to adapt to the online social network, we have assumed that there can be two available modes, a connection is added to the social network or a connection is removed from the social network. These two modes are considered as follows:

1- A new connection is added to the social network: in this case, for the γ^{th} new link that connects two nodes a and b, first the LAs which are residing in vertices a and b (LA_a and LA_b) are added to Active-LA. Second, the actions α_{ab} and α_{ba} which are related to γ^{th} link are activated in the Available Action List and then the action probability vectors of LA_a and LA_b are scaled upon Equation (3). Then, the running step for the Active-LA list is repeated to reconsider the link prediction result. Two Graph G_1 and G_2 in time t_1 and t_2 has been shown in Fig. 7. Edge E_{12} has been added to Graph G_2, so actions α_{12} and $\alpha_{21}$is available in time t_2. Also LA_1 and LA_2 are added to Active-LA and action probability vectors of LA_1 and LA_2 are scaled upon Equation (3).

2- A new connection is removed from the social network: in this case, for the γ^{th} removed link, the two related actions α_{ab} and α_{ba} are disables in Available Action List and the action probability vectors of LA_a and LA_b are scaled upon Equation (3). Then the LAs LA_a and LA_b are activated and added to the Active-LA list. Also, the running step is repeated for the set of LAs in Active-LA to reconsider the link prediction result. It has been shown in Fig. 7(b) that two edges E_{14} and E_{45} is removed in the Graph G_2, so actions $\alpha_{14}, \alpha_{41}, \alpha_{45}, \alpha_{45}$ are not available in time t_2 and LAs LA_1 ,LA_4, and LA_5 are added to Active-LA. Likewise, the action probability vector of LAs LA_1 ,LA_4, and LA_5 are scaled upon Equation (3).

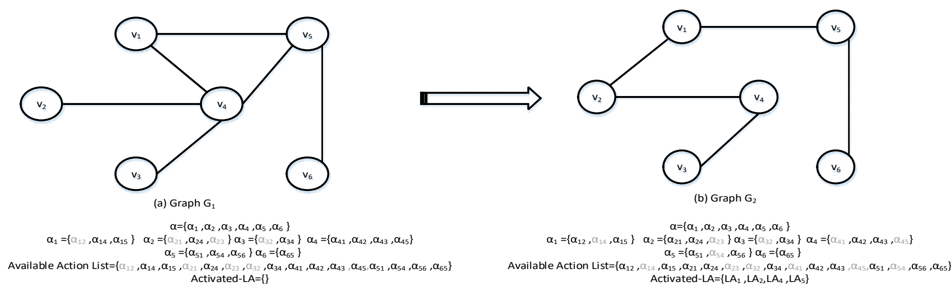


Figure 7. The illustration of adding or removing a connection when the algorithm is proceeding from the time one to time two and the updated Available Action List

As mentioned earlier, the procedure enables us to weigh the OSN's edges according to their participation in forming triangular communities. After the operation of the ICLA is finished, the probability vector of each LA in a cell shows the weights of the social network's edges. Since the probability of selecting a neighborhood v_j in vertex v_i (p_{ij}), differs from the probability of selecting a neighborhood v_i in vertex v_j (p_{ji}), at the end of the ICLA's process, to calculate the weight of an edge such as $E_{ij} = E_{ji}$, it would suffice that it calculated the average of p_{ij} and p_{ji}. Suppose that $G(V, E, W)$ shows the input weight graph so that V is the collection of vertices, E is the collection of edges between a pair of vertices and W is the set of weights of edges, so that for all E_{ij} edges $0 \leq W(i,j) \leq 1$. Based on Fig. 8, for the entire unconnected pairs of vertices (v_i, v_k), in a way that $v_i, v_k \in V$ and $E_{ik} \notin E$, if a vertex such as v_j exists so that $v_j \in V$ and $E_{ij}, E_{jk} \in E$, then the weight of the unconnected link E_{ik} will be calculated as:

