Applied Soft Computing Journal 104 (2021) 107213

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Cross-platform dynamic goods recommendation system based on reinforcement learning and social networks

Gang Ke [a,*], Hong-Le Du [b], Yeh-Cheng Chen [c]

[a] Department of Computer Engineering, Dongguan Polytechnic, Dongguan, China
[b] School of Mathematics and Computer Application, Shangluo University, Shangluo, China
[c] Department of Computer Science, University of California, Davis, CA, USA

ARTICLE INFO

Article history:
Received 26 August 2020
Received in revised form 24 December 2020
Accepted 14 February 2021
Available online 2 March 2021

Keywords:
Recommendation system
Collaborative filtering
Reinforcement learning
Dynamic prediction
Big data

ABSTRACT

Aiming at the problems of cold start, gray sheep and sparsity of the traditional collaborative filtering recommendation system, this paper proposes a cross-platform dynamic goods recommendation system based on reinforcement learning and edge computing. First of all, this system models the current friendship relationship networks and potential friendship relationship networks, it also constructs two layers preference prediction models. Then, we consider the frequent change characteristic of social networks and shopping platforms, we design a dynamic reinforcement learning method and edge computing to learn the minimized entropy loss error. Finally, we finish the validation experiments based on the real datasets, the results show the proposed system realizes better link prediction accuracy, and using our proposed system can obtain an obvious increase in the accuracy compared to the existing of collaborative filtering recommendation systems.

1. Introduction

Nowadays, some large-scale network platforms have gathered a large number of users, and have established certain social relations and social circles among these users. These platforms include: Entertainment short video platform “tiktok”, shopping platform “taobao”, social class platform “Sina Weibo”, knowledge sharing platform “Zhihu” and music player platform “NetEase cloud music” and other [1]. A large number of social information is generated on these platforms every day, such as Sina Weibo, which generates about 100 million new Weibo and 60 million active users every day. How to use the value of these information is a hot research direction of major enterprises [2].

A large number of studies show that the development and utilization of social network information can bring positive effects to many applications, such as: combining the public security prediction of social network [3], combining the distribution design of water supply system of social network [4], combining the knowledge sharing algorithm of social network [5], etc. Peng et al. [6] proposes a cross platform recommendation model, which considers the user’s interest in Sina Weibo, and combines the user’s behavior characteristics in Zhihu platform to complete the recommendation of Weibo friends by combining the user’s interest in all relevant platforms. This method improves the recommendation accuracy and coverage of friends comprehensively by combining the data of multiple social platforms. The integration of multiple social platforms to achieve friend recommendation is essentially to predict the social relationship between users from different dimensions, and its research focus is to extract data from different social platforms. The research focus of this paper is to study the positive impact of cross platform social relations on product recommendation. For example, user Li Si and user Wang Wu are frequent friends on Sina Weibo, and they often publish microblogs about football, and Li Si purchases football related products on Taobao. Then Taobao can recommend the products purchased by Li Si to Wang Wu. On the other hand, if Li Si and Wang Wu are not friends on Sina Weibo, and Li Si and Wang Wu have very similar purchase records on Taobao, then they can be recommended to be friends on Sina Weibo. Therefore, cross platform combination of social network and product recommendation system is a way to benefit each other. Some scholars have carried out similar research: Tong et al. [7] analysis concludes that users with similar consumption records are likely to become friends in the future; Sheng et al. [8] believes that friends with the same interest in social networks are likely to buy goods related to that interest. However, these methods all assume that the structure of social network and users’ shopping interest are static and fixed, which is contrary to the actual social network with drastic changes, and users’ shopping interest will also change with time. Cao et al. [9] proposed a user clustering search algorithm combined with inverted index in the field of information retrieval and adopted “member strategy” to shorten the calculation time

* Corresponding author.
E-mail address: kegang95@126.com (G. Ke).

https://doi.org/10.1016/j.asoc.2021.107213

of nearest neighbor. This method can effectively improve the scalability of collaborative filtering recommendation system on the premise of ensuring the correctness of recommendation. Fang et al. [10] aiming at the poor performance of the traditional collaborative filtering algorithm when the score matrix is sparse, an improved recommendation algorithm based on sigmoid function is proposed. The algorithm can solve the problem of data sparsity and effectively improve the prediction accuracy of the traditional recommendation algorithm.

According to the current research conclusion of collaborative filtering recommendation system [11], sparsity problem, cold start problem and gray sheep user problem are the main problems faced by the recommendation system. The common point of these problems is that a large number of users do not have similar users or similar users are very few. In order to solve the above problems, this paper uses cross platform social relationship prediction method to predict these user preferred products, and establishes a dynamic social network model, and designs a dynamic reinforcement learning(RL) method is used to predict user preferences dynamically, in order to improve the performance of commodity recommendation system and friend recommendation system, and to meet the dynamic changes of social network topology and user shopping interest.

2. Dynamic social network model

2.1. Dynamic social network model

Suppose the recommendation system contains a product set $P = (|P| = N)$ and a user set $U = (|U| = M)$, users have two behaviors of shopping and social interaction, and the social network structure and user preferences change dynamically with time. $G = (U, E)$ represents undirected social network graph, U represents user set in graph, E represents edge set, $e{<}u, w{>}$ represents connection between user u and w, where $u, w \in U$. Users of social networks have a wealth of personal information, such as geographic location, work unit, school, city, and age. For each user, create a vector to save personal files, expressed as $Q =< q_1(u), q_2(u), \ldots, q_m(u)>$, all friend connections of user u are expressed as $F(u) = \{f | f \in U \cap e{<}u, f{>} \in E\}$. Finally, user u is represented as tuple U: u: $<q_1(u), q_2(u), \ldots, q_m(u), F(u)>$, if part of the file information of user u is unknown or has no friends, the corresponding element of the tuple is set to NULL.

