Computer & Robotics

A Link Prediction Method Based on Learning Automata in Social Networks

Sara YounessZadeh [a], Mohammad Reza Meybodi [b,*]

[a] Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

Received 10 August 2017; revised 24 September 2017; accepted 11 October 2017; available online 15 March 2018

Abstract

Nowadays, online social networks are considered as one of the most important emerging phenomena of human societies. In these networks, prediction of link by relying on the knowledge existing of the interaction between network actors provides an estimation of the probability of creation of a new relationship in future. A wide range of applications can be found for link prediction such as electronic commerce and recommender systems or identification of terroristic relations in social networks. In this article, a new idea is presented for the prediction. It is an integration of the two methods of prediction of similarity score based link and prediction of probabilistic link, which is placed in a new category of link prediction methods. This idea acquires the similarity score between nodes from probabilistic techniques and through using learning automata, and provides better results compared to other criteria methods on standard datasets.

Keywords: *Distributed Learning Automata, Similarity Score, Social Networks Analysis, Link Prediction, Online Social Networks.*

1. Introduction

As part of the recent surge of research on large, complex networks and their properties, a considerable amount of attention has been devoted to the computational analysis of social networks. Generally, a social network refers to a network of interactions or relationships in which nodes include actors, and edges are the interactions or relationships between these actors. It is obvious that the concept of social networks is not limited to online social networks such as Facebook; the matter of social networks are often studied in the field of sociology and in terms of generic interactions among a group of actors. Such interactions can be reflected in a common or uncommon form, e.g. face to face interactions, telecommunication, interaction through e-mail or postal mails [1].

Link prediction is a significant activity in social networks analysis and it can be adopted in other fields such as information retrieval, bioinformatics, and electronic commerce. In recent years. Link prediction in social networks has attracted the attention many scholars and various techniques have been presented. These methods are classified in three classes of similarity based methods, maximum likelihood methods and probabilistic methods [2].

* Corresponding author. Email: mmeybodi@aut.ac.ir

In the most common approach, similarity based methods, an index is allocated to each pair of groups which is defined as the similarity score between two nodes. All unobserved links are organized according to their similarity score and links that connect nodes with higher similarity scores are more probable to exist [3]. Nodes similarity can be defined through using the basic features of vertices: two nodes are considered to be similar if they have many mutual features [4] or a close topological structure [5]. There are many similarity metrics including local similarity metrics: Common Neighbors [6], Salton Index [7], Jaccard Index [8], Sorensen [9], Hub Promoted Index [10], Hub Depressed Index [11], Leicht–Holme–Newman Index (LHN1)[10],Preferential Attachment Index [12], Adamic-Adar Index [13] and Resource Allocation Index [14], global similarity metrics: Katz Index [15], Leicht–Holme–Newman Index (LHN2) [10], Matrix Forest Index (MFI) [16] and Quasi local metrics that do not require global topological information but use more information than local indices: Local Path Index [17,18], Local Random Walk [19], Superposed Random Walk [19], Average Commute Time [20], Cos+ [21], random walk with restart [22], SimRank [23].

In many of the learning machine problems such as link prediction, the correct answers to the question - which supervised learning needs to learn - is not available. For this reason, the use of a learning method called boosting has been considered. Boosting uses a combination of dynamic programming and supervisor learning to achieve a powerful machine learning system.

One of the boosting learning methods is learning automata (LA) that is used in this article as a learning mechanism. Learning automata finds the answer to the problem without any information about the optimal actions. An automata action is selected randomly and applied in the environment. Then the received environment response and the probability of the actions are updated according to the learning algorithm and the procedure is repeated repeatedly.

In [24] for the first time, a new link prediction algorithm based on distributed learning automata (DLA) is presented in which a network of a set of learning automata corresponding to the nodes of the problem's graph is created. Each automata is correspondent for a node and each automata's action is correspondent for an edge. DLA output is an order of selected actions generating a path including a number of nodes. This path is assessed by a fitness function and changes the probability of the automata actions corresponding to the nodes located in the path. Finally, after several iteration of the algorithm, DLA probabilities vector is used as the similarity score.

In this article, the first suggested algorithm, DLA-LP1, increases accuracy and tries to make DLA efficient through limiting the number of the actions of each learning automata and also giving initial value to actions probabilities vector and through reducing the search space.

The second suggested algorithm, DLA-LP2, achieves a considerable increase in accuracy measures through using route feature and integrating it with automata probabilities matrix.

The order of other parts of the article is as below: part 2 describes the link prediction problem. Parts 3 and 4 explain the learning automata and the distributed learning automata, respectively. In parts 5 and 6 the first and second suggested algorithms are expressed, and in part 7, problem formulation and evaluation methods are described. Part 8 attends to the used datasets and in part 9 tests results are indicated in terms of accuracy and running time. Part 10 is allocated to conclusion and finally last part refer to further research.

2. Traditional Definition of Link Prediction

If we consider the problem scape as a graph, then, link prediction is prediction of the probability of future relationships between two graphs, knowing that currently there are no relationships between these two nodes. Based on the contractual definition, link prediction can be formulated as below:

Social network graph G($\boldsymbol{V}, \boldsymbol{E}$) is given, edge $\boldsymbol{e} = (\boldsymbol{u}, \boldsymbol{v}) \boldsymbol{\epsilon} \boldsymbol{E}$ indicates the interaction between node v and node u in a defined time interval. For $\leq \boldsymbol{t}'$, we assume that $\boldsymbol{G}[\boldsymbol{t}, \boldsymbol{t}']$ indicates the subgraph of G including all of G edges in the time interval $[\boldsymbol{t}, \boldsymbol{t}']$. After selection of two time interval $[\boldsymbol{t_0}, \boldsymbol{t_0'}]$ and $[\boldsymbol{t_1}, \boldsymbol{t_1'}]$ which $\boldsymbol{t_0'} < \boldsymbol{t_1}$, through accessing the graph $\boldsymbol{G}[\boldsymbol{t_0}, \boldsymbol{t_0'}]$, the link prediction algorithm

should predict edges in output that do not exist in the graph $\boldsymbol{G[t_0, t'_0]}$, but will be formed in the graph $\boldsymbol{G[t_1, t'_1]}$. The interval $\boldsymbol{[t_0, t'_0]}$ is called the train interval and interval $\boldsymbol{[t_1, t'_1]}$ is called the test interval [1].

