

Personalized Recommendation Algorithm Based on Preference Features

Liang Hu, Guohang Song, Zhenzhen Xie, and Kuo Zhao*

Abstract: A hybrid collaborative filtering algorithm based on the user preferences and item features is proposed. A thorough investigation of Collaborative Filtering (CF) techniques preceded the development of this algorithm. The proposed algorithm improved the user-item similarity approach by extracting the item feature and applying various item features' weight to the item to confirm different item features. User preferences for different item features were obtained by employing user evaluations of the items. It is expected that providing better recommendations according to preferences and features would improve the accuracy and efficiency of recommendations and also make it easier to deal with the data sparsity. In addition, it is expected that the potential semantics of the user evaluation model would be revealed. This would explain the recommendation results and increase accuracy. A portion of the MovieLens database was used to conduct a comparative experiment among the proposed algorithms, i.e., the collaborative filtering algorithm based on the item and the collaborative filtering algorithm based on the item feature. The Mean Absolute Error (MAE) was utilized to conduct performance testing. The experimental results show that employing the proposed personalized recommendation algorithm based on the preference-feature would significantly improve the accuracy of evaluation predictions compared to two previous approaches.

Key words: recommendation system; collaborative filtering; user preference

1 Introduction

Personalized service is an important trend in the development of information processing technology. With the continuous rapid development and improvement of Internet technology, there has been an explosive growth of information on the Internet^[1,2]. Although publicly available consumer search engines have become the most effective to search the Internet, these tools fail to satisfy all user demands and preferences. Consequently, personalized service technology has been developed^[3]. Personalized service provides an automatic function that recommends

items by obtaining and analyzing user information; predictions based on the analysis and information are made prior to the user launching a search. The core value of personalized services lies in its recommendation capability. The appropriate use of recommendation algorithms that improve the accuracy of recommendations and algorithms that return results consistent with user interests have become bottlenecks for the use and development of personalized services^[4]. To realize better personalized services, new algorithms continue to be developed. Present there are three mainstream recommendation algorithms: The first type is the personalized time series algorithm. This type of algorithm develops a time series relative to user search behavior and then analyzes potential user needs to provide recommendations. Obviously, this method is employed by administrators rather than users. The administrator only has to embed the most appropriate series at the very beginning of a search to make it work. However, personalized time series algorithms

• Liang Hu, Guohang Song, Zhenzhen Xie, and Kuo Zhao are with the School of Computer Science and Technology, Jilin University, Changchun 130012, China. E-mail: hul@jlu.edu.cn; songguohang1988@sina.com; kappaxie@sina.com; zhaokuo@jlu.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2014-05-07; accepted: 2014-05-12

lack accuracy and do not meet different users' demands. The second type is the user-item association based algorithm. This method generates matching pairs by analyzing the associations between user data and resources and filtering for similarity between users and resources. Even though this type of algorithm is much better than the time series based algorithm, it still demands significant effort on the part of an administrator because, initially, there is insufficient associated user and resource data. In addition, there are a great many noisy and useless resources. Reducing the amount of useless resources is difficult and achieving dynamic updates is virtually impossible. The third type is the collaborative filtering algorithm. This method takes good advantage of similarities between users with similar preferences. It will provide new items that have been previously viewed by other users who have the same preferences or search demands as the active user. By using this method, an administrator only has to match users with similar features. The advantages of this method are its high accuracy and ease of searching items that match user interests. These advantages make the collaborative filtering algorithm a well-received and mainstream technique.

However, there are challenges associated with collaborative filtering algorithms. (1) Data sparsity. When a new user or item first enters the system, finding similar data is difficult due to the lack of information, which gives rise to the cold start problem^[5]. (2) Extendibility. When there are billions of users and millions of items, the time complexity will be very large. Many systems demand extremely rapid response to meet performance agreement requirements without regard to the users' purchasing record or evaluation history. Consequently, a Collaborative Filtering (CF) recommendation system must have high extendibility. (3) Similarity. Some items that are in fact very similar can have different names or contexts, therefore, most recommendation systems would not be able to detect potential associations and would treat similar items as different^[6,7]. To solve these problems, a preference-feature based personalized recommendation algorithm has been proposed. A thorough investigation of CF techniques preceded the development of this algorithm. The proposed algorithm improves the user-item similarity measurement by extracting item features and confirms item feature by calculating the weights of different feature. User preference data is obtained by extracting item ratings. Recommendations are made

according to the feature and user preference, which is expected to improve recommendation accuracy significantly. The proposed algorithm also handles the data sparsity problem and can find potential semantic patterns in user ratings. Collectively, these features would make predictions more understandable and more accurate.

