A Local Trust Inferring Algorithm based on Reinforcement Learning DoubleDQN in Online Social Networks

Xiaodong Zhuang Xiangrong Tong *
School of Computer and Control Engineering, Yantai University
Yantai,Shandong 264005, China

Abstract—**The development of online social network has greatly promoted the interaction between users. Trust plays an important role in social activities of social network, which can effectively avoid the risk from unreliable users. In fact, most users have no direct interaction, i.e., there is no direct connection in the social network, so the indirect trust of the target user can only be evaluated by the friends who contact indirectly. The propagation and aggregation methods of trust affect the results of trust evaluation to a great extent. The existing aggregation methods generally have the problem of low prediction accuracy. In addition, how to find a reliable path to propagate trust is also a major challenge. This paper proposes a DoubleDQNTrust(DDQNTrust) algorithm based on reinforcement learning DoubleDQN to find reliable trust paths. Secondly, based on standard collaborative filtering and considering the similarity between users, a new aggregation method is proposed. The experimental results on Filmtrust online social network data set show that DDQNTrust algorithm can effectively find reliable trust paths, and can evaluate trust with high prediction accuracy.**

Keywords-online social network; trust propagation; trust aggregation; reinforcement learning; trust inference

I. INTRODUCTION

Nowadays, online social networks play a very important role in people's daily life [1], such as Facebook, twitter, MicroBlog, etc. These social networks connect people with the same interests and provide great convenience for interpersonal communication. Due to the complexity and variability of social networks [2], there will be many problems in the interaction process, such as junk information, risk users, etc. Therefore, how to correctly identify whether the interactive users are trusted has become an important challenge.

Trust inference in social networks has received extensive attention in recent years [3]. Many scholars use propagation algorithm to propagate trust along the path and use aggregation algorithm to evaluate the final trust in social networks. TidalTrust [4] considers the shortest and strongest paths to propagate trust and uses weighted mean aggregation method to aggregate multiple shortest paths. SWTrust [5] uses the "small world" theory and adjustable width first search algorithm to find the trust path, and infers the trust value between users according to the relevant topics. However, most studies limit the depth and width of path search, which ignores many valuable path information and is still very time-consuming.

Kim and Song [6] show that using all trust paths can achieve higher prediction accuracy than using the shortest path. They point out that reinforcement learning is particularly suitable for multi-step decision-making problems, and use classical Q-learning algorithm to discover trust paths to propagate trust. Ghavipour *et al.* [7] use distributed learning automata to find reliable trust paths. Since the methods of Q-learning and learning automata need to use Q table and action probability vector to store corresponding Q value and action probability, when dealing with complex large-scale network, it will consume a large amount of storage space. Therefore, the above two methods are only suitable for dealing with small-scale problems. Ziegler *et al.* [8] find that trust between users is closely related to interest similarity. Mao *et al.* [9] show that trust inference based on user interest similarity can improve prediction accuracy.

In order to solve the problem that Q-learning consumes a lot of storage space in dealing with large-scale networks, we propose a full path search DDQNTrust algorithm based on reinforcement learning Double DQN to find reliable trust paths. Aiming at the problems of path dependence[10] and low prediction accuracy in existing aggregation methods, we propose a trust aggregation method based on standard collaborative filtering and user's interest similarity.

In our algorithm, we discover the trust path from the source node to the neighbor node of the target node based on our DDQNTrust algorithm, and then use our proposed aggregation method to aggregate the reliable trust values provided by the direct neighbors to evaluate the trust value of the source node to the target node.

The main contributions of this paper are summarized as follows:

1. DDQNTrust algorithm based on DoubleDQN is proposed to find reliable trust path.
2. Considering the interest similarity between users, a new aggregation algorithm is proposed according to the standard collaborative filtering algorithm, which effectively solves the path dependence problem in trust inference.
3. We conduct several experiments on real Filmtrust datasets. The experimental results show that DDQNTrust algorithm can accurately evaluate the trust value between two users.