3. Learning Automata

A learning automata can be considered as a single object with a finite number of actions. The learning automata acts through selecting an action among its own action sets and enforcing it on the environment. The mentioned action is assessed by a random environment, and automata uses the environment's response for selecting its next step. During this process, automata learns to select the optimum action. The method of using the environment's response to the selected action used for choosing the next action is determined by the learning algorithm. Random learning automata can be indicated with the four components of $= \{\boldsymbol{\alpha, \beta, p, T}\}$, where r is the number of actions, $\boldsymbol{\alpha} = \{\boldsymbol{\alpha_1, \alpha_2, .., \alpha_r}\}$ is the actions set, $\boldsymbol{\beta} = \{\boldsymbol{\beta_1, \beta_2, .., \beta_r}\}$ is the input set, $\mathbf{p} = \{\mathbf{p_1, p_2, .., p_r}\}$ is the probability vector of actions, and $\mathbf{T} = \mathbf{p(n+1)} = \mathbf{T[\alpha(n), \beta(n), p(n)]}$ is the learning algorithm of automata [25].

4. Distributed Learning Automata

Network distributed learning automata is consisted of a number of learning automata cooperating for solving a particular problem. The number of an automata's actions in DLA is equal to the number of learning automata connected to this leaning automata. A DLA can be modeled for a directed graph, as the sets of its nodes can be considered as the set of learning automata and output edges can be considered as the set of learning automata actions corresponding to that node. When an automata selects one of its actions, another automata located at the other end of the edge corresponding to that action becomes activated. Each time, only one learning automata is active in a network. Based on the explanations above, distributed learning automata is defined by graph $\mathbf{DLA} = \mathbf{(V, E)}$ where $\mathbf{V} = \{\mathbf{LA_1, LA_2}, \ldots, \mathbf{LA_n}\}$ is the learning automata set, and n is the number of learning automata and $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$ is the graph edges' set. The edge $\mathbf{(i, j)}$shows the jth action of $\mathbf{LA_i}$ learning automata, and $\mathbf{LA_j}$ becomes activated when the action j of the $\mathbf{LA_i}$ learning automata is selected. The number of $\mathbf{LA_k}$, $\mathbf{k = 1, 2, \ldots, n}$ is equal to the output degree of the node corresponding to $\mathbf{LA_k}$. For deeper investigation of distributed learning automata, [26] can be referred to.

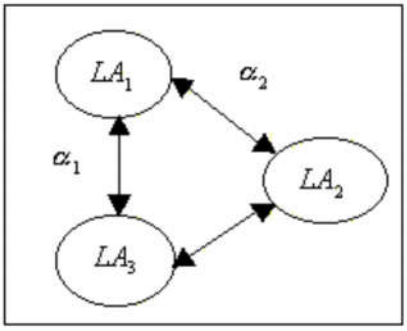


Fig. 1. distributed learning automata (DLA) consisted of 3 learning automata [26]

5. Framework of the DLA_LP1

A network of distributed learning automata (DLA) corresponding with the nodes of logical graph is formed. In this DLA, each learning automata (LA) is corresponding for a node and each LA action is correspondent for an edge. As the graph is complete, the maximum number of each automata actions in each automata in a graph with n nodes is equal to n-1. DLA output is an order of actions selected by automata generating an S route with the length of k links and k+1 nodes. Existence of repeated nodes in the route is permitted. Considering the suitability amount (route fitness function) this route awards the selected automata located in it. Finally, after several iterations, the DLA probabilities vector is used as the similarity score of edges.

5.1. Automata Valuing

For a non-directed and simple network of $G = (V, E)$, if $n = |V|$ and $V = \{v_1, v_2, \ldots, v_n\}$ and $A = [a_{ij}]$ be the adjacency matrix n*n of graph G, the suggested algorithm uses $S = s_0, s_1, \ldots, s_k$ vector for showing a route consisted of selected nodes in the logical graph G′, as $s_i \in V$ is the i^{th} node of the route. First, all S elements are valued with ø which shows a null node. Then s_i is replaced with the value of a real node which is selected by the i^{th} active automata. In case each automata uses equal initial probabilities for selecting their actions, i.e. $P_{ij} = \frac{1}{n-1}$, therefore, because n

(number of nodes) is large, the acquired results in social networks are nearly unacceptable and the algorithm's rate of convergence is very low. For removing this problem, each learning automata can use information such as the number of mutual adjacent nodes of the initial graph for limiting their actions.

5.2. Limiting the Number of Automata's Actions

If we define h (optional) as the maximum number of actions of each of the n automata, first the number of each mutual adjacent of each v_i node with n-1 other nodes (number of routes with length of 2) is found based on the information of the main G graph, as $|CN_{ij}|$ is the number of the mutual adjacent of v_i and v_j. Then, among all $|CN_{ij}|s$ of v_i, we put the number of h nodes with highest amount of $|CN_{ij}|$ in the H_i set.

