

A Mobile Recommendation Algorithm Based on Statistical Analysis of User Data

Chunyong Yin¹, Hui Zhang¹, Jun Xiang¹, Zhichao Yin² and Jin Wang¹

¹*Jiangsu Key Laboratory of Meteorological Observation and Information Processing, School of Computer and Software, Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China*

²*Nanjing No.1 Middle School, Nanjing, 210001, China*

Abstract

Recommendation technology is used to help people solve the problem of information overload. Recent years, it has been widely applied to the movie ratings, e-commerce and many other fields. Researchers have noticed its powerful application prospect. But with the exponential growth of information data, the recommendation systems also have to improve the ability of data processing and this leads to that the traditional collaborative filtering recommendation algorithms cannot meet the needs of the users. To solve the problem, we designed an algorithm based on the theory of statistical analysis. This algorithm classified the data simply firstly, and then system could give users the relatively satisfactory personalized recommendations by the statistical analysis of different attributes on the data sets.

Keywords: *recommendation technology, mobile electronic commerce, user behavior attributes, big data*

1. Introduction

Recent years, recommendation technology has been wisely used in e-commerce, advertising push, digital library and many other fields as a tool to solve problem of information overload for most web users. It is the extension and development of data mining technology. In the early 1990s, the beginning of the web technology, Robert Armstrong proposed some kind of idea similar to the recommendation technology, the idea was to research behaviors of different users to provide some specific customer useful or helpful suggestions. Then, this simple idea had been applied in different fields and gradually developed into the personalized recommendation technology. So far, Recommendation technology has formed a set of mature theories [1, 2].

Personalized recommendation technology was mainly used in the information filtering at an early age such as TAPSTRY system developed by Goldberg, Nicols and Oki. But with the cross development of global network technology and business activities, recommendation technology had been applied to a new field called e-commerce, so many research institutions like IBM Almaden research center, compaq research center and UC Berkeley begun to research this field [3].

A completed recommendation system consists of three modules: user modeling module, target object modeling module and recommendation algorithm module.

The recommendation system matches useful information of interests in user modeling module with the information of features in target object modeling module and uses the recommendation algorithm to screen those information at the same time, then finds the items to the target user.

In the recent ten years, the trend of the development for the Internet is unstoppable and this leads to the exponential growth of information data. In the face of such a large amount of data, most users cannot make a relatively better choice ; This is the problem of information overload. To solve the problem, we need an automated recommendation system, it can find the information we need from the large database according to the automated research on the data.

Though the technology has been developed for many years, the web data size is increasing so fast that the recommendation technology cannot adapt to such a big data processing which causes many problems such as data sparseness, cold boot and so on. To solve the problems, the researchers domestic and overseas had made unremitting efforts. Sarwar used the singular value decomposition (SVD) method for dimension reduction; Karypis designed a collaborative filtering algorithm based on the item to mitigation the problem of data sparseness; Cai Hao took trust model into consideration to design a new collaborative filtering algorithm [4-6].

But in the face of huge amounts, real-time and diversity for the big data, those algorithms still have limitations. For example, online business company Alibaba generates millions of data every day and each data has some different attributes, the traditional methods cannot adapt to this kind of problem.

So this article proposed a specific example of big data, discussed the feasibility of the traditional methods and designed a recommendation algorithm based on statistical analysis of user data. By the comparative analysis of the results for the different methods, the proposed methods improved the recommend quality and had relatively better results.

2. The Proposed Example

The data of the proposed example used in the article contains two sets. One is the data sets from some famous online business company, and it contains over 12000000 records that 10000 users to millions of the online goods, the other is the data set of user_id for the 10000 users.

Symbols are defined as follows: U is the complete set of user_id; I is the complete set of item_id; P is the subset of I ; D is the data set containing over 12000000 records that 10000 users to millions of the online goods, and every record of D is a tuple containing five elements of user_id, item_id, behavior_type, item_category and time. In the tuple, behavior_type includes 'Browse', 'collection', 'add shopping cart' and 'purchase'; We use 1,2,3,4 to label those 4 behaviors respectively.

