Appl Intell
DOI 10.1007/s10489-014-0558-5

A link prediction algorithm based on ant colony optimization

BolunChen · Ling Chen

Abstract The problem of link prediction has attracted considerable recent attention from various domains such as sociology, anthropology, information science, and computer sciences. In this paper, we propose a link prediction algorithm based on ant colony optimization. By exploiting the swarm intelligence, the algorithm employs artificial ants to travel on a logical graph. Pheromone and heuristic information are assigned in the edges of the logical graph. Each ant chooses its path according to the value of the pheromone and heuristic information on the edges. The paths the ants traveled are evaluated, and the pheromone information on each edge is updated according to the quality of the path it located. The pheromone on each edge is used as the final score of the similarity between the nodes. Experimental results on a number of real networks show that the algorithm improves the prediction accuracy while maintaining low time complexity. We also extend the method to solve the link prediction problem in networks with node attributes, and the extended method also can detect the missing or incomplete attributes of data. Our experimental results show that it can obtain higher quality results on the networks with node attributes than other algorithms.

B. Chen (✉)
Department of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
e-mail: chenbolun1986@163.com

L. Chen
Department of Computer Science, Yangzhou University, Yangzhou, 225127, China

L. Chen
State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, 210093, China

Keywords Link prediction · Ant colony optimization · Complex networks

1 Introduction

Many social, biological, and information systems in the real world, from the nervous system to the ecosystem, from road traffic to the Internet, from the ant colony structure to human social relations, can be naturally described as networks, where vertices represent entities and links denote relations or interactions between vertices. As a topology approximation of complex systems, due to limitations of time and space, or experimental conditions, it is inevitable that there will be some errors or redundant links in constructing the complex network, and at the same time, there will also be some undetected potential links. In addition, because of the dynamic evolution of the complex network links over time, we need to predict the missing and potential links according to the known network information, which is the goal of network link prediction problem [1, 2].

Link prediction problem has a wide range of practical applications in different fields. For example, in biological networks, such as protein-protein interactionmetabolic and diseases-gene networks [3], link existing between the nodes indicates they have a interaction relationship. Due to the high experimental costs of revealing the hidden interaction relationships in these networks, the results of link prediction can direct the experiment designing so as to reduce the cost and improve the success rate of experiment. Predicting the loss and suspicious links of diseases-gene networks can help to explore the mechanism of diseases, predict and evaluate their treatment. Furthermore, it can also find new drug targets and open up new ways for drug development [4].

In social network analysis, link prediction can also be used as a powerful supplementary tool to accurately analyze the social network structure. Researches on online social networks analysis have developed very rapidly in recent years. In online social networks, potential friends of the users can be revealed by link prediction, and can be recommended to the users [5]. By analyzing social relations, we can find potential interpersonal links [6, 7]. Link prediction can also be used in the academic network to predict the type and cooperators of an academic paper [8]. Link prediction method can also be directly used for information recommendation, such as the commodity recommendation to customers [9]. Marketers would like to recommend products or services based on existing preferences or contacts. Social networking websites would like to customize suggestions for new friends and groups. For monitoring e-mail communication, link prediction is used to detect the anomalous e-mail [10]. Financial corporations would like to monitor transaction networks for fraudulent activity. In monitoring the network of criminals, link prediction is used to discover the hidden connection between criminals so as to prevent criminal or terrorist activity.

Link prediction not only has a wide range of practical value, but also has important theoretical significance. For example, it is helpful to understand the mechanism of the evolution of complex network [11] .Since the statistical magnitude to describe characteristics of the network structure is very large, it is difficult to compare the advantages and disadvantages of different mechanisms. Link prediction can provide a simple and unified platform for a fair comparison of network evolution mechanisms, so as to promote the theoretical research on a complex network evolution model.

In recent years, many methods on link prediction have been reported. Those methods can be classified into categories such as similarity-based methods, maximum likelihood methods and probabilistic model based methods.

In the similarity-based method, each node pair is assigned an index, which is defined as the similarity between the two nodes. All non-observed links are ranked according to their similarities, and the links connecting more similar nodes are supposed to have higher existence likelihoods. Node similarity can be defined by using the essential attributes of nodes: two nodes are considered to be similar if they have many common features [12] or topological structures [13]. Many studies found that there are substantial levels of topical similarity among users who are close to each other in the social network, such as friendship prediction in [14], which studied the presence of homology in three systems that combine tagging social media with online social networks. Many works exploit topological features of network structures for link prediction tasks. In [15], the overall relations between object pairs are defined as a link pattern, which consists of interaction pattern and connection structure in the network. The structural similarity indices can be classified into three categories: local indices, global indices, and quasi-local indices. Local indices use only neighbor information of the nodes. Typical local indices include Common Neighbors [16], Salton Index [17], Jaccard Index [18], Sorensen Index [19], Hub Depressed Index[20], Hub Promoted Index[21], Leicht-Holme-Newman Index (LHN1) [21], Preferential Attachment Index [22], Adamic-Adar Index [23] and Resource Allocation Index [24]. Global indices require global topological information. Katz Index [25]. Leicht-Holme-Newman Index (LHN2) [21], Matrix Forest Index (MFI) [26] are typical global indices. Quasi-local indices do not require global topological information but make use of more information than local indices. Such indices includes Local Path Index [24, 27], Local Random Walk [28],and Superposed Random Walk [28]. Another group of similarity is based on the random walk, such as Average Commute Time [29], Cos+ [30], random walk with restart [31], and SimRank[32]. Zhou et al. [24, 33] proposed two new local indices, Resource Allocation index and Local Path index. Empirical results show that these two indices outperform all other local indices. In particular, the local path index, requiring a little bit more information than the common neighbors index, provides competitively accurate prediction compared with the global indexes.

Liu and Lv [34] studied the link prediction problem based on the local random walk, and found that the limited step may get a better prediction than the result of global random walk. Rao [35] proposed an algorithm based on the MapReduce parallel computation model that can be applied to large complex networks. Dong [36] proposed an algorithm based on the gravitation of the node, which can improve the prediction accuracy while maintaining a low time complexity.

Another category of link prediction method is based on maximum likelihood estimation. These methods presuppose some organizing models of the network structure, with the detailed rules and specific parameters obtained by maximizing the likelihood of the observed structure. Then, the likelihood of any non-observed link can be calculated according to those rules and parameters. Typical organizing models of the network are the hierarchical structure model [37] and the stochastic block model [38–40] . In [41], a set of simple features are proposed as a structural model that can be analyzed to identify missing links. Hierarchical model has high accuracy in handling with the network of significant levels of the organization, such as the terrorist network and grasslands food chain network. However, since it needs to generate a lot of samples to predict the network, its computational complexity is too high to deal with the large scale networks. Link prediction method based on random block model can predict not only the missing links, but also predict and correct errors in the network, such as the errors links in protein interaction network. From the viewpoint of practical applications, an obvious drawback of the maximum likelihood methods is that it is very time consuming. It will definitely fail to deal with the huge online networks that often consist of millions of nodes.

Another type of link prediction method is based on the probability model. These model based methods aim at abstracting the underlying structure from the observed network, and then predicting the missing links by using the learned model. These methods first create a model containing a set of adjustable parameters, and then use optimization strategy to find the optimal parameter values, such that the resulting model can be better structures and relationships reflecting real network characteristics. The probabilistic model optimizes a target function to establish a model composed of a group of parameters Θ which can best fit the observed data of the target network. Then the probability of a potential link (*i,j*) is estimated by the conditional probability $P(A_{ij} = 1|\Theta)$. There are three mainstream probability based methods, respectively called Probabilistic Relational Model (PRM) [42], Probabilistic Entity Relationship Model (PERM) [43] and Stochastic Relational Model (SRM) [44]. Ramesh R et al. [45] proposed an approach for probabilistic link prediction and path analysis using Markov chains. H. Kashima et al. [46] introduce an approach for link prediction in network structured domains. An advantage of probability model based method is that it can achieve a higher predictive accuracy, but its time complexity and non-universality parameter calculation seriously restrict its application scope.

