EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Personalized project recommendations: using reinforcement learning

Faxin Qi[1], Xiangrong Tong[1*], Lei Yu[1,2] and Yingjie Wang[1]

Abstract

With the development of the Internet and the progress of human-centered computing (HCC), the mode of man-machine collaborative work has become more and more popular. Valuable information in the Internet, such as user behavior and social labels, is often provided by users. A recommendation based on trust is an important human-computer interaction recommendation application in a social network. However, previous studies generally assume that the trust value between users is static, unable to respond to the dynamic changes of user trust and preferences in a timely manner. In fact, after receiving the recommendation, there is a difference between actual evaluation and expected evaluation which is correlated with trust value. Based on the dynamics of trust and the changing process of trust between users, this paper proposes a trust boost method through reinforcement learning. Recursive least squares (RLS) algorithm is used to learn the dynamic impact of evaluation difference on user's trust. In addition, a reinforcement learning method Deep Q-Learning (DQN) is studied to simulate the process of learning user's preferences and boosting trust value. Experiments indicate that our method applied to recommendation systems could respond to the changes quickly on user's preferences. Compared with other methods, our method has better accuracy on recommendation.

Keywords: HCC, Reinforcement learning, Trust, RLS, DQN

1 Introduction

With the development of HCC, recommendation has become common in social networks. Information available in recommendation system is also increasing. User information can be processed and utilized in appropriate ways to enhance user's experience. HCC considers the state of user interaction and how it can be used for practical applications. Among the various information available in HCC, trust is widely used as a user-based subjective information [1]. As a subjective consciousness, trust is the basis of a transaction or exchange relationship. The Internet era has made the interaction between people more frequent. People always establish contacts with friends in social networks which has become research hotspots to make recommendations based on people's trust relationship [2]. It is not uncommon for people to make recommendations to each other in daily life, such as recommending movies and clothing brands to each other. In this recommendation process, in addition to the recommended items (movies, brand, etc.), an important factor that determines whether people will accept the recommendation from their friends is the trust relationship between people. In fact, people are more likely to accept the suggestions and recommendations from their good friends because they trust them more than others. As a result, recommendation based on trust is an important human-computer interaction recommendation application in social networks [3, 4].

In social networks, the trust value between users has been regarded as a basis by many researchers. They predict user's preferences and ratings referring to user's past interaction records, then recommend related items to users. In recent years, scholars have proposed many methods to calculate trust and recommend items in social networks, which are based on different research bases and purposes. In general, most methods focus on the propagation of trust and trust recommendation systems. They treat trust values as static and immutable parameters, which makes it impossible for the recommendation system to make better user-centered personalized recommendation.

* Correspondence: xr_tong@163.com
[1]School of Computer and Control Engineering, Yantai University, Yantai 264005, China
Full list of author information is available at the end of the article

According to the dynamic trust, most of the existing researches are focusing on the changes and states of trust related factors. These researches failed to fully consider the process of trust dynamic changes and the factors affecting the dynamic change of trust. In fact, the change of trust will affect the recommendation results. Dynamic change process will be reflected in recommendation system in real time, also will affect the coefficient of recommendation system. Recommendation results are affected in the same time, so dynamic trust values could lead to completely different recommendations. Therefore, it is necessary to consider dynamic trust to improve the performance of recommendation system. The improved recommendation system could get more accurate results. The real-time performance of recommendation systems will be improved.

For some purposes, when A wants to improve B's trust value, A will communicate with B frequently, which usually start with B's interests and hobbies. If B likes watching movies, A will often recommend B the movies she may like. When B makes a positive evaluation of A's recommendation, it indicates that A's recommendation is consistent with B's preferences in films. B will have more confidence in A's level of appreciation in film, and B's trust in A will increase. On the contrary, it indicates that B doubts A's level of appreciation, and B's trust in A will be decreased. With the increase of the number of interactions that A recommends movies to B, A will know better about B's preferences in movies; thus, A can recommend B's favorite movies accurately. At the same time, B will trust A very much. Essentially, this is a process of *learning their preferences and caters to their likes.*

Our method simulates the above process: in order to boost a user's trust, the recommender chooses an item to recommend to the user. After accepting the recommendation, the user will make an actual evaluation of the item. When the user accepts the item, they will have an expectation. There's a certain difference between the user's actual evaluation and the expectation, which determines user's satisfaction degree on the item. If the actual evaluation is higher than the expectation, the user will give positive feedback to the recommender. Conversely, if the expectation is higher, the user will give negative feedback. Positive feedback indicates the user's recognition of the recommender, and the user's trust in the recommender will increase. Negative feedback indicates that the user doubts recommender's level of recommendation, which leads to the decline of the user's trust in recommender. This paper uses the reinforcement learning method DQN to boost the trust of the users and applies it to recommendation systems to improve the efficiency and accuracy of recommendation results. The DQN method is stable, and it can process large amount of data, so our method could be extended to recommendation systems. The contributions of this paper are shown as follows:

1. The proposed method in this paper combines the reinforcement learning method DQN to learn the change process of trust to boost the users' trust, named DQN trust boost (DQN-TB). This method is based on the difference between user's experience evaluation and expected evaluation, which could obtain more complete information by learning the user preferences, thus to improve the level of individuation and the accuracy of recommendations.
2. The proposed method takes synthetically the user's preferences, direct trust, and indirect trust into consideration, and then selectively screens these factors to deliver more practical results.

This paper is organized as follows. Section 2 reviews the related researches. Section 3 gives definitions related to our research. Section 4 introduces the method of trust transfer and suggestion processing. Section 5 displays the design of our method, including the process of trust boosting by DQN and related parameters. Section 6 elaborates the parameters used and comparison experiments conducted, then analyzes and discusses the experimental results. Conclusion is summarized in section 7.

2 Related work

On the researches of trust, some scholars have made some achievements. Most of these are about the propagation of trust and trust network formation. Xu et al. [5] researched trust through applying reward and penalty to inspire the behaviors of users, which achieved a good performance. Jiang et al. [6] proposed an extraction method of neighborhood sensing trust network, aiming to solve the problem of trust propagation failure in trust networks. This method considered the domain perceived influence of users in online social networks, adopting directed multi-graph to model the multi-trust relationship among heterogeneous trust networks. Then, they designed a domain perceived trust measure method to measure the trust degree between users. Yan et al. [7] proposed an improved social recommendation algorithm based on neighborhood and matrix decomposition, aiming to solve the problems of large-scale, noise and sparsely in relational networks. This method developed a new relational network fitting algorithm to control the propagation and contraction of relationships and generated a separate relational network for each user and item. Some scholars considered the dynamic trust in process of research, and they have proposed some methods, such as Ghavipour et al. [8]. They considered the change of trust value in the process of trust transfer and proposed the heuristic algorithm named DLATrust. The DLATrust based on learning automata, used the improved collaborative filtering and aggregation policy to infer the value of trust. On this basis, Mina et al. [9] proposed a dynamic algorithm named DyTrust, which used distributed learning automaton for random trust propagation. Both DLATrust and DyTrust were aiming to learn to discover reliable paths between users in social networks. You et al. [10] proposed a distributed dynamic management model based on the trust reliability. This method used reliability to evaluate trust to reduce the impact of unreliable data and modify reliability after interaction.

Moreover, many scholars have achieved excellent research results about prediction and recommendation. Qi et al. [11] introduced an instance of Locality-Sensitive Hashing (LSH) into service recommendation and further put forward a novel privacy-preserving and scalable mobile service recommendation approach based on two-stage LSH. In addition, Qi et al. [12, 13], Liu et al. [14] and Gong et al. [15] proposed a variety of recommendation and prediction methods. Furthermore, many scholars have conducted corresponding researches on the related properties of adaptive reputation and trust [16–18]. On the aspect of trust, the dynamic problem of trust has not been considered. The research on dynamic trust is relatively blank. This section will briefly introduce the previous work and DQN method.

2.1 DyTrust

DyTrust algorithm is one of the methods of dynamic trust calculation by using learning algorithm. DyTrust uses distributed learning automaton to obtain the dynamic change of trust in the process of trust propagation, and updates the reliable trust path dynamically according to the change of trust.

As a dynamic trust propagation algorithm, DyTrust algorithm can generate more accurately a trust path. However, DyTrust algorithm only uses the dynamic characteristics of trust, without studying the dynamic process of trust. The dynamic change process of trust will be studied and elaborated in this paper.

2.2 Q-Learning and Deep Q-Learning

Deep Q-Learning [19] is the development of the Q-Learning algorithm [20]; it is also an emerging algorithm that combines deep learning with reinforcement learning to realize learning.

Q-Learning algorithm uses a single neural network to estimate the value function and calculate the practical accumulated experience. Compared with it, DQN uses two equally structured neural networks to evaluate the value function estimates (eval_net) and reality (target_net). The eval_net estimates the value of each action (Q_eval), then selects a final action based on the policy. The environment will return reward value based on the action, and target_net used the reward value for realistic estimation (Q_target).

The weights of target_net are updated more slowly than the weights of eval_net; in other words, target_net tends to be updated every rounds. This method could ensure that DQN avoids the influence of time continuity and gets better results.

At the meanwhile, DQN uses experience playback to train eval_net. After obtaining training parameters of neural network for DQN, the theory of Behrman equation is used to calculate loss function and update weight parameters of eval_net:

$$\text{Loss Function} = \text{Q\_target}-\text{Q\_eval}$$

The target_net calculation is derived from the Markov decision.

