

Cascaded Trust Network-based Block-Incremental Recommendation Strategy

Shujuan Ji^{a,*}, Da Li^b, Qing Zhang^b, Chunjin Zhang^c, and Chunxiao Bao^b

^aKey Laboratory for Wisdom Mine Information Technology of Shandong Province
Shandong University of Science and Technology, Qingdao, 266590, China

^bCollege of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China

^cNetwork Information Center (NIC), Shandong University of Science and Technology, Qingdao, 266590, China

Abstract

Accurate recommendation can effectively bridge sellers and buyers. Because the computation complexity and storage complexity of static data-oriented recommendation algorithms are very high, researchers have recently explored streaming recommendation systems. However, streaming recommendation wastes large quantities of computation resources in quick response and is not suitable for seasonable-dependent situations. Therefore, this paper presents a block incremental recommendation strategy. First, a cascaded trust network construction method is presented, which is realized by using a distrust relationship to purify and predict users' trust relationships. Then, the social regularization is improved by comprehensively considering the cascaded trust relationship, the behavior bias of users and items. Finally, a block-incremental recommendation algorithm called ITDBMF is proposed, which uses the Ebbinghaus forgetting function to decay incremental rating blocks and simultaneously considers incremental social relationships. Experimental results show that the incremental recommendation strategy given in this paper can not only outperform benchmark algorithms in prediction accuracy, but also save storage of remote data and matrix factorization time.

Keywords: personalized recommendation; matrix factorization; social network; trust cascaded network; block incremental; time decay

(Submitted on October 24, 2018; Revised on November 27, 2018; Accepted on December 21, 2018)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

The exponentially growing big data has triggered the design and development of next-generation models, methods, and systems for collecting and analysing consumers' perceptual and objective data [1-2], and hence to formulate more effective marketing and competitive strategies. Personalized recommendation is a popular method for bridging customers and sellers. It is said that the personalized recommendation system boosted Amazon's sales revenue by 29% in 2012 compared to 2011 [3].

Though personal recommendation systems have been used in different styles, most of them first generate conceptual models for users using their historical behaviour data such as browsing, clicking, buying, collecting, and reviewing. Then, the systems make recommendations for users based on their individual models. In traditional recommender systems, static recommendation strategies [4-10] give similar weight to all historical data. However, in real-world scenarios, users' ratings and their associated social networks often are collected in an incremental manner. Moreover, in many real-life applications (such as films, clothes, restaurants, and computer/mobile games), people pay much attention to recent data. For example, women often prefer to be recommended with appropriate fashion clothes in season rather than the ones that were popular in the previous year. Traditional recommendation models that are constructed based on total static data are not suitable in these cases. This is because equal consideration of historical data and recent data may decrease the recommendation accuracy. Therefore, researchers have presented some real-time recommendation strategies (e.g., streaming recommendation) [11-12]. However, in the above cases, streaming recommendation is not needed because of two reasons. One reason is that customers do not need to be recommended frequently or continuously. The other reason is that the computation and storage cost of streaming recommendation strategies are very high for companies. As the requirements for real-time response are not very

* Corresponding author.

E-mail address: jane_ji2003@aliyun.com

strong, block-incremental (i.e., periodical updating) recommendation algorithms that give different weights to historical data need to be designed.

In block recommendation, the open challenges such as data sparsity, cold start, and scalability are more difficult to be solved than in static recommendation situation. That is because (1) the data are cut into increasing blocks, in which ratings are sparser than the ones in static data set, (2) the social network dynamically evolves with time, (3) the users may show different levels of activity in different blocks, and the traditional concept of new user is different in series of blocks. To address the above problems, this paper gives a new block-incremental recommendation strategy with the following characteristics. First, this paper presents a cascaded trust network construction method. The cascaded trust network can be used to predicate users' trust relationships, which makes the social network less sparse. Moreover, it can also help the recommendation algorithm discover trustable friends for new users according to their social relationship. Secondly, this paper improves the social regularization formula by simultaneously considering the rating behaviour deviation of users and items and the rating similarity in the trust cascaded network. This improvement effectively solves the cold start problem caused by the evolution of blocks of social networks. Moreover, this method can solve the problem of scalability caused by different active degrees of users in various blocks. Two sets of experiments are designed and implemented on the Epinions data (<http://www.trustlet.org/datasets/>). Experimental results show that this strategy can relieve the cold start problem of new users, the sparsity problem of social relationships, and the scalability problem of incremental data to some extent. Moreover, this algorithm outperforms the prevalent static data-oriented matrix factorization algorithms in recommendation accuracy.

The rest of the paper is as follows. Section 2 reviews related works. Section 3 gives a framework of the new recommendation algorithm. Section 4 illustrates recommendation algorithms for dealing with block incremental data. Section 5 demonstrates the merits of our strategy with the experimental results. Section 6 concludes this paper with our future work directions.

2. Related Work

2.1. Static Data-Oriented Matrix Factorization Algorithms

Research on recommendations over static data primary aims to better resolve the data sparsity problem and cold start problem. Researchers have introduced social networks, probability models, context, topics, and other factors into the basic matrix factorization algorithm.

Liu and Yuan [13] proposed a local trust network-based recommendation algorithm. Their main contribution is the construction of a local trust network for each user from its friends and the continuous active users. Deng et al. [14] proposed another social-networked based recommendation method, which integrates social network information with random walk algorithm and selects the migration direction of random walk according to users' trust relationship. Fang et al. [15] took full advantage of multiple interpersonal and impersonal aspects of trust and distrust information. Moreover, they used distrust information to refine trust information and improve the prediction accuracy based on the refined trust information.

Considering behaviour constrains factor, Salakhutdinov and Mnih [9] presented a probabilistic matrix factorization (PMF) algorithm. Moreover, Ma et al. [7] gave an algorithm named SoRec by adding social regularization to the basic probabilistic matrix factorization algorithm. These algorithms only add social network information into the probability model. Wu et al. [16] proposed a collaborative topic regression with social trust ensemble model, which comprehensively considers factors such as trust network, topic information, and probabilistic matrix factorization.