Several methods focus on supervised machine learning strategy for link prediction. The target attribute of those methods is a class label indicating the existence or absence of a link between a node pair. The relevance of using weights to improve supervised link prediction is investigated in [47]. In [48], a link propagation method is proposed, which is a semi-supervised learning algorithm for link prediction on graphs based on the popularly-studied label propagation. In [46], a parameterized probabilistic model of network evolution was presented and an incremental learning algorithm for such models was derived. Similar to the maximum likelihood methods, a drawback of machine learning based methods is their high time complexity, which is prohibited in some real applications.

Many studies focus on the link prediction in multidimensional and large-scale social networks. In [49], G. Rossetti et al. presented several predictors based on structural analysis of the multidimensional networks. H. H. Song et al. [50] proposed a method to approximate a large family of proximity measures for link prediction in large-scale networks. Although those methods are designed specifically for the large scale networks, accuracy of the results cannot be guaranteed due to the limit on computation time.

Another link prediction problem of increasing interest revolves around node attributes. Many real-world networks contain rich categorical node attributes, e.g., users in Google+ have profiles with attributes including employer, school, occupation and address. In the attribute inference problem, we aim to populate attribute information for network nodes with missing or incomplete attribute data. This scenario often arises in practice when users in online social networks set their profiles to be publicly invisible or create an account without providing any attribute information. The growing interest in this problem is highlighted by the privacy implications associated with attribute inference as well as the importance of attribute information for applications including people searching and collaborative filtering.

There are two sources of information in networks with node attributes, namely, topological information and attribute information. How to simultaneously incorporate these two sources of information is an important issue in the link prediction on networks with node attributes. The relational learning [51, 52], matrix factorization and alignment [53, 54] based approaches have been proposed to leverage attribute information for link prediction, but they suffer from scalability issues. More recently, Backstrom and Leskovec [55] presented a Supervised Random Walk (SRW) algorithm for link prediction that combines network structure and edge attribute information, but this approach does not fully leverage node attribute information as it only incorporates node information for neighboring nodes. Yin et al. [56, 57] proposed the use of Social-Attribute Network (SAN) to gracefully integrate network structure and node attributes in a scalable way. They focused on generalizing Random Walk with Restart (RWR) algorithm to the SAN model to predict links as well as to infer node attributes.

Ant colony optimization (ACO) is an evolution simulation algorithm proposed by M. Dorigo et al. [58, 60]. Inspired by the behaviors of the real ant colony, they recognized the similarities between the ants' food-hunting activities and travelling salesman problem (TSP), and successfully solved the TSP problems using the same principle that the ants have used to find the shortest route to food source via communication and cooperation. ACO has been successfully used for system fault detecting, job-shop scheduling, frequency assignment, network load balancing, graph coloring, robotics and other combinational optimization problems [61, 67]. ACO has some advantages such as allowing positive feedback, distributed computing, and constructive greedy heuristic search.

In this paper, we propose a link prediction method based on the ant colony optimization. In the algorithm, artificial ants are employed to travel on a logical graph. Each ant chooses its path according to the value of the pheromone and heuristic information on the edges. The paths the ants passing through are evaluated, and the pheromone information on each edge is updated according to the quality of the path it located. Finally, the pheromone on each edge is used as the score of the similarity between the nodes. We use AUC and precision as measurements to evaluate the performance of the algorithms, and compare it with the other link prediction algorithms. The experimental results on a number of real networks show that the accuracy of our algorithm is significantly superior to the other algorithms. We also extend the method to solve the link prediction problem in the networks with node attributes. The pheromones on the edges are used to predict links as well as infer node attributes. Our experimental results show that our algorithm can obtain higher quality results on the networks with node attributes than other algorithms.

The rest of this paper is organized as follows. Section 2 reviews the problem of link prediction and methods for evaluating the results. Section 3 presents the ACO based algorithm ACO_LP, and describes the implementation details of the algorithm. Section 4 extends the method to solve the link prediction problem in the networks with node attributes. Section 5 shows and analyzes the experimental results obtained by ACO_LP, and compares its performance with other similar methods. Section 6 draws conclusions.

2 Problem formulation and evaluation methods

We consider a network represented by an undirected simple network $G(V,E)$, where V is the set of nodes and E is the set of links. Multiple links and self-connections are not allowed in G. Let N=|V| be the number of nodes in G. We use U to denote the universal set containing all $N(N\text{-}1)/2$ possible links. The task of link prediction is to find out missing links (or the links that will appear in the future) in the set of non-existing links $U\text{-}E$.

The purpose of our method is to assign a score, $Score(x,y)$, to each pair of nodes $(x,y)\in U$. This score reflects the similarity between the two nodes. For a nodes pair (x,y) in $U\ E$, the larger $Score(x,y)$ is, the higher probability there will exist a link between nodes x and y.

To test the accuracy of the results of our algorithm, the observed links in E are randomly divided into two parts: the training set, E^T, which is treated as known information, while the probe set (i.e., validation subset), E^P , which is used for testing and no information in this set is used for prediction. $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. As an example, Fig. 1a shows a network with 15 nodes and 21 existing links. Our goal is to predict the potential links in the 84 unconnected node pairs. To test the algorithm's accuracy, we need to select some existing links as probe set, and the other as training set. For instance, we pick 5 links as probe links, which are presented by dash lines in Fig. 1b. Then, an algorithm only makes use of the information contained in the training graph presented by solid lines in Fig. 1b. The algorithm will eventually give a score to each of the 89 node pairs, including 84 non-existing links in $U\text{-}E$, and 5 test links in E^P.

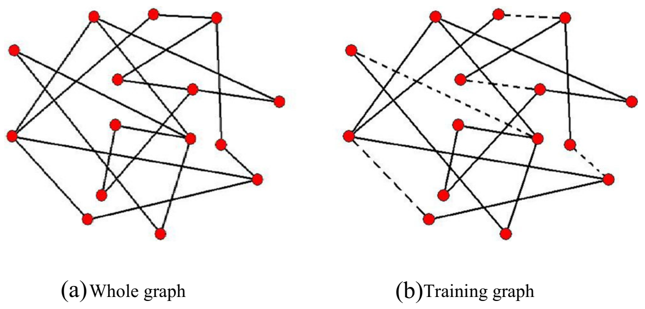

Fig. 1 A network

In principle, a link prediction algorithm provides an ordered list of all non-observed links (i.e., $U\text{-}E^T$) or equivalently gives each non-observed link, say $(x,y) \in U\text{-}E^T$, a score s_{xy} to quantify its existence likelihood. To quantify the accuracy of prediction algorithms, there are three standard metrics: AUC, Precision and Ranking Score.

(1) AUC

AUC (area under the receiver operating characteristic curve) measures the accuracy of link prediction results from the entirety. Provided the rank of all non-observed links, the AUC score can be interpreted as the probability that a randomly chosen missing link (a link in E^P) is given a higher score than a randomly chosen non-existing link(a link in $U\text{-}E$). . 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 non-existing link to compare their scores, if among n independent comparisons, there are n' times the missing link having a higher score and n" times have the same score, the AUC score is:

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

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

(2) Precision

Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the total number of items selected. That is to say, if we take the top-L links as the predicted ones, among which m links are right, then the precision is:

$$precision = \frac{m}{L}$$

Clearly, higher precision means higher prediction accuracy.