We chose a subset of item, divided D to training set and testing set by different time periods, and predicted 'purchase' behaviors in the testing set about the subset. The predicted result was in the form of a tuple including user_id and item_id.

The *precision*, *recall* and *F1* value were used as the evaluation index.

3. The Traditional Recommendation Algorithms in the Recommendation System

As mentioned in the first chapter, a completed recommendation system is consists of three parts: (1) the record module, (2) the analysis model and (3) recommendation algorithm module.

Those three modules have their important role. Among them, the record module collects information about the behaviors of users; the analysis model analyzes the information collected by the record module; the recommendation algorithm module is used to screen the items that users may be interested in. Among those three modules, the recommendation algorithm module is the most important module in the whole recommendation system and it determine the quality of recommendation results, so this paper focuses on the recommendation algorithm module.

There are some traditional recommendation algorithms. Among them, collaborative filtering recommendation algorithm and machine learning recommendation algorithm are mostly used to solve the specific problems.

3.1. Collaborative Filtering Recommendation Algorithm

Collaborative filtering recommendation algorithm [7] is the commonly used method in recommendation system; it can find new items for the target user by comparing target user's records and the records of other users.

A commonly used Collaborative filtering recommendation algorithm is based on users. Its basic idea is to find the nearest neighbor similar to the specific user according to the known historical ratings data to produce the recommendation for the specific user. The whole process is completed by three steps:

3.1.1. The Representation and Pretreatment of Data Sets: The traditional collaborative filtering recommendation algorithm process the data simply and gain a user - item rating matrix $R[m,n]$ firstly. The matrix is shown in Figure 1. Among them, m-row shows users, n-line shows items and R_{ij} shows the score value of user i on item j .

	<i>item</i> ₁	...	<i>item</i> _{<i>j</i>}	...	<i>item</i> _{<i>n</i>}
<i>user</i> ₁	$R_{1,1}$...	$R_{1,j}$...	$R_{1,n}$
⋮	⋮	⋮	⋮	⋮	⋮
<i>user</i> _{<i>i</i>}	$R_{i,1}$...	$R_{i,j}$...	$R_{i,n}$
⋮	⋮	⋮	⋮	⋮	⋮
<i>user</i> _{<i>m</i>}	$R_{m,1}$...	$R_{m,j}$...	$R_{m,n}$

Figure 1. User-item Rating Matrix

3.1.2. Find the Nearest Neighbor: This step is the key step in recommendation algorithm which can determine the system's recommendation quality. In this step, the system calculates the similarity between the target user and other individual users according to the formula of similarity, find the nearest neighbor with highest similarity and get a collection of users $U=\{u_1, u_2, u_3, \dots, u_p\}$, The number order of similarity is from large to small, the Top-N obtained is the nearest neighbor set. In general, there are three methods to calculate the similarity: cosine method, modified cosine method and Pearson correlation method.

Cosine method is the commonly used method to calculate the similarity. Its basic idea is to calculate the included angle cosine of two vectors. It would not be restricted by the dimension, so it can be used to compare any numeric vectors in any dimensions. The formula is as follows:

$$sim(i, j) = \cos(i, j) = \frac{i \cdot j}{\|i\| * \|j\|} \tag{1}$$

Among them, i and j represent two different score vectors of two users respectively.