$$W_j(E_{ik}) = W\left(E_{ij}\right) * W(E_{jk}) \qquad (5)$$

Algorithm 2. The proposed method of weighting the edges based on ICLA	
	*****Modeling step*****
1.	**Let** M be the adjacent matrix for the main social network
2.	**Let** $M_1, M_2, \dots, M_T$ be the adjacent matrix for time *1* through *T*.
3.	**Let** *T* be the total number of time stages, and *I* be the maximum number of iterations for one time stage.

```
Let $i$, $t$ be the iteration counter and time stage counter and initially set to $1$.
Assign $A$ as an LA to each vertex in the main Graph.
Let r be the number of actions for each LA
Let $p_i = (p_{i1}, p_{i2}, \dots, p_{ir})$ Where $p_{ij}$ is the probability that the LA in $i^{th}$ vertex chooses the $j^{th}$ action and initialized to $p_{i1} = p_{i2} = \dots = p_{ir} = \frac{1}{r}$.
Let Active_ LAs be the set of LAs for the next iteration and initially is set to {}.
***Running step***
Begin
While $t < T$ do
    For each vertex in time stage t based on adjacent matrix $M_t$ do
      Calculate the available action of the LA in a vertex
      Calculate the sum of the probability of available actions
     For each action $j \in \{1,2,\dots,r\}$ do
       If ($\alpha_j$ is available action)
         Scale action probability vector $\hat{p}_i(k)$ according to Equation (2)
       End if
     End for
    End for
    While $i < I$ do
       If Active_LA={} then
         Select vertex $v_s$ randomly as the starting vertex.
         Automaton $A_s$ is activated and adds it to Active_LA
       End if
       For each $A_i$ in Active_ LAs do
         Learning automaton $A_i$ select an action based on its action probability vector
         If selected-action of $A_i$ <> prior-action of $A_i$ then
           Active all $A_i$ neighbors and add them to the Active_LA.
           Set configuration be the set of selected actions of $A_i$ and all its neighbors' action.
           Let the selected action by $A_i$ be $\alpha_{ik}$
           If one of the $A_i$'s neighbors like $A_j$ selects action $\alpha_{jk}$ then
             Set $\beta$=1 //reward the $A_i$
           Else
             Set $\beta$=0 // penalty the $A_i$
           End if
           Update the action probability distribution based on generated $\beta$ and the learning algorithm.
           For each action $j \in \{1,\dots,r\}$ of $A_i$ do
             If ($\alpha_j$ is available action)
               Rescale the probability vector of selected available action by Equation (3)
             End if
           End for
         End if
         Remove $A_i$ from Active_LA
       End for
       i=i+1
    End while
***Change step***
   When a change is occurred on the network do
      If link $m$ is appeared then
         Add a new $A_m$ to the ICLA-LP based on the changed occurred on the network
         Add $A_m$ to the Active_LA.
      Else if the $k^{th}$ link is removed then
         Set $A_k$ be the LA corresponds to the remove link
         Active all $A_k$ neighbors and add them to the Active_LA.
         Remove $A_k$ from ICLA-LP structure
      End if
   End when
t=t+1
End while
End Algorithm
```

Figure 8. The pseudo-code of weighting connections in the network by using ICLA

In other words, first, we calculate the weight of connected vertices to a vertex like v_j and, then based on the resulting weight, we calculate the weights of v_j's unconnected neighborhood vertices according to the LA_j

probability distribution and Equation 5. As shown in Fig. 9, vertex v_j has three neighborhoods, v_i, v_k, v_f, and there is no link between vertices v_i and v_k. According to Equation (5), the weight of the unconnected edge E_{ik} can be calculated via the weight of edges E_{ij} and E_{jk} as well as via E_{if} and E_{fk}. Therefore, there are two routes of length two between vertices v_i and v_k via the vertices v_j and v_f.

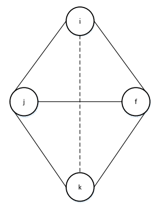


Figure 9. The illustration of using different routes between two unconnected vertices for calculating the score of the unconnected link. Two routes *i j k* and *i f k* are used for calculating the score of unconnected test link E_{ik}.

After calculating the weight of the edge E_{ik}via vertices v_j and v_f by using Equation (5), we add up the resultant weights.

$$w^{'}(E_{ik}) = \sum_{q=1}^{n} W_q(E_{ik}) \qquad (6)$$

If a link existed between vertices v_i and v_k in the absence of vertices v_j and v_f, the weight is given to the unconnected edge E_{ik} can be interpreted according to Equation (6). In the end, the probability of link formation between two unconnected vertices of v_i and v_k in the future will be calculated as the following:

$$prob(E_{ik}) = 1 - \frac{1}{n+w'(E_{ik})} \qquad (7)$$

where *n* denotes the number of length-2 paths between vertices v_i and v_k.

5. Experiments and Results

In the present part, we describe the experiments performed to assess the proposed approach, along with the additionally acquired outcomes. First, we define the social network data utilized and the procedure of experiments (section 5.1). In section 5.2, the evaluation metrics that we use as a part of our analysis have been introduced and then present a set of experiments. In section 5.3., the proposed algorithm's performance is compared with several LP methods.

5.1. Data and Setting

In this article, first, six static datasets called the Internet (INT), electrical power grid of the western US (Grid), protein-protein interaction network (PPI), co-authorships network between scientists (NS), US airport network (USAir), and network of the US political blogs (PB), are used for experiments [61].

The second group of datasets used in this article is datasets related to OSNs, the analysis of three of which is given below:

1. Co-authorship network, where a vertex shows an author and a link indicates that two specific nodes have collaborated to write a paper [62]. More precisely, in co-authorship networks adopted here, each edge stores the co-authored paper's publication year. The co-authorship network data used in our work have been collected from arXiv[2] e-print, which keeps a massive database of electronic scientific papers over several fields. This paper has used three co-authorship networks and has extracted data from the years 1993 to 2003 for the entire datasets. The first one of these networks is composed of authors who collaborated in theoretical high-energy physics (Hep-th).
   In contrast, the second network has been formed by authors who published papers in the field of high-energy physics (Hep-ph). Lastly, the third network is sampled out of collaborating authors in the field of astrophysics (Astro-ph). As co-authorship networks are exceedingly sparse, it is needed to reduce the number of candidate pairs so as to make computation more feasible. To do so, we have chosen only the candidates that have at least two collaborations from the year 1993 up to 2003.
2. The Enron email communication network [63] is an email network in which a node represents an email address, and an edge indicates that address i has sent at least one email to address j. We cover data in 36 months from May 1999 through May 2002. In the Enron email network,
3. College-MSG network, where the users are University of California students. This community aimed to sustain social interaction within students' communities and enlarge their friends' networks. The network dataset covered the time from April through October 2004 [64].