5.3. Initial Valuing of Automata Action Vector

For increasing accuracy, automata actions probability vector is initially valued as below:

If h (optional) is the maximum number of actions of each n automata, first, the number of mutual adjacent of each $\mathbf{v_i}$ with n-1 other nodes (number of routes with length of 2) based on the main G graph, as $|\boldsymbol{CN_{ij}}|$ is the number of the mutual adjacent of $\mathbf{v_i}$ and $\mathbf{v_j}$. Then, among all $|\boldsymbol{CN_{ij}}|$ s of vj, we put the number of h nodes with highest amount of $|\boldsymbol{CN_{ij}}|$ and $\mathbf{Adj_{i,j}}$= 0 – i.e. no direct link between vi and $\mathbf{v_j}$ – in the Hi set. In the next step, we consider $\mathbf{LA_i}$ corresponding to each $\mathbf{v_i}$ nodes and with n-1 actions. If $\boldsymbol{P(v_i, v_j)}$is the selection probability of jth action of $\mathbf{LA_i}$ automata leading to node $\mathbf{v_j} \in \mathbf{H_i}$, then in case:

$$P(\mathbf{v_i}, \mathbf{v_j}) = \left|\boldsymbol{CN_{ij}}\right| / \sum_{j=1}^{h} \left|\boldsymbol{CN_{ij}}\right| \quad , \quad \mathbf{v_j} \in \mathbf{H_i} \tag{1}$$

Otherwise, we have

$$P(v_i, v_j) = 0 \quad , \quad v_j \notin H_i \tag{2}$$

Note: if for node v_i , $|\ H_i\ | = b$ and $0< b < h$, then the number of active actions of the mentioned node is equal to b and the probability of each action will be as follows:

$$P(v_i,v_j) = \left|CN_{ij}\right| / \sum_{j=1}^{b} \left|CN_{ij}\right| \quad , \quad v_j \in H_i \tag{3}$$

5.4. Fitness Function

A fitness function is defined for measuring the quality of each route and for updating the DLA probabilities. A route with more existing links and closely related nodes would have a higher score; because each pair of adjacent nodes would be connected in by a potential link in such a route. Thus, the fitness function of a route can conduct the fitness action from two aspects of nodes' significance and edges significance. Generally, nodes' significance is measured with nodes centrality. Different centralities show different function of nodes in the network, including spreading ability and nodes influence. Degree based centrality is the most criteria among these measures. In general, higher centrality degree of a node shows the higher importance of that node and the higher probability of being linked with other nodes. Thus, a fitness definition for a route can be obtained from the sum of the nodes degrees in that route.

As we have said earlier, after each iteration, the order of automata activation forms the route $S = (s_1, s_2, \dots, s_k)$ with n nodes. For measuring the route fitness score, route nodes degrees are extracted from the problem initial graph (not from the complete graph) and sum them together according to (4).

$$\mathbf{Q(S)} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{degree(s_i)} \tag{4}$$

5.5. Probabilities Update

After each iteration and finding of a route, the probability algorithm updates the activated automata actions based on equations (5) and (6). As first the fitness score of the route is measured and is compared with the fitness score of the best route ever formed by the algorithm. Based on the comparison result, DLA action probability vector is updated, as if the fitness score of the generated route is higher than or equal to the fitness score of the best route ever generated, all activated learning automata award their selected actions based on the learning algorithm. For instance, in case in iteration *(t)*, the generated fitness score is higher or equal to the fitness score of the best route and

DLA learning automata has selected action i among its permitted actions set, the action i selection probability in iteration *(t+1)* would increase based on equation (5), [25].

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (5)$$

And the probability of selecting other learning automata actions in iteration *(t+1)* would be reduced based on equation (6), [25].

$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j, j \neq i \quad (6)$$

5.6. Closure and Output Conditions

Iteration algorithms are stopped when reached to the N_c threshold. The algorithm output of Las actions probabilities matrix is shown with P, as the element P (v_i, v_j) is the probability of selection of node v_j by LA_i, and LA_i is the automata located in node v_i. Since P is asymmetrical, for generation of a symmetrical score matrix in non-directed graphs, we add each element of matrix P to its transposed element, in other words:

$$\text{Score1}(v_i, v_j) = P(v_i, v_j) + P(v_j, v_i) \quad (7)$$

This output is interpreted as the similarity score by the accuracy assessment criteria.

6. Algorithm Improvement Using a Hybrid Method

For increasing the accuracy of the DLA-LP1 algorithm, the number of routes with lengths of 3 between the nodes of the main G graph is used in generating a new score matrix as following:

i. Enforcement of DLA-LP1 algorithm on datasets and generating the score1 output score matrix.

ii. Sum of all Score1 matrix elements with the amount of ε which is an optional positive and small number.

iii. Generation of the number of all routes with lengths of 3 between the nodes of the main G graph and storage in Score2 matrix.

iv. Element to element multiplication of the two Score1 and Score2 matrices as the Score matrix of DLA-LP2 algorithm output according to equation (8).

$$\text{Score} = (\text{Score1} + \varepsilon).*(\beta\,\text{Score2}) \quad (8)$$

$$0 < \varepsilon \ll 1, \quad 0 < \beta \leq 1$$

This output will be interpreted as the similarity score by the accuracy assessment criteria.

7. Problem Formulation and Evaluation Methods

In this article, multiple links and self-connections does not exist, and networks are defined in terms of non-directed and simple graphs of $G = (V, E)$; as V is the set of nodes, E is the set of links and $n = |V|$ is the number of G nodes and U is the universal set including all possible G links.

Link prediction action is finding the missing or non-existing links which will be formed in future. This method aims to allocate $score(x, y)$ to each pair of $U \in (x, y)$ nodes. This score reflects the similarity level between two nodes. For (x, y) pair node in $U - E$, the higher $score(x, y)$ indicates higher probability of existence of link between x and y nodes. For testing the accuracy of algorithm results, the E observed links were randomly divided into two groups:

E^T: instructional set used as the recognized information.

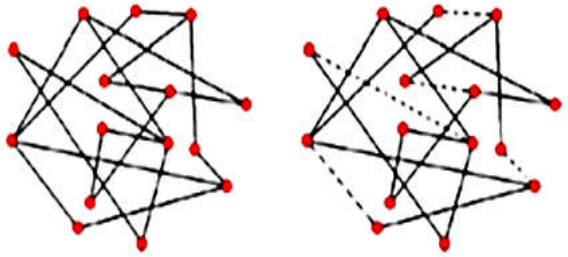
E^P: test set which its information is used for results accuracy testing and not for link prediction.