978-0-7381-0545-1/20/$31.00 ©2020 IEEE

The rest of this paper is organized as follows. Section 2 discusses the related work and the background of reinforcement learning. Section 3 introduces the premieraries. Section 4 defines our problem and section 5 proposes our algorithm for trust reasoning. The relevant experimental analysis is reported in section 6. Finally, Section 7 summarizes our work and provides direction for future work.

II. RELATED WORKS

In this section, we introduce the concepts of trust and trust inference. Subsequently,the related works of trust inference and background of reinforcement learning are briefly surveyed here.

A. Trust and Trust inference

Trust is an interdisciplinary subject, such as sociology, psychology, economics and computer science. Therefore, trust has different definitions in different disciplines. Recently, chojin summarizes trust as "Trust refers to the subjective belief that the trustee will show reliable behavior in order to maximize the benefits of the trustor under the given uncertainty based on the cognitive assessment of the trustee's past experience" [11]. Trust can measure one person how trustworthy to another, so trust plays a very important role in social networks [12].

Trust inference in social networks aims to predict the trust between one user and another without direct interaction experience. People's indirect trust is usually inferred from the trusted interaction between their common friends in social networks. For example, if Alice and Bob are two persons who do not know each other, but they both have a common friend Cathy, then Alice can infer the trust value of Alcie to Bob based on the previous interaction between Bob and Cathy. This process is often called trust inference.

B. Related studies for trust inference

Trust inference based on path search has been widely studied. Golbeck [4] proposes TidalTrust to infer two users who are not directly connected in social network, and finds that the prediction accuracy of using the shortest and strongest trust path is higher. TidalTrust first finds all the shortest path with high trust value in the direct neighbor path from the source node to the target node, and then evaluates the trust value of the source node to the target node by weighted mean of multiple shortest and strongest paths. The algorithm will be seriously affected by the sparsity of trust network. When the trust network is too sparse, it will be difficult to find the path with high trust. Secondly, only using the shortest path will ignore many valuable paths, which will affect the accuracy of prediction.

Kim and Song [6] propose a trust inference model using reinforcement learning method for trust propagation, and compared the accuracy of four different propagation aggregation methods under the shortest paths and all paths. In order to find reliable trust paths, Ghavipour and Meybodi [7] proposed a heuristic algorithm DLATrust based on learning automata, and used the improved collaborative filtering aggregation strategy to infer the trust value between users. Based on this, Ghavipour *et al.* [13] propose DyTrust to conduct dynamic trust inference based on the dynamic change of trust value in the trust propagate process.

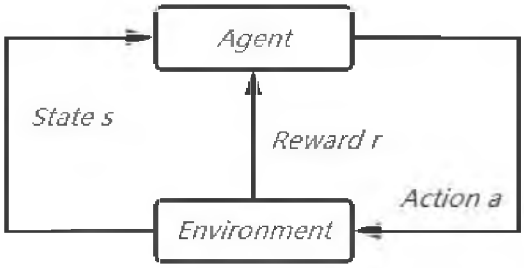

Figure 1. The basic learning model of reinforcement learning.

In addition to trust relationship and topology, other user information, such as interests, topic background [14] and user interaction history can also improve the performance of trust inference. SWTrust proposed by Jiang *et al.* [15] evaluates the social distance between users based on the rating information of product categories, and finds trust paths based on the social distance between users. Mao *et al.* [9] propose three different trust inference strategies based on the topic of interest and topological structure between users. Different weights are set according to the frequency of users appearing in different topics, and the propagation ability of the node is measured according to the in degree and the out degree of the node. The above methods based on user's interest topic can obviously improve the performance of trust inference.

C. Background of reinforcement learning

Reinforcement learning is a kind of learning algorithm that interacts with unknown environment and obtains maximum rewards to achieve specific goals, and has been successfully used in path planning [16] and recommendation system [17].

The basic idea of reinforcement learning is that in the process of interaction with the environment, agents constantly adjust their own strategies to achieve the best decision according to the rewards obtained from the environmental feedback. The learning process is described in Fig. 1.