For a given social network graph $G = (U, E)$, the friend set and attribute vector of user u are expressed as u: $<q_1(u), q_2(u), \ldots, q_m(u), F(u)>$, J represents a subset of users, and s: $<q_1(s), q_2(s), \ldots, q_m(s)>$ represents the attribute vector of the target user s. The goal of this article is to predict and create a set of friends J for a user s with few connections, denoting the connection with low probability as K and the connection with high probability as I. The activity of each user at time t is described as social behavior and shopping behavior, respectively represented as social matrix $S^t \in R^{M \times N}$ and shopping matrix $P^t \in R^{M \times N}$. If user u purchases product e at time t, then P^t_{ue} indicates that he prefers the product, and $P^t_{ue} = 0$ indicates that u is not interested in e at time t. $S^t_{uw} = 1$ indicates that there is a friend connection between users u and w at time t, and $S^t_{uw} = 0$ indicates that there is no connection.

2.2. Direct prediction model

(1) Direct friend relationship prediction

Given a user subset J, the target user s and some users in J have the same attribute information. The basic idea is to search the user subset I similar to s from J, and establish a friend connection according to the probability of similarity. In this paper, a computing method for establishing connection between user $u \in J$ and user s is given. The direct connection prediction model combines multiple social attributes, which are recorded as ω_{us}.

Firstly, social information between several pairs of users is extracted. Each pair of users is represented by social feature y_i and tag l_i. The default value of l_i is 0, and $l_i = 1$ indicates that the pair of users are friends. Then we calculate the probability that u is a friend between u and s according to y_i and l_i, and finally decide whether u belongs to subset K or I according to the probability. The direct connection prediction problem is modeled as an optimization problem as follows:

$$\min F(x) = \frac{1}{2}\|x\|^2 + \partial \sum_{j=1}^{o} \alpha_j \tag{1}$$

where, j represents the jth pair of nodes, o represents the total number of user pairs, ∂ is a constant, and α_j represents the relaxation variable of optimization.

(2) Direct shopping preference prediction

The traditional collaborative filtering model regards the people with similar history as a local group, and assumes that the users prefer the products popular in the local group. But this traditional idea has the problems of sparsity, gray sheep users and cold start, and also has Matthew effect. According to the research conclusions of many experts, the use of developed social networks can enhance the recommendation effect of collaborative filtering. Users' purchase preferences are defined as:

$$P^t_{ue} = (1 - \varphi_u)\, g(u, e, t) + \varphi_u h(u, e, t) \tag{2}$$

where: $0 \le \varphi_u \le 1$, P^t_{ue} indicates the preference of user u to purchase item e. The non negative parameter φ_u is used to balance the relationship between functions h and g, that is, the influence of social relationship and collaborative filtering on user shopping. The φ_u of each user is different, because some users are greatly affected by social platforms, while some users are more affected by their own purchase records.

The function $h(u, e, t)$ indicates the influence of social network on user U's interest in purchasing e at time t, and social influence indicates the communication ability of information in social network, which promotes user u to prefer to buy popular goods in his friend circle. For example, viral marketing is a recommended method to maximize the influence of social communication [12]. Combining social influence and historical purchase record, the function $h(u, e, t)$ is defined as the following formula:

$$h(u, e, t) = \frac{\sum_{t'=1}^{t-1} \sum_{d \in V_u^{t'}} Z^{t'}_{de} S^{t'}_{ud} P^{t'}_{ue}}{\sum_{t'}^{t-1} \sum_{d \in V_u^{t'}} Z^{t'}_{de} S^{t}_{ud}} \tag{3}$$

where: In the formula: the default value of Z^t_{de} is 0, $Z^t_{de} = 1$ means that d purchased goods at time t, and S^t_{ud} means the social influence between user u and d. Assuming that social influence changes with the dynamic change of network structure, Adamic/Adar index is used to calculate social influence, which is a time variable. The calculation method is as follows:

$$S^t_{ud} = \frac{1}{\sum_{b \in V_u^{t-1} \cap V_d^{t-1}} \log\left(\left|V_b^{t-1}\right|\right)} \tag{4}$$

2.3. Potential prediction model

(1) Potential friend relationship prediction

First, extract the friend set of user u and s. If the friend sets of u and s are similar, establish a connection between them, and record it as $e{<}u, s{>}$. This kind of friend relationship is called potential friend relationship. If the friend set attribute of u is similar to s, a potential attribute connection is established for u and s.

Suppose user u has 3 friends e_1, e_2 and e_3, e_1 and s have the same university and company of personal files, and s has a connection with e_2 and e_3 (friends of u). In order to evaluate the potential relationship between s and u, various connections between s and u's friends need to be quantified. The fewer connectionless and more potential connections between s and u's friend sets, the greater the likelihood that s and u will become friends. Therefore, the potential relationship score is defined as:

$$ls = \frac{1}{1 + e^{-\varphi(b-\partial c)}} \tag{5}$$

where, φ is the index parameter. If there is no connection between s and u's friend sets, the potential relationship will be punished to some extent, (b-∂c) indicates the potential relationship between s and u, ∂ is the penalty factor, and c and b are the number of no connection and the number of potential connections respectively. If most attribute information of s and u cannot be obtained, then it is necessary to determine whether the connection $e<u, s>$ exists through the score of the potential relationship.

(2) Forecast of potential shopping preferences

Given item e and user u, u's potential purchase preference at time t is defined as:

$$cp\left(P|X,Y\right) = \prod_{t=1}^{T}\prod_{u=1}^{M}\prod_{e=1}^{N}\gamma\left\{P_{ue}^{t}\left|\left\langle X_u^t, Y_e\right\rangle, \varepsilon_P^2\right.\right\}^{Z_{ue}^t} \tag{6}$$

where, $\gamma(\vartheta, \varepsilon^2)$ represents a normal distribution with mean ϑ and variance ε^2, Z_{ue}^t is the indicator variable, and its default value is 0. If user u purchases item e at time t, then $Z_{ue}^t = 1$. $X_u^t \in R^{Q\times 1}$ represents the potential purchase vector of u. Let $Y \in R^{N\times Q}$ be a potential item matrix and $Y_e \in R^{N\times Q}$ be the potential item vector. Considering the characteristics of user preferences that change dynamically over time, the user's potential preferences are summarized in the form of a matrix sequence, denoted as $X = \{X^1,\ldots, X^t,\ldots, X^T\}$. In order to prevent the over-fitting of the preference prediction, add the following zero mean Gaussian to the potential matrix of the project:

$$cp\left(Y\left|\varepsilon_Y^2\right.\right) = \prod_{e=1}^{N}\gamma\left(Y_e\left|0, \varepsilon_Y^2 E\right.\right) \tag{7}$$