Supposing that ratings cannot be totally explained by the interactions and are also influenced by the combined impact of overall mean, user bias, item bias, and context bias, Wu and Zhao et al. [17] proposed a novel context-aware recommendation model called bias tensor factorization. To address the problems faced by traditional context-aware recommenders with tensor factorization, they proposed a context features auto-encoding algorithm based on regression tree. To recommend routes for drivers according to their individual driving preference, Dai et al. [18] integrated big trajectory data with a multitude of different travel costs, e.g., distances, travel times, and fuel consumption.

2.2. Streaming Data-Oriented Recommendation Algorithms

Data streams present distinct properties such as temporally ordered, continuous, and high-velocity, and they pose tremendous challenges to traditional recommender systems. Huang et al. [19] proposed a general real-time stream recommender system built on Storm.

Following this work, Damaskinos [20] presented a novel similarity metric to enable lightweight similarity computations in an incremental and temporal manner to solve the problem that a huge volume of user-generated data typically results in heavyweight computations for even a single similarity update. Chang [11] proposed a principled framework termed sRec, which provides explicit continuous-time random process models of the creation of users and topics and of the evolution of their interests. A variation of the Bayesian approach called recursive mean-field approximation is proposed, and it permits computationally efficient instantaneous on-line inference.

2.3. Block Incremental Recommendation Algorithms

The above research was conducted on streaming and a real-time recommendation background. However, in many cases, real-time recommendation wastes resources. Therefore, research on block incremental recommendation is also necessary.

To improve the applicability of matrix factorization algorithms in real-world scenarios where users' ratings and their associated networks are always collected in an incremental manner, Nilashi et al. [21] proposed an incremental method for modelling user's preferences. They also used regression-based techniques to predict users' overall assessment of an item based on multi-criteria ratings and used the clustering method to detect subgroups of customers. Therefore, this method solves the problem of cold start and new data to a certain extent.

To deal with the incremental blocks of ratings from users, Luo et al. [22] gave an incremental CF recommender based on the Regularized Matrix Factorization (RMF). They first simplified the training rule of RMF to propose the SI-RMF, which provides a simple mathematic form for further investigation. Then, they designed two incremental RMF models: the Incremental RMF (IRMF) and the Incremental RMF with linear biases (IRMF-B).

To deal with old and potentially obsolete data and avoid unnecessary memory usage and processing time, Vinagre and Jorge [23-24] proposed two forgetting mechanisms: sliding windows and fading factors. Looking at users' activity as a rating block of individual user sessions, they implemented experiments and compared the performance of the forgetting mechanisms in non-incremental and incremental CF algorithms.

3. The Block-Incremental Recommendation Strategy

In the new block-incremental recommendation strategy, three dimensions (i.e., users, items and time windows) are considered. The ratings in a similar time window are regarded as a block. For each block of data, we should execute the following steps 2-8, which is called the TDBMF algorithm. For a series of blocks of data, we should execute the following steps 1-8, which is called the ITDBMF algorithm.

- 1) Using the Hermann Ebbinghaus forgetting function [23] to calculate the decay weight of ratings in various blocks. Then, update the ratings by decaying them. Execute the following offline and online learning steps 2-8, which forms another algorithm called TDBMF.
- 2) Construct a cascaded trust network based on trust/distrust relationship.
- 3) Calculate the similarity between trusted users based on their ratings.
- 4) Construct the social regularization of similar trusted users based on the results obtained from steps 2-3.
- 5) Execute the matrix factorization process over the trust and distrust relationship data.
- 6) Add the user rating-based bias matrix factorization to the results obtained from steps 4-5, and get the objective function.
- 7) Optimize the objective function, and get the optimal parameter.
- 8) Predicate the ratings that a user may give to the unrated items and make recommendations according to these predicated ratings.

The following subsections illustrate the algorithms in the block-incremental recommendation strategy.

3.1. The Construction Algorithm of Cascaded Trust Network

To address the sparsity problem of social relationships, this paper constructs a cascaded trust network for each user based on its social network by considering the propagation property of trust and distrust relationships. We first show the relationship of trust and distrust in social networks. Then, we illustrate the principle of finding trustable users through an example intuitively. Finally, the trust network construction algorithm is given.

Example 1 Suppose Tr is a social network including seven vertexes, i.e., $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \in Tr$. The trust/distrust relationship between these seven vertexes is shown in Figure 1. Among them, the red-solid/blue-dotted arc indicates the trust/distrust relationship. The arrow tail is the trustor, and the arrowhead is the trustee. The users (i.e., v_2, v_3 , and v_4) that are directly trusted or distrusted by the root user are in the first level. The distance between v_1 and these users are 1.

To calculate the new trustworthiness and untrustworthiness of the users, this paper makes another specification, i.e., any user that is distrusted by the users in its upper or similar layer will be considered as untrustworthy and will be deleted from the cascaded trust network of the initial user. The resulting cascaded trust network of v_1 is given in Figure 2. The main idea of the above cascaded trust network construction method is “Assuming that there are several paths from user x to user y . If there exists one path with distrust label, user y is believed to be incompletely trustworthy from x 's viewpoint”. Algorithm 1 formally presents the detailed steps for constructing the cascaded trust network.

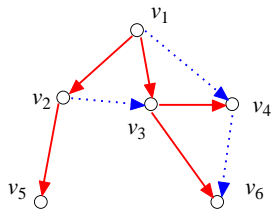


Figure 1. An example social network of seven users

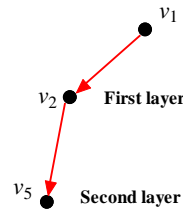


Figure 2. The cascaded trust network of v_1

Algorithm 1 Construction Algorithm of Cascaded Trust Network

Input: (1) Social network Tr ; (2) the maximal layers that trust will propagate Iterlayer (> 1);

