

An improved algorithm for personalized recommendation on MOOCs

Yuqin Wang, Bing Liang and Wen Ji

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Shiwei Wang

Weihai Yuanhang Technology Development Co. Ltd, Shandong, China, and

Yiqiang Chen

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Received 22 August 2017
Accepted 9 September 2017

Abstract

Purpose – In the past few years, millions of people started to acquire knowledge from the Massive Open Online Courses (MOOCs). MOOCs contain massive video courses produced by instructors, and learners all over the world can get access to these courses via the internet. However, faced with massive courses, learners often waste much time finding courses they like. This paper aims to explore the problem that how to make accurate personalized recommendations for MOOC users.

Design/methodology/approach – This paper proposes a multi-attribute weight algorithm based on collaborative filtering (CF) to select a recommendation set of courses for target MOOC users.

Findings – The recall of the proposed algorithm in this paper is higher than both the traditional CF and a CF-based algorithm – uncertain neighbors' collaborative filtering recommendation algorithm. The higher the recall is, the more accurate the recommendation result is.

Originality/value – This paper reflects the target users' preferences for the first time by calculating separately the weight of the attributes and the weight of attribute values of the courses.

Keywords MOOC, Collaborative filtering algorithm, Multi-attribute weight algorithm, Personalized recommendation

Paper type Research paper

1. Introduction

1.1 Background and related work

Recent years have witnessed the rapid development of computer and the internet, changing many aspects in people's daily lives such as the way of receiving education. People all over the world can get access to massive quality courses through the internet. Massive online courses are produced by instructors who are professional in specific fields. These online



courses combine the picture, voice and flash, which greatly arouse the learners' interests (Al-Atabi and Deboer, 2014). This outstanding distance education method is called Massive Open Online Courses (MOOCs). MOOCs have benefited tens of millions of students all over the world. However, when learners are faced with those massive courses, they often need to spend much time looking for the course they really like. It also happens in other similar situations in the age when resources are increasing exponentially. Thus, the personalized recommendation has become a research point both in academic field and in industrial field.

The recommendation technology is first proposed in the early days of e-commerce. The early recommendation system treats all users the same, and it recommends the same resources to every user. It is obvious that the early recommendation system has low efficiency because it ignores the differences between users. As time went by, the personalized recommendation technology emerges to treat every user as a specific person and recommend resources to users according to the user's preferences. Nowadays, there exist three major recommendation approaches: collaborative filtering (CF) (Felfernig *et al.*, 2014; Bobadilla *et al.*, 2012; Jeong *et al.*, 2010), content-based filtering (Pazzani and Billsus, 1997; Adomavicius and Tuzhilin, 2013; Serrano-Guerrero *et al.*, 2011) and knowledge-based recommendation (Felfernig *et al.*, 2006). In this paper, we focus on improving the CF because it is simple and obtains better recommendation results than the other recommendation algorithms (De Campos *et al.*, 2010; Herlocker *et al.*, 1999). The CF has two main problems: one is the sparse matrix problem (Papagelis *et al.*, 2005) and the other is the cold start problem (Herlocker *et al.*, 2004).

In the academic field, many scholars have studied the CF technology and have made a great progress. Sarwar *et al.* (2002) propose SVD (singular value decomposition) method to release the influence of sparse matrix problem of the CF. Ortega *et al.* (2013) use Pareto dominance to eliminate less representative users by pre-filtering process while retaining the most promising neighbors. Huang *et al.* (2010) propose a CF-based prediction algorithm named uncertain neighbors' collaborative filtering recommendation algorithm (UNCF) that selects a trustworthy subset from the original neighbor set. Guo and Deng (2007) propose a new personal technology of recommendation based on traditional CF algorithm to improve the e-commerce recommender system. Zhang *et al.* (2009) present an approach to compute similarity between genres as similarity to lower sparsity of the user-item score matrix. Feng *et al.* (2004) also try to lower sparsity of the user-item score matrix by using a new revised conditional probability expression.

In the industry field, the CF technology is first used in Grouplens system which aims to recommend users news which may be satisfied with the users' preference (Konstan *et al.*, 1997). In the early twentieth century, CF is applied on e-commerce systems such as Amazon, Ebay, Taobao and so on. The CF helps these e-commerce systems obtain great success. Apart from being used in e-commerce, the CF technology is applied on Video Recommender which is a movie recommendation system developed by Bell Core. CF is also applied on the music community such as Last.fm.

1.2 Summary of content and contributions

Although some aspects of the CF have improved by many scholars, the CF still can be improved in other aspects. In this paper, we combine the CF's basic idea with the attributes of MOOC videos' structure's feature to propose a new algorithm of personalized recommendation. The results of this paper can be used in other fields as well.