2 Preference-Feature Model Development

2.1 User preference and item features

2.1.1 User preference features

Users' browser histories are not random; they reflect long-term regular use that corresponds to individual user preferences^[8,9]. The predictions made by traditional CF methods based on user preference obtained by ratings provided for movies that a user has watched are not valid. Users' special interests in particular movies will result in skewed ratings and therefore cannot represent a user's actual preference for such types of movie elements which are the information that included in the movie such as type, actor, and director. It is not representative to determine user preferences based on a limited number of particular ratings. Consequently, the investigation presented in this paper analyzes user movie ratings over a period of time to determine a stable preference and calculates similarity among users to detect neighbor users^[10].

2.1.2 Item features

Different items have different elements that are used to label items and attribute different weights. For example, the movie $i_1 = \langle \text{Titanic} \rangle$ has two genre elements, "drama" and "romance". Even for users who only provide a vague rating for the movie during an evaluation, the result represents the sum of the evaluation of different movie elements. For example, even though movie $i_2 = \langle \text{Notting Hill} \rangle$ has "drama", "romance", and "comedy" elements, the various ratings might only be related to the "comedy" element. For movie $i_3 = \langle \text{Bruce Almighty} \rangle$, which has both "comedy" and "romance" elements, the rating discrepancy for the same user u_1 might be related to both the "comedy" and "romance" elements. If $R_{u_1 i_2} - R_{u_1 i_3}$ is extremely large, that indicates that the "romance" element exerts little influence on the ratings of the two movies. If $R_{u_1 i_2} - R_{u_1 i_3}$ is small, then the "romance" element is pivotal to the evaluation of the two movies. We could calculate

each element's weight for the rated movies using user ratings and make predictions according to the weight combinations. For example, if users show high preferences for "adventure" and "romance" elements, the system would preferentially recommend movies with both elements.

2.2 User preference model development

In this section, we use Term Frequency-Inverse Document Frequency (TF-IDF) to reference user u 's rating R_{ui} for movie I as the weight of the calculation of user preference for elements. The TF-IDF model is commonly used for text character description. It is a statistical evaluation that primarily shows the importance of some particular words to the whole text or a set in a certain set. TF represents the occurrence frequency of a certain word in a given text, and it shows the ability of this text to be distinguished from others. IDF represents the appearance frequency of a particular word. It shows generality and decreases distinctiveness. Here we let TF_{ut} be the occurrence frequency of a particular element in users' ratings and $IDF(t)$ be the occurrence frequency of one particular element in all movies. The following equation provides the definition of user preference $P(u, t)$ for one particular element t in movie items:

$$P(u, t) = TF_{ut} \times IDF(t) \quad (1a)$$

$$TF_{ut} = \frac{\sum_{k \in I_R} R(u, i_k)}{\sum_{s=1}^e T(s, j)} \quad (1b)$$

$$IDF(t) = \log \frac{\sum_{j=1}^N \sum_{s=1}^e T(s, j)}{\sum_{j=1}^N T(t, j)} \quad (1c)$$

$\sum_{k \in I_R} R(u, i_k)$ is the rating sum of all the movies with t

element. $\sum_{s=1}^e T(s, j)$ is the element numbers of all the

rated movies. $\sum_{j=1}^N \sum_{s=1}^e T(s, j)$ is the number of movies

in all the data set. $\sum_{j=1}^N T(t, j)$ is the number of elements

in all the data set. N is the the number of movie in the data set.

The scores and ratings users provided for movies directly show users' subjective preferences. In Eqs. (1), if the element t appears frequently in the movies user

u had watched, but $\frac{\sum_{k \in I_R} R(u, i_k)}{\sum_{s=1}^e T(s, j)}$ is high, it means

that the user u has a high degree of preference for element t , and user u would like to obtain element t through watching movies. If a user watched many

movies with the same element but $\frac{\sum_{k \in I_R} R(u, i_k)}{\sum_{s=1}^e T(s, j)}$ is

low, that indicates that user u is interested in element t but it is hard for them to obtain search results for movies with element t . In other words, the user would benefit from a system that recommends movies with element t . Equation (1c) shows the spread of element t in the set of movie items. This will ensure that the experimental results will not be affected by other ubiquitous elements, such as "drama".