Table 1- Datasets and their statistic information

Dataset	N	M	NUM_C	C	K
USAir	332	2126	332/1	0.74	12.807
PB	1224	19090	1222/2	0.36	31.193
NS	1461	2742	379/268	0.87	3.754
PPI	2617	11855	2375/92	0.38	9.060
Grid	4941	6594	4941/1	0.10	2.669
INT	5022	6258	5022/1	0.03	2.492
Hep-th	9,877	51,971	3729/1	0.3928	8.3328
Hep-ph	12,008	237,010	7044/1	0.5610	23.2533
Astro-ph	18,772	396,160	7816/1	0.5127	59.3352
Email-Enron	87,273	1,148,072	7799/1	0.3303	42.7619
College-MSG	1,899	59,835	1505/1	0.1404	17.6505

Table 1 shows each dataset's topological features' largest connected component on which the experiments have been performed. In this Table, M and N indicate the total number of nodes and links of the entire network, respectively. NUM_C is the number of linked components as well as the number of nodes of the largest connected community. For example, 1222/2 shows that the network has two linked components and that their bigger one has 1222 nodes. Also, C is the clustering coefficient, and K is the network's average degree. Based on experimental results, the values of the reward and punishment parameters were set at 0.01.

5.2. Evaluation Metrics

We use the two evaluation metrics to help us compare our proposed approach with other LP methods as the following [3]:

AUC: Provided that we rank the scores allocated to all non-existence connections, the AUC will be realized as the probability that a randomly selected missing connection has been assigned with a higher score than a randomly chosen non-existent connection. So at each time in the algorithm, we have chosen a missing connection and a non-existent one randomly to compare their scores. If in n independent comparisons, there are n' times missing connections have a higher score and n'' times have the same score, the AUC value is:

[2] Arxiv.org eprint archive

$$AUC = \frac{n' + 0.5n"}{n} \qquad (8)$$

Precision: Provided that we predict that L connections to be connected and L_r connections out of L connections are right; thus, the Precision equals Equation (9).

$$Precision = \frac{L_r}{L} \qquad (9)$$

Table 2. The similarity-based algorithms for experiments. Let the degree of node x is represented by $d(x)$ and Γ(x) denotes neighbors of the node x.

Topological similarity indices	Formula	Description
Common Neighborhood (CN)[3]	$CN(x,y) = \lvert\Gamma(x) \cap \Gamma(y)\rvert$	Two nodes x and y, are more likely to have a link if they have many common neighbors.
Jaccard Index (JC)[38]	$Jaccard(x,y) = \frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$	Measures the probability that a neighbor of x or y is a neighbor of both x and y.
Preferential Attachment (PA)[39]	$PA(x,y) = d(x) \times d(y)$	Gives higher scores to pairs of nodes for which one or both have a high degree.
Adamic-Adar Index (AA)[40]	$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\lvert\Gamma(z)\rvert)}$	This index refines the simple counting of common neighbors by assigning the less-connected neighbors more weight.
Local Path Index[41]	$Local\ Path\ Index(x,y) = A^2 + \varepsilon A^3$	A restricted version of the Katz metric such that only paths of length 1 and 2 are considered.
Katz Index[42]	$Katz(x,y) = \sum_{l=1}^{\infty} \beta^l . \lvert path(x,y)^{<l>}\rvert$	Sum of the number of paths with different lengths, such that shorter paths have more weights.

5.3. LP Comparison

In this section, the introduced ICLA-LP algorithm is compared with classical similarity-based algorithms and a recently presented set of algorithms. In the following, these algorithms are briefly introduced.

1. Similarity-based prediction algorithms: Of this set of algorithms, we have selected local similarity-based algorithms such as PA [39], AA [40], JC [38], CN [3], quasi-local similarity-based algorithms such as local path [41], and global similarity-based algorithm such as Katz [42] to compare with our proposed algorithm. The definition of this set of methods is also shown in Table 2.
2. Supervised LP algorithms: Of this set of algorithms, we have selected a group of methods which have been introduced lately to compare with the ICLA-LP algorithms. These include IP [22], CMA-ES [19], MI-LP [55], ACO-LP [43], LA-TSLP [23], FLP-DLA [25], CALA-LP [24], ICLA-EC-TSLP [31] . It is worth noting that parameters related to the chosen algorithms are similar to the mentioned references.