The rest of this paper is organized as follows. First, we introduce the basic idea of the CF technology and the process of the CF technology. We analyze the process of the CF and find its advantages and shortcomings which have not been resolved completely. Second, we

propose a multi-attribute weight algorithm (MAWA) using the attribute weight and attribute value weight to get accurate users' preferences. At last, we conduct an experiment to verify the MAWA's recommendation result is more accurate than the traditional CF and the UNCF proposed in [Huang et al. \(2010\)](#).

Our main contributions include two aspects:

- (1) using attribute weight and attribute value weight to reflect the users' preferences in both coarse granularity and fine granularity; and
- (2) making up for the two main shortcomings of CF, whereas many other improved approaches solve only one of them.

2. Collaborative filtering technology

In this part, we will introduce CF algorithm, describe the procedure of CF algorithm and analyze the advantages and shortcomings of CF algorithm.

2.1 Introduce to collaborative filtering technology

CF algorithm was proposed by scholars in 1990s to make up for the shortcomings of the content filtering algorithm. Because CF algorithm appeared, CF technology has been a research hotspot because of its solid theoretical basis. The CF technology supposes that the users' preferences will not change over time. Typically, CF can be divided into user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). The difference between them lies in that the UBCF tries to find the collections of similar users and recommend the historical resources of the target user's similar users, whereas the IBCF tries to find the collections of similar resources and recommend the resources which are similar to the target user's resource history. The CF technology can find out the new resources which are similar to the target user's resource history. It means that CF technology can detect the potential resource that the target user may be interested in.

2.1.1 Sub-subheadings. Typeset sub-subheadings in medium face italic and capitalize the first letter of the first word only. Section numbers to be in roman.

2.2 Similarity measurement

Because both user-based collaborating filtering and item-based collaborating filtering need to measure the similarity, we will introduce the three most popular similarity measure methods below. Take the user-based collaborating filtering for example. The user-item score matrix R is an $m \times n$ matrix, which means there are m users and n items. R_{uc} means that the user u gives the item c a score R_{uc} .

2.2.1 Cosine similarity. The user's scores of the items are considered as an n -dimensional vector, and the cosine angle between the users' score vectors represents the similarity between the users. The cosine similarity formula is as follows:

$$\text{sim}(u_i, u_j) = \frac{\sum_{c=1}^n R_{i,c} \cdot R_{j,c}}{\sqrt{\sum_{c=1}^n R_{i,c}^2} \sqrt{\sum_{c=1}^n R_{j,c}^2}} \quad (1)$$

In [equation \(1\)](#), $\text{sim}(u_i, u_j)$ represents the similarity between user u_i and user u_j .

2.2.2 Modified cosine similarity. The cosine similarity has a shortcoming that the user may not score the item, which will result in errors in the results. To make up for this, modified cosine similarity is proposed. The user u_i and u_j have scored two item sets I_i and I_j ,

respectively. The items in both I_i and I_j form a set I_{ij} . The modified cosine similarity formula is as follows:

$$\text{sim}(u_i, u_j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i) (R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - \bar{R}_i)^2} \sqrt{\sum_{c \in I_j} (R_{j,c} - \bar{R}_j)^2}} \quad (2)$$

In equation (2), \bar{R}_i means u_i 's average score of items which have been scored by user u_i .

2.2.3 *Pearson correlation coefficient.* The set I_{ij} contains items that both u_i and u_j have scored. The Pearson's correlation coefficient formula is as follows:

$$\text{sim}(u_i, u_j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i) (R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - \bar{R}_i)^2} \sqrt{\sum_{c \in I_j} (R_{j,c} - \bar{R}_j)^2}} \quad (3)$$

2.3 Process of user-based collaborative filtering algorithm

The main process of UBCF algorithm can be divided into three stages.

2.3.1 *Establish the user model.* Use $m \times n$ matrix R to represent the user-item score matrix, containing m users and n items. The element of matrix is the score that the user has given to the item, ranging from 0 to 10. When the element is 0, it means that the user has not used the item or has not given a score to the item.

2.3.2 *K nearest neighbor query.* First, use the similarity measure methods to measure the similarities between target user and other users. Second, select k users with the highest k similarity degrees to be the neighbors of the target user. Finally, form a neighbor set with the k selected users in non-incremental order of the similarity degrees.

2.3.3 *Generate recommendations.* Use the neighbor set in Step 2 and the formula below to predict the user's score for each item. Recommend N items with top N predicted scores. The predicted scores are calculated as follows:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{n \in N} \text{sim}(u, n) (R_{n,i} - \bar{r}_n)}{\sum_{n \in N} \text{sim}(u, n)} \quad (4)$$

In equation (4), \bar{r}_n means u 's average score for all items and N_u is u 's neighbor set.