(3) Ranking Score

Ranking score (RS) considers the ranks of the similarity scores of the testing edges. Let $H= U\text{-}E^T$ be the set of nonobserved links. Let e_i be an unknown edge in testing set E^P, r_i be the rank of the edge e_i after sorting the edges in descending order of their scores. The ranking score of edge e_i is defined as: $RS_i = r_i/|H|$, and the ranking score of the link prediction result is:

$$RS = \frac{1}{|E^p|}\sum_{i\in E^p} RS_i = \frac{1}{|E^p|}\sum_{i\in E^p}\frac{r_i}{|H|}$$

It can easily be seen that prediction result with higher accuracy will get higher ranking score.

3 Framework of the ACO algorithm for link prediction

3.1 Basic idea of the algorithm

Given an undirected simple network G=(V,E), a complete graph G' called logical graph of G is constructed by adding all the missing links in G. In our algorithm, artificial ants are used to randomly walk on the logical graph G'. We refer the connections between node pairs in original graph G as "link", while those in the logical graph G' as "edge", which includes both existing and non-existing links in G. On each edge of G', we set the pheromone and heuristic information on the edges. The edge has higher probability to have a link will get larger values of pheromone and heuristic.

Each ant travels on the logical graph G' to visit n nodes and forms a path. In one iteration of an ant's walk, some nodes may be selected multiple times, and some nodes may not be selected. In the random walk, the ant at nodev_i chooses the next edge (v_i,v_j) to walk through according to a probability p_{ij}. The value of p_{ij} is defined according to the pheromone and heuristic information on edge (v_i,v_j). The edge with higher tendency to have a link will be assigned larger probability p_{ij}, and the ants will more likely choose this edge to pass through. After the ants finish a round of walk and form the paths, the algorithm evaluates the quality of each path. The path consisting of edges with higher probability to have links will get higher quality score. Then, the quality scores are used to modify the pheromone information on the edges of the path. The edge with higher quality score will obtain larger increment on pheromone information. This pheromone information in turn influences the walk of the ants in the next iteration: the larger amount of pheromone is laid on an edge, the more likely an ant will select this edge. The intensity of pheromone information on each edge could be increased by the ants passing it and decreased by evaporation in each iteration. Communications and cooperations between individual ants by pheromone information enable the algorithm to have strong capability of finding the best paths. Finally, the pheromone τ_{ij} on edge (v_i,v_j) is used as the score reflecting the similarity between the two nodes. The pheromone matrix formed is output as the final score matrix.

3.2 Parameter initialization

For the undirected simple network G=(V,E),let $|V|$ =n, and V={v_1,v_2, ...v_n},n*n matrix A=[a_{ij}] be the adjacent matrix of G. We use a vector S=(s_1,s_2, ...,s_n) to represent a path an ant walking through in the logical graph G', where $s_i \in V$ is the ith node on the path. Initially, we set all the elements in S as Φ, which represents an empty node, and will be replaced by a real node in the ant's random walking.

We set the initial value of pheromone on the edge between the nodes (v_i,v_j) as

$$\tau_{ij} = \lambda * (a_{ij} + \varepsilon) \tag{1}$$

Here, λ and ε are positive constants, namely if (v_i,v_j) $\in E$ the initial value of pheromone τ_{ij} is set as λ(1+ ε),otherwise it is set as $\lambda.\varepsilon$. It is obvious that the edges which have link connection will have higher initial pheromone value. Such pheromone information will guide the ants to walk through the existing links and their neighbors with higher probability.

Denote the set of common neighbors of nodes (v_i,v_j) as:

$$\Gamma(x, y) = \{v|v \in V, (x, v) \in E \wedge (y, v) \in E\}$$

We set the value of heuristic information on the edge between nodes (v_i,v_j) as

$$\eta_{ij} = \gamma * |\Gamma(i, j)| \tag{2}$$

Here, parameter γ is a positive constant. Such heuristic information will direct the ants to walk towards the closely connected nodes.

3.3 Probability for ants' path selection

In each iteration, ant at node v_i selects an edge to reach the next node according to a probability. We define p^k_{ij} as the probability for ant k at node v_i to choose v_j:

$$p^k_{ij} = \frac{\tau^{\alpha}_{ij}.\eta^{\beta}_{ij}}{\sum\limits_{k=1}^{n}\tau^{\alpha}_{ik}.\eta^{\beta}_{ik}} \tag{3}$$

Here,τ_{ij} is the pheromone on the edge between nodes v_i and v_j, η_{ij} is a heuristic function which is defined as the visibility of the edge (v_i,v_j). Parameters α, β determine the relative influence of the pheromone and the heuristic information. Obviously, the edge with larger pheromone and heuristic value will have higher probability to be chosen by the ants.

3.4 The fitness function

After each iteration, the tour of each ant forms a path consisting of n nodes. The path of the k-th ant is denoted as $S(k)$=(s_1,s_2,s_3,...,s_n), here $s_i \in V$ is the ith node in the path. A fitness function is defined to measure the quality of each path, and will be used to update the pheromone information. A path with more existing links and closely connected nodes will have higher quality score, since each pair of adjacent nodes in the path are more likely to be connected by a potential link. Therefore, the fitness of a path can be measured in two aspects, namely, the importance of the nodes and edges on the path.

Generally the importance of a node is measured by its "centrality" under different definitions. Different centricity depicts the different function of nodes in the network, such as the spreading ability, the influence of the node. Degree based centrality is the most simple and direct measure of the node importance. In general, greater centricity degree of a node means that it is more important and more likely to be linked with other nodes. Therefore, the fitness of a path is defined as the summation of the degrees of its nodes :

$$Q(S) = \sum_{i=1}^{n} d(s_i) \tag{4}$$

Here, $d(s_i$ is the degree of the node s_i).

However, not all the nodes with large degree are the most important. The importance of a node is related with the structure of the network and the function of the node. For example, in the communication network, although some nodes have small degrees, they are possibly the hub points for large amount of packets to pass through. Therefore, we use betweenness as a measure of the node's importance. Sociologist Linton Freeman [68] first proposed the concept of betweenness as a measure of both the load and importance of a node. The former is more global to the network, whereas the latter is only a local effect. The betweenness centrality of a node v_i is given by the expression:

$$B(v_i) = \sum_{s \neq i \neq t} \frac{n_{st}^{i}}{g_{st}} \tag{5}$$

where, g_{st} is the total number of shortest paths from node s to node t, and$E' = E_{pp} \cup E_{pa}$ is the number of those paths that pass through v_i.

Based on the node betweenness centrality defined by (5), the fitness of a path is defined as the summation of the betweenness of the nodes on the path:

$$Q(S) = \sum_{i=1}^{n} B(s_i) \tag{6}$$

In addition to the importance of the nodes, the importance of the edges is also a factor in the quality measure of a path. One measurement of the edge importance is the clustering coefficient. The importance of each edge is defined as the number of all triangles associated with it. For an edgee_{ij} = (s_i, s_j), its clustering coefficient is defined as:

$$C(e_{ij}) = \frac{z_{ij} + 1}{\min\left[(d_i - 1), (d_j - 1)\right]} \tag{7}$$

Here, z_{ij} is the number of triangles with edge e_{ij}, d_i and d_j are degrees of nodess_i and s_j respectively. Larger clustering coefficient of an edge between two nodes indicates the higher probability that they are connected by a link. Let S=(s_1,s_2,s_3,...,s_n) be a path, here $s_i \in V$ is the ith node in the path. Denote the ith edge on the path as $e_{i,i+1}$($i = 1, 2, ..., n - 1$). Based on the edge clustering coefficient defined by (7), the fitness of a path is defined as:

$$Q(S) = \sum_{i=1}^{i=n-1} c(e_{i,i+1}) \tag{8}$$

Another measurement of the importance of an edge is the edge betweenness. Similar to node betweenness, the betweenness of the edge is defined as:

$$B(e_{ij}) = \sum_{s \neq t} \frac{n_{st}^{e}}{g_{st}} \tag{9}$$

Here, n_{st}^{e} is the number of the shortest paths from node s to node t passing through the edge. e_{ij}, and g_{st} is the number of the shortest paths from node s to node t. Edge betweenness is the measure of edge's ability of communication, the greater betweenness an edge has, the more important it is in the connectivity of the network.