Output: Cascaded trust network Tr^+ ;

```

1.  for ( $\forall u \in Tr$ )
2.       $S_u^{trust} = \emptyset$ ;  $S_u^{distrust} = \emptyset$ ;
3.      for ( $\forall x \in Direct\_trust_u$ )  $S_u^{trust} = S_u^{trust} + \{x\}$ ; end for
4.      for ( $\forall x \in Direct\_distrust_u$ )  $S_u^{distrust} = S_u^{distrust} + \{x\}$ ; end for
5.  end for
6.  for (iter = 1: Iterlayer-1)
7.      for ( $\forall u \in Tr$  and  $S_u^{trust} \neq \emptyset$ ) // start from the node with zero injection arcs
8.          for ( $\forall x \in S_u^{trust}$ )
9.              for ( $\forall y \in S_x^{trust}$ )
10.                 if ( $y \notin S_u^{trust}$  and  $y \notin S_u^{distrust}$ )  $S_u^{trust} = S_u^{trust} + \{y\}$ ;
11.             end for
12.             for ( $\forall y \in S_x^{distrust}$ )
13.                 if ( $y \notin S_u^{distrust}$ )  $S_u^{distrust} = S_u^{distrust} + \{y\}$ ;
14.                 if ( $y \in S_u^{trust}$ )  $S_u^{trust} = S_u^{trust} - \{y\}$ ;
15.             end for
16.         end for
17.     end for
18. end for
19.  $Tr^+ = (S_1^{trust}, S_2^{trust}, \dots, S_n^{trust})$ ; //  $n = |Tr|$ 

```

3.2. The Cascaded Trust Network-based Matrix Factorization

In real life, most users tend to appeal to his/her trust friends instead of consulting the users who buy the product, other than

the those that he/she distrusts. Moreover, they may believe that the advice from similar trustable friends is most helpful. Comprehensively considering the influence of behaviour bias of users and items [17] and the cascaded trust network between users, this paper defines the loss function as the following Equation (1). $Sim(u, v)$ is the similarity between user u and his trusted user v . The Pearson correlation coefficient is used as the similarity measure between users. Tr_u^+ is the users that are trusted by user u in the cascaded trust network. r_{ui} is the rating that user u rated item i . \hat{r}_{ui} is the estimated rating that user u may give to product i . t_{uv} is the degree that trustor u trusts or distrusts trustee v . \hat{t}_{uv} is the approximate degree that trustor u trusts or distrusts trustee v . The rating bias of the user is denoted by b_{user} . The score bias when users rate items is denoted by b_{item} . I_{ui} is an indicator that equals 1 if the corresponding element in the matrix is not empty; otherwise, it equals 0.

$$Loss = \frac{1}{2} \sum_{u=1}^m \sum_{i=1}^n I_{ui} (r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_t}{2} \sum_{u=1}^m \sum_{v=1}^m I_{uv} (t_{uv} - \hat{t}_{uv})^2 + \frac{\lambda_{sim}}{2} \sum_{u=1}^m \sum_{v \in Tr_u^+} Sim(u, v) \cdot \|p_u - p_v\|_F^2 + \frac{\lambda_b}{2} (\sum_{u=1}^m b_{user}^2 + \sum_{i=1}^n b_{item}^2) + \frac{\lambda}{2} (\sum_{u=1}^m \|p_u\|_F^2 + \sum_{i=1}^n \|q_i\|_F^2 + \sum_{v=1}^m \|f_v\|_F^2) \quad (1)$$

The part

$$\frac{1}{2} \sum_{u=1}^m \sum_{i=1}^n I_{ui} (r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_b}{2} (\sum_{u=1}^m b_{user}^2 + \sum_{i=1}^n b_{item}^2) + \frac{\lambda}{2} (\sum_{u=1}^m \|p_u\|_F^2 + \sum_{i=1}^n \|q_i\|_F^2)$$

is the extended factorization by considering the behaviour bias of users and items.

The part

$$\frac{\lambda_t}{2} \sum_{u=1}^m \sum_{v=1}^m I_{uv} (t_{uv} - \hat{t}_{uv})^2 + \frac{\lambda_{sim}}{2} \sum_{u=1}^m \sum_{v \in Tr_u^+} Sim(u, v) \cdot \|p_u - p_v\|_F^2 + \frac{\lambda}{2} (\sum_{u=1}^m \|p_u\|_F^2 + \sum_{v=1}^m \|f_v\|_F^2)$$

is the extended factorization by considering social matrix factorization based on cascaded trust network.

Optimizing Equation (1) by the stochastic gradient descent method, we can get following optimization objects.

$$\begin{aligned} \frac{\partial Loss}{\partial b_{user}} &= \sum_{i \in I_u} (r_{ui} - \hat{r}_{ui}) + \lambda_b \cdot b_{user} \\ \frac{\partial Loss}{\partial b_{item}} &= \sum_{u \in U_i} (r_{ui} - \hat{r}_{ui}) + \lambda_b \cdot b_{item} \\ \frac{\partial Loss}{\partial p_u} &= \sum_{i \in I_u} (r_{ui} - \hat{r}_{ui}) q_i + \lambda_t \sum_{v \in Tr_u} (t_{uv} - \hat{t}_{uv}) \cdot f_v + \lambda_{sim} \sum_{v \in Tr_u^+} Sim(u, v) \cdot (p_u - p_v) + \lambda p_u \\ \frac{\partial Loss}{\partial q_i} &= \sum_{u \in U_i} (r_{ui} - \hat{r}_{ui}) p_u + \lambda q_i \\ \frac{\partial Loss}{\partial f_v} &= \sum_{i \in I_u} (t_{uv} - \hat{t}_{uv}) p_u + \lambda f_v \end{aligned} \quad (2)$$

3.3. The ITDBMF Algorithm on Block Incremental Data

As time goes by, consumers' historical behavioral data accumulates. In recommendation systems, it is believed that new data contains more effective and valuable information than historical ones. However, the value of data in remote rating blocks is uncertain (i.e., may be useful, partially effective, or even completely unavailable for the current user). Therefore, in the new strategy, the Hermann Ebbinghaus forgetting function given in Equation (3) is used to simulate the pattern that agents' memory of block incremental data fades over time.

$$W_d = \begin{cases} 0, & \text{remote rating block} \\ \frac{e^b}{(t+t_0)^c}, & \text{recent rating block} \end{cases} \quad (3)$$

Where e is the standard mathematical constant, and b and c (the values of which are larger than 0) are the power of e and $(t+t_0)$ respectively. t_0 is the initial time parameter, which is the initial time that is considered by the recommendation system. t is the time of the d^{th} block (to rule out the differences of various ratings in the same block, this paper takes the time of the latest rating in each block as the time of each block). The choice of t_0 and the difference between t_0 and t make the value of W_d adjustable and controllable.

