Trust Inference Path Search Combining Community Detection and Ant Colony Optimization

Yao Ma, Hongwei Lu, Zaobin Gan*, and Yizhu Zhao

Huazhong University of Science and Technology, Wuhan, China
mayaobox@qq.com, {luhw,zgan,missbamboo}@hust.edu.cn

Abstract. Finding trust inference paths for unfamiliar users in online social networks is a fundamental work of trust evaluation. Most existing trust inference path search approaches apply classical brute-force graph search algorithms, which leads to high computation costs. To solve this issue, we propose a trust inference path search approach combining community detection and ant colony optimization. First, the singular value decomposition signs method is utilized to process the trust relationship matrix in order to discovery the trust communities. Then, by taking the communities as different colonies, we use the ant colony optimization to find the optimal trust inference path along which the witness has the maximum deduced referral belief. The released pheromones in previous trust inference path searches help subsequent searches to reuse previous experience and save path search costs. Comparative experiments show that the proposed trust inference path search approach outperforms the existing ones on path search efficiency and trust inference accuracy.

Keywords: Trust inference path search, community detection, ant colony optimization, social network analysis.

1 Introduction

Popularized Online Social Network (OSN) applications, especially the Social Network Sites (SNSs), provide people with great convenience for information sharing, collaborating and interacting. Trust in online social networks plays an important role for users to make trusted decisions when facing unfamiliar copartners or environments. For example, the "Web of Trust" in Epinions.com builds a community of trusted members for users and makes personalized recommendations. It is beneficial for the buyers to evaluate the trustworthiness of the unfamiliar recommenders, sellers or service providers before making purchase decisions. Given a pair of users who have no interaction experience, trust transitivity based trust inference can deduce the trust opinion between them by applying trust discounting and consensus operations to the trust propagation

* Corresponding author.

F. Li et al. (Eds.): WAIM 2014, LNCS 8485, pp. 687–698, 2014.

paths [7]. In the large-scale OSN, there are a great number of such paths and how to efficiently find appropriate trust inference paths emerges as a question.

Most existing trust inference path search approaches ignore the structure characteristics of the trust network, which makes the path search blind and cost expensive. Ant Colony Optimization (ACO)[1], inspired by the pheromone trail laying and following behavior of some ant species, is a metaheuristic for solving hard combinatorial optimization problems. Similar to the ants that find the shortest paths connecting to the food, users in the ONS also want to find the most reliable trust inference path connecting to the target participants. However, the users flock with shared interests, preferences or opinions etc. and they compose the different communities. The structure of social networks attracts much attention and the research of community detection in social networks derives [3]. The ideal communities in trust networks should be like this: the members in the same community trust each other and the distrust relationships do not appear in one community. We try to detect the communities in the trust network by trust and distrust relationships and cluster users as different colonies for ant colony optimization. Since users in one community have similar trusting and distrusting preferences, the clustering can help the path search have a clear sense of direction and tend to find trustworthy recommenders. Moreover, existing trust inference path search approaches do not accumulate and reuse search experience, so they cannot reduce the path search costs even for repeated path search requests. It also inspires us to utilize the ACO to solve this issue.

The main contribution of this paper includes a trust community detection method and an ACO based Trust Inference Path Search algorithm (ACO-TIPS). The proposed trust community detection method utilizes Singular Value Decomposition (SVD) signs method [2] to process the trust relationship matrix to detect communities in the trust network and label them as colonies. For a given pair of source and target users, the ACO utilizes the pheromones of colonies close to the source user as experience information and the distances between the candidates and the target user in the singular vector space as heuristic information to find the optimal or near-optimal trust inference path solutions.