Using the edge betweenness, the quality of a path is defined as:

$$Q(S) = \sum_{i=1}^{i=n-1} B(e_{i,i+1}) \tag{10}$$

We can choose one from formulas (4), (6), (8), (10) as the fitness to evaluate the paths in each iteration. In this paper, we use the node's centrality based measurement in our experiments.

3.5 Pheromone updating

After each iteration, the algorithm updates the pheromone value on each edge according to the formulas as follows:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{11}$$

Here,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{12}$$

and

$$\Delta\tau_{ij}^{k}(t) = Q(S) \tag{13}$$

Obviously, the more ants at v_i select the node v_j , the more increment of pheromone the egde e_{ij} has, and the higher probability the ants select this edge in the next iteration. This forms a positive feedback by the pheromone system. In our experiments, we set the fitness of the path S as

$$Q(S) = C * \frac{1}{n} \sum_{i=1}^{n} d(s_i), \tag{14}$$

where C is a positive constant, $d(s_i)$ is the degree of node s_i.

3.6 Termination conditions and outputs

The algorithm ceases the iterations according to a certain termination condition. We stop the iterations when the pheromone values on each edge obtained in adjacent iterations tend to stabilize. In addition, we also set up a threshold Nc, which is the maximum number of iterations. The iterations should be ended as well when the number of iterations goes beyond Nc.

Finally, the algorithm outputs the pheromone matrix as the score matrix, namely, the final score of nodes pair v_i and v_j is $Score(i, j) = \tau_{ij}$. To evaluate the quality of the link prediction result, we need to rank all the non-existing links in decreasing order according to their scores, and use the AUC and precision to assess the performance of the algorithm.

3.7 Framework of the algorithm

The framework of our ant colony optimization based algorithm for link prediction ACO_LP is as follows.

Algorithm 1 ACO_LP (ACO for Link Prediction)

```
Input:   A:    Adjacency matrix of the network;
         Nc:   The maximum number of iterations;
         ε     The threshold for the error of pheromone
                 information;
Output:  Score: The score matrix;
Begin
1.  t = 1;
2.  Parameter initialization:
        set the initial values of pheromone matrix τ and
          heuristic matrix η according to (1) and (2);
3.  Repeat
4.      For k=1 to m do /* for the m ants*/
5.        Ant k randomly selects a node $s_1$
6.        for $i = 1$ to n-1 do
7.          Ant k selects the next node according to (3);
8.      End for i
9.      Calculate the fitness of the path formed by ant k
          according to (14);
10.   End for k
11.   Update the pheromone values according to (11);
12. Until $\max_{1\le i,j\le n} |\tau_{ij}(t+1) - \tau_{ij}(t)| \le \varepsilon$ or $t > Nc$ ;
13. $Score = \tau$;
14. Output the score matrix Score ;
End
```

Line 2 of algorithm ACO_LP sets the initial values of pheromone matrix τ and heuristic matrix η.To calculate the heuristic information η, we need to calculate the number of common neighbors of all node pairs. Let n be the number of nodes in the network, and k be the average degree of the nodes. For each nodev_i, it takes d^2 time to search for the common neighbors of v_i with other nodes. Therefore, time complexity of this step is O(nk^2).

In lines 3 to 12, the ants travel in the network to form the paths. The time complexity for an ant choosing its path in each iteration is O(n^2). Since there are m ants and N_c iterations, the total time is O(mn^2N_c). Therefore the overall time complexity of the algorithm is O(nk^2+N_cmn^2). Since N_c and m are constants, the time complexity of the algorithm is O(n^2).

4 Link prediction using node attributes

For the networks with node attributes, we also use an undirected graph$G$ =(V,E) to represent a network, where edges in E represent interactions between the n nodes in V. In addition to network structure, we have categorical attributes for the nodes. For instance, in the Google+ social network, nodes are users, edges represent their friendship or some other relationship. Node attributes are derived from user profile information and include fields such as employer, school, and hometown. In this work, we restrict our focus on categorical variables, since the other types of variables, e.g., live chats, email messages, real-valued variables, etc., could be clustered into categorical variables via vector quantization, or directly discredited to categorical variables. We use a binary representation for each categorical attribute. For example, various employers can be treated as separate binary attributes. Hence, for a specific social network, the number of distinct attributes m is finite, though it could be very large. Attributes of a node v_i are then represented as an m-dimensional binary column vector $b_i = (b_{i1}, b_{i2}, ..., b_{im})$. The j-th entry of b_i is defined as

$$b_{ij} = \begin{cases} 1 & v_i \text{ has the } j\text{th attribute} \\ 0 & \text{otherwise} \end{cases}$$

We denote the m*n attribute matrix for all nodes as $B = \left[b_{ij}\right]$.

Given a network G with m distinct categorical attributes, an attribute matrix B, we create an augmented graph G_A by adding m additional nodes to its logical graph G', with each additional node corresponding to an attribute. For each node v in G' with attribute a, we create an undirected link between v and a in the augmented graph G_A. This augmented graph G_A includes the original network interactions, relations between nodes and their attributes. Artificial ants are used to randomly walk on the augmented graph G_A, instead of on the logical graph G'.

There are two types of nodes in the augmented graph G_A =(V$_A$, E_A), namely, the item node set V_p and the attribute node set V_a, where $V_A = V_p \cup V_a$. The set E_A in G_A also consists of two types of edges: $E_A = E_p \cup E_a$, where E_p is the set of edges between the item nodes, and $E_a$is the set of edges between the item nodes and the attribute nodes. There is no edge between two attribute nodes. Let $|V_p| = n$, $|V_a| = m$, then $G_A$is represented by an (n+m)*(n+m) adjacent matrix.

For each edge (v_i,v_j) in set E_p, which is the set of edges between the item nodes, we set the initial pheromone value on the edge (v_i,v_j) as $\tau_{ij} = \lambda * (a_{ij} + \varepsilon)$. Here, λ and ε are positive constants. For an edge (v_i,v_j) in set E_a, which is the set of edges between the item and attribute nodes, we set the initial value of its pheromone as $\tau_{ij} = \lambda * (1 + \varepsilon)$.

It is obvious that the edges which connect an item and its attributes will have higher initial pheromone value. Such pheromone information will guide the ants to walk through the paths between the nodes of items with the identical attributes.

We set the value of heuristic information on the edge between two item nodes (v_i,v_j) as $\eta_{ij} = \gamma * |\Gamma(i, j)|$. Here, parameter γ is a positive constant.

We notice that if an attribute is shared by fewer items, those items are more similar, and are more likely to be linked. Therefore edge connecting an attribute node of lower degree should have higher heuristic value. For each edge (v_i,v_j) in set E_a, where v_i is an item node and v_j is an attribute node, we set the value of its heuristic information as $\eta_{ij} = \mu/d_j$. Here, parameter μ is a positive constant, d_j is the degree of v_j. Such heuristic information will direct the ants to walk between the item nodes through their common attribute nodes with low degrees.