Pearson correlation similarity represents the linear relationship between fixed distance variables. Compared to the cosine similarity and modified cosine similarity, using Pearson correlation coefficient could bring us better recommendation results of collaborative filtering recommendation algorithm if the user-item rating matrix has dense data without the problem of data sparseness, otherwise the method would not have a satisfying performance. The formula is as follows:

$$sim(i, j) = \frac{\sum_{c \in I_{i,j}} (R_{i,c} - R_i)(R_{j,c} - R_j)}{\sqrt{\sum_{c \in I_{i,j}} (R_{i,c} - R_i)^2} \sqrt{\sum_{c \in I_{i,j}} (R_{j,c} - R_j)^2}} \quad (2)$$

Among them, I_{ij} is a set of items that both user i and user j have scores on, R_i represents the average score of user i and R_j represents the average score of user j , $R_{i,c}$ represents the score of user i on item c and $R_{j,c}$ represents the score of user j on item c .

Modified cosine method is developed from cosine method: in cosine method, it do not take the rating scale problem of different users into consideration, so modified cosine method relieves the problem by subtracting the average score of the user on all items. The formula is as follows:

$$sim(i, j) = \frac{\sum_{c \in I_{i,j}} (R_{i,c} - R_i)(R_{j,c} - R_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - R_i)^2} \sqrt{\sum_{c \in I_j} (R_{j,c} - R_j)^2}} \quad (3)$$

Among them, I_{ij} is a set of items that both user i and user j have scores on, I_i represents the set of items that user i have scores on, I_j represents the set of items that user j have scores on, R_i represents the average score of user i and R_j represents the average score of user j , $R_{i,c}$ represents the score of user i on item c and $R_{j,c}$ represents the score of user j on item c .

Among them, i and j represent two different score vectors of two users respectively.

3.1.2. Produce Recommendations: The system predicts items that the target user interested in according to the behavior information of the users in Top-N and recommends those items to the target user [8] [9]. The formula is as follows:

$$P_{i,c} = R_i + \frac{\sum_{m=1}^n sim(i, j) \cdot (R_{j,c} - R_j)}{\sum_{m=1}^n sim(i, j)} \quad (4)$$

Among them, i and j represent two different score vectors of two users respectively.

The system predicts items that the target user interested in according to the behavior information of the users in Top-N and recommends those items to the target user [8, 9]. The formula is as follows:

Among them, R_i represents the average score of user i and R_j represents the average score of user j , and $R_{j,c}$ represents the score of user j on item c . n is the number of the nearest neighbors.

3.2. Machine Learning Recommendation Algorithm

Machine learning recommendation algorithm based on data mining can extract features from the data set; analyze the basic attributes and key words according to the features [10-14]. System can study those interests of the users to produce the recommendation to the target user based on the behavior of other users on the items.

We can implement this algorithm simply by some machine learning tools like WEKA.

The whole process is completed by two steps: the tool chooses the appropriate subset from the data set as the training set to gain training model firstly, and then uses the model to run the testing set to get the recommendation for the target user. This algorithm is

simple to implement and it can avoid the disadvantages of data sparseness and cold boot, but its recommendation for the target user may be useless [15, 16].

4. The Recommend Algorithm Based on the Static Analyze of User Data

In the proposed example, we found some interesting regulars when we do the pretreatment on the data: (1) users will not buy the items after more than four days without any behaviors on those items; (2) different user_behaviors have different influences to the users; (3) users often have the purchase_behavior in the items of the same category.

So, we need the theory of statistical analysis to analyze the data to find the specific expression of these regulars

4.1. Algorithm Analysis

As According to the actual situation of data, we proposed a recommendation algorithm based on statistical analysis of user behavior which could take more about the characteristics of user behavior into consideration.

We divided the data set into a training set and a test set according to the requirements of the problem. We defined a user-item attribute set matrix R :

$$\begin{array}{ccccccc}
 & & item_id_1 & \dots & item_id_j & \dots & item_id_n \\
 user_id_1 & R_{user_id_1,item_id_1} & \dots & R_{user_id_1,item_id_j} & \dots & R_{user_id_1,item_id_n} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 user_id_i & R_{user_id_i,item_id_1} & \dots & R_{user_id_i,item_id_j} & \dots & R_{user_id_i,item_id_n} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 user_id_m & R_{user_id_m,item_id_1} & \dots & R_{user_id_m,item_id_j} & \dots & R_{user_id_m,item_id_n}
 \end{array}$$

Figure 2. User-item Attribute Set Matrix

Among them, m represented m users in the data set, and n represented n kinds of items, R_{ij} represented all information of $user_id_i$'s behavior to $item_id_j$, it was a set which every element was a triad. It contained three attributes, behavior_type, item_category and time.