The union of the two E^T and E^P sets is equal to E and their intersection is equal to $\emptyset$ [8].

For instance, figure 2a shows a network with 15 nodes and 21 links between them. Our purpose is to find the potential links between 84 non-connected pair of nodes. For testing the algorithm accuracy, it is necessary that some of the existing links are selected as the test set and the rest are selected as the instruction set. As a sample, five links were selected as the test set links showed in figure 2b with dotted lines. Then, the algorithm acts only using the information available in the instruction set or the same instruction graph showed with constant lines in figure 2b. Finally, the algorithm gives each one of the 89 pair nodes which include 84 non-existing links member of $U - E$ and 5 test link member of E^P.

Two standard metrics are used to quantify the accuracy of prediction algorithms: area under the receiver operating

characteristic curve (AUC*) and Precision. In principle, a link prediction algorithm provides an ordered list of all non-observed links ($U - E^T$) or equivalently gives each non-observed link, say (x, y) $\in U - E^T$, a score sxy to quantify its existence likelihood. The AUC evaluates the algorithm's performance according to the whole list while the Precision only focuses on the L links with the top ranks or the highest scores. A detailed introduction of these two metrics is as follows [2].



a)The main graph b) instruction graph

Fig. 2. graph chart of a network [3]

7.1. AUC Criteria

Provided the rank of all non-observed links, the AUC value can be interpreted as the probability that a randomly chosen missing link (i.e., a link in E^P) is given a higher score than a randomly chosen nonexistent link (i.e., a link in U −E). In the algorithmic implementation, we usually calculate the score of each non-observed link instead of giving the ordered list since the latter task is more time-consuming. Then, at each time we randomly pick a missing link and a nonexistent link to compare their scores, if among n independent comparisons, there are n ′ times the missing link having a higher score and n ″ times they have the same score, the AUC value is

$$\text{AUC} = \frac{n' + 0.5n''}{n} \qquad (9)$$

If all the scores are generated from an independent and identical distribution, the AUC value should be about 0.5. Therefore, the degree to which the value exceeds 0.5 indicates how much better the algorithm performs than pure chance [2].

7.2. Precision Criteria

Considering the ranking of all of the unseen links, precision is defined as the ratio of the suitable selected items to the total selected items. In other words, if from all of the unseen links, L is considered as the highest link, and m is considered as the accurately predicted link among them, precision is defined as below [2]:

$$\text{precision} = \frac{m}{L} \qquad (10)$$

8. The Used Datasets

In this study, eight popular datasets INT, Grid, PPI, NS, PB, USAir, Jazz[27] and FacBk[28] are used that each representing a particular field. Table 1 indicates the topological features of the largest components connected to each of these datasets on which experiment were conducted. In this table, N and M are the number of nodes and total links of the network, respectively. NUM_c is the number the connected components and also the size (number of nodes) of the largest connected component. For instance, 1222/2 indicates that the network has 2 connected components and its largest includes 1222 nodes. Moreover, C is the clustering ratio and K is the average degree of the network.

Table. 1. Topological features of the largest connected component of datasets

NET	N	M	NUMc	C	K
USAir	332	2126	332/1	0.74	12.8
PB	1224	19090	1222/2	0.36	31.19
NS	1461	2742	379/268	0.87	3.75
PPI	2617	11855	2375/92	0.38	9.06
grid	4941	6594	4941/1	0.10	2.66
INT	5022	6258	5022/1	0.03	2.49
FacBK	4039	88233	4039/1	0.60	43.69
JAZZ	198	5484	198/1	0.61	27.69

* Area Under Curve

9. Test Results

In this part, the effect of the DLA-LP1 and DLA-LP2 suggested algorithms on the eight mentioned popular datasets and also its performance is indicated compared to basic similarity based prediction algorithms such as CN, Salton, jaccard, Sorensen, HPI, HDI, LHN, PA, LP and Katz. All tests are conducted in Microsoft Windows 7 operating system and through Matlab14.0 software.

In each test, the edges existing in each graph are divided randomly into 10 subsets. From these 10 subsets, one subset is preserved as credible data for algorithm test and the other 9 subsets are used as instructional data. Then for investigating the results accuracy, the similarity matrix which includes the automata probabilities matrix is investigated after 1000 iterations through AUC assessment criteria and Precision. The table's values are obtained from measuring the average of AUC and Precision values of 10 independent experiments which was for generating a single estimation.

9.1. Parameters Estimation

Based on empirical results, we set parameters $a = 0.1$, $h = 6$, $\varepsilon = 0.001$, $\beta = 0.1$ in our experiments.

For instance, in figure 3, ROC chart of DLA-LP2 method on Jazz dataset is indicated with different a parameters (learning rate). As it is clear from the figure3, the different a values does not create an obvious change in AUC accuracy.

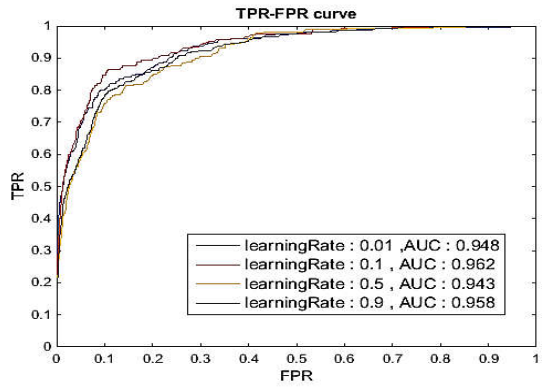


Fig. 3. ROC chart of DLA-LP2 method on Jazz dataset for four different a parameter values

9.2. Investigation of the Accuracy of the Implementation Results of the DLA-LP1 and DLA-LP2 Suggested Algorithms

For investigating the accuracy of the results, similarity matrix which is the automata probabilities matrix is investigated after 1000 iterations through the AUC assessment criteria. The table 2 and figure 3 shows the values obtained from measuring the average of AUC values of 10 independent experiments with purpose of generating a single estimation.