The trust boost algorithm in this paper is combined with DQN for calculation. In fact, Q-Learning algorithm can also achieve similar results for the trust boost process of a single user. But the disadvantages of Q-learning algorithm cannot be ignored. When Q-Learning is used in reality, there are too many state variables and features need to be manually designed. In addition, the quality of results is closely related to the quality of feature design, which makes Q-Learning algorithm unable to be used in recommendation systems to recommend items for a large number of users. At the same time, Q-Learning algorithm needs to use a matrix to store values. When there are too many users, the data volume will be too large, which will lead to a sharp increase in storage space demand. The huge number of user groups in recommendation systems and the complex categories of recommendation items are disasters for the data storage of Q-Learning algorithm.

Therefore, considering the problems of the method proposed in this paper and the advantages of DQN compared with Q-Learning algorithm, this paper uses DQN to design the recommendation model in social networks.

3 Method introductions

3.1 Problem description and basic definitions

This section introduces the basic problem description, user information set, recommender information set, and DQN information set in detail.

3.1.1 Problem description

As shown in Fig. 1, in order to boost user B's trust, user A recommends item i relevant to the content that B is interested in. After B accepted A's recommendation, if B's evaluation of i is higher than B's expected value, B's

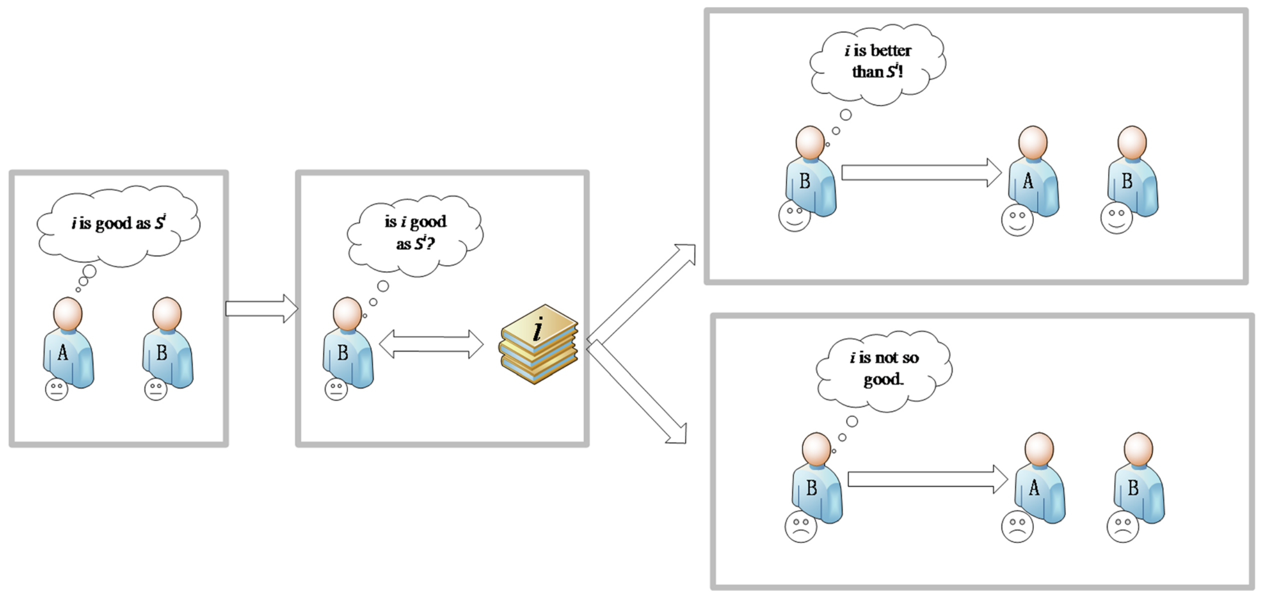


Fig. 1 Relationship between recommendation and trust

trust in A will increase. Otherwise, B's trust in A will decrease.

3.1.2 Basic definitions

Define user set $U = \{u_1, u_2, \ldots, u_k\}$, item set $\mathbb{I} = \{i_1, i_2, \ldots, i_k\}$

.

Definition 1. User Set:

$$\langle T, S, sp \rangle \tag{1}$$

Our method establishes user information set for each user in social network shown by Eq.(1), where T represents user trust matrix, S represents user evaluation matrix, including user's past evaluation and the user's expected evaluation for recommendation items, and sp represents user's actual evaluation for recommendation items.

Definition 2. Recommender Information Set:

$$\langle T, S, \mathbb{I} \rangle \tag{2}$$

After recommending items to other users, the user will transform into recommender. There will be an information set of the recommender to represent his information. T and S are trust matrix and evaluation matrix of the recommender. $\mathbb{I}$ is the set of items.

During the recommendation process, recommender selects items from item sets for recommendation. Users' degree of satisfaction will dynamically affect the trust between users.

Definition 3. *DQN-TB Information Set* $\langle n, a, \pi, r \rangle$

Recommender's status n: A user broadcast in moment τ, recommender chooses action based on a selection strategy. n_τ is recommender's status corresponds to the action. After recommender's action choice, recommender's status updates as $n_{\tau+1}$, and the recommender will wait for the user's next broadcast.

Recommender's action a: Recommender selects the final recommended item to user from item set. The recommended item is recommender's action a.

Action Selection Strategy π: The selection strategy determines the recommended item which is finally selected by the recommender. The selection policy in this paper is same as that used in DQN, namely ε-greedy policy.

Action Reward r: The user will have their satisfaction of the item i provided by the recommender. The influence range of satisfaction on trust is denoted as reward r, and r will affect the recommender's choice at next moment.

In this paper, we use the difference between users' expected evaluation and actual evaluation to represent user satisfaction. Recursive least squares method (RLS) is used to calculate the dynamic mapping relationship between the evaluation difference and the change of trust. In the process, the dynamic change of trust is regarded as the reward given to the recommender in the process of DQN-TB, and the change of trust will affect the recommender's selection behavior of the recommended items.

3.2 Trust calculation and suggestion processing

In this section, formulas of trust and suggestion will be displayed.

3.2.1 Trust calculation and data structure

Trust between users can be obtained through direct trust and indirect trust. Users with interactive experience are direct users. When users have no interactive experience, if a trust path can be found between the two users, they are indirect user of each other [21, 22].

The direct trust of user j to user b is defined as follows:

$$T(j,b) \leftarrow t_{j,b}, \quad t_{j,b} \in [0,1] \tag{3}$$

The indirect trust of user j to user d is defined as follows:

$$T^R(j,d) \leftarrow t^r_{j,d}, \quad t^r_{j,d} \in [0,1] \tag{4}$$

Assume that the path of trust path between users can be obtained by up to six intermediate users. The indirect trust value t^r between user j and d is obtained by the following formula:

$$t^r_{j,d} = \frac{\sum t_{j,x} + \cdots + t_{x,d}}{\left|X^d_j\right|} \quad , \quad x \in X^d_j \tag{5}$$

where X^d_j is the collection of all users between user j and d.

3.2.2 User evaluation suggestion processing

According to the evaluation suggestion s^i_b made by direct user b, the expected evaluation of item i made by user j is calculated as follows:

$$s^i_{j,b} = t_{j,b} \cdot s^i_b, \quad s^i_{j,b} \in S_i \tag{6}$$

According to the evaluation suggestion s^i_d made by indirect user d, the expected evaluation of item i made by user j is calculated as follows:

$$rs^i_{j,d} = t^r_{j,d} \cdot s^i_d, \quad rs^i_{j,d} \in S_i \tag{7}$$

Where S_i is the storage matrix of expected evaluation of item i obtained by user j through different users. The range of user evaluation points is [0, 10].

In reality, user j can receive suggestions from different users around. It is necessary to process all the recommendation evaluation information received by user j to get the final evaluation expectation of user j. Using formula (8) to process the scoring information in S_i, the final expected evaluation s^i_j of user j to item i can be obtained:

$$s^i_j = \frac{\sum t \cdot s^i + \sum t^r \cdot rs^i}{\sum t + \sum t^r} \quad , \quad s^i, rs^i \in S_i, \quad t, t^r \in T \tag{8}$$

After receiving multiple recommendations, user j will accept the item with the highest expected evaluation.

3.3 The method: DQN trust boost process

Some scholars have applied DQN to recommendation systems, which keeps recommendation systems dynamic for a long time. Zheng et al. [23] proposed a deep reinforcement learning framework for news recommendation. This method calculates dynamic reward value according to user characteristics and behavioral feedback, so that the recommendation system can capture the change of user preferences. This method can get long-term rewards and keep users interested in the recommendation system.

This chapter describes the trust boost via Deep Q-Learning process in detail, explains the method and the process of the DQN-TB, and gives the corresponding pseudo code. Figure 2 shows the flow chart framework of DQN-TB process.

It should be noted that a one-to-one relationship can exist in DQN-TB process, that is, only user u_1 expects to increase the trust value of u_2 in him. There can also be a many-to-one relationship, that is, user u_1, u_2... all expect to improve the trust value of user u_3 in them.

3.3.1 Model framework

As shown in Fig. 3, each item in item set represents different actions. In DQN-TB method, a user is regarded as the environment. At each state, DQN network is trained by the data in memory pool, and then an item is selected from the item set and recommended it to the user as the final action. Then, reward returned by the user is obtained. It will be stored in the memory pool with state and action for the next network training update [24].

This process uses the eval_network to expect returns. According to the process flow in Fig. 2 and the frame diagram in Fig. 3, the network update model is as follows:

1. Recommend projects

A user broadcasts at moment τ. According to the user's broadcast of requirements, a recommender selects an item i from the corresponding item set. a_τ is corresponding to the item i, recommender status notes for n_τ.