Firstly, the agent observes the current state s_t and selects action a_t from action space A, the environment feedback the corresponding reward r_{t+1} according to the action chosen by the agent, and transfer to the new state s_{t+1}. the agent adjusts its strategy according to the reward and makes a new decision for the new state. The goal of reinforcement learning is to find an optimal strategy $\pi^* = argmax_\pi E_\pi\{\sum_{k=0}^{\infty}\gamma^k r_{t+k}\}$ (where γ is a discount rate and $0 \leq \gamma < 1$), so that agents can obtain the maximum long-term cumulative reward in any state and any time step.

Q-learning [18] is one of the most popular model-free online learning algorithms. The Belman equation is transformed into an iterative approximation process, and the Q function is updated by the following rules:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

where $\alpha \in (0,1)$ is the learning rate, and $\gamma \in (0,1)$ is the discount factor. Due to Q-learning uses q-table to store q-value, it is only suitable for small-scale problems. When the processing data is too large, it will consume a lot of storage space.

Deep Q Network(DQN)[19] is a new algorithm combining reinforcement learning and deep learning based on Q-learning. DQN uses two neural networks MainNet and TargetNet with the same structure to estimate the function value. MainNet is used to estimate the value of each action and select the final action according to the strategy, the environment returns the corresponding reward value according to the selected action. TargetNet is used to estimate the actual reward value. Finally, according to the idea of Belman equation, the loss function is calculated to update the weight parameters of MainNet network.

$$L(\omega)=E[(r_t+\gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1};\omega^-)-Q(s_t,a_t;\omega))^2] \quad (2)$$

where γ is the discount factor, ω is the network parameter of mainnet, and ω^- is the network parameter of TargetNet. DQN uses neural network to solve the problem of storage space effectively, but there is still the problem of over estimation. In the process of trust inference, overestimation will seriously affect the prediction results.

Double DQN [20] uses the same network based on DQN, decouples the selection and measurement of actions, selects actions in the current MainNet network, and calculates the Q-value in TargetNet with the selected actions. This method can effectively solve the problem of over estimation.

III. Preliminaries

Standard collaborative filtering is to predict a user's rating of a target project based on people with the same interests. Ghavipour *et al.* [7] extend it to trust inference, and predicted the trust value from source user u_s to target user u_t according to the following formula.

$$\tau_{st}^{'}=\overline{\tau}_s+\frac{\sum_{u_k\in Nei_t}W_k(\tau_{kt}-\overline{\tau_k})}{\sum_{u_k\in Nei_t}W_k} \quad (3)$$

where $\overline{\tau}_k$ is the average of trust values provided by u_k. Using the above equation to aggregate direct trust, the predicted results rely on the reliability of the direct trust values provided by all direct neighbors of the target node, and there is still a path dependence. In order to overcome this problem , Ghavipour *et al* modify the above equation as follows:

$$\tau_{st}^{'}=\overline{\tau}_s+\frac{\sum_{u_j\in Nei_t}W_j*(\tau_{jt}-\overline{\tau_j})}{|Nei_t|Max_\tau} \quad (4)$$

where Max_τ represents the maximum trust value provided by the target user's direct neighbor.

There is a positive correlation between trust and user interest similarity. Pearson correlation coefficient is widely used as follows.

$$sim(u_j,u_t)=\frac{\sum_{t_k\in CI_{jt}}(R_{jk}-\overline{R_j})\times(R_{tk}-\overline{R_t})}{\sqrt{\sum_{t_k\in CI_{jt}}(R_{jk}-\overline{R_j})^2}\times\sqrt{\sum_{t_k\in CI_{jt}}(R_{tk}-\overline{R_t})^2}} \quad (5)$$

IV. Problem definition

Given a trust network *G(V, E)* with the set of nodes *V* and the edges' set *E*. Every edge e_{ij} in trust network represents the trust relationship from node v_i to node v_j, and the weight of edge τ_{ij} is the trust value of node v_i to v_j, where $\tau_{ij} \in [0,1]$, where 0 refers to total distrust, 1 refers to full trust. Given source node v_s and target node v_t. Our goal is to infer the trust value of v_s to v_t through trust propagation and trust aggregation.