Definition 1: When there is no intersection between user i and item j , R is an empty set, otherwise R is a nonempty set. We extract all the nonempty sets.

Definition 2: Testing set time window $\Delta t_{testing}$ is the time quantum of 'time' attribute in all data of Testing set, training set time window $\Delta t_{training}$ is the time quantum of 'time' attribute in all data of training set.

It should be noted that the matrix in this section was different from the user-item rating matrix in the previous section. The matrix in this section was just used to define set R more detail and this was convenient for the next static analysis, the element in $R[m,n]$ was also a subset of D , while the element in the user - item rating matrix $R[m,n]$ was a specific quantitative value which could be used in the formula of similarity directly.

The basic idea of the proposed algorithm was to study the interests of the uses according to static analysis for all attributes of all sets in the matrix. All attributes could be divided into two kinds: primary attributes which included behavior_type and time and secondary attribute which included item_category. The primary attributes not only had their own independence but also had the correlation with each other, while the secondary attribute only had the correlation with other attributes. We analyzed the independence of the attributes firstly and then analyze correlation of the attributes.

4.1.1. The Independence of Time

Time is a very important factor to affect the users in recommendation system, because the interests of the users would be influenced by the change of time, but in the most recommendation systems, time was not taken into consideration. So we must pay attention to the influence of time especially in e-commerce.

First we collected statistics of all purchasing behavior in the Δt_{test} , then find out all sets $R_1, R_2, R_3, \dots, R_{k0}$ of $\Delta t_{training}$ in $R[m,n]$ which the user_id and item_id were the same as the user_id and item_id in Δt_{test} at the same time, the 'time' attribute distribution could be got from the statistics above. By parity of reasoning, we could choose different testing set time windows and training set time windows, repeat the above process and get the 'time' attribute distributions, from those distributions, we could find that when the user purchased some item, the user would have some kind of behavior on the item at 15 pm to 23 pm yesterday, so we chose all records at 15 pm to 23 pm in the day before the testing set into the set D_t .

4.1.2. The Independence of Behavior Type

Analyzing the independence of behavior type, the purchasing behavior of users in time window Δt_{test} would be counted, then found out all sets $R_1, R_2, R_3, \dots, R_{k0}$ of $\Delta t_{training}$ in $R[m,n]$ which the user_id and item_id were the same as user_id and item_id in the Δt_{test} at the same time, the 'behavior' attribute distribution could be got from the statistics above. By parity of reasoning, we could choose different testing set time windows and training set time windows, repeat the above process and get the 'behavior' attribute distributions. From the distributions we could find that when a user wanted to buy some item, most users would choose to add the item into shopping cart first, so we extracted the data which have 'add shopping cart' behavior from training set into D_b .

4.1.3. Attribute Correlation

First we considered the correlation between behavior type and behavior time. Let $D_b \cap D_t$, we could get D_{tb} , and then analyzed the correlation of three kinds attributes of D_{tb} . Deleted the records that had bought the same kind of item, put the rest records into data set D_{tb}' , finally let $D_{tb}' \cap D'$, then we get the result D_{result}' .

For example, user A added the item X into shopping cart at 17 o'clock of December 17th, we used $D_1 (D_1 \in D_{tb})$ to represent this behavior, and item X belongs to a category. But we could find another record that user A bought item Y at 18 o'clock of December 17th, which also belongs to a category, here we used $D_2 (D_2 \notin D_{tb})$ to represent the behavior, and then we delete D_1 from D_{tb} .