In the case of limited purchase information of target users, the goal of this paper is to model the dynamic change process of potential purchase matrix X of users. Each user's social influence and their historical shopping preferences affect their future potential shopping preferences. Establish the following two potential purchase preferences for each user:

$$cp\left(X_u^t\right) = \gamma\left(X_u^t\left|\overline{X_u^t}\right., \varepsilon_Y^2 E\right) \tag{8}$$

$$\overline{X_u^t} = (1-\vartheta_u)X_u^{(t-1)} + \vartheta_u \sum_{\omega\in\gamma_u^{(t-1)}} \frac{S_{u\omega}^{t-1}}{N_u^{t-1}} X_\omega^{t-1} \tag{9}$$

where $\forall u \in X$, $0 \le \vartheta \le 1$. $S_{u\omega}^{t-1}$ represents the influence of user ω on u at time t-1, and N_u^{t-1} represents the normalized constant of all friends of u, which constant satisfies $\sum_{\omega\in\gamma_u^{t-1}} \frac{S_{uw}^{t-1}}{N_u^{t-1}} = 1$. Social network's social influence score $S_{u\omega}^{t-1}$ reflects the similarity between two users, and a potential structure factor F_u^{t-1} is defined for each user. If the ability of two users to resist social influence is stronger, the potential structure factor of the two users is more similar. The anti-social influence indicators are defined as:

$$S_{u\omega}^{t-1} = j\left(\left\langle F_u^{t-1}, F_\omega^{t-1}\right\rangle\right) = \frac{1}{1+\exp\left(-\left\langle F_u^{t-1}, F_\omega^{t-1}\right\rangle\right)} \tag{10}$$

where, logistic function $j(a) = 1/(1+ex(\text{-}a))$ is used to normalize the influence index to the interval of [0, 1]. In the initial stage ($t = 1$), the social network has not been established yet, so the purchase preference of each user is directly estimated without considering the social influence. Assuming that the potential preference vector is subject to the gaussian distribution of zero mean, the potential purchase preference matrix is converted into the following equation:

$$cp\left(X\left|\varepsilon_X^2, \varepsilon_{X1}^2\right.\right) = \prod_{u=1}^{X}\gamma\left(X_u^1\left|0\right., \varepsilon_{X1}^2 E\right)\prod_{t=2}^{T}\gamma\left(X_u^t\left|\overline{X_u^t}\right., \varepsilon_Y^2 E\right) \tag{11}$$

(3) Problem model of user grouping

In order to realize the user's connection prediction and purchase preference prediction, a dynamic reinforcement learning technology is designed to dynamically predict. Suppose the input of the predictor is a sample vector ω, and its output is a class label, denoted as $DL_i(\omega)$.

For the input vector ω, the probability of its class label is calculated as:

$$p\left(label = DL_j|\omega\right) = \frac{e^{DL_j}}{\sum_i e^{DL_j}} \tag{12}$$

By minimizing the entropy loss [13] -$\sum_j F_j \log l_j$, we learn the class labels of social connections, where F_j and l_j are the predicted class labels and connection values respectively, and then use the collaborative filtering framework to predict the recommended list of target users.

3. Prediction method based on dynamic reinforcement learning

In order to satisfy the dynamic change of social network structure and users' purchasing preference, this paper designs a dynamic reinforcement learning method.

3.1. Dynamic reinforcement learning environment

Fig. 1 shows the schematic diagram of RL algorithm. In the figure, S, Q, A and R represent state, action state value prediction function, action and reward respectively [14–16]. RL is composed of two entities, environment and agent. The interaction between agent and environment is maintained. In Fig. 1, the external loop describes the interaction of state, action and reward between agent and environment, and the internal loop describes agent evaluation state action [17–19]. This paper assigns an agent to the target connection to be predicted, and the state S_t represents the class label of the connection. At time t, the agent analyzes the current state, strategy (state to action mapping function) and prior information, and then decides to choose action A_t, that is, to adjust the purchase preference. At time t+1, the agent analyzes the current action, the last action and the environment, then outputs a new state $S_{t+1} = S(S_t, A_t)$ and generates reward R_t [20–23]. The agent learns the effect of previous actions according to the reward, and then tries other actions to improve the reward [24–27].

The environmental state is defined as the entropy loss of the connection, which is S (A) in Fig. 1. Setting the environmental state as a separate entropy loss has the following advantages: the dimension of state variables is low, the learning process is adjusted quickly and accurately, and the results are highly interpretable and easy to debug [28,29], which is conducive to the initialization of RL model [30,31]. Using a discrete action space, the agent sets up a total of 20 user attribute combinations. The reward is the cross-entropy loss value, which is R (S, A) in Fig. 1.