In our algorithm, we first find the trust path from source node v_s to all neighbor nodes Nei_t directly connected with target node v_t, and determine the reliability of trust value provided by the direct neighbor of target node. The path strength obtained from the propagation function is considered as the reliability of the direct neighbor trust value. Finally, we use aggregation function to aggregate multiple trust values from direct neighbors to calculate the estimated value $\tau_{st}^{'}$.

V. Trust inference algorithm: DDQNTrust

This section will introduce the details of trust inference algorithm based on Double DQN. There are three steps: pre-processing, discovering reliable trust path and aggregating direct trust. The trust propagation algorithm DDQNTrust as follows.

Algorithm DDQNTrust : Trust inference based Double DQN

Input Trust network *G(V,E)*, Source user u_s, Target user u_t, a minimum trust threshold λ, *Neighbors(target user* u_t *) Nei*$_t$

Output Final estimated trust $\tau_{st}^{'}$

1. **Begin**
2. **Phase 1. Discovering reliable trust path**
3. **Repeat**
4. Training path samples(u_s, a_j, r, u_k) into memory pool
5. **Until** (Reach maximum training times)
6. **Repeat**
7. **Repeat**
8. Choose next trusting user u_k in *A(u$_i$) using* ε - greedy policy
9. **Until** (find u_k in Nei_t)
10. Path strength W_k ← min trust in current path
11. Max_w ← max (W_k)
12. **Until** (All neighbor users Nei_t are visited)
13. **Phase 2. Aggregating direct trust**
14. Compute the final estimated trust $\tau_{st}^{'}$ based on our aggregation function
15. **End**

A. Pre-processing

In order to discover trust path in reinforcement learning model, we need to define state, action and reward function of path selection strategy.

Definition 1(State). Given a trust network *G(V,E),* we define each node v_i (i.e. each user) as a state, and state set *U* is defined as:

$$U = \{u_1, u_2, \ldots, u_n\} \tag{6}$$

where *U* is the set of all users and ***n*** is the number of users in the trust graph. In the following, we use $u_k \in U$ instead of nodes to represent state.

Definition 2(Action). In the trust network *G(V,E),* we define the action as selecting the next trusted user from the current user. Only when the trust value of the user is higher than the threshold λ, it will be considered as a candidate action. Action set $A(u_i)$ is defined as:

$$A(\mathrm{u}_i) = \{a_1, a_2, \ldots, a_m\}, \forall_j \tau_{ij} > \lambda, j = 1, 2, \ldots, m \tag{7}$$

where $A(\mathrm{u}_i)$ is the candidate trust user set of user u_i, τ_{ij} is the trust value of u_i to u_j, and λ is the trust threshold.

Definition 3(Reward function). Based on many experiments and previous experience, when r is set to 100, the experimental results are the best. So in the process of path search, if a new user has been visited in the current path or ends at the final user, i.e., the user has no candidate trusted users, it will receive -100 as a penalty. Otherwise it will receive 100 as an immediate reward. Therefore, the reward function is defined as:

$$r = \begin{cases} -100 & \text{if } u_j \text{ is final user} \\ -100 & \text{if } u_j \text{ visited in current path} \\ 100 & \text{else} \end{cases} \tag{8}$$

B. Discovering reliable trust path

The purpose of this section is to find the reliable trust paths of all the direct neighbors Nei_t from the source node v_s to the target node v_t.

Step 1. Path sampling

The first choice from the source user u_s selects the next user u_t randomly, according to the reward function return reward value. In the follow-up action, the next user is selected by using the ε - greedy policy and the repeat the process. The current state u_s, the selected action a_j, the next state u_k and the reward are stored in the memory pool as the network input.

Step 2. Network training and learning to discover trust path

The learning process of DoubleDQN uses the data in memory pool to select the path in MainNet, and measures the Q value of selecting the path in TargetNet. MainNet is updated according to the difference between MainNet and TargetNet. After repeated iterative learning, the direct neighbor $u_n \in Nei_t$ of the target node is found,i.e., a trust path is found.

Step 3. Path evaluation

In this part, Using the Min propagation method as the path strength W_k of the trust path found in the previous step. Repeat step 2, if a new trust path is found and the path strength W_k is larger or equal to the current maximum path strength Max_w. Then the node of the path is rewarded and the current maximum path strength Max_w is updated. Until all direct neighbors of the taget node are visited, i.e., all trust paths are found.