2. Trust update

After the user received and accepted the recommendation, the difference between expected evaluation and actual evaluation will affect the user's trust in the recommender. Treating the user as the environment of DQN-TB process, changing range Δt of user's trust will be used as the reward value of DQN-TB process.

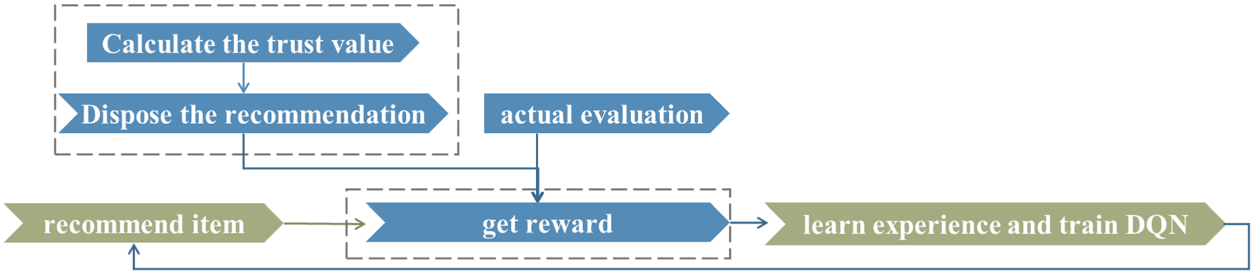

Calculate the trust value
Dispose the recommendation
actual evaluation
recommend item
get reward
learn experience and train DQN

Fig. 2 Flowchart of DQN-TB

Δt, action a_{τ}, recommender's status n_{τ}, and the future state $n_{\tau+1}$ are stored in the memory pool, as input of network.

3. Train network

The DQN-TB process uses data in memory pool and the eval_network for action anticipation selection. DQN-TB simulates user reality through the target_network, and updates eval_network based on the difference between eval_network and target_network.

4. Repeat steps (1)–(3)

Considering the reality, it will not stop unconditionally once the recommendation process is established. If user's trust value in recommender is too low, suggestions provided by the recommender will not be adopted by the user. Therefore, if recommender continues to

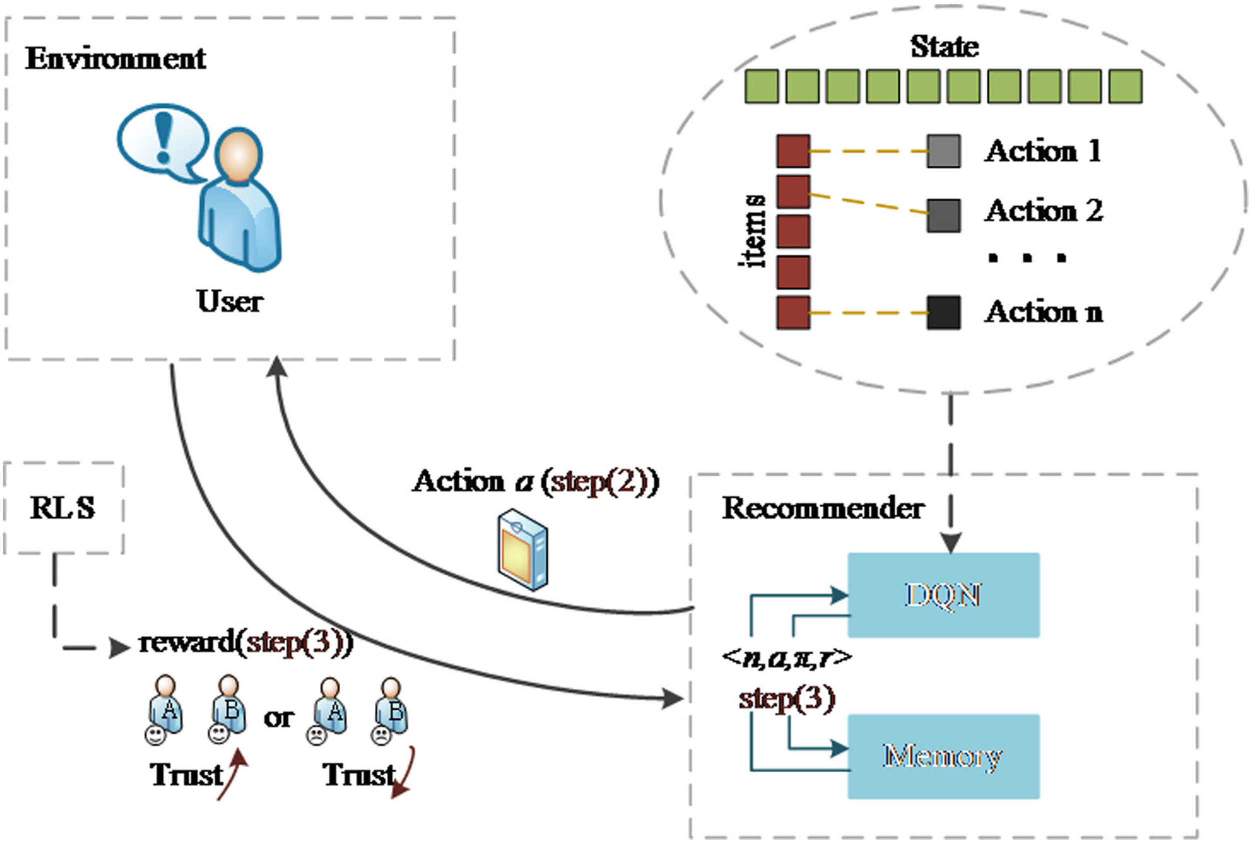


Fig. 3 Framework steps of DQN-TB

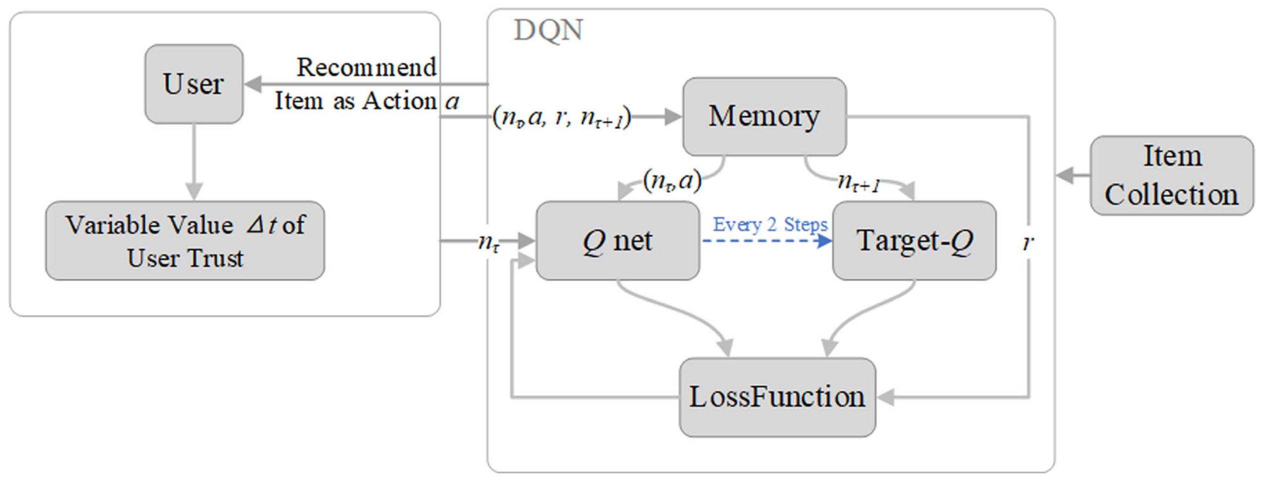


Fig. 4 Data transmission process of DQN-TB

receive negative rewards or recommender has too many referrals fail, the cycle will be terminated. In order to provide more opportunities for recommender's learning, considering the actual situation, it is not practical to recommend when the trust value is too low. Through experimental verification, it was found that recommendation could obtain better results when terminated condition is set as $t<$ 0.2.

3.3.2 DQN-TB design

Considering dynamic trust and the concrete process of item recommendation, our method uses the eval_net to estimate recommender's reward by choosing a certain item for recommendation (that is, the action a_τ). After taking the degree of trust change as reward value, the reward of item selection can be modeled as:

$$Q(n_\tau, a_\tau; \omega) \approx Q^\pi(n_\tau, a_\tau) \tag{9}$$

where ω is the weight parameter of eval_network and π is the selected strategy. There may be no interactive experience between recommender and user, so the recommendation strategy shall be stipulated as follows:

1. *Initial recommendation*: All alternative items are equally likely to be selected. Recommendation items are randomly to be selected.
2. *Follow-up recommendation*: By using the ε-greedy policy, selection probability changes dynamically according to recommended results and finally converges.

On the basis of Markov property, the state at the next moment in random state is only related to the current state. Thus, after computing action probability and making action choice by eval_net, DQN-TB process will get reward Δt and next status $n_{\tau+1}$. Meanwhile, actual return is calculated by target_net:

$$Q(n,a) = \Delta t_\tau + \lambda \ \max_{a_{\tau+1}} Q(n_{\tau+1}, a_{\tau+1}; \overline{\omega}) \tag{10}$$

where λ is the discount coefficient, which is used to balance the immediate returns and the future returns. Δt_τ is the reward of action a_τ. $\overline{\omega}$ is the weight parameter of target_net. After target_net calculated actual returns, the target function is calculated by bellman equation:

$$L(\omega) = E\left[\left(\Delta t_j + \lambda \underset{a'}{\max} Q\left(n', a', \overline{\omega}\right) - Q(n,a,\omega)\right)^2\right] \tag{11}$$

The eval_net parameter ω is updated by using stochastic gradient descent on the objective function $L(\omega)$. Eval_net has the same structure with target_net. The weight parameter of target_net is same as that of eval_net, and is copied from eval_net every two steps.