Use normalization computing to calculate Eqs. (1) and let $\sum_{t=1}^m w_{u,t} = 1$, w is user u 's preference weight for some particular movie element t . Let $w_{u,t} \in (0, 1)$ and m be the number of elements in the movie dataset. For user u , the preference vector will be $w_u(w_{u,1}, w_{u,2}, w_{u,3}, \dots, w_{u,m})$. To process all the element preferences of user set U , we employ user preference matrix P (see Table 1).

2.3 Item feature model development

The weight of an item element corresponds to the user preference. Even though these two are similar, essentially user preference represents the subjective attitude of users and the item element reflects the objective truth of the user group, which is also an objective item property. In this section, we will explain how to calculate the element feature's weight to the item using the user preference matrix. TF_{it} represents the rating weights of element t in all the rated

Table 1 User preference matrix.

	t_1	t_2	t_3	t_m
u_1	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,m}$
u_2	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,m}$
u_3	$w_{3,1}$	$w_{3,2}$	$w_{3,3}$	$w_{3,m}$
u_4	$w_{4,1}$	$w_{4,2}$	$w_{4,3}$	$w_{4,m}$

movies. $IDF(t)$ represents the occurrence frequency of one particular element in all the movies. Thus, $P(u, t)$ can be defined as the user preference for some particular element in movie items. We define $Q(i, t)$ as the weight of element t to item i :

$$\begin{aligned} Q(i, t) &= TF_{it} \times IDF(t), \\ TF_{it} &= \frac{\sum_{j \in I_R} R(u_j, i) \times w_{u_j, t}}{\sum_{s=1}^e T(u, s)}, \\ IDF(t) &= \log \frac{\sum_{j=1}^N \sum_{s=1}^e T(s, j)}{\sum_{j=1}^N T(t, j)} \end{aligned} \quad (2)$$

$R(u_j, i)$ means the user u_j 's rate to the movie i . $w_{u_j, t}$ is user u_j 's preference for element t . $\sum_{j \in I_R} R(u_j, i) \times w_{u_j, t}$

is the rating sum of movies with element t . $\sum_{s=1}^e T(u, s)$ is the number of the types of the movies which have been rated by the users who have rated this movie. The movie ratings directly show user preference. Equation (2) shows that, if the movies user u watched had more

element t , and $\frac{\sum_{k \in I_R} R(u, i_k)}{\sum_{s=1}^e T(s, j)}$ is high, then element

t constitutes the majority in the movie item; in other words, this element is the primary feature.

Use normalization computing to calculate Eq. (2) and let $\sum_{t=1}^m v_{i, t} = 1$. v is the preference weight of some particular element t to all movies items. Let $v_{i, t} \in (0, 1)$ and m is the number of elements obtained by all the movie items. For item i , the item feature vector would be $v_i (v_{i, 1}, v_{i, 2}, v_{i, 3}, \dots, v_{i, m})$. To process all the element preferences of user set U , we employ item feature matrix Q (see Table 2).

3 Preference-Feature Based Top-N Recommendations

3.1 Preference-feature similarity calculation

Similarity is the standard that measures the relationship between user-user, item-item. The common

Table 2 Item feature matrix.

	t_1	t_2	t_3	t_m
i_1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	$v_{1,m}$
i_2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	$v_{2,m}$
i_3	$v_{3,1}$	$v_{3,2}$	$v_{3,3}$	$v_{3,m}$
i_4	$v_{4,1}$	$v_{4,2}$	$v_{4,3}$	$v_{4,m}$

measurements are as follows^[11,12]:

Cosine similarity. In the users-item rating matrix, $R_{u,i}$ denotes the ratings of item I_t rated by user u .

$$\text{sim}_{u,u'} = \frac{\sum_{i \in I} (R_{u,i} R_{u',i})}{\sqrt{\sum_{i \in I} (R_{u,i})^2} \sqrt{\sum_{i \in I} (R_{u',i})^2}}.$$

Constrained cosine similarity. Let u and u' be the users who have rated the same movie. In the intersection of these two sets, $I' = (I_u \cap I_{u'})$ is the set of items that were rated by both users. Then the similarity of u and u' is:

$$\text{sim}_{u,u'} = \frac{\sum_{i \in I} (R_{u,i} - \bar{R}_u)(R_{u',i} - \bar{R}_{u'})}{\sqrt{\sum_{i \in I} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I} (R_{u',i} - \bar{R}_{u'})^2}}.$$

\bar{R}_u is user u 's average rate for all items. The value of $\text{sim}_{u,u'}$ is in the range $[0,1]$. The larger $\text{sim}_{u,u'}$ is, the greater the similarity of user u is to user u' .