4.2. Steps of the Algorithm

Step 1 Input: the data set D of all user behavior, and the data set D' associated with the item subset P .

Step 2 Divide the data set D into several sub data sets according to the time.

Step 3 Put all the records which need to be predicted between 15 o'clock and 23 o'clock into data set D_t .

Step 4 Put all the data which contains behavior 3(add shopping cart) into data set D_b .

Step 5 Let $D_b \cap D_t = D_{tb}$.

Step 6 Delete the records that have bought the same kind of item, put the rest records into data set D_{tb}' .

Step 7 Let $D_{tb}' \cap D' = D_{result}'$, extract user_id and item_id.

5. Experimental Results

We divide D into two parts, a part is the data set which will be used as the testing set, and the rest of the data as the training set. According to the prediction results (D_{result}) obtained by the training data set, compared to the true results (D_{result}), The *precision*, *recall* and *F1* value were used as the evaluation index. Specific calculation formula is as follows:

$$Precision = \frac{|\cap (Prediction\ Set, ReferenceSet)|}{|Prediction\ Set|} \tag{5}$$

$$Recall = \frac{|\cap (Prediction\ Set, ReferenceSet)|}{|ReferenceSet|} \tag{6}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

The PredictionSet is a data set which contains the algorithm’s predicted data on buying. The ReferenceSet is a true data set. We take the *F1* value as the final evaluation criteria. We use the different commodity subsets to calculate precision, recall and *F1*.

First, we analyzed the traditional collaborative filtering algorithm. The traditional collaborative filtering algorithm is often used in Score predicts, so it relies on user-item rating matrix in a single score value, but there are some attributes which cannot be quantified in this paper’s data set such as behavior time, item categories and so on. It is very difficult to integrate all these attributes into a single quantitative value, especially according to our analysis, the effect of time attribute is a large proportion. Secondly, because of the problem of large amount of data, and the user and most of the items do not have any intersection and it will cause the matrix sparse and other issues which leads to a negative impact. So the traditional collaborative filtering algorithm has the defects of solving the problem in this paper.

Then we used the machine learning recommendation algorithm and recommendation algorithm proposed in the paper to get the different recommendation results, compared the results.

The following Tables are the results of the algorithm presented by machine mining and the next one is presented in this paper.

Table 1. Machine Mining

Prediction	Precision	Recall	F1
12.18	0.024927577	0.022722646	0.023774097
12.17	0.032486726	0.035857143	0.034088827
12.16	0.022437811	0.024235955	0.023302246
12.15	0.033393443	0.033513514	0.033453371
12.14	0.023333333	0.027729977	0.025342375

Table 2. Recommendation Algorithm based on Statistical Analysis of User Data

Prediction	Precision	Recall	F1
12.18	0.043313725	0.053435115	0.047844995
12.17	0.052208835	0.066326531	0.058426966
12.16	0.046332046	0.053932584	0.049844237
12.15	0.052427184	0.060810811	0.056308655
12.14	0.043071161	0.052631579	0.047373841

The results are accurate to nine decimal places, but in the tables above, it is hard to compare the difference between the data, so we express the data in the form of following Figures.