Figure 4 shows the data transmission process of DQN-TB.

3.3.3 Reward parameter

Since the evaluation error and trust in the actual trust updating process are based on a single user, there is always a trust value between users who can be recommended, $t(0)$ is the original trust between them in the reward calculation process, and $P(0) = \delta^{-1}I$.

u_1,u_2 are the user and the presentee, respectively, taking u_2 recommends item i to u_1 in the nth round of recommendation as an example, RLS method is used to simulate the trust change and update process:

The estimated error e is the gap between the actual evaluation and the expected evaluation:

$$e = sp^i_{u_1} - s^i_{u_1,u_2} = sp^i_{u_1} - t_{u_1,u_2} \cdot s^i_{u_2} \tag{12}$$

The cost function $J(n)$ is expressed as:

$$J(n) = \sum \lambda |e|^2 = \lambda \left| sp^i_{u_1} - t_{u_1,u_2} \cdot s^i_{u_2} \right|^2 \tag{13}$$

where λ is the forgetting factor, in this paper, we set $\lambda = 1$ for stability.

In order to minimize the cost function, it can be obtained by differentiating the weight vector t:

$$\begin{aligned} \frac{\partial J(n)}{\partial t} &= \frac{\partial}{\partial t} \sum \left[sp^i_{u_1} - s^i_{u_1,u_2} \right] \times \left[sp^i_{u_1} - s^i_{u_1,u_2} \right] \\ &= R(n) \cdot t_{u_1,u_2} - r(n) \end{aligned} \tag{14}$$

known that $R(n) = \sum \left(s^i_{u_2}\right)^2$, $r(n) = \sum sp^i_{u_1} \cdot s^i_{u_2}$.

Set $P(n) = R^{-1}(n)$:

$$P(n) = P(n-1) - s^i_{u_2} \cdot k(n) \cdot P(n-1) \tag{15}$$

then $k(n) = \frac{k(n) \cdot P(n-1)}{\left(s^i_{u_2}\right)^2 \cdot P(n-1)}$.

By $t_{u_1,u_2} = R^{-1}(n) r(n)$, after simplify, it is known that $\Delta t = k(n) e^*$.

If the trust relationship between u_1 and u_2 is direct trust, the trust update between them is expressed as:

$$t_{u_1,u_2} \leftarrow t_{u_1,u_2} + k(n) e^* \tag{16}$$

Similarly, if the trust relationship between u_1 and u_2 is indirect trust, the update between them is expressed as:

$$t^r_{u_1,u_2} \leftarrow t^r_{u_1,u_2} + k(n) e^* \tag{17}$$

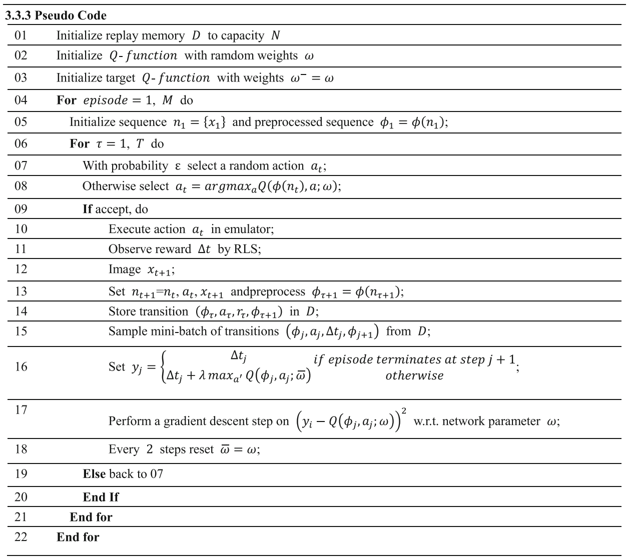
After trust update, user u_1's evaluation $s^i_{u_1}$ of item i will be updated to the actual evaluation $sp^i_{u_1}$ in evaluation matrix S:

$$s^i_{u_1} \leftarrow sp^i_{u_1} \tag{18}$$

Reward that recommender u_2 gets is $0.01 g(u_2) = \Delta t$.

In DQN-TB, reward value will be entered into the network as a parameter for future calculation.

3.3.4 Pseudo code



```
3.3.3 Pseudo Code
01 Initialize replay memory $D$ to capacity $N$
02 Initialize $Q\text{-}function$ with ramdom weights $\omega$
03 Initialize target $Q\text{-}function$ with weights $\omega^- = \omega$
04 For $episode = 1,\ M$ do
05   Initialize sequence $n_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(n_1)$;
06   For $\tau = 1,\ T$ do
07     With probability $\varepsilon$ select a random action $a_t$;
08     Otherwise select $a_t = argmax_a Q(\phi(n_t), a; \omega)$;
09     If accept, do
10         Execute action $a_t$ in emulator;
11         Observe reward $\Delta t$ by RLS;
12         Image $x_{t+1}$;
13         Set $n_{t+1}$=$n_t, a_t, x_{t+1}$ andpreprocess $\phi_{\tau+1} = \phi(n_{\tau+1})$;
14         Store transition $(\phi_\tau, a_\tau, r_\tau, \phi_{\tau+1})$ in $D$;
15         Sample mini-batch of transitions $(\phi_j, a_j, \Delta t_j, \phi_{j+1})$ from $D$;
16         Set $y_j = \begin{cases} \Delta t_j & if\ episode\ terminates\ at\ step\ j+1 \\ \Delta t_j + \lambda\, max_{a'}\, Q(\phi_j, a_j; \overline{\omega}) & otherwise \end{cases}$;
17         Perform a gradient descent step on $\left(y_i - Q(\phi_j, a_j; \omega)\right)^2$ w.r.t. network parameter $\omega$;
18         Every 2 steps reset $\overline{\omega} = \omega$;
19     Else back to 07
20     End If
21   End for
22 End for
```

4 Experiments

4.1 Experimental device

In this section, the simulation experiments are conducted to evaluate the performance of the proposed model:

1. The ratio of recommendation trust to direct trust conversion in DQN-TB and the learning rate of trust update when calculating reward parameters will be determined through experiments.
2. The trust boost effect of DQN-TB will also be illustrated.
3. The performance of recommendation systems with DQN-TB is compared with other recommendation systems, including success rate of recommendation and abilities to sense dynamic changes in user preference.

ANR is defined as the number of rounds required by recommendation system to respond changes in user preference, SUMR represents the total number of recommended rounds, and SEN represents the sensitivity of corresponding changes of user preference in recommendation system. Then SEN can be calculated by Eq.(19):

$$\mathrm{SEN} = 1 - \frac{\mathrm{ANR}}{\mathrm{SUMR}} \tag{19}$$

4.2 Basic introduction

User review data and movie project category in *douban* are used in experiment, including 10 user's evaluation data on different movies in 11 film project categories. The total number of movies watched by all users is 300, and the scale of movie review data is 510. In the experiment, users are randomly selected from all users as target users for making recommendations. The environment of experiments is based on OpenAI Gym, where the calculation process of reward value is written separately, then transmitted to DQN-TB. As DQN-TB is carried out for individuals, the structure of eval_net in this experiment is shown in Fig. 5.

4.3 Markov decision process parameters

Table 1 shows the relevant definitions of Markov decision process of DQN-TB,

The recommendation process will not stop unconditionally, so the number of user states will increase with recommendation process. Actions in recommendation process are recommended items; optional actions are related to the number of items in item set. Through consulting relevant literature and reference materials, the value of γ is set at 0.9.

4.4 Time complexity

In this algorithm, the time complexity of RLS method is $O(n)$. And every iteration in the process of DQN-TB should use RLS method to calculate the reward value. DQN-TB uses the stochastic gradient descent to update parameters, so the time complexity of our algorithm is calculated by $T(n) = (C + n) * n * n * n \approx T(n^4) = O(n^4)$. *T*he time complexity is polynomial.

5 Results and analysis

5.1 Conversion of indirect trust and direct trust

During the item recommendation process, when an item is recommended by recommender with indirect trust, there will be a direct trust relationship between the user and the recommender. In real situation, direct trust and indirect trust are not the same. Direct trust tends to have more impact. In order to conform better to real situation, indirect trust value needs to be converted into direct trust value with a certain discount so as to conduct subsequent calculation. The indirect trust discount factor is represented by μ.

To determine the value of μ, we use four sets of small data to analyze the result changes caused by different discount factors. The four data sets correspond to high indirect trust and high recommendation evaluation, high indirect trust and low recommendation evaluation, low indirect trust and high recommendation evaluation, and low indirect trust and low recommendation evaluation. All of the other data in the four data sets are the same. The data sets used for analysis are given in Table 2. The effect of discount factor μ on results of different data is reflected by the whole evaluation term in Fig. 6. The comparison evaluation is the expected evaluation calculated by data of Direct trust1-Direct trust 4 only.