The weight of remote ratings is assigned 0, meaning we can completely neglect these remote ratings. In contrast, the weight of recent rating blocks follows the forgetting curve. Starting from the current rating block, the higher the historical rating block decay, the lower the weight of the historical rating block. Equation (4) defines the integrated ratings that a user gave to an item. If he/she once rated an item multiple times in all the considered rating blocks, the ratings given by him/her will be the weighted average of these ratings. If a user only rated an item one time in the historical rating blocks, the rating also should be weighted according to the Ebbinghaus forgetting curve function.

$$r_{ui}^{(s)} = \begin{cases} \frac{\sum_{d=1}^{s-1} W_d \cdot r_{ui}^{(d)} + W_s \cdot r_{ui}^{(s)}}{\sum_{d=1}^s W_d}, & r_{ui} \text{ appear in the recent } s \text{ rating blocks} \\ W_d \cdot r_{ui}^{(d)}, & r_{ui} \text{ only appear in the } d^{\text{th}} (d=1, \dots, s) \text{ rating block} \end{cases} \quad (4)$$

Where the rating blocks are named 1, 2, ..., s from far remote. The s^{th} rating block is the current rating block. It should be noted that the rating time in one rating block is regarded as identical. The weight of the current rating block is 1, i.e., $W_s = 1$.

Based on the above definitions of integrated ratings, the ITDBMF recommendation strategy given in section 3 can be formally represented as Algorithm 2.

Algorithm 2 ITDBMF Algorithm and its Optimization

Input: (1) Users' social relationship matrix Tr ; (2) users' cascading trust network Tr^+ ; (3) user-item's scoring matrix R ; (4) latent feature dimension K ; (5) learning rate η ; (6) number of iterations $Iteration$

Output: (1) User latent feature matrix P ; (2) item latent feature matrix

1. Match the latest block of user-item matrix to the previous block of user-item matrix.
 2. **if** the corresponding elements of these two user-item matrixes are not empty
 3. **if** the latest user-item matrix intersects with the previous user-item matrix
 4. $r_{ui}^{(s)} = \frac{\sum_{d=1}^{s-1} W_d \cdot r_{ui}^{(d)} + W_s \cdot r_{ui}^{(s)}}{\sum_{d=1}^s W_d}$; //decay data according to Equations (3) and (4)
 5. **else**
 6. $r_{ui}^{(s)} = W_d \cdot r_{ui}^{(d)}$; //decay data according to Equations (3) and (4)
 7. **end if**
 8. **end if**
 9. Update cascade social trust network Tr^+ according to Algorithm 1;
 10. Update the similarity coefficient $Sim(u, v)^{(d)}$ between users;
 11. Randomly generate a latent feature matrix F representing the user's latent feature matrix P , the item's latent feature matrix Q , and a consumer's trusted user;
 12. **for** $iter = 1: Iteration$ // optimize the object function according to Equation (2)
 13. $p_u \leftarrow p_u + \eta \frac{\partial Loss}{\partial p_u}$; $q_i \leftarrow q_i + \eta \frac{\partial Loss}{\partial q_i}$; $f_v \leftarrow f_v + \eta \frac{\partial Loss}{\partial f_v}$; $b_{user} \leftarrow b_{user} + \eta \frac{\partial Loss}{\partial b_{user}}$; $b_{item} \leftarrow b_{item} + \eta \frac{\partial Loss}{\partial b_{item}}$;
 14. **if** ($Loss$ converge to an acceptable value) **break** ;
 15. **end for**
 16. **Return** P, Q
-

4. Experimental Results

4.1. Experimental Data

The extended Epinions data set is used in the experiments. This Epinions version enables users to review products by adding text comments and issuing numerical ratings in the range of [1, 14]. Users can also explicitly specify other users as trust or distrust based on whether the reviews and ratings of others are consistently valuable or useless for the user. In the extended Epinions data set, the trust value is labelled as 1 and distrust as -1.

Since the proposed ITDBMF algorithm involves the time decay of incremental rating blocks, this paper considers dividing the training set and the test set of the experiments in chronological order. First, the original extended Epinions data set is pre-processed to ensure that the time span of extracted users' trust data and user rating data are consistent in the time span. After pre-processing, the time spans of ratings range from February 1, 2001 to December 31, 2001. Moreover, in order to address the cold start problem of new users, it is necessary to ensure that each rating block includes consistent items when dividing multiple rating blocks. Considering the above constraints, we divide the 11-month of rating data into five blocks that have approximately equal time spans. Table 1 lists these five blocks in chronological order. Taking the intersection of all the items from these five blocks, we get 469 items and correspondingly 3,030 users and 8,101 ratings. The data sparseness is 0.57%. Tables 1 and 2 list the detailed characteristics of total ratings, users, appended users, and distribution of various rating values in each block respectively.

Table 1. Blocks of users' ratings after pre-processed

Rating blocks	Records #	Total users #	New appended users #	New users #/Total users #
1	1705	1227	—	—
2	1665	1226	792	64.60%
3	1661	1082	462	42.70%
4	1492	994	309	31.09%
5	1578	1020	240	23.53%

Table 2. Distribution of users' ratings after pre-processed

Rating values	Ratings #	Ratings # in block 1	Ratings # in block 2	Ratings # in block 3	Ratings # in block 4	Ratings # in block 5
1	10	1	1	4	3	1
2	170	30	39	41	29	31
3	452	86	98	89	77	102
4	1406	270	304	262	283	287
5	6063	1318	1223	1265	1100	1157
Summary	8101	1705	1665	1661	1492	1578

In processing the trust data of users, five blocks of trust data are also obtained according to the same time span of rating data. The 3,030 users who give ratings are selected. Between them, 31,971 social relationship records are obtained, including 27,328 trust relationship (85.48%) and 4,643 distrust relationship (14.52%). The sparsity of social network data is 0.09%. The detailed characteristics of trust/distrust information between users are listed in Table 3.

Table 3. Trust/distrust relationship between users after pre-processed

Relationship blocks	Records #	Trust #	Distrust #
1	4118	3540	578
2	1220	1052	168
3	1173	1019	154
4	1144	912	232
5	694	539	155
Summary	8349	7062	1287

4.2. Experimental Settings

In order to verify the performance of the ITDBMF algorithm, we design two sets of experiments. To verify the influence of the number of latent features and the level of cascaded trust network propagation on the recommendation system and to compare the performance of comparison recommendation algorithms over static data, we design the first experiment for the TDBMF algorithm, which does not consider the dynamic incremental part of the ITDBMF algorithm. To verify the scalability ability of the ITDBMF algorithm, we design a second set of experiments over incremental blocks of data. Two kinds of dynamic increment data sets (i.e., the blocks that are rolling incremented, the blocks that are accumulatively incremented) are considered. Recommendation algorithms such as PMF [9], SoRec [7], RSTE [6], SocialMF [4], SoReg [8],

and TrustMF [10] are selected as benchmarks. The TDBMF algorithm is also chosen as a benchmark in the second set of experiments.