Here, we use a Pearson correlation coefficient to calculate similarity. The process for calculating the Pearson correlation between user u and user i is as follows. First, calculate user preference for items that had not been rated:

$$\text{sim}(u, u') = \frac{\sum_{i \in I} (w_{u,t} - \bar{w}_u)(w_{u',t} - \bar{w}_{u'})}{\sqrt{\sum_{i \in I} (w_{u,t} - \bar{w}_u)^2} \sqrt{\sum_{i \in I} (w_{u',t} - \bar{w}_{u'})^2}}.$$

Then calculate the feature similarity between items:

$$\text{sim}(i, i') = \frac{\sum_{i \in I} (v_{i,t} - \bar{v}_i)(v_{i',t} - \bar{v}_{i'})}{\sqrt{\sum_{i \in I} (v_{i,t} - \bar{v}_i)^2} \sqrt{\sum_{i \in I} (v_{i',t} - \bar{v}_{i'})^2}}.$$

The sum of $i \in I$ is the number of items that had been rated by both users u and u' . \bar{w}_u is the average preference weight of user u . \bar{v}_i is the average type weight of item i .

3.2 Prediction of user rating

We assign different weights σ and $1 - \sigma$ for the user preferences and item feature similarity to calculate

the integrated similarity. The first part represents the subjective feelings of the users and the second represents the objective elements of the items. Through the weighted feature similarity calculation, we can derive the prediction equation for a user and an item:

$$p_{u,i} = \sigma \left(\frac{\overline{R}_u + \frac{\sum_{u' \in N} \text{sim}(u, u')(R_{u'i} - \overline{R}_{u'})}{\sum_{u' \in N} \text{sim}(u, u')}}{\overline{R}_i + \frac{\sum_{i' \in N} \text{sim}(i, i')(R_{ui'} - \overline{R}_{i'})}{\sum_{i' \in N} \text{sim}(i, i')}} \right) + (1 - \sigma) \left(\frac{\overline{R}_i + \frac{\sum_{i' \in N} \text{sim}(i, i')(R_{ui'} - \overline{R}_{i'})}{\sum_{i' \in N} \text{sim}(i, i')}}{\overline{R}_u + \frac{\sum_{u' \in N} \text{sim}(u, u')(R_{u'i} - \overline{R}_{u'})}{\sum_{u' \in N} \text{sim}(u, u')}} \right).$$

4 Evaluation

MovieLens was used as the database for the evaluation experiments^[13]. MovieLens is a movie recommendation system developed by GroupLens in 1997. We selected a portion of the data containing 980 users, 100 000 rating records, and 1734 movie items (each user has more than 20 ratings). Primarily, we tested the Mean Absolute Error (MAE) of the proposed user preference-feature based hybrid collaborative filtering algorithm^[14] by calculating the average value of the absolute error between the prediction ratings and actual ratings. n is the total number of user ratings. $p_{u,i}$ is user u 's predicted rating for item i . $R_{u,i}$ is user u 's actual rating for item i . The lower the MAE value is, the more accurate the prediction is.

$$\text{MAE} = \frac{1}{n} \sum_{u,i} |p_{u,i} - R_{u,i}|.$$

Due to data sparsity and the selection of neighbor user, which would greatly affect the accuracy of the algorithm, we paid particular attention to two points.

First, we selected weights in an appropriate proportion and the experimental results show that when the weight is 0.5, MAE is minimum, which is also the most appropriate weight ratio value. Experimental results are shown in Fig. 1.