In tables 2 and 3, the results of AUC and Precision comparison of DLA-LP1 and DLA-LP2 methods with ten other popular techniques on eight standard datasets are briefly presented, respectively.

9.2.1.AUC Accuracy Check and Implementation Results

Considering table 2 and with comparison to AUC of DLA-LP1 and DLA-LP2 method, we see that except for the NS dataset, the second method shows better performance on other datasets, although NS values are very close.

For having a better comparison of the table 2 results, AUC values resulted from comparing the

DLA-LP2 method with AUC maximum and minimum of executing 10 base algorithms and also DLA-LP1 algorithm on eight standard datasets are indicated in figure 4. Based on this chart, in the three cases of PB, Jazz, and INT, the AUC value of the second suggested method is higher than the maximum values of other methods. This amount is considerable for INT and has increased the AUC value from 0.64 to 0.93. in USAir, Ns, PPI, and FacBk, AUC accuracy of the DLA-LP2 method is placed in the second rank.

Table. 2. Comparison results of AUC of the DLA-LP1 and DLA-LP2 methods with 10 base algorithms on 8 standard datasets

AUCs		ALGHORITHMS											
		CN	**Salton**	**Jaccard**	**Sorens**	**HPI**	**HDI**	**LHN**	**PA**	**LP**	**Katz.01**	**DLA-LP1**	**DLA-LP2**
DATASETS	**USAir**	0.9562	0.9284	0.9157	0.915	0.8832	0.907	0.779	0.916	0.955	0.9537	**0.845**	**0.954**
	PB	0.923	0.881	0.8774	0.8775	0.858	0.871	0.764	0.9109	0.936	0.9343	**0.615**	**0.94**
	NS	0.9919	0.9921	0.9919	0.9922	0.9921	0.991	0.991	0.7356	0.997	0.9991	**0.956**	**0.941**
	PPI	0.9183	0.9155	0.9154	0.9164	0.9147	0.915	0.911	0.8694	0.971	0.9727	**0.755**	**0.966**
	Grid	0.6212	0.6218	0.6218	0.623	0.6225	0.623	0.622	0.5782	0.695	0.9618	**0.62**	**0.634**
	INT	0.0649	0.0023	0.00298	0.6457	0.6458	0.646	0.646	0.6456	0.646	0.6462	**0.56**	**0.937**
	Jazz	0.9566	0.9664	0.962	0.9611	0.9489	0.952	0.905	0.7719	0.951	0.9432	**0.754**	**0.992**
	FacBk	0.993	0.9924	0.9909	0.9912	0.9872	0.989	0.958	0.8325	0.993	0.6123	**0.886**	**0.947**

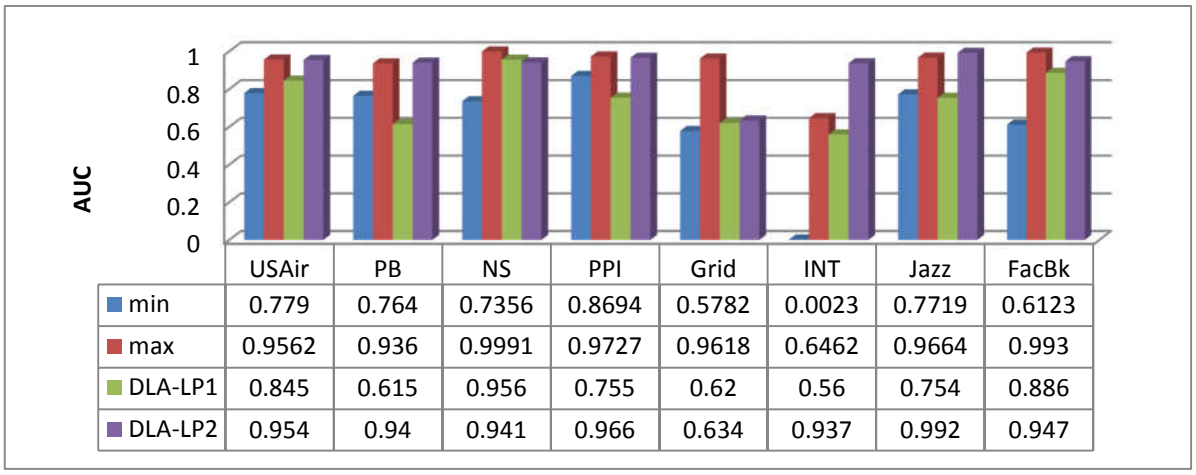


Fig. 4. bar chart of comparing AUC of the DLA-LP1 and DLA-LP2 methods with maximum and minimum amount of AUC resulted from conducting 10 base algorithms on 8 standard datasets