According to Fig. 6, when μ is low, the overall expected evaluation value is lower than the contrast evaluation. When μ is higher, the evaluation of indirect user can positively stimulate the total results. The data used in this

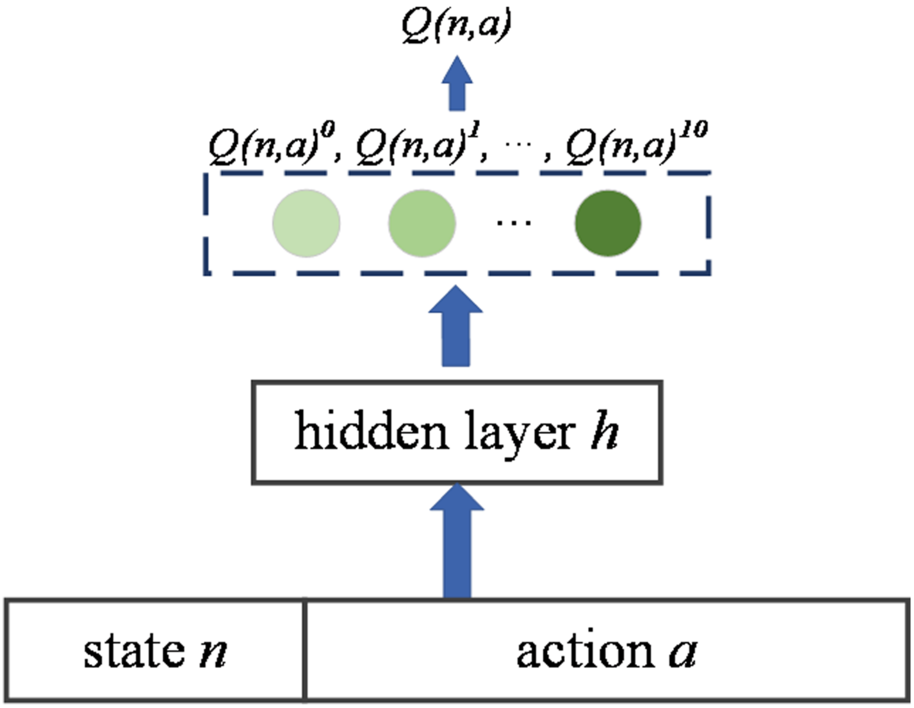


Fig. 5 Q network structure

Table 1 Parameters of Markov decision process

Parameter	Numerical definition
γ	0.9
Action space	$\lvert I \rvert$
Immediate reward	Δt
State space	∞

comparison experiment cannot represent the whole reality, but it still can reflect the influence of discount factor of indirect trust on results. Considering the practical factors, when recommenders make recommendation by direct trust for the first time, it will still have the corresponding impact on users' evaluation. In order to minimize the interference of μ, and let the users' evaluation be mainly influenced by trust values, we set $\mu = 0.8$.

5.2 Dynamic change of trust

According to the DQN-TB process, after the presentee learned relevant experience through the recommendation results, the recommendation choices will be further adjusted to meet the relevant interests and the preferences of the recommended users. Figure 7 shows the correlation between trust update and evaluation error using RLS method. The horizontal axis represents the recommend rounds. The left axis represents the trust value. The right axis represents the difference of evaluation. It can be seen that the error range is narrow and the convergence speed is fast when using RLS method for trust update. With the increase of recommendation times, trust increases steadily and shows down to a steady level. At the same time, the evaluation error continues to decrease. The decrease of error has fluctuation, which is caused by action selection strategy, but the general trend remains unchanged. This shows that with the increase of the number of recommendations, the recommender can learn more about user's preference and make more accurate recommendations.

Figures 8, 9, and 10 show the result of trust update and evaluation error by using LMS method. Same as Fig. 7, the horizontal axis represents the recommend rounds, the left axis represents the trust value, and the right axis represents the difference of evaluation. The convergence speed of the LMS method is related to the learning rate η. Figures 8, 9 and 10, respectively, correspond to the change of trust value and error when the learning rate is 0.1/0.2/0.35. It can be seen that when $\eta = 0.1$, trust changes smoothly, and the error fluctuation is not drastically very much. The effect is good. When $\eta = 0.2$, the change of trust has begun to fluctuate, but it still in the normal state. When $\eta = 0.35$, error changes drastically, trust fluctuates greatly, even value overflow appears (trust value bigger than 1). Comprehensive considering several factors, RLS method has good effect in some cases. But the effect has a strong correlation with η, which brings some limitations in the application of this method. Therefore, our method uses RLS method to calculate the trust update.

Figure 11 shows the broking line graph of the average change of trust with each turn in the DQN-TB process based on recommendation trust. The initial recommendation trust value between the presentee and the user is 0.67. User trust decreased in the second round, which was caused by the recommendation trust-direct trust transformation in 5.2. As shown in Fig. 11, when the number of rounds is low, the DQN-TB process is in the exploration stage, and the experience in the memory pool is not rich enough, so the trust changes slowly. With the increase of recommendation rounds, the recommender has a deeper understanding of the recommended user's preferences, and the trust value among users continues to rise. Moreover, due to the increase of successful experience, the trust value between users in the subsequent recommendation rounds is always stable and the fluctuation range is small.

DQN-TB process can accurately describe the user's trust change state and achieve better results. The dynamic change of user trust can reflect the change of user preference and social relationship in real time, so the dynamic study of DQN-TB is meaningful, which also brings more flexibility for the application of DQN-TB in recommendation system.

5.3 DQN-TB in recommendation system

The experiment in 5.4 verifies that DQN-TB has an accurate description of dynamic change and trust boosting, so this method can also be applied to recommendation system to provide accurate recommendations for users in the system. In this section, we compare DQN-TB with CSIT proposed by Li et al. [25] and confidence-based recommendation (CBR) proposed by Faezeh et al. [26]

Table 2 Trust value and score value

	Direct Trust 1	Direct Trust 2	Direct Trust 3	Direct Trust4	Recommend Trust1	Recommend Trust 2	Recommend Trust 3	Recommend Trust 4
Trust value	0.5	0.9	0.6	0.8	0.7	0.7	0.5	0.5
Score value	6.5	7	7	6.5	9	5.5	9	5.5

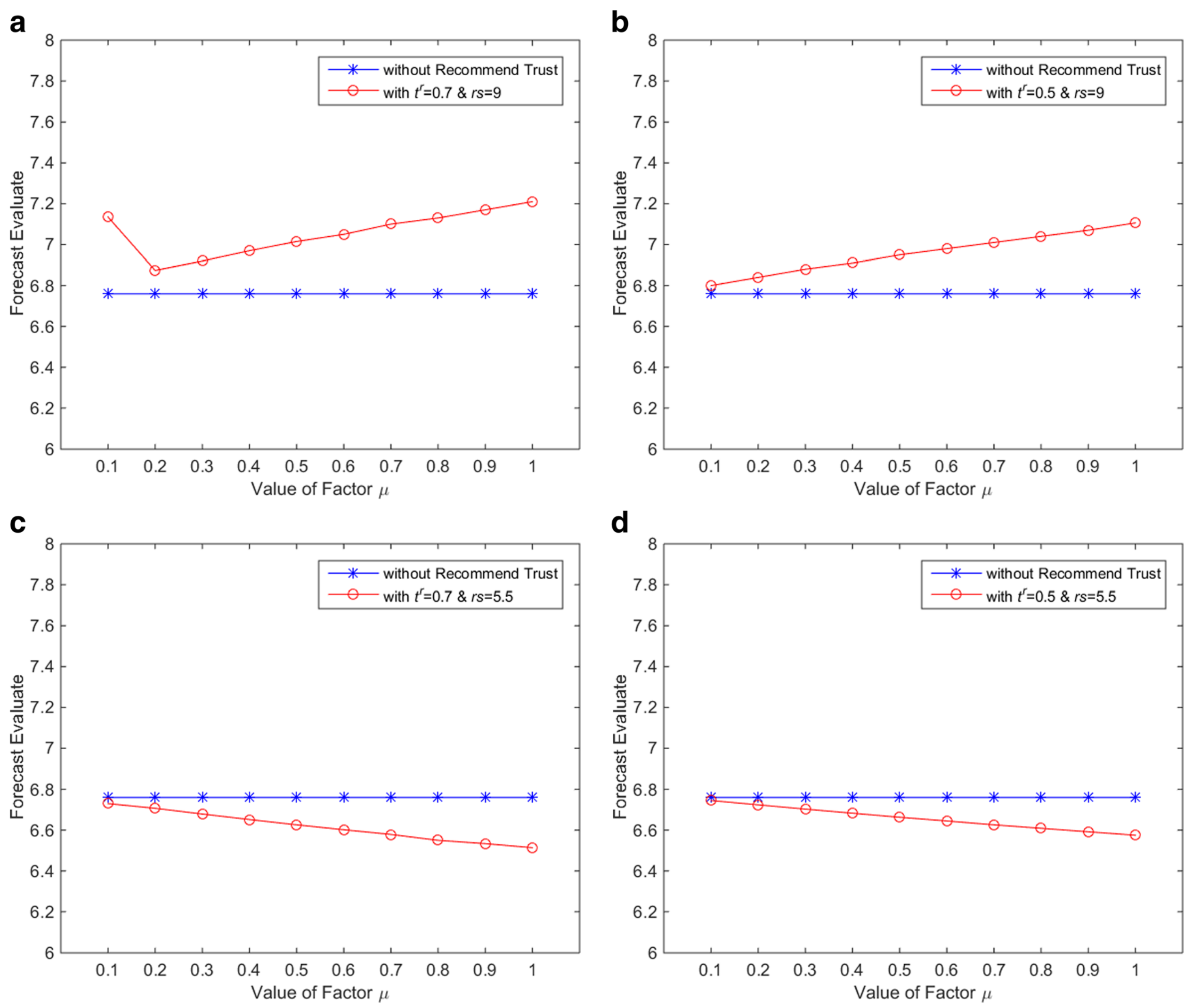


Fig. 6 Performance comparison on different μ. **a** The results of high recommend trust value and high score value. **b** The results of high recommend trust value and low score value. **c** The results of low recommend trust value and high score value. **d** The results of low recommend trust value and low score value

in success rates of recommendation and their response sensitivity of changes in user preference.