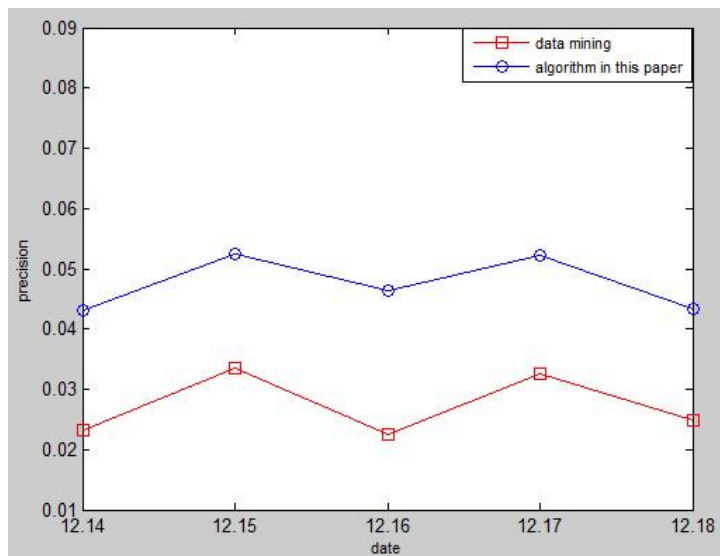


Figure 3. Comparison of Two Methods for *Predicting Accuracy*

In Figure 3, through the comparison of the *predicting accuracy*, we can see that the accuracy of traditional machine mining algorithm in the data set is not high, this method in our paper significantly is better than the traditional methods, and improving the accuracy through statistical analysis of user behavior.

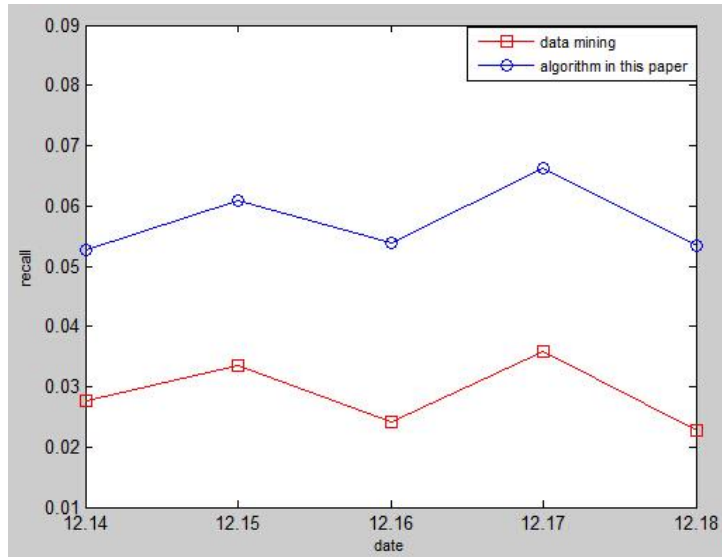


Figure 4. Comparison of Two Methods for Predicting Recall

In Figure 4 we can see that the *recall* rate is higher than the traditional method, although daily predicting results are not very stable, but overall is still in the 5% - 7%, the effect is superior to the traditional method.

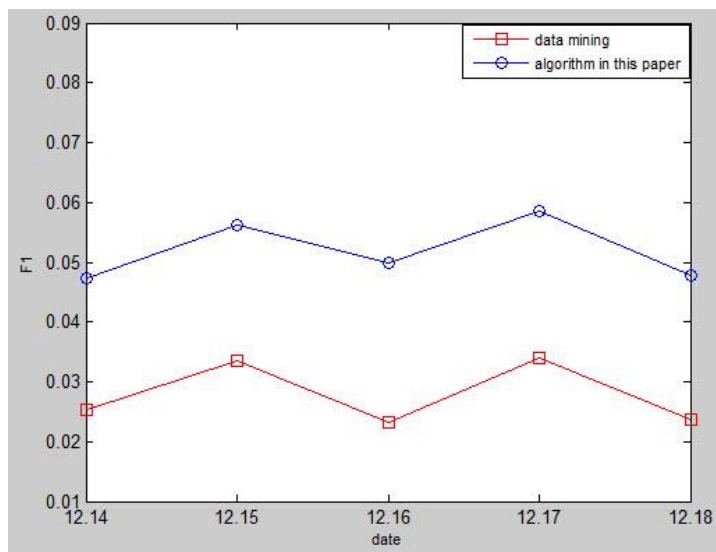


Figure 5. Comparison of Two Methods for Predicting F1 Value

Through the comparison of the final *F1* values in Figure 5, it is not hard to find that the algorithm proposed in this paper is better than the traditional mining algorithm.