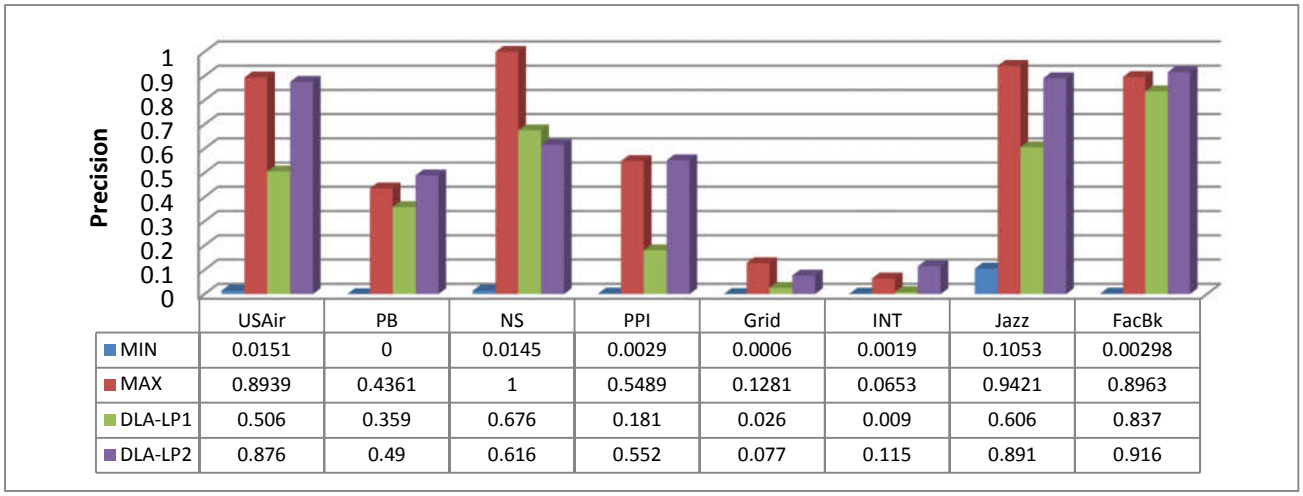
9.2.2.Investigation of the Precision Accuracy of the Implementation Results

Considering table 3, in comparison of the precision of the DLA-LP1 and DLA-LP2 methods, we realize that DLa-LP1 has only had a stronger performance on the NS dataset.

Table. 3. Comparison results of precision of the DLA-LP1 and DLA-LP2 methods with 10 base algorithms on 8 standard datasets

Precisions		ALGHORITHMS											
		CN	**Salton**	**Jaccard**	**Sorens**	**HPI**	**HDI**	**LHN**	**PA**	**LP**	**Katz.01**	**DLA-LP1**	**DLA-LP2**
DATASETS	**USAir**	0.894	0.015	0.033	0.033	0.818	0.033	0.015	0.752	0.882	0.894	0.506	0.876
	PB	0.417	0.000	0.000	0.000	0.173	0.000	0.000	0.099	0.411	0.436	0.359	0.490
	NS	0.861	0.679	0.680	0.680	1.000	0.700	0.337	0.015	0.599	0.598	0.676	0.616
	PPI	0.360	0.010	0.016	0.016	0.344	0.016	0.003	0.232	0.357	0.549	0.181	0.552
	Grid	0.075	0.010	0.011	0.011	0.128	0.011	0.010	0.001	0.070	0.070	0.026	0.077
	INT	0.065	0.002	0.002	0.002	0.023	0.002	0.002	0.022	0.064	0.064	0.009	0.115
	Jazz	0.911	0.937	0.937	0.937	0.379	0.942	0.105	0.321	0.900	0.868	0.606	0.891
	FacBk	0.896	0.018	0.023	0.023	0.480	0.024	0.003	0.036	0.889	0.554	0.837	0.916

For having a better comparison of the results of table 3, the precision values obtained from comparing the DLA-LP2 method with maximum and minimum precision amount with conducting 10 base algorithm and also the DLA-LP1 algorithm on eight standard datasets are shown in figure 5. As it is clear in this figure, the precision values of the new method in four cases of PB, PPI, INT, and FacBk have the maximum amount and except for the NS dataset, have little difference with the maximum value.



	USAir	PB	NS	PPI	Grid	INT	Jazz	FacBk
MIN	0.0151	0	0.0145	0.0029	0.0006	0.0019	0.1053	0.00298
MAX	0.8939	0.4361	1	0.5489	0.1281	0.0653	0.9421	0.8963
DLA-LP1	0.506	0.359	0.676	0.181	0.026	0.009	0.606	0.837
DLA-LP2	0.876	0.49	0.616	0.552	0.077	0.115	0.891	0.916

Fig. 5. bar chart of comparing Precision of the DLA-LP1 and DLA-LP2 methods with maximum and minimum amount of Precision resulted from conducting 10 base algorithms on 8 standard datasets

9.3. Investigation of the DLA-LP2 Algorithm Running Time

For having a better investigation of the suggested method, we compared the algorithm running times of different algorithms and the DLA-LP2 algorithm on the studied standard datasets. The running time of different methods based on second is briefed in table 4. For having a better comparison, the chart in figure 6 is drawn according to the values of table 4.

Table. 4. Comparing the running time of different algorithms on datasets based on second

	USAir	PB	NS	PPI	Power	Router	FaceBk	Jazz
CN	0.00891	0.074163	0.003757	0.023419	0.004614	0.010145	0.191396	0.008136
Salton	0.008499	0.176781	0.414851	0.760567	2.253574	3.946123	1.527609	0.002272
Jaccard	0.023426	0.278759	0.2642	0.816342	0.748399	4.533376	1.158189	0.01025
Sorensen	0.057538	0.640881	0.752351	1.893128	5.958586	10.8433	3.414912	0.030178
HPI	0.063945	0.686866	0.773222	1.751337	6.434335	11.22152	6.727208	0.032443
HDI	0.065079	0.687104	0.773998	1.622303	6.965168	11.52238	5.431655	0.032929
LHN	0.040269	0.351165	0.337494	0.604276	1.677344	3.987214	1.295553	0.024883
PA	0.032639	0.258785	0.241365	0.540526	2.063694	3.526532	2.985415	0.018578
LP	0.058913	0.665976	0.504242	1.163504	4.333509	8.619344	5.516589	0.037948
Katz	0.07268	1.244949	0.335564	2.451238	2.51668	6.797508	86.72485	0.038553
DLA-LP2	0.56735	1.375909	0.535508	2.183487	5.130428	5.217084	5.351587	0.784262