5.3.1. Experiments for static social networks datasets

In this section, to do the experiments, each dataset's existing vertices are randomly divided into ten subcategories. Of these ten subcategories, one is preserved as valid data for testing the algorithm, and the remaining nine subcategories are used as training data. The Precision and AUC criteria are used to compare the accuracy of results obtained using ICLA-LP with results obtained using other algorithms.
Tables 3 and 4 show AUC and Precision's results from 10 independent experiments that have been averaged to create a single estimation. Table 3 shows the average AUC points derived from ICLA-LP compared to 8 different algorithms on six datasets. In this Table, the highest AUC point for each dataset has been illustrated in boldface. In previous research, it has been shown that generally, the Katz index has the best AUC performance on datasets, but as you can see in Table 3, out of 9 algorithms, the ICLA-LP algorithm gets the highest points over five datasets, and

that the results from 1 other dataset are acceptable regarding the points of other algorithms, including Katz. Algorithm ICLA-LP outdoes algorithm ACO-LP in all datasets because the proposed algorithm uses the strength of links. The proposed algorithm outdoes algorithm MI-LP in all datasets except INT. The reason is that most of the pairs of nodes in INT lack common neighborhoods, and since our approach is based on weighting the edges that participate in triangular communities, it gains lower AUC and Precision compared to MI-LP. But the MI-LP gives better results in this dataset because it performs based on mutual information. As we have compared Tables 1 and 3, we have noticed that the datasets' AUC values are in harmony with their clustering coefficient and average network degree. The algorithm produces better results on datasets with a higher clustering coefficient and average network degree.

Table 3. Comparing algorithms' accuracy based on AUC for static datasets

Dataset/Method	USAir	PB	NS	PPI	Grid	INT
CN [3]	0.9210	0.9650	0.8860	0.9302	0.5845	0.5393
JC [38]	0.9066	0.9293	0.8891	0.9273	0.5872	0.5404
AA [40]	0.9146	0.9658	0.8896	0.9329	0.5863	0.5410
PA [39]	0.8889	0.9572	0.6209	0.8039	0.4206	0.4106
Local Path [41]	0.9280	0.9695	0.8985	0.9356	0.6454	0.6239
Katz [42]	0.9565	0.9703	0.9009	0.9385	0.6464	0.6271
ACO-LP [43]	0.9135	0.9527	0.8736	0.9227	0.5804	0.5402
MI-LP [21]	0.9224	0.9322	0.8746	0.9265	0.6076	**0.8317**
ICLA-LP	**0.9700**	**0.9950**	**0.9450**	**0.9750**	**0.6550**	0.5750

Table 4. Comparing algorithms' accuracy based on Precision for static datasets

Dataset/Method	USAir	PB	NS	PPI	Grid	INT
CN	0.3212	0.1325	0.6212	0.6537	0.1161	0.1021
JC	0.2925	0.1241	0.6125	0.6112	0.1104	0.2112
AA	0.4735	0.2114	0.6305	0.7112	0.2278	0.2325
PA	0.1014	0.0120	0.5847	0.5431	0.0670	0.0271
Local Path	0.3680	0.0630	0.7190	0.2900	0.4500	0.6230
Katz	0.5570	0.1320	0.7341	0.2931	**0.5142**	**0.6271**
ACO-LP	0.6412	0.2785	0.6425	0.6211	0.1435	0.2012
MI-LP	0.7625	0.3743	0.7543	0.7235	0.3749	0.2170
ICLA-LP	**0.7784**	**0.4012**	**0.8012**	**0.8278**	0.2567	0.1103

5.3.2. Experiments for OSNs datasets

5.3.2.1. Experiment one

In this experiment, we consider Hep-ph, Astro-ph, Hep-th, Enron-email, and College-MSG datasets. For three co-authorship datasets, each year has been considered as a time period. For Enron-email each three-month has been considered as one time period. For College-MSG, each month has been considered as a time period. For weighting the social network connections and predicting test links for time $T+1$, graphs of time 1 to T $(G_1, G_2, \ldots, G_T)$ has been used. For this purpose, in co-authorship networks, the years 1993 through 2002 have been used for weighting the real connections, and test connections of time 1996 to 2003 have been predicted. For Enron-email, we have considered January 2000 to December 2001 for weighting the social network, May 2000 through April 2002 for predicting test links. For the College-MSG dataset, April through September 2004 has been considered for weighting social networks, and we have predicted connections for Jun through October 2004. We have shown the result based on the AUC metric in Fig. 10. The result of ICLA-LP has been compared with JC, CN, AA, PA, LA-TSLP, and ICLA-EC. Fig. 10 illustrates that the AUC of ICLA-LP is much better than similarity-based methods and the LA-TSLP method in all datasets; however, ICLA-LP has close results to ICLA-EC. ICLA-EC is better than ICLA-LP due to the fact that each genome in a cell uses not only local information of neighbor cells but also each cell enjoys the experience of other cells' genome as well as previous experience.