C. Aggregating direct trust

In this section, the direct trust values of multiple neighbors of the target node are aggregated to evaluate the trust. In order to improve the prediction accuracy, we further consider the rating of common items to measure the interest similarity among two users.

We use Pearson correlation coefficient to calculate the similarity between the target and the user's interest.If the similarity between user u_j and user u_t is higher, the weight between the two users will be greater, then u_t will be more willing to choose opinions from u_j. Based on this principle, a new aggregation function called SMCFAvg is proposed:

$$\tau'_{st} = \overline{\tau_s} + \frac{\sum_{u_j \in Nei_t} W_j * sim(u_j, u_t) * (\tau_{jt} - \overline{\tau_j})}{|Nei_t| Max_\tau} \tag{9}$$

Based on this formula, the trust value is calculated to solve the problem of path dependence. The target users prefer to trust the opinions provided by users with higher interest similarity, which further improves the accuracy of the prediction results.

D. Algorithm evaluation

We analyze the time complexity of the proposed algorithm. In our proposed algorithm for trust estimates, given the total number of actions $|A|$, the number of iterations is ***e***. The sampling step of our algorithm traverses the whole trust network from the source user to search the accessible users, so the time complexity is $O(|A| \cdot e)$ in the worst case. Given the depth of the state space $d = d(u_s, u_t)$, the time complexity of Step 2 is $O(|E| \cdot d)$. Thus, the total time is $O(|A| \cdot e + |E| \cdot d)$.

VI. Experiment

In this section, we evaluate the performance of our proposed algorithm DDQNTtrust and verify that our proposed aggregation function is more accurate. We test the performance of our trust inference algorithm on real Filmtrust datasets.

A. Dataset

In order to evaluate the proposed method, we apply the proposed algorithm to Filmtrust social network. The social network has 571 users, including 1853 trust relationships and 35497 movie rating data. In the original dataset, trust relations are represented as 1 to 10. We normalize the original data to map trust relationships to continuous value intervals [0,1]. 0 means no trust, 1 means full trust. In Filmtrust dataset, each user will rate and score some movies. We calculate the similarity between users based on the rates of two users for common movies.

B. Evaluation method

In this experiment, we use the leave one out method, which ignores the known trust value of the source user u_s to the target user u_t, and then uses our algorithm to infer the trust according to other indirect relations. Finally, we compare the actual trust value with our estimated value to measure the performance of our algorithm.

C. Evaluation Metrics

In our experiment, we mainly consider the prediction accuracy and coverage. Prediction accuracy indicates whether the predicted user is trusted or not. Coverage represents the percentage of predictable trust relationships, i.e., at least one path can be found between two users to infer trust values. The different evaluation metrics are as follows:

Mean absolute error (*MAE*): let E' denote the edge set of predictable trust, τ_{ij} is the actual trust value, τ'_{ij} is the predicted trust value.

$$MAE = \frac{\sum_{e_{ij} \in E'} \left| \tau'_{ij} - \tau_{ij} \right|}{\left| E' \right|} \tag{10}$$

Precision.This measures the ratio of predicted trusted users to actual trusted users. T_d is the number of edges in which u_s directly trust u_t, T_c is the number of trust edges estimated by our algorithm.

$$Precision = \frac{T_d \cap T_c}{T_c} \tag{11}$$

Recall.This metric measures the fraction of users are actually trusted and are successfully predicted.

$$Recall = \frac{T_d \cap T_c}{T_d} \tag{12}$$

Fscore.This measures the accuracy of combined recall and precision.

$$Fscore = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{13}$$

TABLE I. COMPARISON BETWEEN DIFFERENT AGGREGATION

Strategy	MAE	Precision	Recall	FScore
SMCFAvg	0.110	0.890	0.963	0.925
MCFAvg	0.120	0.880	0.961	0.918
WAvg	0.490	0.665	0.952	0.783
Max	0.350	0.677	0.943	0.788

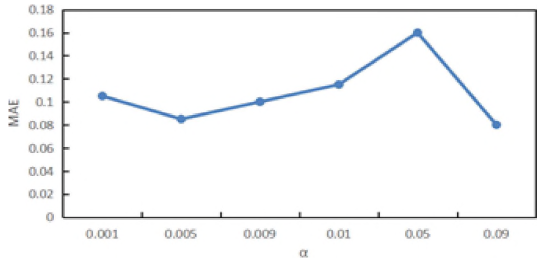

Figure 2. The effect of α on MAE when γ is 0.5.