Each ant travels on the augmented graph G_A to visit n+m nodes and forms a path. In each iteration of an ant's walk, some nodes may be selected multiple times, and some nodes may not be selected. In the random walk, the ant at each node chooses the next edge (v_i,v_j) to walk through according to a probability p_{ij}. defined in (3). The edges in E_p with higher tendency and the edges in $E_a$linked with lower degree attribute node will be assigned larger probability p_{ij}, and ants will more likely to choose those edges to pass through. After the ants finish a round of walk and form the paths, the algorithm evaluates fitness of the paths according to (14). Then, the fitness scores are used to modify the pheromone information on the edges of the path according to (11). Finally, the pheromone τ_{ij} on each edge (v_i,v_j) in set E_p is used as the score reflecting the similarity between the two nodes. Also, the pheromone τ_{ij} on each edge (v_i,v_j) in set E_a is used to detect the potential attributes of item node v_i. Suppose an edge (v_i,v_j) in set E_a connects an item node v_i with an attribute node v_j , if the pheromone τ_{ij} on edge (v_i,v_j) is greater than a threshold, node v_i probably has the attribute represented by node v_j.

5 Experimental results

In this section, we empirically demonstrate the effectiveness of our proposed algorithm ACO_LP on real world networks. We also compare its performance against the traditional similarity based link prediction algorithms such as CN Salton, Jaccard Sorensen, HPI, HDI, LHN_I, PA, LP and Katz. We focus on the accuracy of the results and the algorithms' computing time. All experiments have been conducted on Microsoft Windows 7 operating system, and the results are visualized on Matlab 6.0. Based on our experience in ACO applications, we set parameters $\alpha = 0.8$, $\beta = 0.7$, $\varepsilon = 0.01$, $C = 0.95$ in our experiments.

5.1 Data Sets

In this paper we consider six benchmark data sets [69] representing networks drawn from disparate fields: protein-protein interaction network (PPI), coauthorships network between scientists (NS),electrical power grid of the western

US (Grid), network of the US political blogs (PB),Internet (INT),and US airport network (USAir). For each dataset, we test on its largest connected component. Table 1 summarizes the topological features of the largest components of those networks. In the table, N and M are the total numbers of nodes and links, respectively. NUM_C is the number of the connected components in the network and the size of the largest one. For example, 1222/2 means that this network has 2 connected components and the largest one contains 1222 nodes. In the table, e is the efficiency of the network [70], C and r are clustering coefficient [71] and assortative coefficient [72], respectively. K is the average degree of the network.

5.2 Test on quality of the results

First, we test the accuracy of the results by the algorithms using AUC score and precision as measurements. To evaluate the accuracy of the results a random 10-fold cross validation (CV) is used. In 10-fold cross-validation, the original nodes are randomly partitioned into 10 subsets. Of the 10 subsets, a single subset is retained as the validation data for testing the algorithms, and the remaining 9 subsets are used as training data. The cross-validation process is then repeated 10 times. We calculated standard deviation of the results on each data set, and find that all of the standard deviations are less than 0.024. The 10 results from the folds are averaged to produce a single estimation. Table 2 presents the average AUC scores on 10-fold CV tests by different algorithms. In the table, the highest AUC scores for the data sets by the 11 algorithms are emphasized in bold-face.

As shown in Table 2, we can see that among all the 11 algorithms, ACO_LP has the highest AUC scores on all the datasets. Even for the most difficult data set Grid, algorithm ACO_LP gets the highest AUC score 0.9985, while the other algorithms get AUC scores from 0.4677 to 0.6375. Comparing Tables 1 and 2, we can find that the AUC scores of the data sets are roughly proportional to their clustering coefficients, an algorithm can get better results on data sets with larger clustering coefficients. Our algorithm sets an initial value to those logical edges where a link does not exist in real network, it can increase the diversity of ants search.Therefore, our algorithm still achieves better results on the networks with low clustering coefficients. This shows that the algorithm ACO_LP can achieve high quality results and strong robustness.

Based on Table 2, we use Wilcoxon signed-rank test to show that the AUC scores of the results by ACO_LP are statistically different from those by other methods. Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when two related samples or matched samples are compared. It does not make assumptions about the distribution of the data. We calculate the W-values of precisions by ACO_LP with those by other methods, and the results are as shown in Table 3. We let the confidence level α=0.05, and the number of samples n=6. From the table of criteria W value, we know W(0.05, 6)=2. Since all the W values are greater than W(0.05, 6), there are significant differences between the AUC scores by ACO_LP and other ten methods. Therefore, the quality of results by our algorithm ACO_LP is obviously higher than those by other methods.

Table 1 Topological features of the giant components in the six networks tested

Networks	N	M	$NUMc$	e	C	r	K
USAir	332	2126	332/1	0.440	0.749	-0.228	12.807
PB	1224	19090	1222/2	0.397	0.361	-0.079	31.193
NS	1461	2742	379/268	0.016	0.878	0.462	3.754
PPI	2617	11855	2375/92	0.180	0.387	0.454	9.060
Grid	4941	6594	4941/1	0.056	0.107	0.004	2.669
INT	5022	6258	5022/1	0.167	0.033	-0.138	2.492

Table 2 Comparison of the algorithms' accuracy quantified by AUC

	USAir	PB	NS	PPI	Grid	INT
CN	0.9366	0.9218	0.9404	0.8987	0.5896	0.5451
Salton	0.9230	0.8739	0.9377	0.8980	0.5896	0.5552
Jaccard	0.8854	0.8714	0.4903	0.7691	0.4926	0.7770
Sorensen	0.8876	0.8720	0.9267	0.8980	0.5987	0.5640
HPI	0.8602	0.8502	0.9504	0.8969	0.5957	0.5512
HDI	0.8698	0.8681	0.9248	0.8979	0.5896	0.5569
LHN_I	0.7194	0.7541	0.9391	0.8941	0.5896	0.5631
PA	0.8233	0.8179	0.6147	0.7817	0.4677	0.6183
LP	0.9329	0.9267	0.9312	0.9395	0.6205	0.6230
Katz	0.9110	0.9232	0.9272	0.9196	0.6375	0.3732
ACO_LP	0.9373	0.9664	0.9985	0.9852	0.9985	0.9915

Table 3 W-values of AUC scores by ACO_LP with those by other methods

Our Method	Other Method Compared	W-value
ACO_LP	CN	5.3331
	Salton	5.4298
	Jaccard	7.3285
	Sorensen	5.3965
	HPI	5.5878
	HDI	5.5221
	LHN_I	6.365
	PA	7.6181
	LP	4.6006

Table 4 Comparison of the algorithms' accuracy quantified by precision

	USAir	PB	NS	PPI	Grid	INT
CN	0.6585	0.2356	0.6545	0.2009	0.0364	0.0128
Salton	0.0976	0.0012	0.7055	0.0034	0.0121	0
Jaccard	0.1037	0.0407	0	0.0017	0	0.0064
Sorensen	0.0976	0.0024	0.7055	0.0034	0.0121	0
HPI	0.0366	0.0018	0.2364	0.0265	0.0091	0
HDI	0.0732	0.0144	0.6945	0.0034	0.0121	0
LHN_I	0.0122	0.0005	0.3273	0.0017	0.003	0
PA	0.1037	0.0317	0.0073	0.0051	0	0
LP	0.5671	0.2261	0.3055	0.3897	0.0091	0.0096
Katz	0.5488	0.2105	0.3055	0.2214	0.0061	0.0256
ACO_LP	0.9756	0.3923	0.9927	0.6684	1	0.5783

Next, we test and compare the precisions of the algorithms on different datasets, the results are shown in Table 4.

As shown in Table 4, we can also see that the ACO_LP algorithm has higher precision than all the other algorithms. The experimental results show that our algorithm ACO_LP performed significantly better than all the other algorithms in all the data sets. The reason for ACO_LP achieving high quality results is that it uses both the pheromone and heuristic information, which reflect both local and global structure of the network. By the ants' touring on the network, global topological information is transformed and accumulated by the pheromone on each edge. Therefore the final pheromone information is more accurate as a similarity score between the nodes than other similarity measurements based on local information.