2 Related Work

Classical brute-force graph search based approaches are the mainstream in the field of the trust inference path search. Jøsang et al. [7] used the Depth First Search algorithm to find all the possible paths connecting the source and target participants in the trust network. Similarly, Hang et al. [5] proposed CertProp with three search strategies (*shortest, fixed, selection*) to find the paths for trust evaluation. The trust inference requires to find the best path connecting to each witness. Although the search algorithm is not detailed, the Depth First Search algorithm is obviously preferred to find all the possible paths. TidalTrust [4] utilizes a modified Breadth First Search to first find the trust inference path with the minimum depth and continue to find any other paths at the minimum depth. The trust inference paths with the maximum strength will be used in the calculation for inferring trust. In [13], the Breadth First Search algorithm is also used to find the trust propagation paths within the minimum depth for further trust evaluations. Ma et al. [11] proposed a bidirectional path search approach based on Dijkstra's algorithm to find the trust inference path with the minimum deduced uncertainty.

There are also stochastic trust inference path search approaches. TrustWalker [6] performs random walks on the trust network to solve the recommendation issues for cold start users. Repeated random walks take into account both the trust values of the neighbors and the similarities between the target item and the items rated by the neighbors. Thus, it makes a good combination of trust based and collaborative filtering based recommendation. Liu et al. [10] modeled the optimal social trust path selection as the classical Multi-Constrained Optimal Path (MCOP) selection problem and proposed the Heuristic Social Context-Aware trust Network discovery algorithm (H-SCAN) based on the K-Best-First Search. This method shows better performance than Time-To-Live Breadth First Search, Random Walk Search and High Degree Search.

The brute-force search based approaches are computation costly and the path search experience cannot be accumulated and reused for all the methods mentioned above. So, given a pair of source and target participants, repeated requests for the trust path between them will lead to repeated path searches at the same or similar computation cost, unless the previous search results are saved. Obviously, it is infeasible to save such paths for the dynamic large-scale trust networks.

3 Proposed Trust Inference Path Search Approach

3.1 Trust Communities Detection

The trust network can be formally described by a directed graph $G =< V, E >$, where V represents the set of participants and E represents the set of trust relationships. Binary trust relationships (i.e. trust and distrust relationships) are considered in this paper. So, $\forall e_{v_1 \to v_2} \in E, \exists | s_{e_{v_1 \to v_2}} \in \{1, -1\}$ and $v_1, v_2 \in V$. Here $s_{e_{v_1 \to v_2}}$ is the sign of the trust relationship, $s_{e_{v_1 \to v_2}} = 1$ means v_1 trusts v_2, $s_{e_{v_1 \to v_2}} = -1$ means v_1 distrusts v_2.

The trust relationship matrix noted by $T_{|V| \times |V|} = (t_{ij})$ is a $|V| \times |V|$ sparse matrix, where $t_{ij} = s_{e_{v_i \to v_j}}$ and $1 \leq i, j \leq |V|$. This matrix is different with the adjacent matrix because it contains the -1 elements. So it not only describes how participants connect with others but also shows their opinions on trust worthiness. By decomposing this matrix with truncated SVD, we can cluster the participants by how they trust and distrust others and/or how they are trusted and distrusted by others with less dimensions. The decomposed trust relationship matrix with rank k can be represented by:

$$T'_{|V| \times |V|} = U_{|V| \times k} S_{k \times k} V^T_{|V| \times k} \tag{1}$$

Here T' is the best possible rank k approximation to T and $k < rank(T)$. The value of k can be chosen by plotting the descending ordered singular values of T and finding the turning point of the line. Thus, the entries of S are the k dominant singular values and the rows of U and V can be regarded as the coordinates of the participants in the k dimensional spaces.

The *SVD signs* [2] is a clustering method which makes the singular value deposition of the adjacent matrix of the undirected graph and uses the sign patterns of the singular vectors to cluster the entries. In this paper, we apply this method to process the trust relationship matrix T so as to detect trust communities in the trust network. Since the matrix T is asymmetric, clustering methods by rows of U or V have different meanings. If the rows of U that have the same sign patterns on the k dimensions are classified into one cluster, this may lead to up to 2^k clusters. It clusters the participants by how they trust and distrust others. Similarly, the sign patterns of the rows of V are also applicable and this clusters the participants by how they are trusted and distrusted by others.