Through the analysis, we found that although traditional machine mining algorithm can extract more features to avoid the cold start problem, when single attribute elements in excessive (such as item categories), its effect is still poor, and this method is not suitable for large scale data processing.

The algorithm of this paper can be divided into primary and secondary attributes, which can be used to consider the correlation of attributes, and the result is better than the traditional recommendation algorithm.

In general, from the recommendation results, each indicator in the recommendation algorithm proposed is better than the indicator in the traditional recommendation algorithms, even collaborative filtering recommendation algorithm is not suitable for this

example proposed. So the recommendation algorithm proposed has the better performance in the propose example.

6. Possible Improvements

But the recommendation algorithm proposed in the paper is still not perfect because its recommendation results are still low of precision. We analyze the algorithm to figure out the reasons.

We find that when the whole data was pretreated, we deleted a part of the users that who will have the purchase_behavior. We call those users casual users, because those users have no behaviors on the items before they have purchase_behavior. And we also assume that although the traditional recommendation algorithms have a poor performance, we can still make full use of the traditional recommendation algorithms to get the better recommendation results.

So we assume that better results will be got if we take those users into consideration or design a hybrid recommendation algorithm.

7. Conclusions and Future Work

As an important research direction, personalized recommendation has been widely applied to many virtual domains like e-commerce, e-news and so on. With the continuous expansion of the scale of data, some of the traditional recommendation algorithm has exposed many drawbacks. To solve these problems, this paper analyzed the traditional recommendation algorithm and its problems, a recommendation algorithm based on statistical analysis of user behavior was designed, and finally we got a high quality result without data sparse and cold start problems.

Next step, we will take adding the traditional algorithm into consideration to design a new hybrid recommendation algorithm, take those casual users into consideration. More experiments need to be done for the further research to achieve a better result.

Acknowledgments

This paper is a revised and expanded version of a paper entitled “A New Mobile Recommendation Algorithm Based on Statistical Theory” presented at AITS 2015, Harbin, China, August 21-23, 2015. This work was funded by the National Natural Science Foundation of China (61373134, 61402234), and by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Trade, Industry and Energy (MOTIE) Korea. It was also supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), Jiangsu Key Laboratory of Meteorological Observation and Information Processing (No.KDXS1105) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET). Prof. Jin Wang is the corresponding author.

References

- [1] A. Wei, L. Qihua and Z. Liyi, “Research Progress on the Diversity of Personalized Recommendation System”, *Library and Information Service*, vol. 57, no. 20, (2013), pp. 127-135.
- [2] M. Hongwei, Z. Guangwei and L. Peng, “Overview of Collaborative Filtering Recommendation Algorithm”, *Journal of Chinese Computer Systems*, vol. 30, no. 7, (2009), pp. 1282-1288.
- [3] K. Goldberg, T. Roeder, D. Gupta, et al., “Eigen taste: A Constant Time Collaborative Filtering Algorithm”, *Information Retrieval*, vol. 4, no. 2, (2001), pp. 133-151.
- [4] Z. Ting and X. Ruliang, “A Personalized Recommendation Algorithm for Roulette Wheel based on the Combined Effect of Time”, *Journal of Computer Applications*, vol. 34, no. 4, (2014).
- [5] A. Yue, L. Bin and Y. Ruitai, “Personalized Recommendation Research based on Content of Popular Micro Topics”, *Library and Information Service*, vol. 33, no.2, (2014), pp. 155-160.
- [6] Z. H. Yan, D. Feng and J. L. Zhong, “A collaborative filtering recommendation method based on fuzzy clustering”, *Computer Simulation*, vol. 22, no. 8, (2005), pp. 144-147.