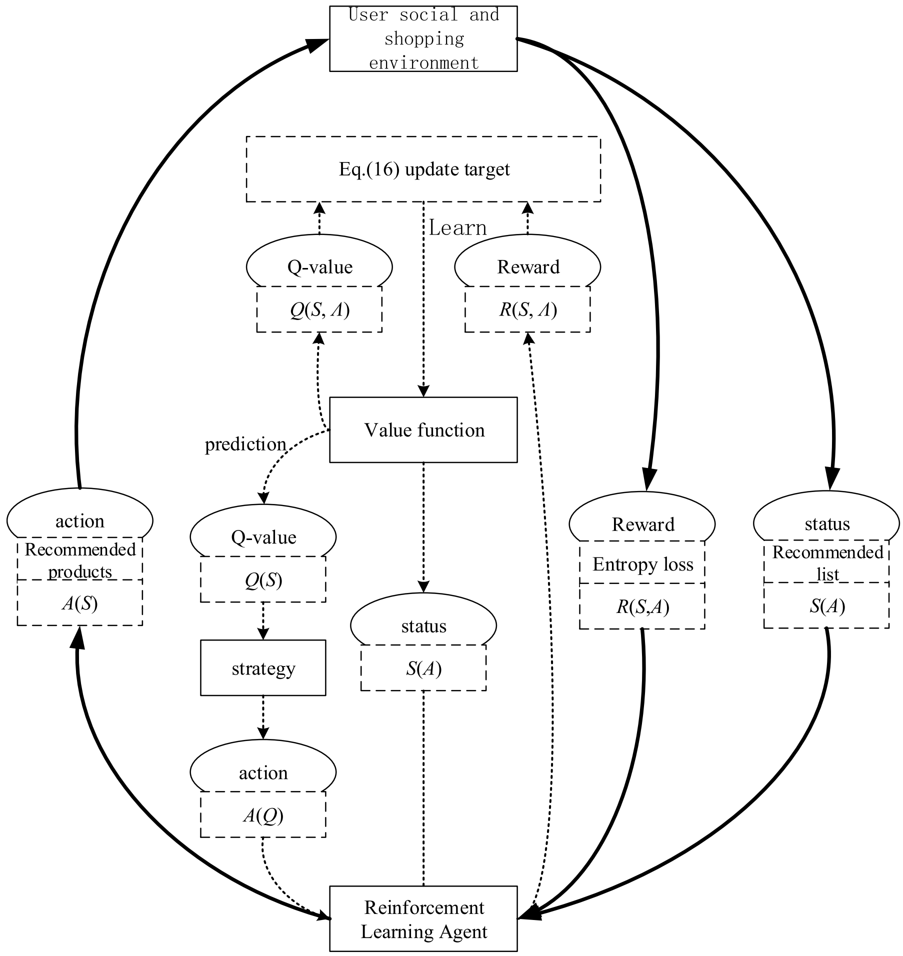


Fig. 1. Schematic diagram of the RL algorithm.

3.2. Agent for dynamic reinforcement learning

In order to complete the task of optimization, agent uses a function model and a strategy set. The nonlinear fitting ability of radial basis function (RBF) is better than the simple gradient descent method [32,33], and the convergence speed is faster than the artificial neural network model. Therefore, this paper uses RBF network as the function approximation model. Fig. 2 shows the learning method of reinforcement learning agent [34–36].

The experiment uses the Scikit-learning library based on Python to implement the RBF function [37–39], which is approximately the following formula:

$$x = \frac{\sqrt{2}}{\sqrt{k}} cos\left(w^{RBF} s + c^{RBF}\right) \tag{13}$$

where, $w^{RBF} \sim N(0, \sqrt{2\gamma^{RBF}})$, $c^{RBF} \sim U(0, 2\pi)$, x is the transformation eigenvector of the result [40,41], s is the state variable of the input, k is the number of original features of Monte Carlo samples [42], w_{RBF} is a k-element vector of random RBF weight, c_{RBF} is a k-element vector of the random RBF offset value, γ^{RBF} is the variance parameter of a normal distribution. In the experiment, $\gamma^{RBF} = \{5,2,1,0.5\}$, the number of k is set to 500, and the parameters of RBF are quickly determined by the grid search algorithm.

Using RBF to transform a state variable to generate a total of 2000 features, a simple linear regression model is used to regularize the 2000 features, the calculation formula is as follows:

$$Q\ (w_a, s) = w_a RBF(s) = w_a x \tag{14}$$

where w_a is the weight vector of action a, s is the state variable, $RBF()$ is the radial basis function, x is the result eigenvector, Q is the value of action a in state s.

The Agent has 20 actions in total. The agent selects an action based on the current policy, which is $A(Q)$ in Fig. 1. In this paper, Boltzmann-Q strategy is used to implement *A(Q)*, and its calculation formula is:

$$\pi^B = (s) = P\left[A_t = a | S_t = s\right] = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_{a' \in A} e^{\frac{Q\left(s,a'\right)}{\tau}}} \tag{15}$$

where, A is the set of all actions, a' represents all actions except a, τ is the temperature parameter of Boltzmann distribution, τ controls the balance between global search and local development of RL. The action range of Boltzmann Q strategy is [max(argmax$_a Q(s, a)$-3, 0), min(argmax$_a Q(s, a)$-3, 20)]. After the agent selects an action, it needs to select the action at the next time point. At this time, it looks for the action that meets the environmental conditions and the cross entropy loss is better than the current time.

The value objective function of RL is defined as:

$$Q\ (S_t, A_t) = Q\ (S_t, A_t) + \alpha \left[R_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right] \tag{16}$$

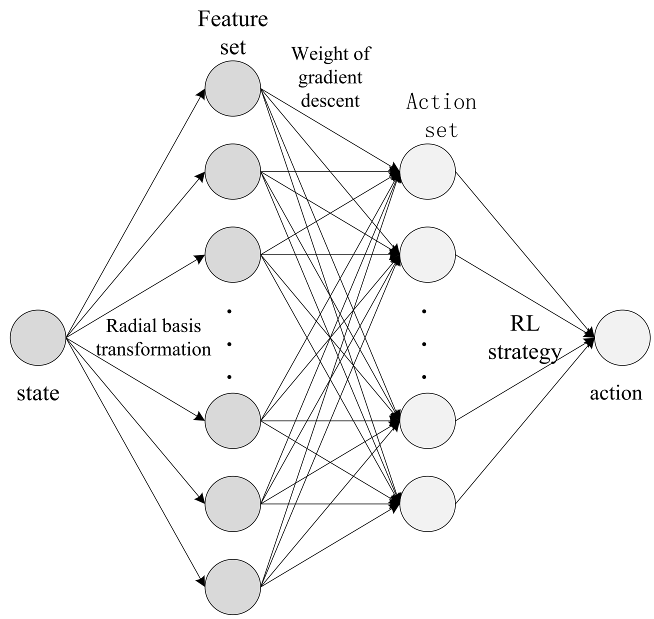


Fig. 2. Learning method of RL action.

where, γ is the discount factor, α is the learning rate, $Q(s, a)$ is the action value function, Q function describes the expected discount reward when action a is adopted in state s, and its error is as follows:

$$R_t + \gamma \max_a Q\left(S_{t+1}, a\right) - Q\left(S_t, A_t\right) \tag{17}$$

The weight updating rules of RL gradient descent method are as follows:

$$w_\alpha \leftarrow w_a + \alpha \left[R_t + \gamma \max_a Q\left(S_{t+1}, a\right) - Q(S_t, A_t)\right] S_t \tag{18}$$

where, $Q(S_t, A_t)$ is the current prediction result of the model, R_t+γ maxaQ(S_{t+1}, a) is the target, and S_t is the gradient of weight.