For examples shown in Fig.1, trust and distrust relationships are distinguished as solid and dotted arrows. The left example shows that A and B both trust V_1, V_3, V_n and distrust V_2, V_{n-1}, and they are probably classified into the same cluster by using rows of U. In the example on the right side of Fig.1, A and B are both trusted by V_1, V_3, V_n and distrusted by V_2, V_{n-1}. Thus, they are probably classified into the same cluster by using rows of V. For simplicity, we only use the sign patterns of the rows of U and ignore the rows of V. Given a pair of participants who have the similar trusted and distrusted participants, they may be classified into the same cluster, or to say, colony.

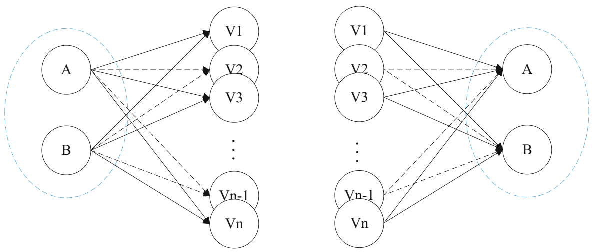


Fig. 1. The meanings of clusterings by rows of U or V

The instance in Fig.2 illustrates that, by building the trust relationship matrix and set $k = 2$, the ten vertices in the trust network are classified into three clusters according to the sign patterns ($(-,+)$, $(+,-)$ and $(-,-)$) of the rows of $U_{10\times 2}$ (on the right side of Fig.2). In this clustering result, the trust relationships lie between vertices in the same cluster and the distrust relationships lie between vertices in different clusters, which satisfies the expectation of ideal trust community. Thus, the SVD sign based trust community detection method is feasible.

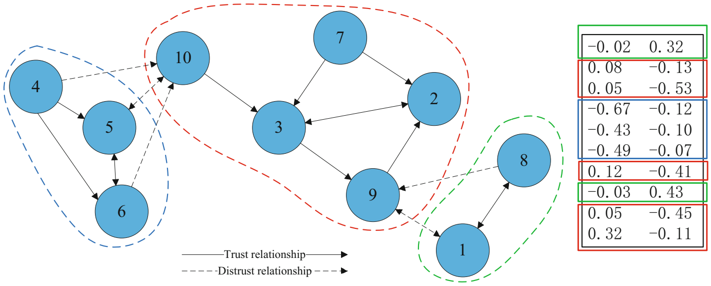


Fig. 2. SVD sign based trust community detection

3.2 ACO with Multiple Colonies

Following the subjective logic proposed by Jøsang et al. [8], we infer the trust relationship of a pair of unfamiliar participants by applying the trust inference path (e.g. $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_n$) with trust discounting operators $\omega_n^{1:\ldots:n-1} = \omega_2^1 \otimes \omega_3^2 \otimes \ldots \otimes \omega_n^{n-1}$, where ω is the *subjective opinion* and $\otimes$ denotes the *trust discounting operator*. The Uncertainty Favoring Discounting operator (noted as $\otimes_1$) and Opposite Belief Favoring Discounting (noted as $\otimes_2$) operator are also introduced in [8]. Here, only the last hop of the path is on *functional trust* and the former ones are on *referral trust*. Modified ACO is utilized in this paper to find appropriate trust inference paths.