5.3 Test on the time requirement by the algorithms

Computational complexity is another important concern in the designing of link prediction algorithm. In our experiments, we also compared the time complexity of ACO_LP algorithm with the other algorithms, and the results are depicted in Fig. 2. We can see from Fig. 2 that the computational time required by ACO_LP algorithm is much less than those of the algorithms CN, Salton, Jaccard, Sorensen, HPI, HDI, LHN_I and PA, and slightly more than those of the algorithms LP and Katz.

Table 5 Topological features of the giant components in the eight networks tested

Networks	N	M	NUM_C	e	C	r	K
ACM	1465	1960	16/688	0.0014	0.3621	0.5570	1.6505
CAIP	2563	2505	178/631	0.0026	0.6811	0.0424	1.8662
CISIS	2122	2385	287/433	0.0078	0.7811	0.1491	3.8883
ICANN	3786	3463	166/807	0.0023	0.6898	0.1586	3.1759
ICICS	888	1066	187/208	0.0139	0.7484	0.2726	3.1486
ICML	2640	2213	1733/302	0.0719	0.6470	0.0132	3.6201
NLDB	847	1041	96/225	0.0072	0.7130	0.2112	2.8595
WWW	5400	3421	1995/897	0.0182	0.7592	0.3724	3.9430

Let n be the number of nodes in the network, and k be the average degree of the nodestime complexity of ACO_LP algorithm is O(n^2). In CN algorithm, for each node v_i, it takes d^2 time to search for the common neighbors of v_i with other nodes. Therefore, time complexity of algorithm CN is O(nk^2). Similarly, other common neighbor based algorithms, such as Salton, Jaccard, Sorenson, HPI, HDI, LHN-I and PA, also have the same time complexity of O(nk^2). Although the common neighbor based algorithms have lower time complexity than ACO_LP algorithm, the large hidden constants make those algorithms much slower than ACO_LP. All the experiment results show that ACO_LP algorithm can achieve high quality prediction results in less computational time.

5.4 Test on networks with node attributes

We also test our extended method on networks with node attributes. We test on eight data sets [73] representing networks drawn from Digital Bibliography Library Project (DBLP) which is a computer science bibliography website

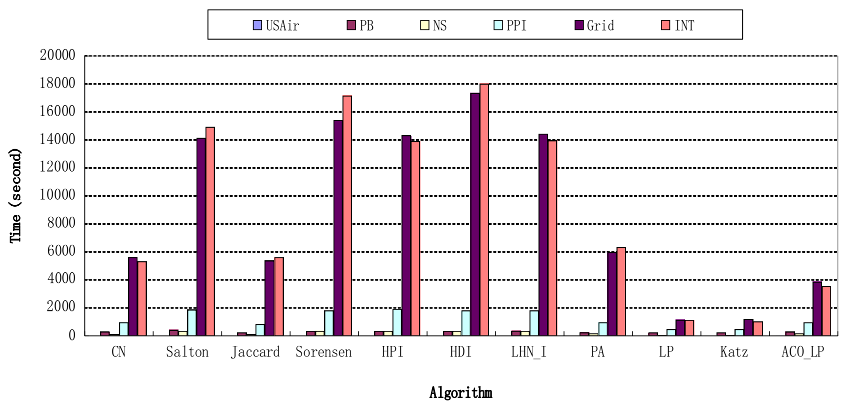

Fig. 2 Running time of the algorithms

Table 6 Comparison of the algorithms' accuracy quantified by AUC

	ACM	CAIP	CISIS	ICANN	ICICS	ICML	NLDB	WWW
CN	0.8635	0.9237	0.9618	0.9350	0.9598	0.9236	0.9214	0.9505
Salton	0.7932	0.9169	0.9620	0.9350	0.9603	0.9238	0.9217	0.9417
Jaccard	0.4223	0.3240	0.4475	0.4150	0.3401	0.5223	0.3352	0.4426
Sorensen	0.8552	0.9141	0.9620	0.9350	0.9573	0.9236	0.9217	0.9525
HPI	0.8222	0.9141	0.9621	0.9350	0.9503	0.9238	0.9216	0.9305
HDI	0.8263	0.9141	0.9619	0.9350	0.9602	0.9235	0.9216	0.9505
LHN_I	0.8552	0.9140	0.9618	0.9349	0.9342	0.9234	0.9214	0.9504
PA	0.5541	0.5080	0.5842	0.5535	0.5335	0.5767	0.5154	0.5846
LP	0.8301	0.9178	0.9696	0.9363	0.9586	0.9284	0.9204	0.9539
Katz	0.8219	0.9175	0.9693	0.9361	0.9273	0.9126	0.9197	0.9492
ACO_LP	0.9455	0.9998	0.9699	0.9412	0.9630	0.9579	0.9236	0.9613

hosted at Universität Trier, Germany. It has existed at least since the 1980s, and has listed more than 2.3 million articles on computer science. All important journals on computer science are tracked. Proceedings papers of many conferences are also tracked. It consists of major data bases of publications in computer science. In our experiment, we test on eight datasets including Computer Science Conference (ACM), Applications of Natural Language to Data Bases (NLDB), Complex, Intelligent and Software Intensive Systems (CISIS), International Conference on Information and Communication Security (ICICS), International Conference on Machine Learning (ICML), International World Wide Web Conferences (WWW), Computer Analysis of Images and Patterns (CAIP) and International Conference on Artificial Neural Networks (ICANN). We test on part of the authors in each dataset. For instance, in dataset ACM, we only test the authors in the ACM conference in the years from 1986 to 1996. For each dataset, we construct a network where each node represents an author. Co-authorship between two authors is mapped to the link between their corresponding person nodes. For each author, we get his entire publication history. Terms in the paper titles are considered as the attributes for the corresponding author. Since networks of some database are not connected, we test only on the largest component. Table 5 summarizes the topological features of the largest components of those networks. In the table, N and M are the total numbers of nodes and links, respectively. NUM_C is the number of the connected components in the network and the size of the largest one. For example, 1995/897 means that this network has 897 connected components and the largest one contains 1995 nodes. In the table, e is the efficiency of the network [70], C and r are clustering coefficient [71] and assortative coefficient [72], respectively. K is the average degree of the network.

Table 5 Topological features of the giant components in the eight networks tested

On those eight datasets, we test the accuracy and the computing time of our algorithm, and compare its performance with the link prediction algorithms CN, Salton, Jaccard, Sorensen, HPI, HDI, LHN_I, PA, LP and Katz.

First, we test the accuracy of the results by the algorithms using AUC score and precision as measurements. To evaluate the accuracy of the results, a random 10-fold cross validation is used. Table 6 presents the average AUC scores on 10-fold CV tests by different algorithms. In the table, the highest AUC scores for each data set by the 11 algorithms are emphasized in bold-face.

We can see from Table 6 that among all the 11 algorithms, ACO_LP has the highest AUC scores on all the datasets. For instance, for dataset ACM, algorithm ACO_LP gets the highest AUC score 0.9455, while the other algorithms get AUC scores less than 0.8635. This shows that the algorithm ACO_LP can achieve high quality results and strong robustness.