CSIT is a matrix factorization and context-aware recommender method with superior performance. The author provided GMM method to enhance and deal with classification context and continuous context. CBR used trust of user's opinion and certainty of opinion to describe user's confidence, introduced user's confidence into trust model, and provided a series of recommendations to user through implicit trust model. At present, recommendation systems recommend relevant items only by use user's social relations and other user's preference in relational network, which cannot reflect the change of user's trust. Moreover, the change of user's preference can only be passively obtained from other users in the relational network, resulting in systems' untimely and inaccurate on making feedback on changes of users' preferences.

Figure 12 shows the comparison of average success rate of DQN-TB, CSIT, and CBR corresponding to different rounds with in 20 rounds with no dynamic change of user trust, that is, user's trust is static trust. The DQN-TB method cannot obtain the reward under the situation of static trust, which leads to the inability to learn experience and poor recommendation effect. Both CSIT method and CBR method are calculated based on static trust. Therefore, when user's trust and other information remain unchanged, their recommendation success rate is higher and their performances are better than DQN-TB.

Figure 13 shows the comparison of average success rates of DQN-TB, CSIT, and CBR corresponding to

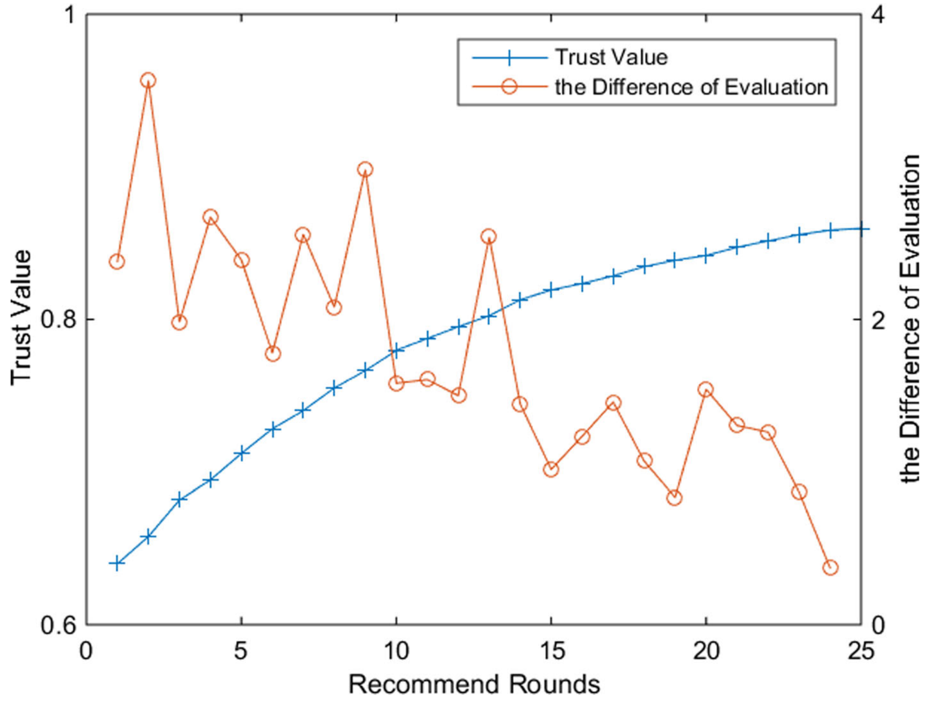


Fig. 7 Correlation graph of evaluation and trust (with RLS)

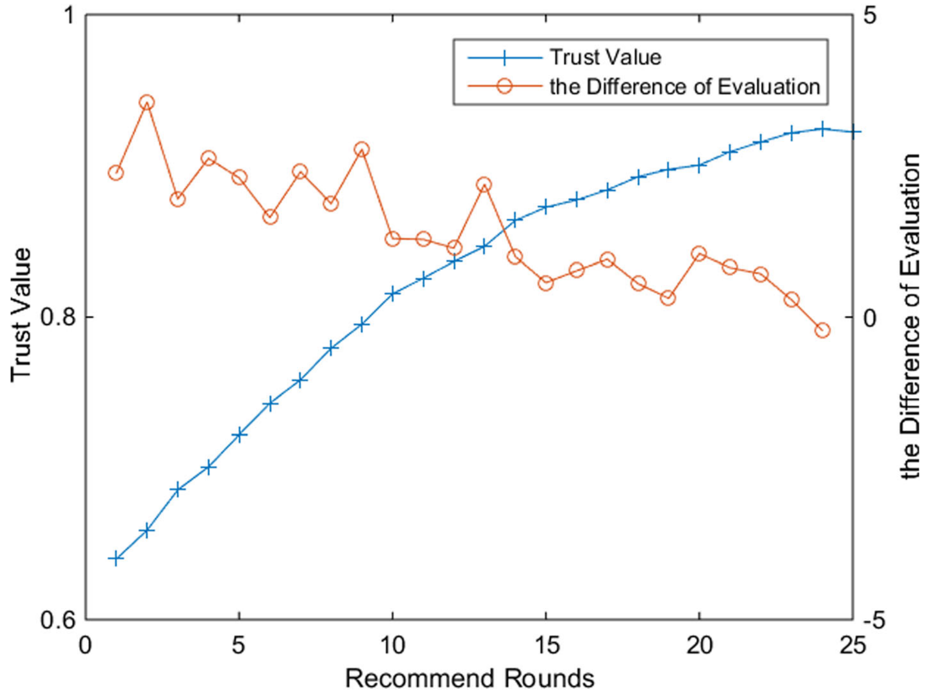


Fig. 8 Correlation graph of evaluation and trust (with LMS and $\eta = 0.1$)

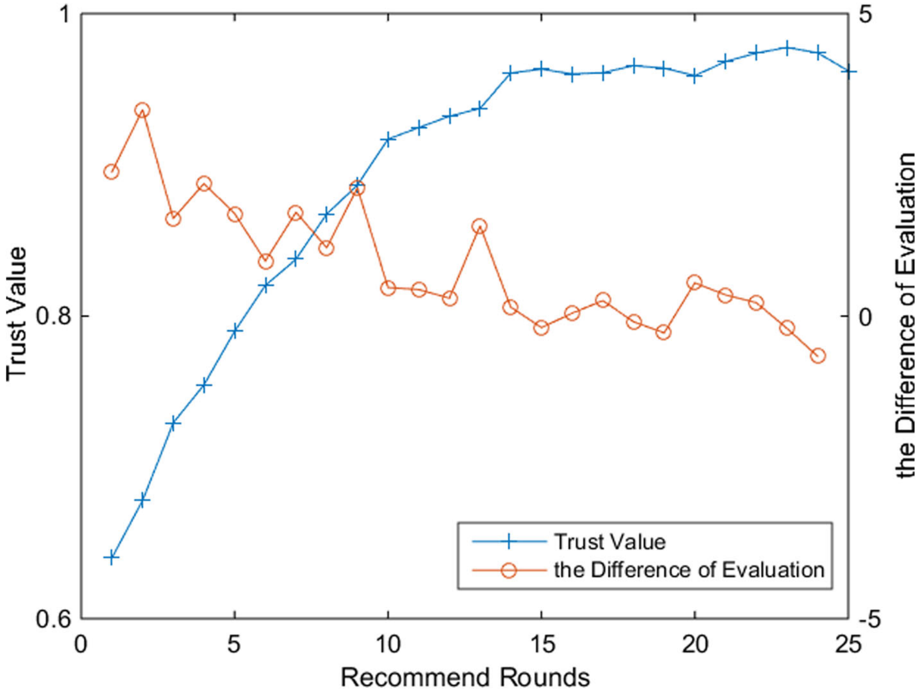


Fig. 9 Correlation graph of evaluation and trust (with LMS and $\eta = 0.2$)

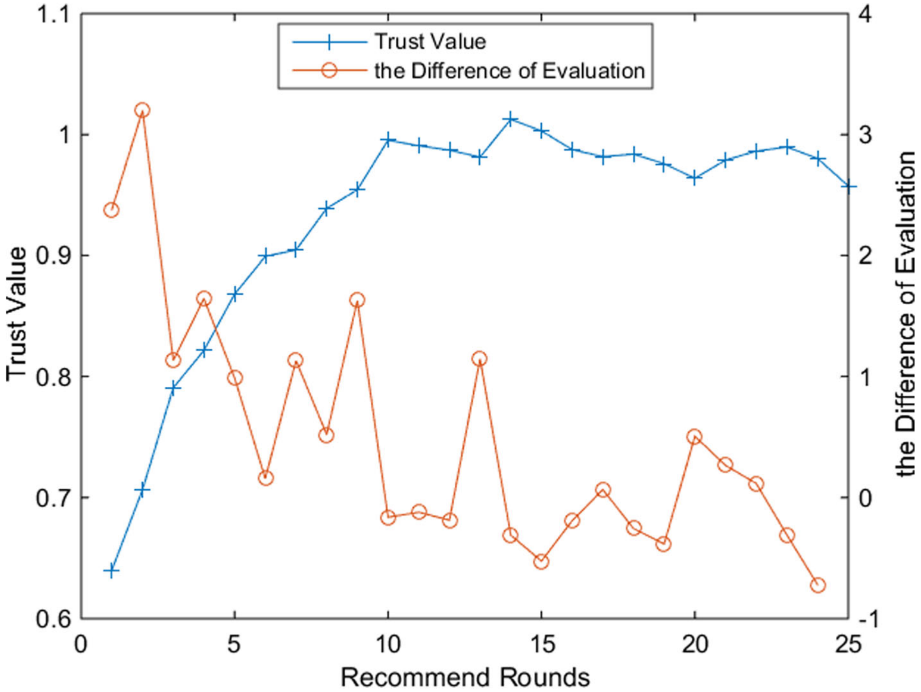


Fig. 10 Correlation graph of evaluation and trust (with LMS and $\eta = 0.35$)