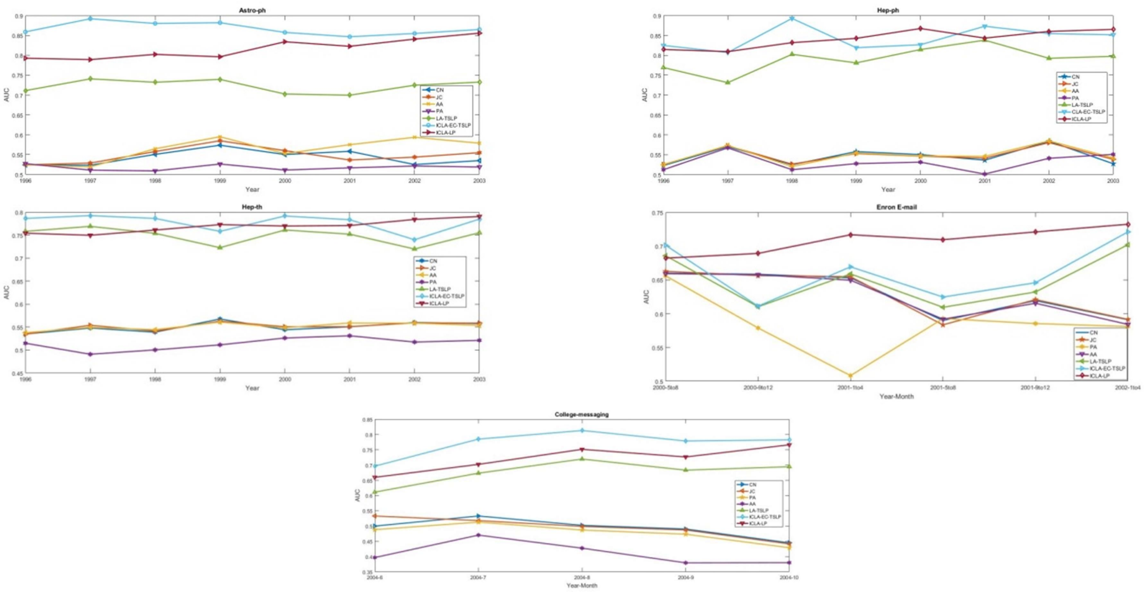


Figure 10. Comparison of *AUC* of the *ICLA-LP* with some Methods on *Astro-ph, Hep-ph, Hep-th, Enron-Email, and College-MSG* Datasets

5.3.2.2. Experiment two

To perform experiments on Astro-ph, Hep-ph, as well as, Hep-th we take data from 1993 to 2002 as our training data (each year serving as a time period) and then the year 2003 as our test data. For the Enron-email network, we consider every three months as a time period. Also, for the College-MSG network, we consider each month as a time period.

The results gained for the proposed algorithm, as well as their results for other algorithms for co-authorship, Email-Enron, College-MSG datasets using AUC and Precision criteria, are reported in Tables 5 and 6. The best outcomes are given in bold. The results which are reported in this section show that the ICLA-LP algorithm achieves the highest Precision and AUC criteria values for various datasets compared to some similarity-based methods such as CN, AA, PA, and JC. The reason why this algorithm is superior is that the ICLA-LP algorithm examines the likelihood of a connection between the two nodes in the future based on the participation of nodes in the formation of local communities at different time intervals. The proposed algorithm also produces better results than do LP and Kats since the ICLA-LP algorithm uses link strength in different time periods to predict links, while LP and Kats use link length. Also, given Tables 5 and 6, it can be seen that ICLA-LP achieves better AUC and Precision than the IP algorithm. Due to the fact that both algorithms use the information of time-series graphs, the better results produced by the ICLA-LP are since the proposed algorithm estimates path strength using communities that exist in time-series networks. In contrast, the IP algorithm uses a prediction model for estimating future connections.

The ICLA-LP algorithm has achieved superior results than MI-LP. The reason for that is tapping link communities while moving along different time-series, whereas MI-LP only uses nodes' mutual information to predict links. The proposed algorithm has achieved better results than ACO-LP in OSNs, because the latter, unlike the former, does not consider neighborhood and community changes over different time spans.

The results lead us to conclude that the ICLA-LP algorithm outdoes CMA-ES because the proposed algorithm uses knowledge drawn at different times. In contrast, CMA-ES only uses a combination of similarity criteria. Since the proposed algorithm is much better than similarity-based methods, it is not surprising that it outdoes CMA-ES. The ICLA-LP is superior to LA-TSLP since we utilize community dynamicity in the dynamic social network. At the same time, LA-TSLP uses local similarity-based methods to predict future connections. The ICLA-LP is better than FLP-DLA because FLP-DLA only employs the strength of test links to predict future relations. CALA is a little better than ICLA-LP because it has considered different similarity-based criteria through different times with different coefficients. Also, ICLA-EC is better than our proposed method because our proposed method uses local information of neighbor vertices for weighting the real connections in the network, so the structural feature of networks affects it. On the other hand, ICLA-EC considers both the local information of each cell and its previous experience related to the resident genome in a cell for predicting connections.