Agent uses different actions to access each state, the learning rate is set as the decreasing change, different update function is used to improve the approximation quality, and the final estimate function converges to the actual value function of the problem. The formula for calculating the decline of learning rate is:

$$\alpha_t = \frac{\alpha_0}{t^{in}} \tag{19}$$

where, α_t is the learning rate of time t, α_0 is the initialization learning rate, and *in* is the decreasing factor of the learning rate.

4. Simulation experiments and results analysis

4.1. Experimental data and experimental environment

Peng et al. [6] proposes a cross-platform data collection method, and uses the API provided by sina weibo platform and taobao platform to obtain various information corresponding to users by passing in different user ids, thus effectively solving the problem that it is difficult to obtain information from different platforms. Since taobao and sina weibo are strategic partners, taobao provides an entry point for "weibo account login", crawls the personal profile information of taobao users who log in through the "weibo account", and crawls out the score and comment content of these users on the products.

Weibo 's social network data only extracted the two relationships of "follow each other" and "friends". These two relationships are considered as friend relationships (there is a connection), and other relationships such as "one-way attention" are excluded. The ID and personal profile of each weibo user are extracted, personal files include: occupation, current city, education and gender. Taobao's shopping data extracts the shopping records of users within 12 months (June 2018 to June 2019). Due to the limited machine performance of the experiment, 100 users with rich personal files (more than 15 valid attributes of personal data, high activity of microblog) and purchase records (more than 30 shopping records in a year) are selected for simulation experiment. 14 of these 100 people failed to establish good friend connection between June 2018 and December 2018, and established friend connection between January 2019 and June 2019. Shopping information includes data such as users' shopping history, user ratings and comments, and taobao's "favorable" products are considered as users' purchasing preferences.

Table 1
Parameter settings for dynamic reinforcement learning.

Parameter	Value
γ	0.95
α_0	$1\times e^{-5}$
τ	5

The algorithm of this paper is implemented using Python programming, and the RBF function of reinforcement learning is implemented based on Python's Scikit-learning library. The experimental environment is Intel (R) Core (TM) i7-2600K CPU 3.40 GHz, 16 GB memory. After some pretreatment experiments, the parameters shown in Table 1 were used.

4.2. Performance evaluation method

AUC (area under the curve), accuracy, recall rate and F-measure are used as four commonly used evaluation indexes. AUC evaluates the accuracy rate of the predicted user's purchase preference. The calculation method is as follows:

$$AUC = \frac{(p + 0.5q)}{r} \tag{20}$$

where, p is the number of existing friends (or buying preference), q is the number of times two users give the same score to the same product, and r is the number of times two users compare scores separately.

The precision is denoted as P, and the calculation method is:

$$P = \frac{n}{H} \tag{21}$$

where, n is the number of user pairs in the user subset T_H, and H is the number of user pairs that prefer the same product.

The recall rate is denoted as R, and the calculation method is as follows:

$$R = \frac{n}{\mathrm{N}} \tag{22}$$

where, N is the total number of friend connections, and n is the predicted number of friend connections.

F-measure is defined as:

$$F\text{-}measure = \frac{2 \times P \times R}{P + R} \tag{23}$$

where, P is precision, R is recall rate.

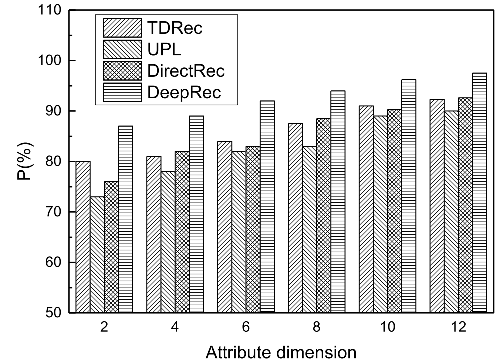
4.3. Friendship prediction experiment

Due to the lack of research on combining social network and commodity recommendation system at the same time, two social network friend relationship prediction algorithms are used as a comparison method, which are the enhanced social recommendation algorithm (TDRec) [43] and the dynamic friend prediction algorithm (UPL) [44]. TDRec is a friend recommendation algorithm for social networks, which fully considers the one-way attention connection and trust propagation relationship between social users. UPL only analyzes the profile of social users, and analyzes the similarity between users through probability distribution. The model of this method includes two methods: direct friend prediction and potential friend prediction. In this paper, the recommendation method containing only direct friend prediction is called DirectRec for short, and the recommendation method containing both direct friend prediction and potential friend prediction is called DeepRec for short.