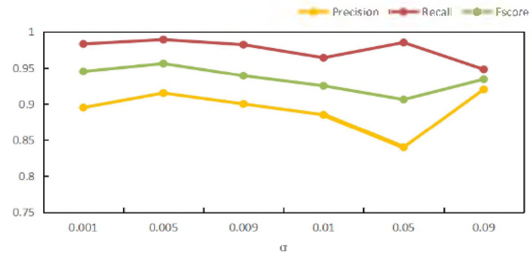

Figure 3. The effect of α on Precision, Recall and FScore when γ is 0.5.

D. Experimental Result

In order to discuss the effect of the proposed aggregation method and DDQNTrust algorithm, we compare it with four well-known aggregation strategies. We also compare the proposed algorithm with the famous TidalTrust and MoleTrust algorithms where the trust threshold λ is 0.5.

The purpose of this experiment is to study the influence of learning rate α on our algorithm. After many experiments, the result is best when γ is 0.5, we study the effect of α on the experimental results when γ is 0.5, the experimental results are shown in Fig. 2 and Fig. 3.

Fig.2 show that MAE increases first and then decreases with the increase of α. When α is greater than 0.05, MAE decreases with the increase of α, but when α is too high, the algorithm may not converge. MAE is 0.08 when α is 0.09, which is the best value compared with other values. However, Fig.3 show that FScore is the best when α is 0.005. Comprehensive consideration, we set α to 0.005 in the next experiments.

In order to verify the accuracy of our aggregation function, we compare it with the other three aggregation function based on Min propagation straegies. The comparison results are shown in Table 1.

From this table, we can see that MCF and SMCF have similar trust inference results, but the MAE and FScore of our SMCFAvg aggregation function are obviously better than MCF aggregation function. Moreover, the effect of the Max and WAvg aggregation function are obviously worse than MCF and SMCF aggregation function. So our aggregation function performs the best on Filmtrust dataset.

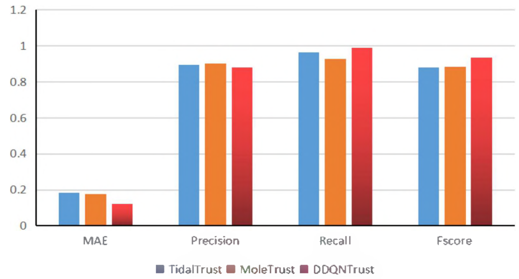

Figure 4. Comparison between different strategies on FilmTrust dataset.

In this experiment, we compare the proposed DDQNTrust algorithm with the famous TidalTrust and MoleTrust algorithms, and the experimental results are shown in Fig. 4.

Because of considering all trust paths and only selecting users whose trust value is highter than 0.5.According to the results, DDQNTrust algorithm obtains the higher estimation accuracy than the strategies using only shortest trust paths.

VII. Conclusion

Previous work mainly considered the shortest path and restricted search range, which reduced many available trust information. In the process of trust propagation, path dependence and trust decay are common problems. In view of the existing problems, we propose a trust propagation algorithm based on reinforcement learning DoubleDQN algorithm to find reliable trust path and predict the indirect trust value between users without direct contact. In order to overcome the path dependence problem and improve the prediction accuracy,we also propose an aggregation algorithm based on standard collaborative filtering algorithm and interest similarity. Experimental results on real data sets show that our algorithm achieves higher prediction accuracy than previous algorithms.

In future work, we will try to propose a new trust propagation algorithm to effectively overcome the trust decay problem, and also consider dynamic trust to propose a new dynamic trust inference algorithm.