Also, the results have been compared based on topological features in Table 1. As explained before, the experiments have been performed on the largest connected component of the network. The experiments lead us to conclude that better results have been achieved over datasets with greater clustering factors and average degrees.

Table 5. Comparing algorithms' accuracy based on AUC for dynamic datasets

Dataset/Method	Hep-th	Hep-ph	Astro-ph	Email-Enron	College-MSG
CN [3]	0.5577	0.5275	0.5350	0.5911	0.4451
JC [38]	0.5587	0.5385	0.5550	0.5916	0.4414
AA [40]	0.5537	0.5420	0.5788	0.5841	0.4612
PA [39]	0.5212	0.5510	0.5188	0.4283	0.3294
Local Path [41]	0.6370	0.6112	0.6322	0.6554	0.5334
Katz [42]	0.6478	0.6527	0.6602	0.6827	0.5529
ACO-LP [43]	0.5692	0.6253	0.5872	0.6122	0.5197
CMA-ES [19]	0.6581	0.6311	0.6476	0.6702	0.5597
IP [22]	0.6894	0.6478	0.6634	0.6733	0.5665
MI-LP [21]	0.7634	0.8034	0.7985	0.7078	0.6994
LA-TSLP [23]	0.7553	0.7947	0.7325	0.7022	0.6945
FLP-DLA [25]	0.7795	0.8201	0.8297	0.7212	0.7395
CALA [24]	**0.7930**	**0.8527**	**0.8745**	**0.7395**	**0.7795**
ICLA-EC-TSLP [31]	**0.7849**	**0.8515**	**0.8650**	**0.7212**	**0.7825**
ICLA-LP	**0.7905**	**0.8650**	**0.8558**	**0.7322**	**0.7662**

Table 6. Comparing algorithms' accuracy based on Precision dynamic datasets

Dataset/Method	Hep-th	Hep-ph	Astro-ph	Email-Enron	College-MSG
CN	0.3998	0.3622	0.4012	0.4112	0.2015
JC	0.3325	0.3225	0.3516	0.3125	0.1999
AA	0.3548	0.3675	0.3699	0.4011	0.2289
PA	0.3312	0.3128	0.3396	0.3018	0.1732
Local Path	0.4236	0.4399	0.4512	0.4599	0.3418
Katz	0.4682	0.4765	0.4698	0.5036	0.4199
ACO-LP	0.3524	0.3601	0.3655	0.3795	0.2399
CMA-ES	0.4511	0.4639	0.4601	0.4789	0.3316
IP	0.4329	0.4599	0.4511	0.4839	0.3916
MI-LP	0.4678	0.4725	0.4712	0.4899	0.4011
LA-TSLP	0.4401	0.4612	0.4622	0.4725	0.3954
FLP-DLA	0.4974	0.4998	0.4869	0.5122	0.4324
CALA	0.5426	0.5601	0.5219	0.5786	0.4468
ICLA-EC-TSLP	0.5414	0.5597	0.5293	0.5614	0.4533
ICLA-LP	0.4994	0.5027	0.4906	0.5212	0.4315

6. Conclusion

In this paper, we introduced an ICLA-LP model based on network structure. The central part of this model is comprised of a probabilistic matrix formed through the degree of participation of the Graph's edges in building a triangular community. In the beginning, using graph proximity matrix, a weighted graph is created based on the evolution of the ICLAs probability vector over time. Then, the probability of a link between two unconnected nodes is calculated based on the weighted Graph.

The first phase of this algorithm is to estimate the quality of the existing links in the network based on their participation in forming communities over different time periods. Then, in the next phase, the likelihood of a connection between non-existent links is calculated based on existing relationships' quality.

To evaluate the proposed algorithm based on AUC and Precision evaluation criteria, static and OSN datasets are used. The results show that the ICLA-LP algorithm has outdone other algorithms in the case of many of the datasets used in this research. In some datasets, it has achieved results comparable to the best current results.

Since the datasets' structure plays a crucial role in the results obtained, the results given in various studies show that no structure-based LP algorithm exists which would achieve the best results in all of the datasets. The use of ICLAs in this algorithm means that the ICLA-LP algorithm can be implemented entirely in parallel. For future work, also, ICLA can be used to weight graph edges considering more complicated communities.

7. Availability of data

The static datasets described in section 5.1 and support this study's findings are openly available at http://www.linkprediction.org/index.php/link/resource/data [61]. Online datasets Astro-ph, Hep-ph, Hep-th, and Email-Enron, are available at http://konect.uni-koblenz.de/networks [62,63]. Also, the College-MSG dataset is available at https://snap.stanford.edu/data [64].