2.4 The advantages and shortcomings of collaborative filtering

Both the UBCF and the IBCF have achieved great success in many fields in practice. Based on the theory and process of CF introduced above, the advantages and shortcomings of CF are summarized as follows:

2.4.1 The advantages of collaborative filtering

- CF is easy. As we can see from this paper, CF only needs a user-item score matrix and conducts simple calculation based on this matrix, so the CF is very easy to be applied in practice.

- CF can be applied on many different fields. The CF can be applied to both structured and unstructured resources, so the CF can be applied widely.
- CF can detect users' new interest points. As mentioned above, CF can recommend the new resources to target users to help users detect their new interest points.

2.4.2 The shortcomings of collaborative filtering

- *Cold start problem*: The CF generates recommendation set according to the resources that the users have used and scored. So only if one resource has been used and scored by at least one user, it can be recommended to other users. However, when there is a new resource, it has not been scored by any users, so it has no chance to be enrolled in the recommendation set.
- *Sparse matrix problem*: As mentioned above, the CF calculates on user-item score matrix. The matrix is sparse because the average amount of items that one user has used is less than 1 per cent of the total and many users do not score the items initiatively. So the similarity measurements between users are not accurate and the neighbor set is not reliable, resulting in low recommendation efficiency.

3. Multi-attribute weight algorithm

In Section 2, we introduce CF technology and analyze its shortcomings. To overcome these shortcomings, we propose MAWA.

3.1 Introduce to multi-attribute weight algorithm

Suppose that an item can be described by some constant attributes and the target user likes one item because it has some attributes that the user likes. Based on these two assumptions, we recommend some resources to the target user according to the attributes. Based on CF technology and the assumption, we propose a MAWA which uses attribute weight to represent the user's preference on attribute and attribute value weight to represent the user's preference on attribute value under an attribute.

The MAWA calculates on the user-attribute value matrix R . The $M \times N$ matrix R means that there are M users and N attribute values for an attribute. The element of R is the number that the user has visited the attribute value. If the item can be described by T constant attributes, there will be T matrices in total.

The MAWA makes up for the shortcomings of CF. With regard to CF's cold start problem which means the new resources have no chance to be recommended, the MAWA uses the attribute value instead of item in basic matrices. As the amount of the old resources is big enough, the new resource contains at least one common attribute values with the old resources. Because the MAWA recommends a resource in accordance with the resource's attribute value instead of the entire resource, the new resources containing the old attribute values have the chance to be recommended. Thus, the MAWA does not have the cold start problem. As to the sparse matrix problem, because the number of attributes and the amount of attribute values are both constant and they are far less than the amount of the items, the dimension of the MAWA's basic matrix is far fewer than the dimension of the CF's basic matrix. Under the same data set, the MAWA's basic matrix with fewer dimension is far denser than the CF's basic matrix. With the amount of the resources increasing, the MAWA's basic matrix gets denser, whereas the CF's basic matrix gets sparser. Thus, the MAWA doesn't have the sparse matrix problem.

The MAWA calculates on the user-attribute value matrices to find every attribute value's neighbor attribute values by the means of CF and recommend the resources which has the target user's favorite attribute and attribute value or their neighbors.

3.1.1 *Process of multi-attribute weight algorithm.* The main process of MAWA is as follows:

- Calculate the similarity between attribute values.
- If the item can be described by T constant attributes, then there will be T user-attributes value matrices R_1, R_2, \dots, R_S to R_T . For each user-attribute value matrix R_S , use the cosine similarity measurement to calculate the attribute values' similarity with other attribute values of the attribute S.

$$sim(S_a, S_b) = \frac{\sum_{u=1}^n R_{u,S_a} \cdot R_{u,S_b}}{\sqrt{\sum_{u=1}^n R_{u,S_a}^2} \sqrt{\sum_{u=1}^n R_{u,S_b}^2}} \quad (5)$$

In equation (5), $sim(S_a, S_b)$ represents the similarity between attribute value S_a and S_b . R_{u,S_a} means that the user U has visited the resources with S_a for R_{u,S_a} times.

- Find the neighbor set of attribute value.
- Select K attribute values with top k similarity degrees to be the neighbors and form a neighbor set with the k selected items in non-incremental order of the similarity degrees.
- Get the recommended attribute value set.
- For every attribute value, predict the times that the target user uses it. Select L recommended attribute values with top L prediction values to form a recommended attributed value set. The prediction formula is as follows:

$$P_{u,S_a} = \overline{R_{S_a}} + \frac{\sum_{S_b \in C} sim(S_a, S_b) (R_{u,S_b} - \overline{R_{S_b}})}{\sum_{S_b \in C} sim(S_a, S_b)} \quad (6)$$

In equation (6), P_{u,S_a} represents the predicted amount that user U uses attribute value S_a . $\overline{R_{S_a}}$ is the average amount that all users use attribute value S_a . The C means the neighbor set of attribute value S_a .