Transition possibilities of ants. With each participant labeled by a unique colony identifier, we can obtain the coordinates of the colony centers in the k dimensional singular vector space. Given a pair of source and target participants v_S and v_T, the corresponding colony labels $label(v_S)$ and $label(v_T)$ (belong to $\{label_i | 1 \leq i \leq |\{label_i\}|\}$), and the coordinates of the colony centers $center(label_i)$. For each round of ACO, numbers of ants perform random walks from the source vertex. Each ant chooses its next hop by computing the transition possibilities for the successors of the current vertex. Without loss of generality, given the current vertex v_A and its successors $suc(v_A)$, the transition possibility for the successor v_B ($v_B \in suc(v_A)$) can be obtained by the following equation

$$p_{v_A \rightarrow v_B} = \frac{\tau_{v_A \rightarrow v_B}^{\alpha} \cdot \eta_{v_A \rightarrow v_B}^{\beta}}{\sum\limits_{v_i \in suc(v_A)} \tau_{v_A \rightarrow v_i}^{\alpha} \cdot \eta_{v_A \rightarrow v_i}^{\beta}} \tag{2}$$

where $\tau_{v_A \rightarrow v_B}$ is the total amount of pheromone on $e_{v_A \rightarrow v_B}$ and $\eta_{v_A \rightarrow v_B}$ is the value of heuristic information. Parameters α and β determine the relative influence of the pheromone trails and the heuristic information. Since we consider multiple colonies, the pheromones of similar colonies are also utilized.

$$\tau_{v_A \rightarrow v_B} = \sum_{1 \leq i \leq |\{label_i\}|} \psi_{(v_A \rightarrow v_B, label_i)} \cdot w_{label_i} \tag{3}$$

Here $\psi_{(v_A \to v_B, label_i)}$ is the amount of pheromone on $e_{v_A \to v_B}$ for $label_i$ and w_{label_i} is the weight of the pheromone for $label_i$. This pheromone weight is related to the distances between the source participant and each colony center in the k dimensional space.

$$w_{label_i} = \frac{1/||pos(v_S) - center(label_i)||}{\sum\limits_{1 \le j \le |\{label_j\}|} 1/||pos(v_S) - center(label_j)||} \tag{4}$$

where $pos(v_S)$ are the coordinates of v_S. Moreover, the value of heuristic information is also estimated by the distance between the coordinates of the successors and the target participant.

$$\eta_{v_A \to v_B} = 1/||pos(v_B) - pos(v_T)|| \tag{5}$$

After the computation of transition possibilities for the successors, a random hop can be determined and the current vertex is updated. If the current vertex is the target participant, the ant stops and the path is recorded for further path selection.

Selection of the optimal trust inference path. At the end of each round, all the found paths are compared by computing the deduced referral trust about the last recommender (i.e. the witness) as the quality of the path. Given a connecting path $path_i$ denoted as $[v_{(i,1)}, v_{(i,2)}, \ldots, v_{(i,n_i)}]$, where n_i is the number of vertices along $path_i$, $v_{(i,1)} = v_S$, $v_{(i,n_i)} = v_T$ and $1 \le i \le |\{path_i\}|$, the inferred v_S's opinion about the witness $v_{(i,n_i-1)}$ on referral trust would be $\omega_{(i,n_i-1)}^{(i,1):\ldots:(i,n_i-2)}$. The first element in the round bracket denotes the index of the path and the second element denotes the index of the vertex along this path.

For each round of the ACO search, the optimal trust inference path among all the connecting paths found in this round is the path $path_k$ with the maximum indirect referral belief $b_{(k,n_k-1)}^{(k,1):\ldots:(k,n_k-2)}$. The optimization problem can be formally described as to find $path_k$ that

$$b_{(k,n_k-1)}^{(k,1):\ldots:(k,n_k-2)} = \max_{path_i}\{b_{(i,n_i-1)}^{(i,1):\ldots:(i,n_i-2)}\} \tag{6}$$

where $1 \le k \le |\{path_i\}|$ and $b_{(i,n_i-1)}^{(i,1):\ldots:(i,n_i-2)}$ can be obtained by $\omega_{(i,n_i-1)}^{(i,1):\ldots:(i,n_i-2)} = \omega_{(i,2)}^{(i,1)} \otimes \omega_{(i,3)}^{(i,2)} \otimes \ldots \otimes \omega_{(i,n_i-1)}^{(i,n_i-2)}$ with all hops on referral trust.