From the portion of the MovieLens database selected for use in the experiments, we used a portion as training data and the remainder as test data. We adopted the control variable method: Let neighbor user set k be stable. Then, we analyzed the proposed algorithm's performance by comparing situations when data sparsity changed. With the MovieLens database, the proposed preference-feature based hybrid CF algorithm showed different feature to different features. Let

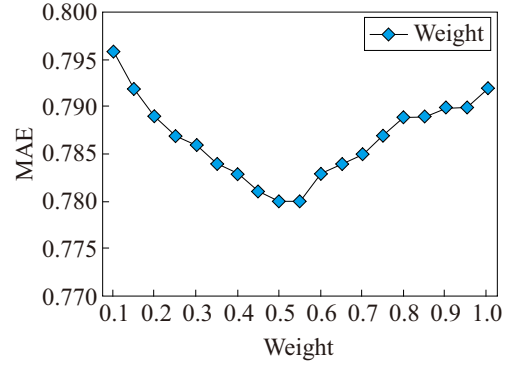


Fig. 1 The selection of weight.

neighbor user set k be stable and change the training set. Table 3 shows the experimental results for each training set when the neighbor set $k = 30$. Line graphs of the results are shown in Fig. 2.

In Fig. 2, UPIF represents the preference-feature CF algorithm. IF is the item feature CF algorithm. BI is the traditional item-based CF algorithm. The MAE of the preference-feature CF algorithm is less than both the IF and BI algorithms, which means that the UPIF algorithm's recommendation performance is significantly greatly better than both the BI and IF algorithms.

From the portion of the MovieLens database selected, we used a portion as training data. As before, we adopted the control variable method: Let data sparsity

Table 3 The proportion of different training sets results.

The proportion of training set	BI	IF	UPIF
0.1	0.8536	0.8400	0.804
0.2	0.8422	0.8280	0.802
0.3	0.8400	0.8200	0.797
0.4	0.8410	0.8050	0.8050
0.5	0.8370	0.8000	0.7773
0.6	0.8300	0.7860	0.7730
0.7	0.8200	0.7800	0.7700

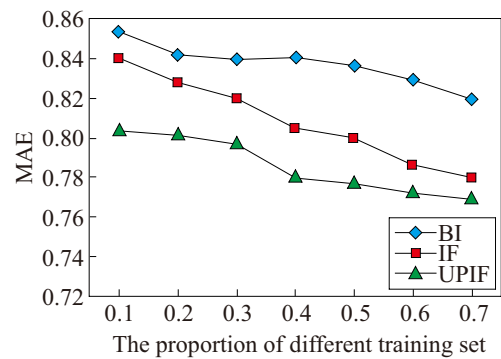


Fig. 2 The proportion of different training sets results

be stable and change the neighbor user set. Then, we analyzed the algorithm performance by comparing the results with the experimental results. To ensure the equity and avoid occasionality, we used the same division method and divided the MovieLens database into five sets and chose the average value of the results from these five sets as the final result. Table 4 shows the results for different proportions of the training set. A line graph of these results is shown in Fig. 3.

When neighbor user $k = 30$, the MAE for the three algorithms all achieved the minimum value and showed the highest efficiency. When k took other values, the MAE of algorithm is much smaller than UPIF for both the IF and BI algorithms. This also indicates that the UPIF algorithm's recommendation performance is significantly better than that of the BI and IF CF algorithms.

5 Discussion

The user preferences-item feature collaborative filtering algorithm proposed in this paper is more accurate compared with previous traditional methods. This conclusion is supported by the following evidence. The current data acquisition model fails to obtain all the

information. It is only able to reflect a particular feature of the user or item. The artificial division of item feature also increased the deviation significantly. Thus, the main purpose of the newly proposed algorithm is to reveal the hidden relationship between the user model and the item model. It is evident that detailing of information would greatly improve recommendation accuracy. Detailing of information will be achieved by extracting the potential semantics of information. To some extent, data sparsity can be solved by detailing the information, and detailing the information has become a new research direction to address the challenging puzzles inherent in the development recommendation systems.

Acknowledgements

This work was supported in part by the National High-Tech Research and Development (863) Program of China (No. 2011AA010101), the National Natural Science Foundation of China (Nos. 61103197 and 61073009), the Science and Technology Key Project of Jilin Province (No. 2011ZDGG007), the Youth Foundation of Jilin Province of China (No. 201101035), and the Fundamental Research Funds for the Central Universities of China (No. 200903179).