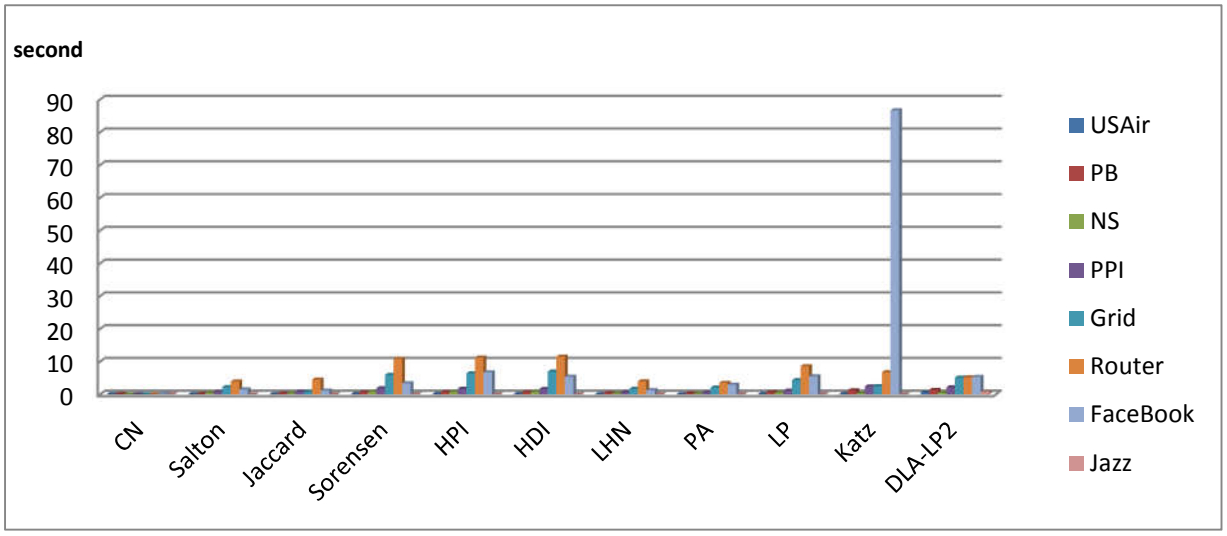


Fig. 6. bar chard of comparison of running time of different algorithms on datasets based on second

With conduction of the Friedman test, which results are briefly expressed in table 5, the DLA-LP2 method is placed in the 10th position in terms of running time within 11 algorithms that have little difference from average. Also, considering the 7th rank of the LP method and the eight rank of Katz method which are still among the best presented algorithms, this rank can be acceptable.

Table. 5. Friedman mean rank values of running time of DLA-LP2 algorithm and other algorithms

Running time of methods	Friedman mean rank	score
CN	1.25	1
Salton	3.25	2
PA	3.25	3
Jaccard	3.38	4
LHN	4.13	5
Sorens	7.50	6
LocalP	7.50	7
Katz.01	8.63	8
HPI	8.88	9
DLA_LP2	9.00	10
HDI	9.25	11

9.4. Friedman Test Results for Accuracy Ranking of the DLA-LP2 Method Alongside the Current Methods

As you can see in table 6, it is observed the DLA-LP2 suggested method has gained the second position in terms of AUC accuracy criteria and also in terms of Precision accuracy criteria as well, which this ranking shows the high progression of the accuracy of the second suggested algorithm. Moreover, compared to the first suggested method, 16.50 percent of the improvement were obtained in AUC accuracy and 16.662 percent improvement were obtained in Precision accuracy.

10. Conclusion

In this study, a new idea of link prediction is presented which is an integration of the two similarity score based link prediction and probabilistic link prediction and is placed in a new category of prediction methods. The aforesaid method acquires the similarity score between nodes through a probabilistic method and through using the learning automata.

Table. 6. Friedman mean rank values of AUC and Precision scores obtained from running the DLA-LP2 algorithm and other algorithms

AUC of methods	Friedman mean rank	score
LocalP	9.25	1
DLA_LP2	8.13	2
Katz.01	7.63	3
CN	6.81	4
Sorens	6.63	5
Salton	6.25	6
HDI	5.38	7
Jaccard	5.13	8
HPI	4.44	9
LHN	3.50	10
PA	2.88	11

Precisions of methods	Friedman mean rank	Score
CN	9.31	1
DLA_LP2	9.00	2
LocalP	7.63	3
Katz.01	7.56	4
HPI	7.50	5
HDI	5.63	6
Jaccard	5.00	7
Sorens	5.00	8
PA	4.25	9
Salton	3.44	10
LHN	1.69	11

The work basis is that a network of distributed learning automata is formed corresponding to the graph nodes. In this DLA, each learning automata is correspondence of a node and each action of a LA represents an edge. As the graph is complete, maximum number of the actions of both automata in a graph with n nodes equals to n-1. The DLA output is an order of actions selected by automata which generates the S route with k length and with k+1 nodes. After each iteration and finding a route, the algorithm updates the actions of

the activated automata, in this way that the fitness score of the route is calculated and is compared with the fitness score of the best route ever generated by the algorithm. Based on the comparison result, the probability vector of DLA action will be updated. If the generated fitness score of the route is larger or equal to the score of the best route ever generated by algorithm, all learning automata are activated and award their selected action based on the learning algorithm. Finally, after several iterations, the DLA probabilities vector will be used as edges similarity score.

Based on this idea, two different link prediction algorithms are presented as below:

The first algorithm increases the accuracy value through limiting the number of each learning automata's actions and initial valuing of probabilities vector of the learning automata through reducing the search space.

The second algorithm considerably increases the accuracy criteria by using the route feature and integrating it with automata probabilities matrix.

After enforcing the first algorithm (DLA-LP1) on the eight standard datasets and comparing the results with 10 popular link prediction methods, through conduction of the Friedman statistical test on experiments' results, this algorithm was placed in the 11^{th} ranking in terms of AUC accuracy and in the 5^{th} ranking in terms of precision accuracy.