References

1. Martínez V, Berzal F, Cubero J-C (2017) A survey of link prediction in complex networks. ACM Computing Surveys (CSUR) 49 (4):69 %@ 0360-0300
2. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications 390 (6):1150-1170 %@ 0378-4371
3. Al Hasan M, Zaki MJ (2011) A survey of link prediction in social networks. In: Social network data analytics. Springer, pp 243-275
4. Samad A, Qadir M, Nawaz I, Islam MA, Aleem M (2020) A Comprehensive Survey of Link Prediction Techniques for Social Network. EAI Endorsed Trans Indust Netw & Intellig Syst 7 (23):e3
5. Kaya B (2020) A hotel recommendation system based on customer location: a link prediction approach. Multimedia Tools and Applications 79 (3):1745-1758 %@ 1573-7721
6. Kurt Z, Ozkan K, Bilge A, Gerek ON (2019) A similarity-inclusive link prediction based recommender system approach. Elektronika IR Elektrotechnika 25 (6):62-69 %@ 2029-5731
7. Kovács IA, Luck K, Spirohn K, Wang Y, Pollis C, Schlabach S, Bian W, Kim D-K, Kishore N, Hao T (2019) Network-based prediction of protein interactions. Nature communications 10 (1):1-8 %@ 2041-1723
8. Lim M, Abdullah A, Jhanjhi NZ (2019) Performance optimization of criminal network hidden link prediction model with deep reinforcement learning. Journal of King Saud University-Computer and Information Sciences %@ 1319-1578

9. Bhattacharyya P, Garg A, Wu SF (2011) Analysis of user keyword similarity in online social networks. Social network analysis and mining 1 (3):143-158 %@ 1869-5450
10. Anderson A, Huttenlocher D, Kleinberg J, Leskovec J Effects of user similarity in social media. In, 2012. ACM, pp 703-712 %@ 1450307477
11. Akcora CG, Carminati B, Ferrari E (2013) User similarities on social networks. Social Network Analysis and Mining 3 (3):475-495 %@ 1869-5450
12. Daud NN, Ab Hamid SH, Saadoon M, Sahran F, Anuar NB (2020) Applications of link prediction in social networks: A review. Journal of Network and Computer Applications:102716 %@ 101084-108045
13. Liben‐Nowell D, Kleinberg J (2007) The link‐prediction problem for social networks. Journal of the American society for information science and technology 58 (7):1019-1031 %@ 1532-2882
14. Valverde-Rebaza J, de Andrade Lopes A (2013) Exploiting behaviors of communities of twitter users for link prediction. Social Network Analysis and Mining 3 (4):1063-1074 %@ 1869-5450
15. Liu H, Hu Z, Haddadi H, Tian H (2013) Hidden link prediction based on node centrality and weak ties. EPL (Europhysics Letters) 101 (1):18004 %@ 10295-15075
16. Qiu B, Ivanova K, Yen J, Liu P Behavior evolution and event-driven growth dynamics in social networks. In, 2010. IEEE, pp 217-224 %@ 1424484391
17. Yang S-H, Long B, Smola A, Sadagopan N, Zheng Z, Zha H Like like alike: joint friendship and interest propagation in social networks. In, 2011. ACM, pp 537-546 %@ 1450306322
18. Dong Y, Tang J, Wu S, Tian J, Chawla NV, Rao J, Cao H Link prediction and recommendation across heterogeneous social networks. In, 2012. IEEE, pp 181-190 %@ 1467346497
19. Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. Journal of Computational Science 5 (5):750-764 %@ 1877-7503
20. Huang Z, Lin DKJ (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS Journal on Computing 21 (2):286-303 %@ 1091-9856
21. Tan F, Xia Y, Zhu B (2014) Link prediction in complex networks: a mutual information perspective. PloS one 9 (9):e107056 %@ 101932-106203
22. Rossetti G, Guidotti R, Pennacchioli D, Pedreschi D, Giannotti F Interaction prediction in dynamic networks exploiting community discovery. In, 2015. IEEE, pp 553-558 %@ 1450338542
23. Moradabadi B, Meybodi MR (2017) A novel time series link prediction method: Learning automata approach. Physica A: Statistical Mechanics and its Applications 482:422-432 %@ 0378-4371
24. Moradabadi B, Meybodi MR (2016) Link prediction based on temporal similarity metrics using continuous action set learning automata. Physica A: Statistical Mechanics and its Applications 460:361-373 %@ 0378-4371
25. Moradabadi B, Meybodi MR (2017) Link prediction in fuzzy social networks using distributed learning automata. Applied Intelligence 47 (3):837-849 %@ 0924-0669X
26. Moradabadi B, Meybodi MR (2018) Link prediction in weighted social networks using learning automata. Engineering Applications of Artificial Intelligence 70:16-24 %@ 0952-1976
27. Moradabadi B, Meybodi MR (2018) Link prediction in stochastic social networks: learning automata approach. Journal of computational science 24:313-328 %@ 1877-7503
28. Clauset A, Moore C, Newman MEJ (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453 (7191):98 %@ 1476-4687
29. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences 106 (52):22073-22078 %@ 20027-28424
30. Menon AK, Elkan C Link prediction via matrix factorization. In, 2011. Springer, pp 437-452
31. Manshad MK, Meybodi MR, Salajegheh A (2020) A new irregular cellular learning automata-based evolutionary computation for time series link prediction in social networks. Applied Intelligence:1-14 %@ 1573-7497