After the determination of the optimal trust inference path, we update the pheromones for colony $label(v_S)$ along this path by the following equation and ignore the rest found paths.

$$\psi_{(v_A \to v_B, labelv_S)} = \rho \cdot \psi_{(v_A \to v_B, labelv_S)} + \frac{e}{1 - b_{(k,n_k-1)}^{(k,1):\ldots:(k,n_k-2)}} \tag{7}$$

where $e_{v_A \to v_B}$ is an arbitrary edge of $path_k$, ρ is the evaporation rate of pheromones and e is the reinforcement factor to enhance the pheromones of the best path since the start of the algorithm.

3.3 ACO Based Trust Inference Path Search Algorithm

The overall trust inference path search algorithm first clusters the users by the trust community detection method. Then, by taking the clustered users as different colonies, multiple rounds of ACO are performed to find the optimal or near-optimal trust inference path. The ACO-TIPS algorithm can improve the performance of each round of search gradually by utilizing, releasing and updating pheromones. In order to simplify the trust inference path search problem, only one optimal trust propagation path is found for each search without considering the fusions of multiple paths. The detailed algorithm is described in the Algorithm 1.

Algorithm 1 ACO based Trust inference path search algorithm

Require: Trust network $G = (V, E)$, source and target participants v_S, v_T.
Ensure: $path = [v_S, \ldots, v_T]$ {if the path does not exist, it returns null}.

Get the trust relationship matrix $T_{|V|\times|V|}$ and its singular values.
Determine k, make the truncated SVD of $T_{|V|\times|V|}$ and cluster the participants by the SVD sign method.
while search round $round \leq$ the maximum $round_{max}$ **do**
  Set the current vertex $v_c(ant_i)$ to v_S for each ant_i.
  while current path depth $depth \leq 7$ **do**
    for $ant_i \in \{ant_i | ant_i.state == active\}$ **do**
      if $v_c(ant_i)$ has no successors **then**
        $ant_i.state \leftarrow inactive$
      end if
      Compute the transitive possibilities $p_{v_c(ant_i)\rightarrow v_s(ant_i)}$ by Eq.(2) where $v_s(ant_i)$ belongs to the successors of $v_c(ant_i)$.
      Scale the possibilities of the edges without pheromones by the exploring factor θ and the ones with pheromones by $1 - \theta$.
      Choose one successor $v_s(ant_i)$ randomly and replace the current vertex $v_c(ant_i) \leftarrow v_s(ant_i)$.
      if $v_c(ant_i) == v_T$ **then**
        $ant_i.state \leftarrow inactive$, and $ant_i.found \leftarrow success$
      end if
    end for
    $depth \leftarrow depth + 1$
  end while
  Assemble the paths $\{path_i\}$ passed by ant_i where $ant_i.found == success$ and determine the optimal $path_k$ by Eq.(6).
  if $b_{(k,n_k-1)}^{(k,1):\ldots:(k,n_k-2)} > b_{best}(v_S, v_T)$ **then**
    $b_{best}(v_S, v_T) \leftarrow b_{(k,n_k-1)}^{(k,1):\ldots:(k,n_k-2)}$, and $e \leftarrow e_{strength}$
  end if
  Update $\psi_{(v_A\rightarrow v_B, labelv_S)}$ by Eq.(7) where $e_{v_A\rightarrow v_B}$ belongs to the edges of $path_k$.