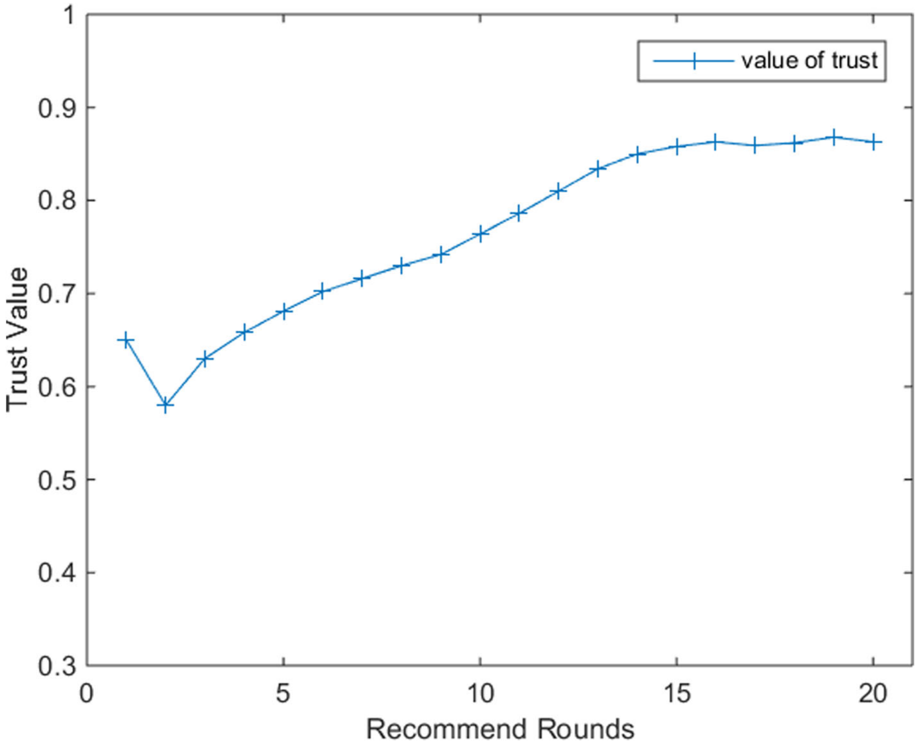


Fig. 11 The line chart of dynamic change of trust value

different rounds with in 20 rounds. After the 4th, 9th, 14th, and 19th rounds of recommendation, user's trust changes and feedback according to the recommendation results, while other rounds of user's trust remain unchanged. When user's trust changes, the DQN-TB method obtains experience to train its neural network and has a further understanding of user's preference, the recommendation results are more inclined to user's preference, which can improve the success rate. Correspondingly, in CSIT method and CBR method, user's trust has changed dynamically, but the user's trust and other information have not been changed. The recommend results obtained from the original trust data will be more and more deviated from the user's preference.

Figure 14 shows the comparison of average success rate of DQN-TB, CSIT, and CBR corresponding to different rounds within 20 rounds under the dynamic change of user trust and preference. With the increase of recommended rounds, DQN-TB has higher accuracy, which is guaranteed by the flexibility of DQN-TB in response to dynamic changes. Meanwhile, due to the influence of CSIT and CBR on the calculation of perception of user preference, the accuracy of the two methods decreased with the dynamic change of conditions. The rate of decrease of accuracy of ordinary recommender methods varies with the change range of user preference, but it is also enough to explain the superiority of DQN-TB.

5.4 Response sensitivity

Implicit trust model and context matrix decomposition rely too much on user's neighbors and similar users. When the preference changes for several times, the analysis of relevant information will lose its accuracy and user's preference will not be timely feedback. CBR recommends multiple items for users at the same time, so it has certain coverage, and its sensitivity of response is better than CSIT. Table 3 shows the response sensitivity of each recommendation system after the user preference changes for 60 times in 450 rounds.

The data in Table 3 verify that DQN-TB has a good sensitivity to user preferences, which is related to dynamic rewards and experience learning in DQN-TB. Therefore, the application of DQN-TB to recommendation system can timely perceive the change of user preferences and adjust the selection of recommendation items accordingly.

6 Discussion

Based on the experiment, it can be seen that when the user's trust changes, our method can better capture the user's trust changes and give personalized recommendations that are more in line with user preferences. Comparing with other recommendation systems, we can see that our method has strong perception ability in dynamic trust. This method focuses on the dynamic