4.2.1. Data Description in the First Set of Experiment

In the first set of experiments (i.e., the verification experiment of the TDBMF algorithm), the total data of the first four rating blocks are taken as the training set, and the data of the fifth rating block are used as the test set (see Table 1). Therefore, in the data set of users' ratings, the training set contains 6523 rating records, 1562 users, and 469 items. In contrast, in the test set, there are 1578 rating records, 1020 users, and 469 items. There is no crossover data in the training set and test set. The ratio of data in the training set to data in the test set is about 4:1. However, in the data set of the trust/distrust relationship between users, the total trust/distrust relationships of the first four rating blocks is used as the input data of the experiment, without being divided into training set and test set. This is because our goal in this article is to improve the predicting accuracy of users' ratings other than the trust/distrust relationship between users. Therefore, the users' trust/distrust relationship is taken as supplementary information for accuracy prediction of users' ratings. The input trust/distrust relationship data contains a total of 7655 records, among which there are 6523 trust relationships and 1132 distrust relationships. In addition, since only one fixed training set and one fixed test set are used in the first set of experiments, it is inevitable that the results have the characteristics of randomness and fortuitousness. To rule out the influence of randomness and fortuitousness, this paper tests 100 times for each set of experimental parameters and takes the mean of 100 test results as the final experimental results.

4.2.2. Experimental Settings in the Second Set of Experiments

Similar settings are adopted in the second set of experiments. Differently, two kinds of dynamic increment data sets (i.e., the blocks are rolling incremented, the blocks are accumulatively incremented) are designed as follows.

For the user rating data, the data of rating blocks 1, 2, 3, and 4 are used as the training set of the experiments respectively. The data of rating blocks 2, 3, 4, and 5 are sequentially used as the test set of the experiment correspondingly. However, it is believed that users' trust/distrust relationship is continuous and not easily changed data. Therefore, corresponding to the four training user rating data blocks, trust/distrust relationships are accumulated in each experiment. The trust/distrust relationships in rating blocks 1 and 2 are used with rating block 2. The trust/distrust relationships in rating blocks 1, 2, 3, and 4 are cumulated and used as input for rating block 4. Figure 3 provides a vivid illustration of process details over the rolling rating blocks. For the user rating data, rating block 1, the accumulation of rating blocks 1 and 2, the accumulation of rating blocks 1, 2, and 3, and the accumulation of rating blocks 1, 2, 3, and 4 are sequentially used as a training set of experiments. Correspondingly, rating blocks 2, 3, 4, and 5 are served as the experimental test set. The user's trust/distrust relation data are also accumulated and taken as the input of the experiment. Figure 4 illustrates the process details over accumulated incremental rating blocks and relationship blocks.

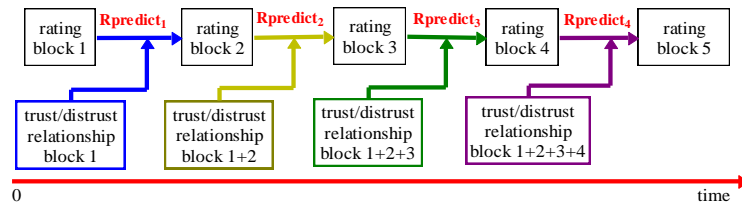


Figure 3. The illustration of process details over the rolling rating blocks

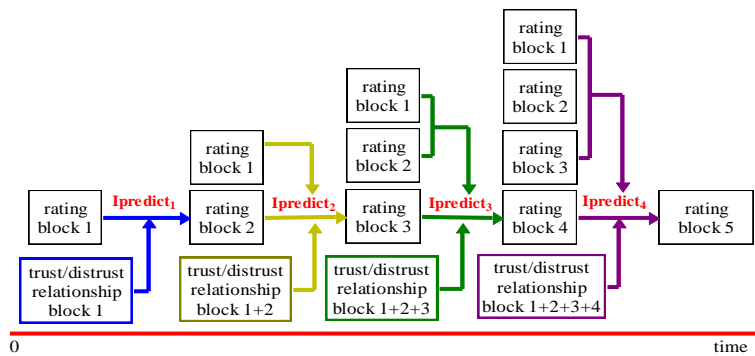


Figure 4. The illustration of process details over the accumulated incremental blocks

4.3. Evaluation Criteria

This paper adopts two commonly used evaluation criteria, RMSE and MAE, in comparing the prediction accuracy of the proposed algorithm with related algorithms.

4.4. Experimental Results and Analysis about TDBMF Algorithm

To trade off the computation complexity and the approximating accuracy, we must choose an appropriate value for K . Figure 5 shows the prediction results of the TDBMF algorithm with various values of latent feature K . In the experiments, the value of K ranges from 5 to 30 for step 5.

From Figure 5, we can find that the values of RMSE and MAE decrease first and then increase respectively with the increase in the latent feature K . When the value of the latent feature K is 20, the TDBMF algorithm has the best prediction effect. In the following experiments, without special statement, the latent feature K is assigned a value of 20.

In the experiments, the layers of cascade trust network are assigned 1 to 5 respectively. Table 4 lists the number of trust relationships generated from the one layer to five layers' propagation of trust between users. From Table 4, we can see that with an increase in propagation layers, the number of trust relationships obtained by the cascaded trust network also increases. When the propagation layer is four and five, the cascaded trust relationship is similar, which means that the number of cascaded trust relationships gradually becomes stable when the propagation layer is larger than 3.

Table 4. The cascaded trust network relationship in total relationship data

Trust propagation layer	The number of cascaded trusts
1	6523
2	69874
3	282266
4	342593
5	342605

Figure 6 lists the RMSE and MAE values when the propagation layer of the cascaded trust network ranges from 1 to 5. It can be seen that the RMSE and MAE values tend to slightly decrease as the propagation layers of the cascaded trust network increase. When the cascaded network is propagated to layer 4, the RMSE and MAE values obtained become stable.