  $round = round + 1$
end while
return $path_k$

In this algorithm, we address the *exploit-vs-explore* dilemma by introducing an exploring factor θ ($0 < \theta < 1$). When computing the the transitive possibilities, the edges without pheromones share the possibility that equals to θ and the ones with pheromones share the possibility that equals to $1 - \theta$. This mechanism is disabled when all the edges from the current vertex do or do not have pheromones. It protects the algorithm from the premature convergence at the initial rounds of searches and makes the subsequent searches able to find better paths. Moreover, the found path with the maximum deduced referral belief since the start of algorithm is rewarded by a strengthened reinforcement factor $e_{strength}$. This can help the pheromones of the global best found path avoid to be submerged. Generally, the number of ants starts with a great number and then decreases gradually. After several rounds of trust inference path searches, it can find the trust inference path with high quality by one round of search with a small number of ants.

If we denote the number of ants as m and the maximum path depth as d (in this paper $d = 7$), in the worst case, the times of vertex scan would be $m \cdot (d-1)$ in one round and the time complexity for one round of search is $O(m)$.

4 Experiments and Analysis

In this section, experiments are carried out on the Epinions data set to compare the performance of the proposed ACO-TIPS approach with the representative TidalTrust[4], CertProp(Sel.)[5] and H-SCAN[10] approaches on the path search efficiency and the applicability to the trust inference.

4.1 Data Set Description

Epinions is a consumer reviews web site that helps people make informed buying decisions by valuable consumer insight and personalized recommendations. The extended Epinions data set released by [12] is available at trustlet.org which describes the trust and distrust relationships among users and their ratings on other user's articles. The sampling method based on random walk introduced in [9] is utilized to scale down the original data set and the data set for experiments contains 33036 users who issued 84141 trust and distrust statements.

4.2 Methodology and Metrics

First, the subjective opinion between arbitrary pair of users should be obtained. Given a pair of users (v_A and v_B), we count the number of the source user's ratings on target user's articles as the number of observations (noted as $n_{rating} \geq 0$) and get the mean rating (noted as $m_{rating} \in [1, 5]$). Then, $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$ can be obtained by:

$$\begin{cases} b_B^A = (m_{rating} - 1) \cdot (1 - u_B^A)/4 \\ d_B^A = (5 - m_{rating}) \cdot (1 - u_B^A)/4 \\ u_B^A = 2/(2 + n_{rating}) \\ a_B^A = 0.5 \end{cases} \tag{8}$$

Before the trust inference path search, the users are clustered by the trust community detection method with the left singular vector. The trust relationship matrix is reordered so that the rows corresponding to the users within the same cluster are together and the same reordering is also applied to the columns. The reordered trust relationship matrix is plotted in Fig.3. The blue dots represent trust relationships and the red ones represent distrust relationships. The transverse lines are the borders of clusters. In this figure, we can find that there are clusters where the users mainly trust each other in the same cluster and overall distrust the users in some other clusters.