Based on Table 6, we also use $Wilcoxon$ signed-rank test to show that the AUC score of the result by ACO_LP is

Table 7 W-values of AUC scores by ACO_LP with those by other methods

Our Method	General Methods	the test statistic W
ACO_LP	CN	1.5081
	Salton	2.1588
	Jaccard	21.0611
	Sorensen	1.6356
	HPI	2.0154
	HDI	1.8555
	LHN_I	1.7567
	PA	15.1941
	LP	1.7507

Table 8 Comparison of the algorithms' accuracy quantified by precision

	ACM	CAIP	CISIS	ICANN	ICICS	ICML	NLDB	WWW
CN	0.7355	0.6739	0.5230	0.5648	0.5071	0.5418	0.7623	0.6995
Salton	0.24	0.3043	0.6586	0.5000	0.8429	0.4121	0.7049	0.7531
Jaccard	0	0	0.0097	0	0.0286	0.0042	0	0
Sorensen	0.4215	0.2065	0.6566	0.5	0.8400	0.4205	0.7049	0.7568
HPI	0.3554	0.1712	0.2397	0.1993	0.3714	0.1527	0.1721	0.2685
HDI	0.3140	0.2065	0.6634	0.5	0.8571	0.4268	0.6700	0.7800
LHN_I	0.2893	0.1929	0.3971	0.2392	0.5214	0.1862	0.4262	0.4200
PA	0	0.0020	0.0097	0	0.0071	0	0.0050	0.0028
LP	0.1983	0.1640	0.1792	0.1860	0.1739	0.1172	0.1650	0.5200
Katz	0.1818	0.1467	0.1792	0.1827	0.1429	0.1160	0.1840	0.4600
ACO_LP	0.7507	0.9674	0.5649	0.6578	0.8365	0.6054	0.8361	0.8121

statistically different from those by other methods. We calculate the W-values of AUC scores by ACO_LP with those by other methods, and the results are as shown in Table 7. We also let the confidence level α=0.05, and the number of samples $n = 8$. From the table of criteria W value, we know $W(0.05, 8)$= 6. From the table we can see that the W values of AUC scores by ACO_LP with those by other methods except $Jaccard$ and PA are greater than $W(0.05, 8)$, there are significant differences between the AUC scores by ACO_LP and other eight methods. Therefore, the quality of results by our algorithm ACO_LP is obviously higher than other eights methods.

Next, we test and compare the precisions of the algorithms on different datasets, the results are shown in Table 8.

As shown in Table 8, we can see that the algorithm ACO_LP has higher precision than all the other algorithms in six datasets other than CISIS and ICICS. For datasets CISIS and ICICS, precisions of ACO_LP are very close to the highest ones.

We also compared the time complexity of algorithm ACO_LP with the other algorithms, and the results are depicted in Fig. 3. We can see from Fig. 3 that the computational time required by ACO_LP algorithm is less than or very close to the other algorithms in most of the datasets. But for datasets WWW and ICANN, ACO_LP consumes more computation time than some other methods. Since these two datasets have large number of attributes, they create lots of additional nodes in the augmented graph, and consume more computation time. Since our algorithm ACO_LP can also discover the potential attributes of the items, this excess computation time is due to the cost of attribute detecting for the item nodes. To reduce the time cost for the datasets with large number of attributes, it is necessary to eliminate the nonessential attributes by dimension reduction in data preprocessing.

6 Conclusions

With the large amount of network data available in electric form today, link prediction has become a popular subarea in data mining. From the perspective of swarm intelli-

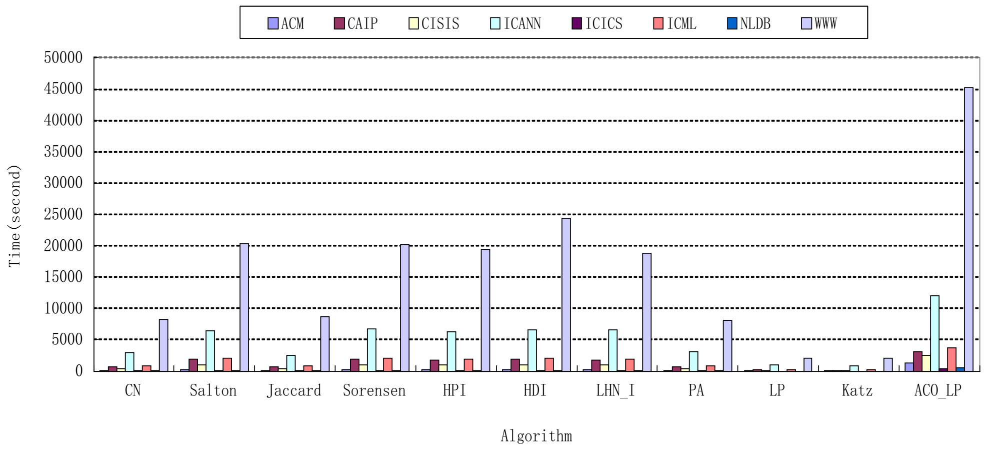

Fig. 3 Running time of the algorithms

gence, a new link prediction method based on ant colony optimization is proposed in this paper. In the algorithm, artificial ants are used to randomly walk on the logical graph. Each ant chooses its path according to the value of the pheromone and heuristic information on the edges. The initial value of pheromone on each edge on the logical graph is set according to the link connection on the network. The initial value of heuristic information on each edge is set according to the common neighbors of the two nodes it connects.

The paths obtained by the ants' walking are evaluated, and the pheromone information on each edge is updated according to the quality of the path it located. Finally, the pheromone on each edge is used as the final similarity score of the node pair. Empirical results show that our algorithm can achieve higher quality results of link prediction using less computation time than other algorithms. We also extend our method to solve the link prediction problem in networks with node attributes. We expend the logical graph by adding attribute nodes, and use the ant colony optimization algorithm on this augmented graph to perform link prediction using both topologic information and attributes on the nodes. . Our experimental results show that such extended method also can precisely detect the missing or incomplete attributes of data.

There are two reasons for ACO_LP achieving high quality results. One is that it uses both the pheromone and heuristic information reflecting both local and global structure of the network. Another reason is that ACO_LP considers both attribute and structure informationour experimental results show that it can obtain higher quality results on those networks with node attributes.

When the datasets have large number of attributes, since our algorithm ACO_LP creates lots of additional nodes in the augmented graph, it consumes more computation time. It is our further work to develop an efficient way to eliminate the some nonessential attributes to reduce the time cost.

Acknowledgments This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61379066, 61070047, 61379064. Natural Science Foundation of Jiangsu Province under contracts BK20130452,BK2012672BK2012128, and Natural Science Foundation of Education Department of Jiangsu Province under contract 12KJB520019,13KJB520026.Foundation of Graduate Student Creative Scientific Research of Jiangsu Province under contract CXZZ13_0172.

References

1. Lichtenwalter RN (2010) New precepts and method in link prediction. In: Proceedings of ACM KDD'10, pp 243–252
2. Lü L, Zhou T (2011) Link prediction in complex networks:A survey. Phys A 390:1150–1170
3. Airoldi EM, Blei DM, Fienberg SE (2006) Mixed membership stochastic block models for relational data with application to protein-protein interactions. In: Proceedings of the international biometrics society annual meeting
4. Guimera R, Sales-Pardo M (2010) Missing and spurious interactions and the reconstruction of complex networks. Proc. Natl Acad Sci USA 106(52):22073–22078
5. Papadimitriou A, Symeonidis P, Yannis M (2012) Fast and accurate link prediction in social networking systems. J Syst Softw 85(9):2119–2132
6. Hossmann T, Nomikos G, Spyropoulos T, Legendre F (2012) Collection and analysis of multi-dimensional network data for opportunistic networking research. Comput Commun 35(13):1613–1625
7. Jahanbakhsh K, King V, Shoja GC (2012) Predicting missing contacts in mobile social networks. Pervasive Mob Comput 8(5):698–716
8. Sun Y, Barber R, Gupta M (2011) Co-author relationship prediction in heterogeneous bibliographic networks. In: IEEE 2011 international conference on advances in social networks analysis and mining (ASONAM), pp 121–128
9. Li X, Chen H (2013) Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. Decis Support Syst 54(2):880–890
10. Huang Z, Lin DKJ (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS J Comput 21:286–303
11. Liu HK, Lü LY, Zhou T (2011) Uncovering the network evolution mechanism by link prediction (in Chinese). Sci Sin Phys Mech Astron 41:816–823
12. Lin D (1998) An information-theoretic definition of similarity. ICML 98:296–304
13. Sun D, Zhou T, Liu J-G, Liu R-R, Jia C-X, Wang B-H (2009) Information filtering based on transferring similarity. Phys Rev E 80:017101
14. Aiello LM, Barrat A, Schifanella R (2012) Friendship prediction and homophily in social media. ACM Trans Web (TWEB) 6(2):9
15. Gao S, Denoyer L, Gallinari P (2012) Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks. J China Univ Posts Telecommun 19:172–181
16. Newman MEJ (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64:025102
17. Salton G, McGill MJ (1983) Introduction to modern information retrieval
18. Jaccard P (1901) Etude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Science Naturelles 37:547–579
19. Sorensen T (1948) A method of establishing groups of equalamplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol Skr 5(4):1–34
20. Ravasz E, Somera AL, Mongru DA (2002) Hierarchical organization of modularity in metabolic networks. Science 297(5586):1553–1555
21. Leicht EA, Holme P, Newman MEJ (2006) Vertex similarity in networks. Phys Rev E 026120:73