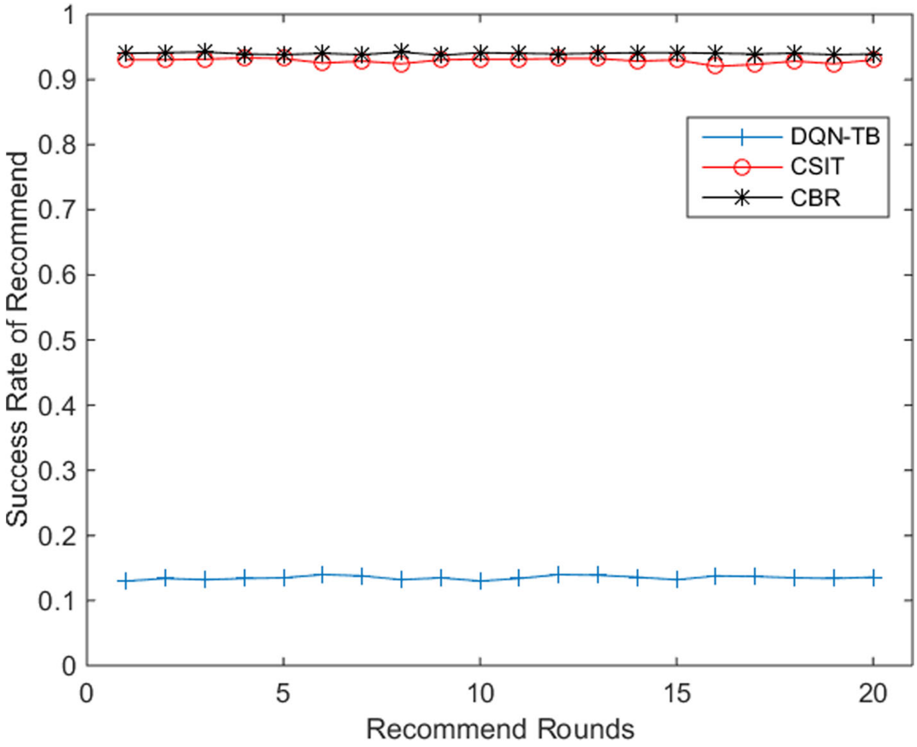


Fig. 12 Change of success rate with no dynamic change

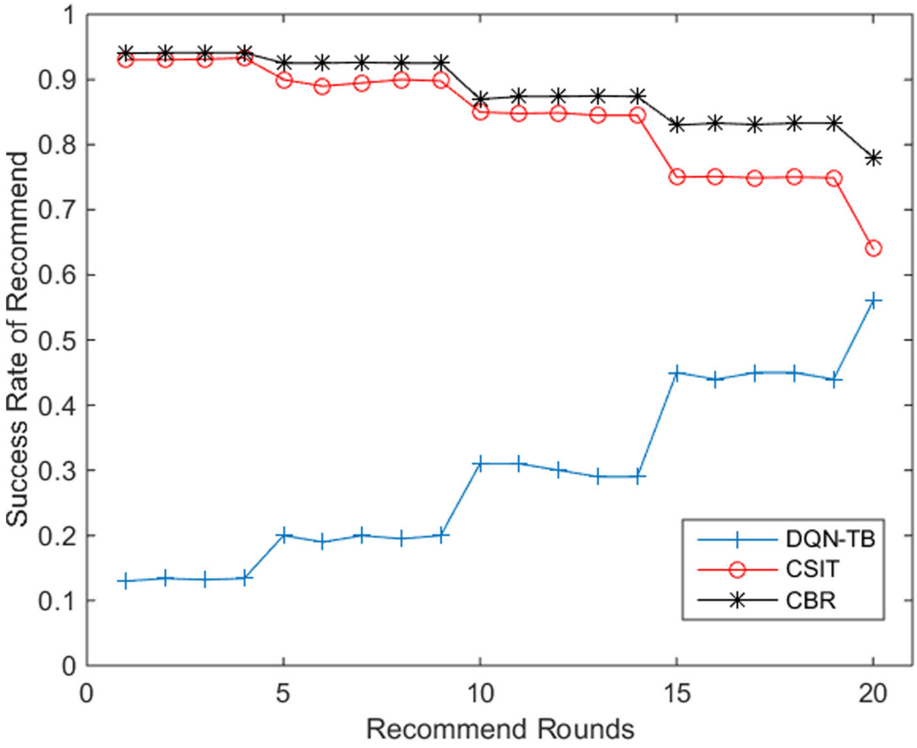


Fig. 13 Change of success rate with dynamic change a few times

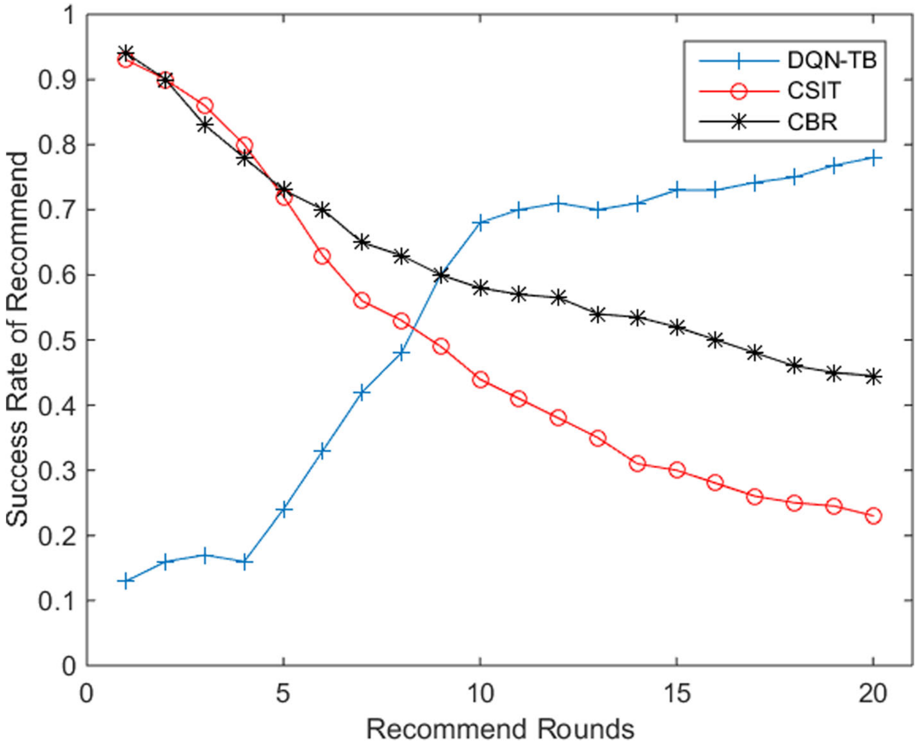


Fig. 14 Change of success rate in the recommend system

changes of user trust. It should be noted that in the state set of DQN-TB method, the factor affecting the selection of the next action are only the reward value obtained by the last action, and the correlation between states is not considered. In the future, this aspect will be studied and improved. At the same time, the calculation methods of trust and suggestions will be improved to make results of recommender method more accurate and effective.

7 Conclusions

In this paper, we propose a trust boost method based on dynamic trust and combined with reinforcement learning. This method senses the change of user preference through the dynamic change of user trust and learns from recommendation experience to provide more accurate recommendations, so as to increase and maintain user trust at a high level. Experiment shows that our method is efficient and accurate. Our method can also be applied to recommendation system to sensing change of user preference and making accurate recommendation.

Table 3 Response sensitivity of each recommendation system

Recommendation system	ANR	SEN
DQN-TB	86	0.809
CSIT	314	0.302
CBR	247	0.451

Abbreviations

CBR: Confidence-based recommendation; CSIT: A context-aware social recommender system via individual trust among users; DLATrust: A heuristic algorithm based on learning automata; DQN: Deep Q-Learning; DQN-TB: Trust boost via Deep Q-Learning; DyTrust: A dynamic algorithm based on distributed learning automata; HCC: Human-centered computing; RLS: Recursive least squares

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments as well as the National Natural Science Foundation of China (61572418), the China Postdoctoral Science Foundation under grant (2019T120732) for providing partial support to this research.

Authors' contributions

FXQ is in charge of the major theoretical analysis, algorithm design, and experimental simulation and wrote the manuscript. The others were in charge of part of the theoretical analysis and given suggestions to the manuscript writing. All authors read and approved the final manuscript.

Funding

The work was supported by the National Natural Science Foundation of China (61572418).

Availability of data and materials

Not available online. Please contact corresponding author for data requests.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer and Control Engineering, Yantai University, Yantai 264005, China. [2]Department of Computer Science, State University of New York at Binghamton, Binghamton, NY, USA.

Received: 31 August 2019 Accepted: 8 December 2019
Published online: 21 December 2019

References

1. Y.J. Wang, G.S. Yin, Z.P. Cai, A trust-based probabilistic recommendation model for social networks. J. Netw. Comput. Appl. **55**, 59–67 (2015)
2. J.L. Xu, S.G. Wang, B.K. Bharat, F. Yang, A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing. IEEE Trans. **15**(6), 3538–3547 (2019)
3. B.X. Zhao, Y.J. Wang, Y.S. Li, Y. Gao, X.R. Tong, Task allocation model based on worker friend relationship for mobile crowdsourcing. Sensors **19**(4), 921 (2019)
4. J. Li, Z. P. Cai, M. Y. Yan, Y. S. Li, Using crowdsourced data in location-based social networks to explore influence maximization. The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016), (2016)
5. J.L. Xu, S.G. Wang, N. Zhang, F.C. Yang, X.M. Shen, Reward or penalty: aligning incentives of stakeholders in crowdsourcing. IEEE Trans. **18**(4), 974–985 (2019)
6. C. Q. Jiang, S. X. Liu, Z. X. Lin, G. Z. Zhao, R. Duan, K. Liang, Domain-aware trust network extraction for trust propagation in large-scale henterogeneous trust networks. Knowl.-Based Syst.. 111(C), 237-247(2016)
7. S.R. Yan, K.J. Lin, X.L. Zheng, An approach for building efficient and accurate social recommender systems using individual relationship networks. IEEE Trans **29**(10), 2086–2099 (2017)
8. G. Mina, M.R. Meybodi, Trust propagation algorithm based on learning automata for inferring local trust in online social networks [J]. Knowl.-Based Syst. **143**, 307–316 (2018)
9. M. Ghavipour, M.R. Meybodi, A dynamic algorithm for stochastic trust propagation in online social networks: learning automata approach. Comput. Commun. **123**, 11–23 (2018)
10. J. You, J.L. Shangguan, L.H. Zhuang, N. Li, Y.H. Wang, An autonomous dynamic trust management system with uncertainty analysis. Knowl.-Based Syst. **161**, 101–110 (2018)
11. L.Y. Qi, X.Y. Zhang, W.C. Dou, C.H. Hu, C. Yang, J.J. Chen, A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Futur. Gener. Comput. Syst. **88**, 636–643 (2018)
12. L. Y. Qi, Y. Chen, Y. Yuan, S. C. Fu, X. Y. Zhang, X. L. Xu, A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web Journal. (2019) DOI: https://doi.org/10.1007/s11280-019-00684-y
13. L.Y. Qi, X.Y. Zhang, W.C. Dou, Q. Ni, A Distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. IEEE Journal on Selected Areas in Communications. **35**(11), 2616–2624 (2017)
14. H. W. Liu, H. Z. Kou, C. Yan, L. Y. Qi, Link prediction in paper citation network to construct paper correlated graph. EURASIP Journal on Wireless Communications and Networking. (2019)
15. W.W. Gong, L.Y. Qi, Y.W. Xu, Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wirel. Commun. Mob. Comput. **2018**, 1–8 (2018)
16. G.F. Liu, Y. Liu, A. Liu, Z.X. Li, K. Zheng, Y. Wang, Context-aware trust network extraction in large-scale trust-oriented social networks. World Wide Web **21**(3), 713–738 (2018)
17. Z.Q. Liu, J.F. Ma, Z.Y. Jiang, Y.B. Miao, FCT: a fully-distributed context-aware trust model for location based service recommendation. SCIENCE CHINA Inf. Sci. **08**, 101–116 (2017)
18. E.K. Wang, Y. Li, Y. Ye, S.M. Yiu, L.C.K. Hui, A dynamic trust framework for opportunistic mobile social networks. IEEE Trans **15**(1), 319–329 (2018)
19. V. Mnih, K. Kavukcuoglu, D. Silver, et al., Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
20. J.C.H.W. Christopher, *Learning from Delayed Rewards* (King's College, Cambridge, 1989)
21. Y.J. Wang, Z.P. Cai, Z.H. Zhang, Y.J. Gong, X.R. Tong, An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing. IEEE Trans **6**(3), 414–429 (2019)
22. Y.J. Wang, Z.P. Cai, G.S. Yin, Y. Gao, X.R. Tong, An incentive mechanism with privacy protection in mobile crowdsourcing systems. Comput. Netw. **102**, 157–171 (2016)
23. G. J. Zheng, F. Z. Zhang, Z. H. Zheng, Y. Xiang, N. J. Yuan, X. Xie, Z. H. Li, DRN: A deep reinforcement learning framework for news recommendation. https://doi.org/10.1145/3178876.3185994.
24. Z. J. Duan, W. Li, Z. P. Cai, Distributed auctions for task assignment and scheduling in mobile crowdsensing systems. The 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017). (2017)
25. J. Li, C.C. Chen, H.L. Chen, C.F. Tong, Towards context-aware social recommendation via individual trust. Knowl.-Based Syst. **2017**(127), 58–66 (2017)
26. F.S. Gohari, F.S. Aliee, H. Haghighi, A new confidence-based recommendation approach: combining trust and certainty. Inf. Sci. **2018**(422), 21–50 (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.