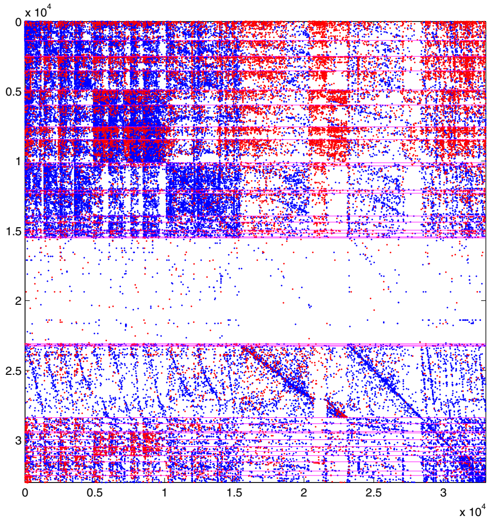


Fig. 3. Reordered trust relationship matrix after clustering

In order to validate the performance of ACO-TIPS in terms of trust inference path search efficiency and trust inference accuracy, we use a standard Leave-one-out evaluation technique with 500 randomly chosen sample user pairs. Before the trust inference path search starts, the original trust statement is masked. All the comparative path search approaches are performed to find the trust inference path connecting the source user and the target user. When the path searches terminate, the number of the found paths and the scanned vertices per path are recorded as the metrics for the trust inference path search efficiency. The approach with higher path discovery rate and lower scanned vertices per path shows better path search performance. Then, the trust inference is performed by applying the found trust inference paths with trust discount operators $\otimes_1$ and $\otimes_2$ respectively. The deduced subjective opinion is compared with the original subjective opinion obtained by the ground truth in terms of P-error and B-error introduced in [5] as the metrics for trust inference accuracy. The lower errors in trust inference reflect better applicability of the path to trust inference.

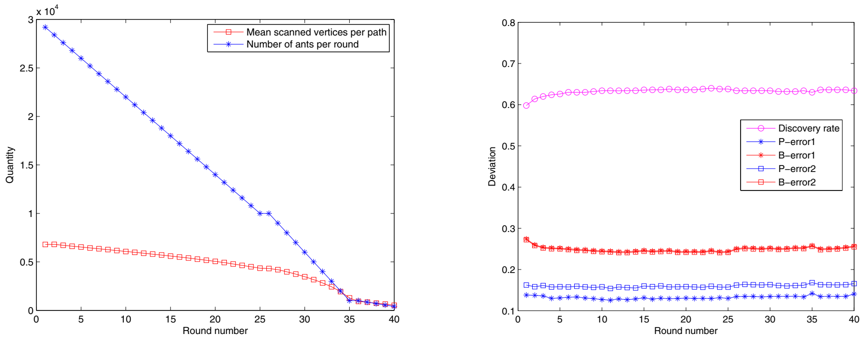
In order to give a computation bound in the experiments, the maximum number of scanned vertices is set to 5000 and the maximum depth of path is set to

7 according to the small world theory. For TidalTrust, CertProp(Sel.) and H-SCAN, repeated searches for the same user pair yield the same or similar results and costs, and thus they are performed only once. The proposed ACO-TIPS is performed in 40 rounds with decreased number of ants. We choose the mean performance of the last 5 rounds to make comparisons.

4.3 Results and Analysis

Experiment results of the ACO-TIPS are illustrated in Fig.4a and Fig.4b. As we decrease the number of ants for each round, the number of scanned vertices per path also decreases. However, the path discovery rate (the number of searches that find at least one path divided by the number of samples), the P-errors and B-errors for $\otimes_1$ and $\otimes_2$ are all floating at a stable level.

The mean performance of the last 5 rounds of ACO-TIPS are compared with the performance of TidalTrust, CertProp(Sel.) and H-SCAN in Table.1. On trust inference path search, ACO-TIPS reaches the highest discovery rate 63.56% (9.59% higher than TidalTrust's 58%) and the lowest mean scanned vertices per path is 735.538 (38.9% less than TidalTrust's 1204). It means that ACO-TIPS can find the trust inference paths for the most number of samples with the lowest mean search cost. On trust inference, the P-errors and B-errors of ACO-TIPS for $\otimes_1$ and $\otimes_2$ are the lowest ones among those of the four trust inference path search approaches (P-error1, B-error1, P-error2 and B-error2 of ACO-TIPS are 27.25%, 21.61%, 15.55% and 21.88% lower than those of TidalTrust respectively). This implies that the trust inference with the path found by ACO-TIPS reaches higher trust inference accuracy.