References

- [1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, vol. 40, no. 3, pp.169-174, 1997.
- [2] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, Probabilistic memory-based collaborative filtering, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no.1, pp. 56-69, 2004.
- [3] G. Linden, B. Smith, and J. York, Amazon.com recommendations: Item-to-item collaborative filtering, *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, 2003.
- [4] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, Recommender systems for large-scale E-commerce: Scalable neighborhood formation using clustering? in *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT'02)*, 2002, pp. 1-7.
- [5] G.-R. Xue, C. Lin, Q. Yang, W. S. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in *Proceedings of the ACM SIGIR Conference*, Salvador, Brazil, 2005, pp. 114-121.
- [6] S. K. Jones, A statistical interpretation of term specificity and its applications in retrieval, *Journal of Documentation*, vol. 28, no. 1, pp. 11-21, 1972.

Table 4 The results of different numbers of the Neighbors.

Number of neighbors	BI	IF	UPIF
0	0.8756	0.8720	0.8370
10	0.8342	0.8301	0.8060
20	0.8220	0.8220	0.7990
30	0.8010	0.8050	0.7930
40	0.7960	0.8040	0.7900
50	0.7940	0.8030	0.7850
60	0.7920	0.8030	0.7830
70	0.7910	0.8010	0.7820
80	0.7900	0.8000	0.7810

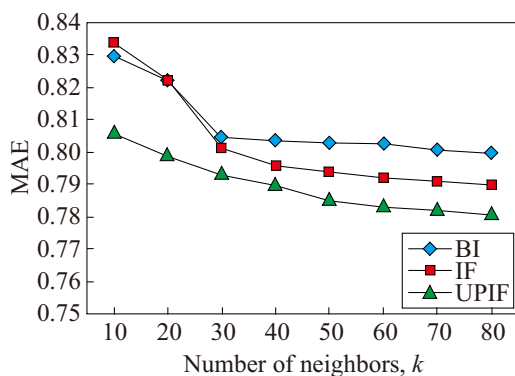


Fig. 3 The results of different numbers of the Neighbors.

- [7] P. Resnick and H. R. Varian, Recommender systems, *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.
- [8] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, Scalable collaborative filtering using cluster-based smoothing, in *Proc. the ACM SIGIR Conference*, Minneapolis, MN, USA, 2000, pp. 158-167.
- [9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, An algorithmic framework for performing collaborative filtering, in *the Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, USA, 1999, pp. 175-186.
- [10] X. Su, T. M. Khoshgoftaar, X. Zhu, and R. Greiner, Imputation-boosted collaborative filtering using machine learning classifiers, in *the 23rd Annual ACM Symposium on Applied Computing (SAC'08)*, Ceara Fortaleza, Brazil, 2008, pp. 175-186.
- [11] A. Nakamura and N. Abe, Collaborative filtering using weighted majority prediction algorithms, in *Proc. the 15th International Conference on Machine Learning (ICML'98)*, Madison, WI, USA, 1998, pp. 395-403.
- [12] X. Su and T. M. Khoshgoftaar, Collaborative filtering for multi-class data using belief nets algorithms, in *Proc. the 18th International Conference on Tools with Artificial Intelligence (ICTAI'06)*, Arlington, VA, USA, 2006, pp. 497-504.
- [13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, Grouplens: An open architecture for collaborative filtering of netnews, in *Proc. the ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 1994, pp. 175-186.
- [14] G. Karypis, Evaluation of item-based top-N recommendation algorithms, in *Proc. the International Conference on Information and Knowledge Management (CIKM'01)*, Atlanta, GA, USA, 2001, pp. 247-254.

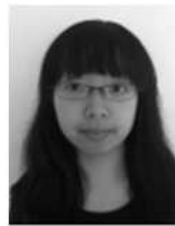


Liang Hu received his MEng and PhD degrees in computer science from Jilin University in 1993 and 1999. Currently, he is a professor and doctoral supervisor at the College of Computer Science and Technology, Jilin University, China. His research areas are network security and distributed computing, including theories,

models, and algorithms of PKI/IBE, IDS/IPS, and grid computing. He is a member of the China Computer Federation.



Guohang Song is a PhD candidate of College of Computer Science and Technology, Jilin University, China. His current research areas are Internet of Things information processing and semantic analysis of recommendation system.



Zhenzhen Xie is a master candidate of College of Computer Science and Technology, Jilin University, China. Her current research areas are digital forensic technology for cloud computing environments and behavior modeling for digital forensics.



Kuo Zhao received his BS degree in computer software in 2001 from Jilin University, followed by his MEng degree in computer architecture in 2004 and PhD degree in computer software and theory from the same university in 2008. He is currently an associate professor in the College of Computer Science and Technology, Jilin University. His research interests are in operating systems, computer networks, and information security.