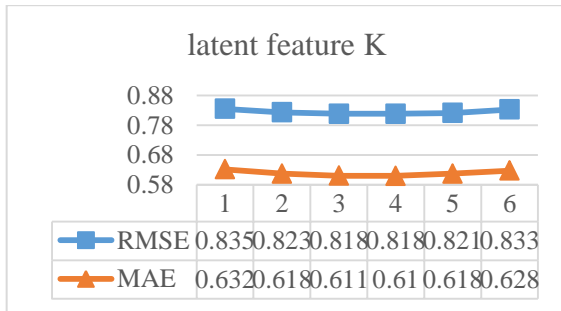


Figure 5. RMSE and MAE values when latent feature ranges from 5 to 30

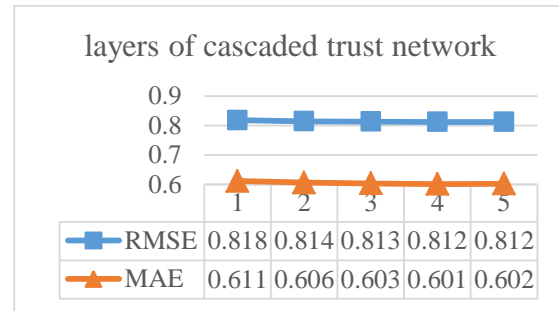


Figure 6. RMSE and MAE values when layer of cascaded trust network ranges from 1 to 5

To compare the performance of the TDBMF algorithm and benchmark algorithms, the parameter K in these algorithms are set with 10, 15, and 20 respectively. Table 5 illustrates the RMSE and MAE values we obtained under these settings. From Table 5, we can find that the TDBMF algorithm obtains the lowest RMSE and MAE values, which indicates that the TDBMF algorithm outperforms other recommendation algorithms. It is obvious that the PMF algorithm has the poorest prediction accuracy among the seven comparison algorithms. This is because PMF algorithm only considers rating information for predicting users' interest. Experimental results show that the consideration of the trust relationship between users can improve the performance of recommendation algorithms. The TrustMF algorithm achieves the second lowest RMSE and MAE values. The TDBMF algorithm outperforms TrustMF because the TDBMF algorithm considers the untrustworthy relationship between users, while the TrustMF algorithm does not. Experimental results show that the distrust relationship can further improve the performance of recommendation algorithms.

Table 5. The comparison of various recommendation algorithms

RMSE							
	PMF	SoRec	SocialMF	RSTE	SoReg	TrustMF	TDBMF
K=20	1.0278	0.9846	0.9476	0.8760	1.0144	0.8457	0.8121
K=15	1.0277	0.9747	0.9465	0.8752	1.0185	0.8433	0.8132
K=10	1.0373	0.9987	0.9808	0.9216	0.9708	0.8565	0.8183
MAE							
	PMF	SoRec	SocialMF	RSTE	SoReg	TrustMF	TDBMF
K=20	0.8140	0.8011	0.7769	0.6990	0.7741	0.6221	0.6014
K=15	0.8078	0.7947	0.7765	0.6977	0.7708	0.6257	0.6021
K=10	0.8277	0.8175	0.8100	0.7313	0.8197	0.6554	0.6107

4.5. Experimental Results and Analysis about ITDBMF Algorithm

As the second set of experiments are implemented over blocks of data other than the total rating data and relationship data, it is necessary to verify the influence of layers on the cascaded trust relationship in various social relationship blocks. Table 6 summarizes the correlations between cascaded trust relationships and the layers in each social relationship block. From this table, we can see that the cascaded trust relations stabilize at the fourth and fifth layer in all of these five relationship blocks. Similar to the TDBMF algorithm that adopts the total social relationship data, the ITDBMF algorithm can reach optimal recommendation performance when the trust relationship propagates four layers.

Table 6. Cascaded trust network relationships in various social relationship blocks

Trust relationships # Blocks	Trust propagation layers	1	2	3	4	5
1		3540	29846	190929	258136	258241
2		4592	43053	244271	309572	309678
3		5611	58064	289463	354447	354553
4		6523	69874	282266	342593	342605
5		7062	76750	293062	354130	354143

According to the setting of rolling blocks, the prediction of users in each block is a continuous process, with 1 followed by 2, 3 following 2, and so on. In the sequential prediction of blocks 2, 3, 4 and 5, only the current rating block is used in training; therefore, the weight of W_d in each “current” rating block is 1, and the weight in the remote rating block (historic blocks excepting for the current rating block) is 0. The result of the rolling rating blocks is shown in Figure 7. From Figure 7, it can be seen that the predicted RMSE and MAE values fluctuate with the rolling of rating blocks. This fluctuation may be caused by the difference of experiment data. Though they fluctuate with the rolling of the rating blocks, the RMSE and MAE gained by using the ITDBMF algorithm are minimized, which indicates that the prediction performance of the ITDBMF algorithm is better than that of other algorithms. The performance of other algorithms is similar to the ones in Table 5.

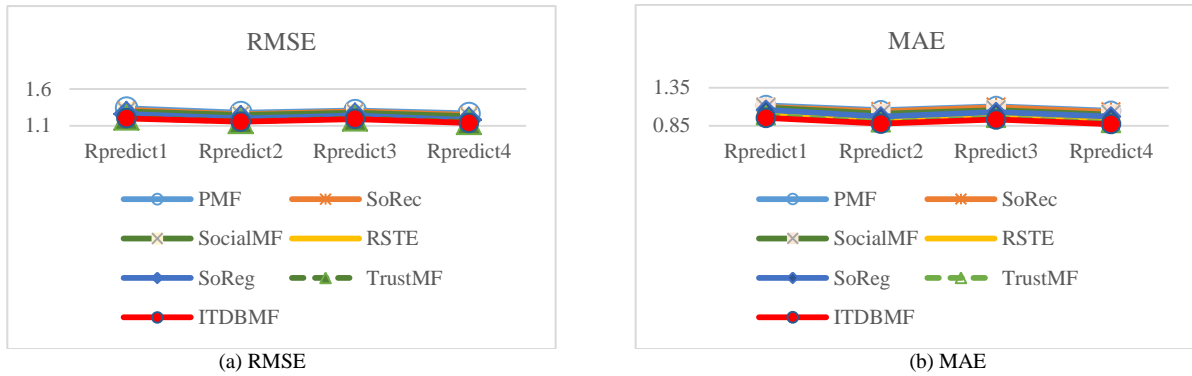


Figure 7. Prediction accuracy over rolling blocks

Figure 8 shows the RMSE and MAE of comparison algorithms over accumulative incremental rating blocks given in Figure 4. From the experimental results, we can see that the RMSE and MAE obtained by various recommendation algorithms decrease with the accumulation of historical data over time. It should be noted that the RMSE and MAE values obtained by the TDBMF algorithm and the ITDBMF algorithm are the smallest. This means these two algorithms outperform other ones in prediction accuracy. Compared with the values obtained in Figure 7, we can conclude that the use of sufficient historical data in training can increase the prediction accuracy.