(a) Path search costs for ACO-TIPS **(b)** Discovery rate, P-errors and B-errors of ACO-TIPS

Fig. 4. Experimental results of ACO-TIPS

Thanks to the pheromones released by the previous rounds of searches in ACO-TIPS, the subsequent rounds of searches can easily discover the trust inference paths with much less costs than previous ones. It implies that ACO-TIPS

Table 1. Performance comparisons of TidalTrust, CertProp(Sel.), H-SCAN and ACO-TIPS

Term	TidalTrust	CertProp(Sel.)	H-SCAN	ACO-TIPS
Discovery Rate	0.58	0.452	0.478	**0.6356**
Scanned vertices per path	1204	4963.02	793.2	**735.54**
P-error1	0.1868	0.3323	0.2109	**0.1359**
B-error1	0.3225	0.5696	0.3799	**0.2528**
P-error2	0.1936	0.3784	0.2392	**0.1635**
B-error2	0.3209	0.567	0.3785	**0.2507**

can accumulate and reuse former path search experience to speed up the path search. In ACO-TIPS, only the optimal path of each round is qualified to add pheromones. Furthermore, the optimal path since the start of the algorithm even gets additional pheromones. Therefore, ACO-TIPS tends to find better and better trust inference path after rounds of searches and make the trust inference more and more accurate until it finds the optimal path.

5 Conclusions and Future Work

In this paper, we propose a trust inference path search approach combining the trust community detection and the ACO with multiple colonies. The SVD sign clustering method is applied to detect trust communities according to how users trust and distrust others. Then, the trust communities are regarded as colonies in the ACO. By selecting the found path with the maximum deduced referral belief, the ACO-TIPS can accumulate and reuse path search experience to find the optimal or near-optimal trust inference path efficiently. This optimal path can connect the source user to the target user with the most trustworthy witness, which leads to accurate trust inference results.

Since the trust network in the experiments is static, the so-called “exploit” or “explore” period is transient. In fact, the evaporation of the pheromones and stochastic routing can make the ACO-TIPS applicable to the dynamic trust network, which needs to be further validated with suitable experiment environments. Also, the idea of accumulating and reusing experience in trust inference path search can be applied to the trust inference based recommendations for better accuracy and lower costs.

Acknowledgments. This research is funded by the National Nature Science Foundation of China (No. 61272406) and the Fundamental Research Funds for the Central Universities (HUST:2013TS101).

References

1. Dorigo, M., Stttzle, T.: Ant colony optimization: Overview and recent advances. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 146, pp. 227–263. Springer, US (2010)
2. Douglas, E.P.: Computing and applying trust in web-based social networks. Master's thesis, College of Charleston (2008)
3. Fortunato, S.: Community detection in graphs. Physics Reports 486(3), 75–174 (2010)
4. Golbeck, J.A.: Computing and applying trust in web-based social networks. Ph.D. thesis, University of Maryland College Park (2005)
5. Hang, C., Wang, Y., Singh, M.P.: Operators for propagating trust and their evaluation in social networks. In: Proc. of the 8th Intl. Conf. on Autonomous Agents and Multiagent Systems, pp. 1025–1032 (2009)
6. Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: Proc. of the 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pp. 397–406. ACM (2009)
7. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: Proc. of the 29th Australasian Computer Science Conf., pp. 85–94. Australian Computer Society, Inc. (2006)
8. Jøsang, A., Marsh, S., Pope, S.: Exploring different types of trust propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F., et al. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)
9. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proc. of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pp. 631–636. ACM (2006)
10. Liu, G., Wang, Y., Orgun, M.A., Liu, H.: Discovering trust networks for the selection of trustworthy service providers in complex contextual social networks. In: Proc. of IEEE 19th Intl. Conf. on Web Services (ICWS 2012), pp. 384–391. IEEE (2012)
11. Ma, Y., Lu, H., Gan, Z.: Trust inference path search with minimum uncertainty for e-commerce. In: Proc. of the 10th Conf. on Web Information Systems and Applications, pp. 133–137. IEEE (2013)
12. Massa, P., Avesani, P.: Trust-aware bootstrapping of recommender systems. In: Proc. of ECAI Workshop on Recommender Systems, pp. 29–33. IOS (2006)
13. Shekarpour, S., Katebi, S.: Modeling and evaluation of trust with an extension in semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 8(1), 26–36 (2010)