- [7] H. Guoyan, L. Youchao, G. Jianpei and C. Xuliang, "Recommendation Algorithm based on Item Attributes of User Clustering Collaborative Filtering", *Computer Engineering and Design*, vol. 31, no. 5, (2010), pp. 1038-1041.
- [8] C. Hao, "Research on Collaborative Filtering Recommendation Method based on User Trust Model", *Computer Engineering and Application*, vol. 46, no. 35, (2010), pp. 148-151.
- [9] S. Yuan and C. Cheng, "Ontology-based personalized couple clustering for heterogeneous product recommendation in mobile marketing", *Expert Systems with Applications*, vol. 26, no. 4, (2004), pp. 461-476.
- [10] C. Yin, "Towards accurate node-based detection of P2P botnets", *Scientific World Journal*, vol. 20, no. 14, (2014), pp. 425491-425491.
- [11] C. Yin, M. Zou and D. I. J. Wang, "Botnet Detection Based on Correlation of Malicious Behaviors", *J. International Journal of Hybrid Information Technology*, vol. 6, no. 6, (2014), pp. 291-300.
- [12] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano and S. Li, "Incremental, 'Support Vector Learning for Ordinal Regression', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, (2015), pp. 1403-1416.
- [13] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman and S. Li, "Incremental learning for v-Support Vector Regression", *Neural Networks*, no. 67, (2015), pp. 140-150.
- [14] K.-Y. Lee, J.-J. Kang, J.-W. Yoon, Y.-G. Jung, G.-S. Choi, S.-B. Park and E.-Y. Kang, "A Summary and Recommendation System based on Bio-Text Analysis", *IJBSBT*, vol. 6, no. 2, (2014), pp. 117-128.
- [15] D. Tomar and S. Agarwal, "A Survey on Data Mining approaches for Healthcare", *IJBSBT*, vol. 5, no. 5, (2013), pp. 241-266.
- [16] J. Wang, J.-U. Kim, L. Shu, Y. Niu and S. Lee, "A distance-based energy aware routing algorithm for wireless sensor networks", *Sensors*, vol. 10, no. 10, (2000).
- [17] J. Wang, Y. Yin, J. Zhang, S. Lee and R. S. Sherratt, "Mobility based energy efficient and multi-sink algorithms for consumer home networks", *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, (2013).
- [18] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering", In *Workshop on Recommendation Systems at the Fifteenth National Conference on Artificial Intelligence*, (1998) July 26-30, pp. 178-181, Madison, Wisconsin, USA.

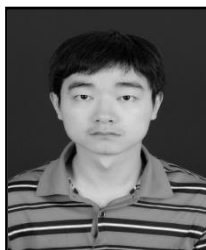
Authors



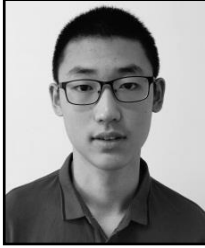
Chunyong Yin, is currently an associate Professor and Dean with the Nanjing University of Information Science & Technology, China. He received his Bachelor (SDUT, China, 1998), Master (GZU, China, 2005), PhD (GZU, 2008) and was Post-doctoral associate (University of New Brunswick, 2010). He has authored or coauthored more than twenty journal and conference papers. His current research interests include privacy preserving and network security.



Hui Zhang, obtained his B.E. degree in the Measurement and Control from Nanjing University of Information Science and technology, China in 2014. Currently he is a graduate student at the School of Computer and Software of Nanjing University of Information Science & Technology. His current research interests are in big data mining and recommendation technology.



Jun Xiang, received his bachelor degree in 2014 from Nanjing University of Information Science & Technology. He is studying for his master's degree in Nanjing University of Information Science & Technology. His main research interests include data mining, data security and privacy protection.



Zhichao Yin, is studying in Nanjing No.1 Middle School. His current research interests include network security and mathematical modeling.



Jin Wang, received the B.S. and M.S. degree from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree from Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and technology. His research interests mainly include routing method and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.