32. Beigy H, Meybodi MR (2004) A mathematical framework for cellular learning automata. Advances in Complex Systems 7 (03n04):295-319
33. Wolfram S (1994) Cellular automata and complexity: collected papers, vol 1. Addison-Wesley Reading, MA,
34. Thathachar MA, Sastry PS (2011) Networks of learning automata: Techniques for online stochastic optimization. Springer Science & Business Media,
35. Esnaashari M, Meybodi MR (2015) Irregular cellular learning automata. IEEE transactions on cybernetics 45 (8):1622-1632
36. Zadeh PM, Kobti Z A Knowledge Based Framework for Link Prediction in Social Networks. In, 2016. Springer, pp 255-268
37. Rezvanian A, Meybodi MR (2010) Tracking extrema in dynamic environments using a learning automata-based immune algorithm. In: Grid and Distributed Computing, Control and Automation. Springer, pp 216-225
38. Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat 37:547-579
39. Newman ME (2001) Clustering and preferential attachment in growing networks. Physical review E 64 (2):025102
40. Adamic LA, Adar E (2003) Friends and neighbors on the web. Social networks 25 (3):211-230
41. Zhou T, Lü L, Zhang Y-C (2009) Predicting missing links via local information. The European Physical Journal B 71 (4):623-630 %@ 1434-6028
42. Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18 (1):39-43 %@ 0033-3123
43. Sherkat E, Rahgozar M, Asadpour M (2015) Structural link prediction based on ant colony approach in social networks. Physica A: Statistical Mechanics and its Applications 419:80-94 %@ 0378-4371
44. Dhote Y, Mishra N, Sharma S Survey and analysis of temporal link prediction in online social networks. In, 2013. IEEE, pp 1178-1183 %@ 1467362174
45. Juszczyszyn K, Musial K, Budka M Link prediction based on subgraph evolution in dynamic social networks. In, 2011. IEEE, pp 27-34 %@ 1457719312
46. Lichtenwalter RN, Chawla NV Vertex collocation profiles: subgraph counting for link analysis and prediction. In, 2012. ACM, pp 1019-1028 %@ 1450312292
47. Huang Z (2010) Link prediction based on graph topology: The predictive value of generalized clustering coefficient. Available at SSRN 1634014
48. Zhang Q-M, Lü L, Wang W-Q, Zhou T (2013) Potential theory for directed networks. PloS one 8 (2):e55437 %@ 51932-56203
49. Potgieter A, April KA, Cooke RJ, Osunmakinde IO (2009) Temporality in link prediction: Understanding social complexity. Emergence: Complexity & Organization (E: CO) 11 (1):69-83
50. Huang Z, Lin DK (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS Journal on Computing 21 (2):286-303
51. Wu X, Wu J, Li Y, Zhang Q (2020) Link prediction of time-evolving network based on node ranking. Knowledge-Based Systems:105740 %@ 100950-107051
52. Mallek S, Boukhris I, Elouedi Z, Lefèvre E (2019) Evidential link prediction in social networks based on structural and social information. Journal of computational science 30:98-107 %@ 1877-7503
53. Özcan A, Öğüdücü ŞG Multivariate temporal link prediction in evolving social networks. In: Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on, 2015. IEEE, pp 185-190
54. Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. Journal of Computational Science 5 (5):750-764
55. Tan F, Xia Y, Zhu B (2014) Link prediction in complex networks: a mutual information perspective. PloS one 9 (9):e107056

56. Thathachar MA, Sastry PS (2002) Varieties of learning automata: an overview. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 32 (6):711-722
57. Torkestani JA, Meybodi MR Approximating the minimum connected dominating set in stochastic graphs based on learning automata. In, 2009. IEEE, pp 672-676 %@ 076953595X
58. Akbari Torkestani J, Meybodi MR (2010) Learning automata-based algorithms for finding minimum weakly connected dominating set in stochastic graphs. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 18 (06):721-758 %@ 0218-4885
59. Torkestani JA, Meybodi MR (2012) Finding minimum weight connected dominating set in stochastic Graph based on learning automata. Information Sciences 200:57-77 %@ 0020-0255
60. Thathachar MAL, Harita BR (1987) Learning automata with changing number of actions. IEEE transactions on systems, man, and cybernetics 17 (6):1095-1100 %@ 0018-9472
61. Lü L, Pan L, Zhou T, Zhang Y-C, Stanley HE (2015) Toward link predictability of complex networks. Proceedings of the National Academy of Sciences 112 (8):2325-2330 %@ 0027-8424
62. Barabâsi A-L, Jeong H, Néda Z, Ravasz E, Schubert A, Vicsek T (2002) Evolution of the social network of scientific collaborations. Physica A: Statistical mechanics and its applications 311 (3-4):590-614
63. Shetty J, Adibi J (2004) The Enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California 4 (1):120-128
64. Ahn Y-Y, Han S, Kwak H, Moon S, Jeong H Analysis of topological characteristics of huge online social networking services. In: Proceedings of the 16th international conference on World Wide Web, 2007. ACM, pp 835-844