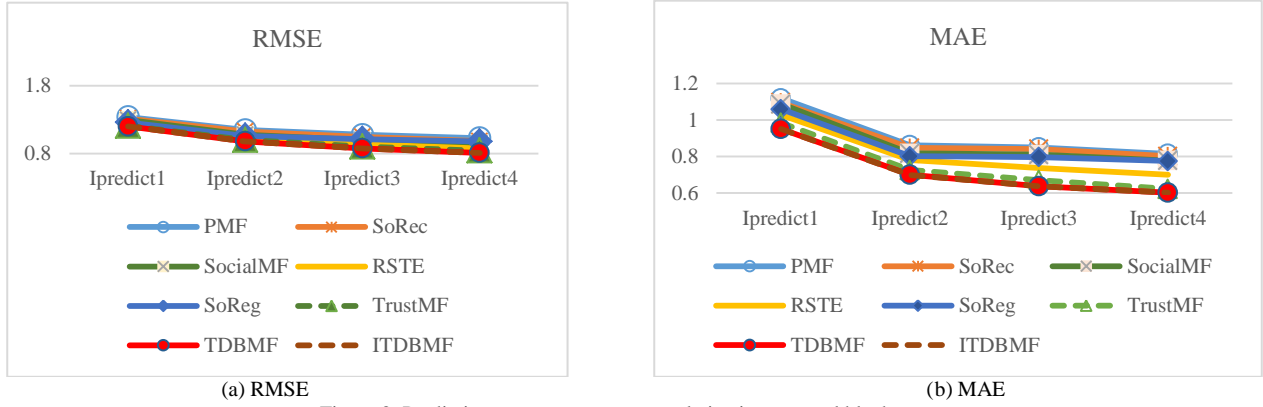


Figure 8. Prediction accuracy over accumulative incremental blocks

When comparing the ITDBMF and TDBMF algorithms, we can find that the ITDBMF algorithm only improves the TDBMF algorithm slightly (see Table 7). In this set of experiments, when predicting users' interests in blocks 2, 3, 4, and 5, the TDBMF algorithm directly uses rating data in block 1, blocks 1-2, blocks 1-3, and blocks 1-4. In contrast, the ITDBMF algorithm predicts users' interest in block 3 by incrementally considering the ratings in block 2 based on the ones in block 1 (denoted as 1+2). Similarly, when predicting users' interest in block 5, the ITDBMF algorithm incrementally considers the ratings in block 4 based on the ones obtained by incremental computation of blocks 1, 2, and 3.

Table 7. RMSEs and MAEs of TDBMF and ITDBMF

	RMSE		MAE	
	TDBMF	ITDBMF	TDBMF	ITDBMF
Ipredict1	1.2003	1.2003	0.9511	0.9511
Ipredict2	0.9820	0.9807	0.6998	0.6984
Ipredict3	0.8779	0.8764	0.6370	0.6354
Ipredict4	0.8121	0.8107	0.6014	0.6002

5. Conclusions

To solve the problems of data sparseness and cold starts in incremental data situations, we present a block-incremental recommendation algorithm called ITDBMF. This paper first gives a cascaded trust network construction algorithm, which uses the distrust relationship in purifying trust relationships. Then, based on the users' rating behaviour matrix and integration of the social network information, this paper proposes a static batch data recommendation algorithm named TDBMF. It considers the matrix factorization based on the social network and the social regularization based on the cascade trust network. Then, on the basis of the TDBMF algorithm, we propose a block-incremental recommendation algorithm called ITDBMF, which uses the Ebbinghaus forgetting function to decay historically incremental rating blocks and simultaneously consider the incremental social relationships. In order to verify the performance of the ITDBMF algorithm, this paper designs and implements two experiments based on a typical public data set. Experimental results show that the proposed algorithm outperforms all the other six typical static data-oriented matrix factorization algorithms in prediction accuracy. Moreover, the ITDBMF algorithm is helpful to alleviate the three inherent problems of the collaborative filtering algorithm and further improve the recommendation accuracy of matrix factorization algorithms.

Though the ITDBMF algorithm outperforms the benchmarked static data-oriented matrix factorization recommendation algorithms in prediction accuracy, there is still some literature that needs to be compared. First, other block-incremental matrix factorization algorithms such as those given in [21-22, 24] should be compared in the future. Moreover, the streaming recommendation algorithm should also be compared on trade of computation/storage complexity and recommendation accuracy. The idea of trust cascaded network and decay of historical ratings can also be applied in the calculation of a trustable advisor list in spam ratings detection/filtering field, which we have done recently [25].

Acknowledgements

This paper is supported in part by the Natural Science Foundation of China (No. 71772107, 71403151, 61502281, 61433012), Qingdao Social Science Planning Project (No. QDSKL1801138), National Key R&D Plan (No. 2018YFC0831002), Humanity and Social Science Fund of the Ministry of Education (No. 18YJAZH136), Key R&D Plan of Shandong Province (No. 2018GGX101045), Natural Science Foundation of Shandong Province (No. ZR2018BF013,

ZR2013FM023, ZR2014FP011), Shandong Education Quality Improvement Plan for Postgraduate, Leading Talent Development Program of Shandong University of Science and Technology, and Special Funding for Taishan Scholar Construction Project.