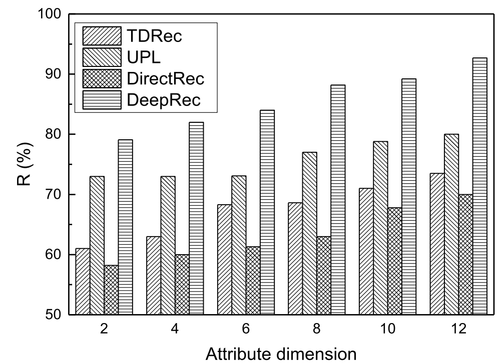
The goal of the friend connection prediction problem is to identify potential friend relationships that may become friends in the future. The data of 100 users in the experimental data were tested from June 2018 to December 2018, and the data of 100 users in January 2019 to June 2019 were used as the verification data. Each group of experiments was repeated for 20 times independently, and the average value of each performance index was counted for 20 times as the experimental result. Fig. 3 shows the comparison results of these join prediction models, and the dimensions of user attributes are selected as {2, 4, 6, 8, 10, 12}. The figure shows that the prediction accuracy of TDRec is higher, but the recall rate is lower; UPL has obtained a higher recall rate by analyzing the recommendation list generated by the user profile, but the accuracy is also lower. According to the F-measure value and AUC value, the direct prediction algorithm DirectRec in this paper does not show obvious advantages, but the DeepRec algorithm in this paper shows obvious advantages. In the end, it can be concluded that the performance of friend recommendation can be effectively improved through the prediction of potential friends, and the cross-platform recommendation system in this paper helps to improve the performance of friend relationship prediction in social networks.

4.4. Product recommendation experiment

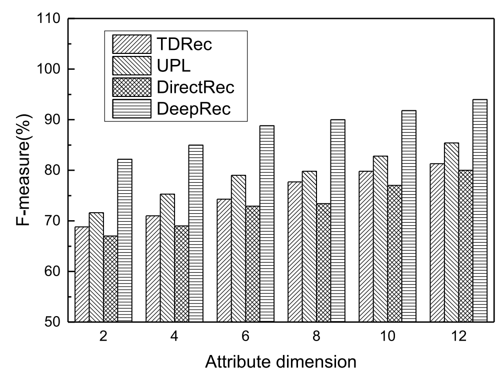
At present, there are few researches on combining social network and commodity recommendation system at the same time. Two single platform recommendation algorithms are used as



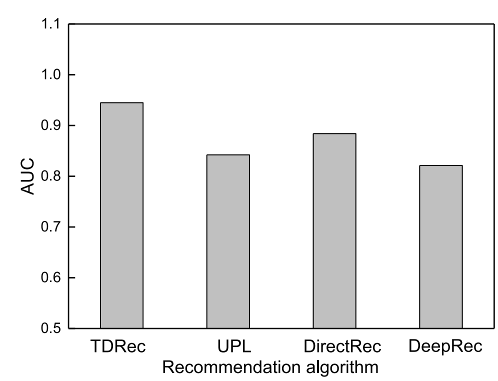
(a) Average prediction accuracy



(b) Average forecast recall rate

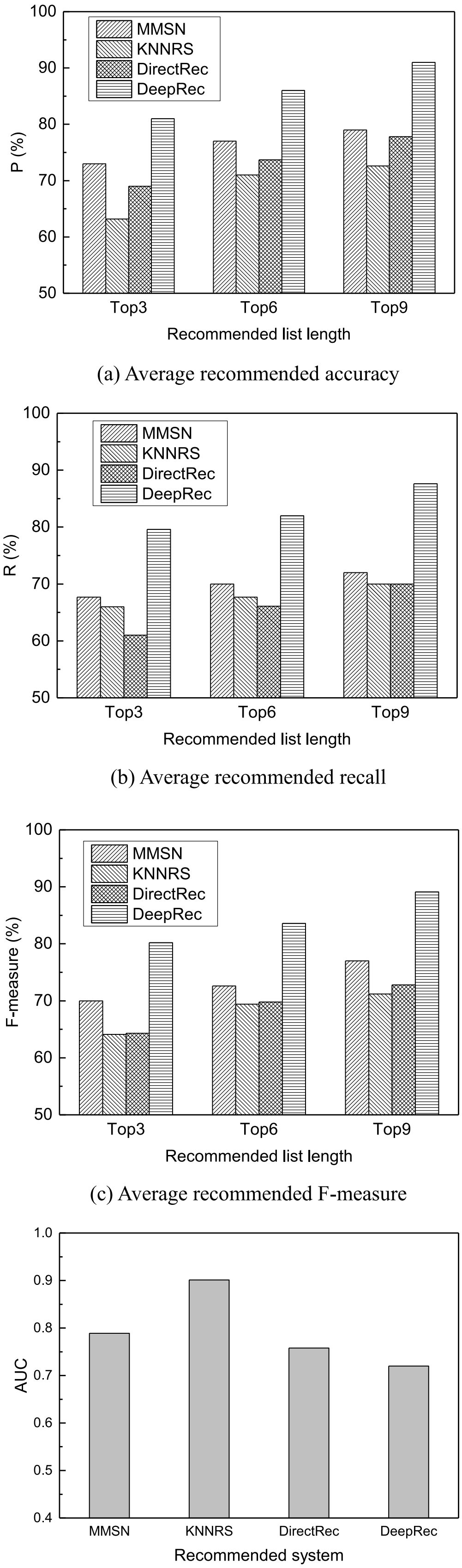


(c) Average predicted F-measure value



(d) Average predicted AUC value

Fig. 3. Experimental results of friend relationship prediction.



(a) Average recommended accuracy

(b) Average recommended recall

(c) Average recommended F-measure

(d) Average recommended AUC

Fig. 4. Experimental results of product recommendation.

comparison methods, namely, Markov social model based recommendation algorithm (MMSN) [45] and adaptive collaborative filtering recommendation system (KNNRS) [46]. MMSN is a recommendation algorithm for e-commerce websites considering weak social relations. This algorithm considers weak social relations among users in e-commerce website, and uses Markov random field to model weak social relations and predict future similarity among users. KNNRS is an adaptive collaborative filtering recommendation algorithm. An adaptive k-neighbor classifier is designed to predict the user's preferences in real time. This model includes two methods: direct shopping preference prediction and potential shopping preference prediction. In this paper, the recommendation method which only includes direct shopping preference prediction is called DirectRec for short, and the recommendation method which includes both direct shopping preference prediction and potential shopping preference prediction is called DeepRec for short. The goal of the shopping preference prediction problem is to identify the potential commodities that may be purchased in the future, test the data of 100 users of the experimental data in the period from June 2018 to December 2018, and take the data of 100 users in the period from January 2019 to June 2019 as the validation data. Each group of experiments was repeated for 20 times independently, and the average value of each performance index was counted for 20 times as the experimental result. The comparison results of these recommendation systems are shown in Fig. 4, and the experimental results with the recommended list length of 3, 6 and 9 are calculated respectively. As shown in the figure, MMSN has high prediction accuracy and high recall rate. KNNRS is an adaptive collaborative filtering recommendation system with low precision and recall rate, which shows that the use of social relations can effectively enhance the performance of the recommendation system. Comparing MMSN with the algorithm in this paper, the recommendation performance of MMSN is slightly better than that of the direct recommendation algorithm in this paper, and MMSN is similar to the direct preference prediction algorithm in this paper. The predictive performance of DeepRec in this paper is obviously better than that of MMSN and DirectRec, and the DeepRec system in this paper also supports dynamic application scenarios, while MMSN only supports static and stable application scenarios.