22. Barabasi A-L, Albert R. (1999) Emergence of scaling in random networks. Science 286(5439):509–512
23. Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Netw 25(3):211–230
24. ZHOU T, LÜ L, ZHANG Y C (2009) Predicting missing links via local information. Eur Phys J B 71(4):623–630
25. KATZ L (1953) A new status index derived from sociometric analysis. Psychometrika 18(1):39–43
26. Chebotarev P, Shamis EV (1997) The matrix-forest theorem and measuring relations in small social groups. Autom Remote Control 58:1505
27. Lü L, Jin C-H, Zhou T (2009) Similarity index based on local paths for link prediction of complex networks. Phys Rev E 80:046122
28. Liu W, Lü L (2010) Link prediction based on local random walk. EPL (Europhys Lett) 89(5):58007
29. Klein DJ, Randic M (1993) Resistance distance. J Math Chem 12(1):81–95
30. Fouss F, Pirotte A, Renders JM (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans Knowl Data Eng 19(3):355–369
31. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Comput Netw ISDN Syst 30(1):107–117
32. Jeh G, Widom J (2002) SimRank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 538–543
33. Lü L, Jin C-H, Zhou T (2009) Similarity index based on local paths for link prediction of complex networks. Phys Rev E - Stat Nonlinear Soft Matter Phys 80(4):046122
34. Liu W-P, Lü L (2010) Link prediction based on local random walk. Eur Phys Lett 89:58007
35. RAO J, WU B, Yu-Xiao D (2012) Parallel link prediction in complex network using map reduce. J Softw 23(12):3175–3186
36. Yu-xiao D, Qing KE, WU B (2011) Link prediction based on node similarity. Comput Sci 38(7):162–164
37. Clauset A, Moore C, Newman MEJ (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453:98
38. White HC, Boorman SA, Breiger RL (1976) Social structure from multiple networks I: block models of roles and positions. Am J Sociol 81:730
39. Doreian P, Batagelj V, Ferligoj A (2005) Generalized blockmodeling. Cambridge University Press
40. Airoldi EM, Blei DM, Fienberg SE, Xing XP (2008) Mixed-membership stochastic block models. J Mach Learn Res 9:1981
41. Fire M, Tenenboim L, Lesser O (2011) Link prediction in social networks using computationally efficient topological features. In: 2011 IEEE third international conference on privacy, security, risk and trust (passat), and 2011 IEEE third international conference on social computing (socialcom). IEEE, pp 73–80
42. Friedman N, Getoor L, Koller D (1999) Learning probabilistic relational models. IJCAI 99:1300–1309
43. Heckerman D, Meek C, Koller D (2007), Probabilistic entity-relationship models, PRMs, and plate models. Introduction to statistical relational learning
44. Yu K, Chu W, Yu S (2006) Stochastic relational models for discriminative link prediction. NIPS, pp 1553–1560
45. Sarukkai RR (2000) Link prediction and path analysis using Markov chains. Comput Netw 33(1):377–386
46. Kashima H, Abe N (2006) A parameterized probabilistic model of network evolution for supervised link prediction. In: IEEE sixth international conference on Data Mining, 2006. ICDM'06, pp 340–349
47. Rodrigues H, Prudencio RBC (2011) Supervised link prediction in weighted networks. In: The 2011 International Joint Conference on neural networks (IJCNN). IEEE, pp 2281–2288
48. Raymond R, Kashima H (2010) Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. Machine learning and knowledge discovery in databases. Springer, Berlin / Heidelberg, pp 131–147
49. Rossetti G, Berlingerio M, Giannotti F (2011) Scalable link prediction on multidimensional networks. In: 11th international conference data mining workshops (ICDMW), 2011. IEEE, pp 979–986
50. Song HH, Cho TW, Dave V (2009) Scalable proximity estimation and link prediction in online social networks. In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, pp 322–335
51. Miller KT, Griffiths TL, Jordan MI (2009) Nonparametric latent feature models for link prediction. In: NIPS, vol 9, pp 1276–1284
52. Pieter BTMFW, Koller AD (2003) Link prediction in relational data[J]
53. Menon AK, Elkan C (2011) Link prediction via matrix factorization. Machine learning and knowledge discovery in databases, Berlin / Heidelberg, pp 437–452
54. Scripps J, Tan PN, Chen F (2009) A matrix alignment approach for collective classification. In: International conference on advances in social network analysis and mining, 2009. ASONAM'09. IEEE, pp 155–159
55. Backstrom L, Leskovec J (2011). ACM, pp 635–644
56. Yin Z, Gupta M, Weninger T (2010) LINKREC: a unified framework for link recommendation with user attributes and graph structure. In: Proceedings of the 19th international conference on World wide web. ACM, pp 1211–1212
57. Yin Z, Gupta M, Weninger T (2010) A unified framework for link recommendation using random walks. 2010 International conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 152–159
58. Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization: artificial ants as a computational intelligence technique. IEEE Comput Intell Mag 11:28–39
59. Blum C (2009) Ant Colony Optimization, Proceedings of The 2009 Genetic and Evolutionary Computation Conference, Montreal, pp 2835–2851
60. Blum C, Ant colony optimization: Introduction and recent trends (2005) Phys Life Rev 2:353–373
61. Wu J, Abbas-Turki A, El Moudni A (2012) Cooperative driving: an ant colony system for autonomous intersection management. Appl Intell 37(2):207–222
62. Lu M, Xu B, Sheng A (2014) Modeling analysis of ant system with multiple tasks and its application to spatially adjacent cell state estimate. Appl Intell:1–17
63. Khan SA, Engelbrecht AP (2012) A fuzzy particle swarm optimization algorithm for computer communication network topology design. Appl Intell 36(1):161–177
64. Rivero J, Cuadra D, Calle J (2012) Using the ACO algorithm for path searches in social networks. Appl Intell 36(4):899–917
65. Zhang N, Feng ZR, Ke LJ (2011) Guidance-solution based ant colony optimization for satellite control resource scheduling problem. Appl Intell 35(3):436–444
66. Shuang B, Chen J, Li Z (2011) Study on hybrid PS-ACO algorithm. Appl Intell 34(1):64–73
67. Merkle D, Middendorf M (2003) Ant colony optimization with global pheromone evaluation for scheduling a single machine. Appl Intell 18(1):105–111

68. Freeman L (1977) A set of measures of centrality based on betweenness. Sociometry 40:35–41
69. http://www.linkprediction.org/index.php/link/resource/data
70. Latora V, Marchiori M (2001) Efficient behavior of small-world networks. Phys Rev Lett 67:198701–198704
71. Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. Nature 393(6684):440–442
72. Newman MEJ (2002) Assortative mixing in networks. Phys Rev Lett 89:208701–208705
73. http://www.informatik.uni-trier.de/~ley/db/