References

1. A. Abbasi, R. Y. K. Lau, and D. E. Brown, "Predicting Behaviour," *IEEE Intelligent Systems*, Vol. 30, No. 3, pp. 35-43, 2015
2. J. L. Zhao, S. Fan, and D. Hu, "Business Challenges and Research Directions of Management Analytics in the Big Data Era," *Journal of Management Analytics*, Vol. 1, No. 3, pp. 169-174, 2014
3. "Amazon's recommendation secret," (Available at <http://fortune.com/2012/07/30/amazons-recommendation-secret/>, Last accessed on March 3, 2018)
4. M. Jamali and M. Ester, "A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks," in *Proceedings of ACM Conference on Recommender Systems*, Vol. 45, pp. 135-142, 2010
5. T. Y. Ji, T. Z. Huang, X. L. Zhao, T. H. Ma, and G. Liu, "Tensor Completion using Total Variation and Low-Rank Matrix Factorization," *Information Sciences*, Vol. 326, No. C, pp. 243-257, 2016
6. H. Ma, I. King, and M. R. Lyu, "Learning to Recommend with Social Trust Ensemble," in *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp. 203-210, 2009
7. H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social Recommendation using Probabilistic Matrix Factorization," in *Proceedings of ACM Conference on Information & Knowledge Management, CIKM 2008*, Vol. 28, pp. 931-940, Napa Valley, California, USA, October 2008
8. H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender Systems with Social Regularization," in *Proceedings of Fourth International Conference on Web Search and Web Data Mining, WSDM 2011*, pp. 287-296, Hong Kong, China, February 2011
9. R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in *Proceedings of International Conference on Neural Information Processing Systems*, Curran Associates Inc, Vol. 29, pp. 1257-1264, 2007
10. B. Yang, Y. Lei, J. Liu, and W. Li, "Social Collaborative Filtering by Trust," in *Proceedings of International Joint Conference on Artificial Intelligence*, AAAI Press, Vol. 39, pp. 2747-2753, 2013
11. S. Y. Chang, Y. Zhang, J. L. Tang, D. W. Yin, Y. Chang, M. A. Hasegawa-Johnson, et al., "Streaming Recommender Systems," in *Proceedings of International Conference on World Wide Web*, pp. 381-389, 2017
12. Y. Wang, Y. Zhang, Y. Yin, D. Yi, and B. Wei, "A Cluster-based Incremental Recommendation Algorithm on Stream Processing Architecture," in *Proceedings of International Conference on Digital Libraries: Social Media and Community Networks*, Springer International Publishing, pp. 73-82, 2013
13. B. Liu and Z. Yuan, "Incorporating Social Networks and User Opinions for Collaborative Recommendation: Local Trust Network based Method," in *Proceedings of the Workshop on Context-Aware Movie Recommendation*, ACM, pp. 53-56, 2010
14. S. Deng, L. Huang, and G. Xu, "Social Network-based Service Recommendation with Trust Enhancement," *Expert Systems with Applications*, Vol. 41, No. 18, pp. 8075-8084, 2014
15. H. Fang, G. Guo, and J. Zhang, "Multi-Faceted Trust and Distrust Prediction for Recommender Systems," *Decision Support Systems*, Vol. 71, No. C, pp. 37-47, 2015
16. H. Wu, K. Yue, Y. Pei, B. Li, Y. Zhao, and F. Dong, "Collaborative Topic Regression with Social Trust Ensemble for Recommendation in Social Media Systems," *Knowledge-based Systems*, Vol. 97, No. C, pp. 111-122, 2016
17. W. Wu, J. Zhao, C. Zhang, F. Meng, Z. Zhang, and Y. Zhang, "Improving Performance of Tensor-based Context-Aware Recommenders using Bias Tensor Factorization with Context Feature Auto-Encoding," *Knowledge-based Systems*, Vol. 128, No. C, pp. 71-77, 2017
18. J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized Route Recommendation using Big Trajectory Data", in *Proceedings of International Conference on Data Engineering*, pp. 543-554, 2015
19. Y. X. Huang, B. Cui, W. Y. Zhang, J. Jiang, and Y. Xu, "TencentRec: Real-Time Stream Recommendation in Practice," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 227-238, 2015
20. G. Damaskinos, R. Guerraoui, and R. Patra, "Capturing the Moment: Lightweight Similarity Computations," in *Proceedings of International Conference on Data Engineering*, pp. 747-758, 2017
21. M. Nilashi, D. Jannach, O. B. Ibrahim, and N. Ithnin, "Clustering- and Regression-based Multi-Criteria Collaborative Filtering with Incremental Updates," *Information Sciences*, Vol. 293, No. 293, pp. 235-250, 2015
22. X. Luo, Y. Xia, and Q. Zhu, "Incremental Collaborative Filtering Recommender based on Regularized Matrix Factorization," *Knowledge-based Systems*, Vol. 27, No. 3, pp. 271-280, 2012
23. J. Vinagre and A. M. Jorge, "Forgetting Mechanisms for Scalable Collaborative Filtering," *Journal of the Brazilian Computer Society*, Vol. 18, No. 4, pp. 271-282, 2012
24. J. Vinagre, A. M. Jorge, and J. Gama, "Online Bagging for Recommendation with Incremental Matrix Factorization," 2016
25. S. J. Ji, H. Y. Ma, Y. Q. Liang, H. Leung, and C. Zhang, "A Whitelist and Blacklist-based Co-Evolutionary Strategy for Defending Against Multifarious Trust Attacks", *Applied Intelligence*, Vol. 48, No. 7, pp. 1-17, 2017

Shujuan Ji is an associate professor in the College of Computer Science and Engineering at Shandong University of Science and Technology. She received her B.S., M.S., and Ph.D. degrees from Shandong University of Science and Technology. Her main research interests include artificial intelligence and intelligent business information processing.

Da Li received his bachelor's degree from Qingdao University of Science and Technology. He is currently a Master's candidate at Shandong University of Science and Technology. His main research interests include artificial intelligence and intelligent business information processing.

Qing Zhang received her B.S. and M.S. degrees in computer science and engineering from Shandong University of Science and Technology. Her research interests include artificial intelligence and intelligent business information processing.

Chunjin Zhang is an engineer in the Center of Modern Education at Shandong University of Science and Technology. He received his B.S. degree in computer science and technology from the College of Information Science and Engineering at Shandong University of Science and Technology and his M.S. degree in control theory and control engineering from the College of Information and Electrical Engineering at Shandong University of Science and Technology.

Chunxiao Bao is a Master's student in the College of Computer Science and Engineering at Shandong University of Science and Technology. Her research interests include game theory and intelligent business information processing.