5. Conclusion

This paper proposes cross-platform user friend prediction and user shopping preference prediction algorithms. Direct friend relationship and potential friend relationship models are established, as well as direct shopping preference model and potential shopping preference model. In order to meet the characteristics of dynamic changes in shopping preferences of social networks and users, the ability to correct erroneous prior knowledge and dynamic evolution using reinforcement learning was implemented to achieve dynamic friend prediction and shopping preference prediction. The experimental results based on the actual data show that the cross platform mechanism effectively improves the performance of the recommendation system, and has the ability of dynamic learning.

With the expansion of the scale of social networks and the increase of the number of users, the computing cost of this algorithm is higher. In the future, we will focus on the research of distributed computing, and improve the scalability and time efficiency of this algorithm.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Dongguan Social and Technology Development Project (No. 2020507156684, No. 2020507156694); Technical Skills Expert Project of Dongguan Polytechnic (No. 2019JY05); Scientific Research Fund of Dongguan Polytechnic (No. 2020a11). This work was also supported by the natural science foundation research project of Shanxi Province Foundation of China (No. 2020KRM156)); Science and technology plan project of Shangluo of China (No. SK2019-84); Science and Technology Research Project of Shangluo University of China (No. 18SKY014). Science and Technology Innovation Team Building Project of Shangluo University of China (No. 18SCX002).