In the second suggested algorithm (DLA-LP2) which uses the hybrid method, very suitable results were obtained and both criteria of AUC and Precision accuracy of the algorithm gained the second position among other 10 algorithms in the Friedman statistical test. Moreover, a 16.50% improvement of AUC accuracy and a 16.662% improvement of Precision accuracy were obtained compared to the first suggested algorithm. Considering the significance of the Precision criteria, it can be said that DLA-LA2 is one of the best link prediction algorithms ever presented which has provided acceptable results on different types of datasets.

This performance is caused by utilizing mutual adjacent information and topologic graph and also adaption of random distributed learning automata which can revise and correct its performance according to feedback received from its surrounding environment and guide the problem to its suitable solution.

A wide range of applications can be found for our Project such as electronic commerce and recommender systems or identification of terroristic relations in online social networks.

11. Recommendations for Further Research

In this part the actions that future researchers can take as a result of our Project is presented. This report recommends further work to:

- Apply other types of linear and nonlinear learning algorithms to DLA.

- Use criteria such as Betweenness and Closeness to limit the number of automated actions.

- Use other criteria of centrality such as Betweenness and Closeness to calculate path fit.

- Applying the algorithms presented on the directed networks with regard to the orientation of the vector of automated actions.

- Apply the algorithms presented on the weighted networks.

- Applies the algorithms provided on networks with attribute nodes.

- Provide solutions for increasing the speed of proposed algorithms.

References

[1] Liben-Nowell, D.; Kleinberg, J., "The link-prediction problem for social networks", Journal of the American Society for Information Science and Technology, vol. 58, no. 7, pp. 1019-1031, (2007).

[2] Lu, L.; Zhou, T., "Link prediction in complex networks: A survey", Physica A: Statistical Mechanics and its Applications, vol. 390, no. 6, pp. 1150-1170, (2011).

[3] Chen, B.; Chen, L., "A link prediction algorithm based on ant colony optimization", Applied Intelligence, vol. 41, no. 3, pp. 694-708, (2014).

[4] Lin, D., "An information-theoretic definition of similarity", ICML, vol. 98, pp. 296-304, (1998).

[5] Sun, D.; Zhou, T.; Liu, J.; Liu, R.; Jia, C.; Wang, B., "Information filtering based on transferring similarity", Physical Review E, vol. 80, no. 1, (2009).

[6] Newman, M., "Clustering and preferential attachment in growing networks", Physical Review E, vol. 64, no. 2, (2001).

[7] Harter, S., "Introduction to modem information retrieval (Gerard

Salton and Michael J. McGill)", Education for Information, vol. 2, no. 3, pp. 237-238, (1984).

[8] Jaccard, P., "Etude comparative de la distribution florale dans une portion des Alpes et des Jura". Bulletin de la Soci´et´e Vaudoise des Science Naturelles 37: pp. 547–579Dd, (1901).

[9] Sorensen, T., "A method of establishing groups of qualamplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons", Biol Skr 5(4), pp. 1–34Dd, (1948).

[10] Leicht, E.A.; Holme, P.; Newman, M.E.J., "Vertex similarity in networks", Phys Rev E 026120:73, (2006).

[11] Ravasz, E.; Somera, A.L.; Mongru, D.A., "Hierarchical organization of modularity in metabolic networks", Science 297(5586): pp.1553–1555, (2002).

[12] Barabasi, A-L.; Albert, R., "Emergence of scaling in random networks", Science 286(5439): pp. 509–512, (1999).

[13] Adamic, L.; Adar, E., "Friends and neighbors on the Web", Social Networks, vol. 25, no. 3, pp. 211-230, (2003).

[14] Zhou, T.; Lü, L.; Zhang, Y., "Predicting missing links via local information", The European Physical Journal B, vol. 71, no. 4, pp. 623-630, (2009).

[15] L. Katz, "A new status index derived from sociometric analysis", Psychometrika, vol. 18, no. 1, pp. 39-43, (1953).

[16] Chebotarev, P.; Shamis, E., "The matrix-forest theorem and measuring relations in small social groups", ArXiv Preprint math/0602070, (2006).

[17] L¨U, L.; Jin, C-H.; Zhou, T., "Similarity index based on local paths for link prediction of complex networks", Phys Rev E 80:046122, (2009).

[18] Zhou, T.; Zhang, Y.C., "Predicting missing links via local information", Eur Phys J B 71(4), pp. 623–630, (2009).

[19] Liu, W.; Lü, L., "Link prediction based on local random walk", EPL (Europhysics Letters), vol. 89, no. 5, p. 58007, (2010).

[20] Klein, D.J.; Randić, M., "Resistance distance", J. Math. Chem, vol.12 no. 1, pp. 81–95, (1993).

[21] Fouss, F.; Pirotte, A.; Renders, J.; Saerens, M., "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation", IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 355-369, (2007).

[22] Brin, S.; Page, L., "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107-117, (1998).

[23] Jeh, G.; Widom, J., "SimRank: a measure of structural-context similarity", in Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, (2002).

[24] Youneszadeh Jalili, S.; Meybodi, M.; Moradabadi, B., "presenting a novel method of distributed automata based link prediction in social networks", 6th conference of artificial intelligence and robotic and the 8th international symposium, Qazvin, Islamic Azad University, Qazvin Branch, (2016).

[25] Narendra, S.; Thathachar, M., "Learning Automata: An Introduction". New York: Prentice-Hall, (1989).

[26] Beigy, H.; Meybodi, M., "Utilizing distributed learning automata to solve stochastic shortest path problems", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 14, no. 05, pp. 591-615, (2006).

[27] "Resource", Linkprediction.org, 2017. [Online]. Available: http://www.linkprediction.org/index.php/link/resource/data. [Accessed: 01- Feb- 2017]. Latora, V.; Marchiori, M., "Efficient Behavior of Small-World Networks", Phys. Rev. Lett., vol. 87, no. 19, (2001).

[28] "Stanford Large Network Dataset Collection", Snap.stanford.edu, 2017. [Online]. Available: https://snap.stanford.edu/data/. [Accessed: 01- Feb- 2017].