Acknowledgment

The authors are thankful for the financial support of the National Natural Science Foundation of China under Grants 62072392 and 61972360. The authors are grateful to all of the reviewers for suggestions and insights that improved the paper.

References

[1] D.M. Boyd, N.B. Ellison, “Social network sites: Definition, history, and scholarship,” Journal of Computer - Mediated Communication, vol. 13, pp. 210-230, October 2007.

[2] C. Thovex, F. Trichet. “Static, dynamic and semantic dimensions: towards a multidisciplinary approach of social networks analysis,” in Proc. 4th International Conference on Knowledge Science, Engineering and Management, 2010. pp. 618-628.

[3] D. Mandaglio, A. Tagarelli, "Generalized Preference Learning for Trust Network Inference," IEEE Access, vol. 7, pp. 174583-174596, December 2019.

[4] J. Golbeck, “Computing and applying trust in web-based social networks,” PhD thesis, Dept. Comput. Sci., College Park, Md,Univ. of Maryland, 2005.

[5] W. Jiang, G. Wang, and J. Wu, “Generating trusted graphs for trust evaluation in online social networks,” Future Generation Com-put. Syst, vol. 31, pp. 48–58, February 2014.

[6] Y.A. Kim, H.S. Song, “Strategies for predicting local trust based on trust propagation in social networks,” Knowledge-Based Syst, vol. 24, pp. 1360–1371, December 2011.

[7] M. Ghavipour, M.R. Meybodi, “Trust Propagation Algorithm based on Learning Automata for Inferring Local Trust in Online Social Networks,” Knowledge-Based Syst, vol. 143, pp. 307–316, March 2018.

[8] C.-N. Ziegler, J. Golbeck, “Investigating interactions of trust and interest similarity,” Decis. Support Syst, vol. 43, pp. 460-475, March 2007.

[9] C.Y. Mao, C.F. Xu, Q. He, “A cost-effective algorithm for inferring the trust between two individuals in social networks,” Knowledge-Based Syst, vol. 164, pp. 122–138, January 2019.

[10] K. Akilal, M. Omar, H. Slimani, “Characterizing and using gullibility, competence, and reciprocity in a very fast and robust trust and distrust inference algorithm for weighted signed social networks,” Knowledge-Based Syst, vol. 192, March 2020.

[11] J.-H. Cho, K. Chan, S. Adali, “A survey on trust modeling,” ACM Comput. Surv, vol. 48, pp. 1-40, October 2015.

[12] Y. Ruan, A. Durresi, “A Survey of Trust Management Systems for Online Social Communities – Trust Modeling, Trust Inference and Attacks,” Knowledge-Based Syst, vol. 106, pp. 150-163, August 2016.

[13] M. Ghavipour, M.R. Meybodi, “A dynamic algorithm for stochastic trust propagation in online social networks: Learning automata approach,” Comput Commun, vol. 123, pp. 11–23, June 2018.

[14] X. Chen, Y. Yuan, M.A. Orgun, L. Lu, “A topic-sensitive trust evaluation approach for users in online communities,” Knowledge-Based Syst, vol. 194, April 2020.

[15] W. Jiang, G. Wang, and J. Wu, “Generating trusted graphs for trust evaluation in online social networks,” Future Generation Comput. Syst., vol. 31, pp. 48–58, February 2014.

[16] S.J. Liu, X.R. Tong, “Urban transportation path planning based on reinforcement learning,” Journal of Computer Applications, vol. 51,pp. 1-9, July 2020(in Chinese).

[17] F.X. Qi, X.R. Tong, Y. Lei, “Agent Trust Boost via Reinforcement Learning DQN,” Journal of Computer Research and Development, vol. 06, pp.1227-1238, 2020(in Chinese).

[18] C.J.C.H. Watkins, P. Dayan, “Q-learning,” Mach Learn, vol. 8, pp. 279-292, May 1992.

[19] V. Mnih, K. Kavukcuoglu, D. Silver et al, “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529-533, February 2015.

[20] H.V. Hasselt, A. Guez, D. Silver, “Deep Reinforcement Learning with Double Q-learning,” Computer ence, September 2015.