References

[1] A.H. Li, J. Wang, Y.Q. Wang, Research on CSER rumor spreading model in online social network, in: International Conference on Emerging Internetworking. 2018.
[2] C. Liu, X. Lu, Analyzing hidden populations online: topic, emotion, and social network of HIV-related users in the largest Chinese online community, BMC Med. Inf. Decis. Mak. 18 (1) (2018) 2.
[3] H. Hosseinmardi, S.A. Mattson, R.I. Rafiq, et al., Prediction of cyberbullying incidents on the instagram social network, Mucosal Immunol. 2 (2) (2015) 144–155.
[4] B.M. Brentan, E. Campbell, G.L. Meirelles, et al., Social network community detection for DMA creation: criteria analysis through multilevel optimization, Math. Probl. Eng. 2017 (7) (2017) 1–12, 2017, (2017-02-20).
[5] S.W. Lin, Y.S. Lo, Mechanisms to motivate knowledge sharing: integrating the reward systems and social network perspectives, J. Knowl. Manage. 19 (2) (2015) 212–235.
[6] Peng Jian, Wang tuntuntun, Chen Yu, Liu Tang, Xu Wenzheng, User recommendation based on cross-platform online social networks, J. Commun. 39 (3) (2018) 147–158.
[7] Z. Tong, J. Hu, P. He, et al., Exploiting homophily-based implicit social network to improve recommendation performance, in: International Joint Conference on Neural Networks, 2014.
[8] W. Sheng, M.S. Haghighi, C. Chao, et al., A sword with two edges: Propagation studies on both positive and negative information in online social networks, IEEE Trans. Comput. 64 (3) (2015) 640–653.
[9] H.J. Cao, K. Fu, Research on clustering search method in collaborative filtering recommendation system, CEA 50 (5) (2014) 16–20.
[10] Y.N. Fang, Y.F. Guo, H.C. Hu, etal, Improved collaborative filtering recommender algorithm based on sigmoid function, Appl. Res. Comput. 30 (6) (2013) 1688–1691.
[11] A. Roy, S. Banerjee, M. Sarkar, et al., Exploring new vista of intelligent collaborative filtering: A restaurant recommendation paradigm, J. Comput. Sci. 27 (2018) 168–182.
[12] Xia Xiufeng, Wang Na, Li Xiaoming, Prediction model for recommended effectiveness of core group in web viral marketing, J. Chin. Comput. Syst. 35 (2) (2014) 244–248.
[13] Cheng Jianhua, Mao Shiyun, Wang Dehui, Bayesian estimation of parameter of Poisson distribution under weighted balanced entropy loss function, J. Jilin Univ. Sci. (4) (2019) 839–843.
[14] F. Luo, W. Guo, Y. Yu, et al., A multi-label classification algorithm based on kernel extreme learning machine, Neurocomputing 260 (2017) 313–320.
[15] Y. Niu, J. Chen, W. Guo, Meta-metric for saliency detection evaluation metrics based on application preference, Multimedia Tools Appl. 77 (20) (2018) 26351–26369, Online Publication.
[16] Y. Niu, W. Lin, X. Ke, et al., Fitting-based optimisation for image visual salient object, IET Comput. Vis. 11 (2) (2017) 161–172.
[17] J. Wang, X.M. Zhang, Y. Lin, et al., Event-triggered dissipative control for networked stochastic systems under non-uniform sampling, Inform. Sci. (2018) S0020025518301749.
[18] X. Huang, G. Liu, et al., Obstacle-avoiding algorithm in X-Architecture based on discrete particle swarm optimization for VLSI design, ACM Trans. Des. Autom. Electron. Syst. 20 (2) (2015) 28, http://dx.doi.org/10.1145/2742143, Article 24.
[19] Z. Huang, Y. Yu, J. Gu, et al., An efficient method for traffic sign recognition based on extreme learning machine, IEEE Trans. Cybern. 47 (4) (2017) 920–933.
[20] B. Lin, W. Guo, X. Lin, Online optimization scheduling for scientific workflows with deadline constraint on hybrid clouds, Concurr. Comput. Pract. Exp. 28 (11) (2016) 3079–3095.
[21] K. Guo, W. Guo, Y. Chen, et al., Community discovery by propagating local and global information based on the MapReduce model, Inform. Sci. 323 (2015) 73–93.
[22] K. Guo, Q. Zhang, Fast clustering-based anonymization approaches with time constraints for data streams, Knowl.-Based Syst. 46 (2013) 95–108.
[23] W. Guo, G. Chen, Human action recognition via multi-task learning base on spatial–temporal feature, Inform. Sci. 320 (2015) 418–428.
[24] W. Guo, G. Liu, et al., A hybrid multi- objective PSO algorithm with local search strategy for VLSI partitioning, Front. Comput. Sci. 8 (2) (2014) 203–216, http://dx.doi.org/10.1007/s11704-014-3008-y.
[25] X. Huang, W. Guo, G. Liu, G. Chen, FH-OAOS: A fast 4-step heuristic for obstacle-avoiding octilinear architecture router construction, ACM Trans. Des. Autom. Electron. Syst. 21 (3) (2016) 30, http://dx.doi.org/10.1145/2856033, Article 48.
[26] Fei Ye, Evolving the svm model based on a hybrid method using swarm optimization techniques in combination with a genetic algorithm for medical diagnosis, Multimedia Tools Appl. (2016).
[27] S. Zhang, Y. Xia, Two fast complex-valued algorithms for solving complex quadratic programming problems, IEEE Trans. Cybern. 46 (12) (2016) 2837–2847, MathSciNet.
[28] S. Zhang, Y. Xia, J. Wang, A complex-valued projection neural network for constrained optimization of real functions in complex variables, IEEE Trans. Neural Netw. Learn. Syst. 26 (12) (2015) 3227–3238.
[29] S. Zhang, Y. Xia, W. Zheng, A complex-valued neural dynamical optimization approach and its stability analysis, Neural Netw. 61 (2015) 59–67.
[30] S. Zhang, Y. Xia, Two fast complex-valued algorithms for solving complex quadratic programming problems, IEEE Trans. Cybern. 46 (12) (2016) 2837–2847.
[31] Y. Zhang, S. Wang, P. Phillips, G. Ji, Binary pso with mutation operator for feature selection using decision tree applied to spam detection, Knowl.-Based Syst. 64 (2014) 22–31.
[32] S. Zhong, T. Chen, F. He, et al., Fast Gaussian kernel learning for classification tasks based on specially structured global optimization, Neural Netw. 57 (2014) 51–62.
[33] W. Zhu, W. Guo, Z. Yu, H. Xiong, Multitask allocation to heterogeneous participants in mobile crowd sensing, Wirel. Commun. Mob. Comput. (2018) 10, Article 7218061.
[34] C. Zou, Y. Xia, Restoration of hyperspectral image contaminated by Poisson noise using spectral unmixing, Neurocomputing 275 (2018) 430–437.
[35] J. Zou, L. Dong, W. Wu, New algorithms for the unbalanced generalise, IET Inf. Secur. 12 (6) (2018) 527–533.
[36] L.H. Yang, Y.M. Wang, Q. Su, et al., Multiattribute search framework for optimizing extended belief rule-based systems, Inf. Sci. 370 (2016) 159–183.
[37] Y. Yang, M. Ma, Conjunctive keyword search with designated tester and timing enabled proxy reencryption function for E-Health clouds, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016) 746–759.
[38] D. Ye, Z. Chen, et al., A novel and better fitness evaluation for rough set based minimum; attribute reduction problem, Inf. Sci. 222 (3) (2013) 413–423.
[39] L. Hu, Y. Miao, G. Wu, M.M. Hassan, I. Humar, iRobot-factory: An intelligent robot factory based on cognitive manufacturing and edge computing, Future Gener. Comput. Syst. 90 (2019) 569–577.
[40] C. Yoon, M.M. Hassan, H. Lee, W. Ryu, E.N. Huh, Dynamic collaborative cloud service platform: Opportunities and challenges, ETRI J. 32 (4) (2010) 634–637.
[41] A. Enayet, M.A. Razzaque, M.M. Hassan, A. Alamri, G. Fortino, A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities, IEEE Commun. Mag. 56 (2) (2018) 110–117.
[42] Zhang Yichun, Chen Xiaosu, He Miao, et al., Interleaving algorithm method for Monte Carlo sample event mixing in daya bay neutrino experiment, J. Univ. Sci. Technol. China 43 (3) (2013) 246–252.
[43] Z. Zhou, H. Lu, C. Deng, et al., User preference learning for online social recommendation, IEEE Trans. Knowl. Data Eng. 28 (9) (2016) 1.
[44] T. Bai, Y. Bo, L. Fei, TDRec: Enhancing social recommendation using both trust and distrust information, in: European Network Intelligence Conference, 2015.
[45] Y. Bhawsar, G.S. Thakur, R.S. Thakur, User recommendation system using Markov model in social networks, Int. J. Comput. Appl. 86 (9) (2014) 33–39.
[46] V. Subramaniyaswamy, R. Logesh, Adaptive KNN based recommender system through mining of user preferences, Wirel. Pers. Commun. 97 (4) (2017) 119.

Gang Ke received the B.S. degree in computer science and technology from Hubei Normal University, Huangshi, China, in 2007 and the M.S. degree in computer application technology from Guangdong University of Technology, Guangzhou, China, in 2010. His current research interests include cloud computing, computing network security, machine learning and IoT.

Hong-Le Du received the B.S. degree in computer science and technology from Henan University, Kaifeng, China, in 2004 and the M.S. degree in computer application technology from Guangdong University of Technology, Guangzhou, China, in 2010. His current research interests include machine learning, pattern recognition and network security.

Yeh-Cheng Chen is a Ph.D. at the Department of Computer Science, University of California, Davis, CA, USA. His research interests are radio-frequency identification (RFID), data mining, social network, information systems, wireless network artificial intelligence, IoT and security.