- Calculate the weight of attribute values.
- For each attribute value, calculate its weight for target user U . If the attribute value is not included in recommended attribute value set of target user U , its weight is 0. The weight formula of attribute values is defined as follows:

$$W_{u,S_a} = \begin{cases} \frac{P_{u,S_a}}{\sum_{S_i \in L_S} (P_{u,S_i})} & S_a \in L_S \\ 0 & S_a \notin L_S \end{cases} \quad (7)$$

In equation (7), $W_{(u,S_a)}$ represents the weight of attribute value S_a for user U . L_S is the recommended attribute value set of attribute S :

- Calculate the weight of attributes.
- For target user U , calculate the standard deviation for each attribute to represent the interest distributions. The bigger the standard deviation is, the more concentrated the target user's interests are. The weight formula of attributes is as follows:

$$Q_{u,S} = \sqrt{\frac{1}{n_S} \sum_{k=1}^{n_S} (R_{u,S_k} - \overline{R_{u,S}})^2} \quad (8)$$

In equation (8), $Q_{u,S}$ represents the standard deviation of attribute S for target user U . The $\overline{R_{u,S}}$ is user U 's average use times for attribute S . And n_S is the number of attribute values of attribute S .

- Generate the recommendation set.
- For target user U , calculate the value of all attributes for each resource. Select the N recommended resources with highest N recommendation values to form a recommendation set. Recommend the resources in recommendation set to target user. The recommendation value formula is as follows:

$$V_{u,r} = \sum_{S=1}^T W_{u,S} * \left(\frac{Q_{u,S}}{M_S} \right) \quad (9)$$

In equation (9), the $V_{u,r}$ represents the resourcer's recommendation value for user U . The M_S is a controlling coefficient that can help modify the attribute weight to achieve a better recommendation efficiency.

3.2 Complexity analysis

The traditional item-based CF's time complexity and space complexity are $O(N^2N)$, in which N means the amount of resources. The MAWA's time complexity is $O(N^2M)$, in which N^2 means the largest amount of attribute values among T attributes and M means the number of users. The analysis of MAWA's time complexity is as follows. In Step 1, MAWA calculates the similarities between attribute values for each attribute, so the time complexity in Step 1 is $O(N^2N)$. In Step 2, MAWA sort the similarities for each attribute, so the time complexity is $O(N \log N^2)$. In Step 3 and Step 4, MAWA calculates the prediction value and calculates the weight of every attribute value for every user, thus the time complexities in Step 3 and Step 4 are $O(N^2M)$. In Step 5, calculate the standard deviation for every user, thus the time complexity in Step 5 is $O(N^2+M)$. The time complexity of Step 6 is $O(T)$, in which T means the number of attributes and T is constant. As the number of users (M) gets larger and larger and N^2 is constant, the time complexity of MAWA is $O(N^2M)$. The space complexity of MAWA is $O(N^2N)$, which is far smaller than that of the item-based CF.

4. Experimental results and discussion

In Section 2 and Section 3, we respectively introduce the CF technology and MAWA. In this part, we compare the efficiencies of the two algorithms using true data set from MOOC College.

4.1 Data set and evaluating indicator

In this paper, we use the web crawler to get the true data set from MOOC College. In the data set we use contains 978 MOOC videos, 30 users and 2026 score records. The scarcity of user-item score matrix is 6.9 per cent. We select 4 constant attributes including language, platform, school and video score to represent the MOOC videos. In more detail, the language contains Chinese and English. The videos are originally from 18 different platforms and 73 different schools. The scores vary from 0 to 10.

The recall in the field of information retrieval can be used to evaluate the effect of the CF, the UNCF and the MAWA. The higher the recall is, the more accurate the recommendation results are. First, division coefficient “a” is utilized to divide the raw data into two parts, one of which is the test set, whereas the other one is the training set. For example, ‘a’ ranges from 0.1 to 0.9 by 0.1. The 2026* ‘a’ resources are regarded as the training set and the rest are test set. Second, use the CF, the UNCF and the MAWA respectively to generate the recommendation set. Finally, use [equation \(10\)](#) to calculate the recall rate:

$$recall = (recommendation\ set \cap testing) / |testing| \quad (10)$$

4.2 Experimental environment

First, we use python 2.7.12 to write the web crawler to get data set from the MOOC College and the data downloaded from the web is stored in mysql 5.7.17. Second, we set the controlling coefficients to be 5 5 2 and 10 for the four attributes. Then we use the MATLAB 8.3.0.532 to generate the recommendation sets using the CF, the UNCF and the MAWA and calculate the recalls for nine times with varying ‘a’. As for the coefficients of UNCF, we choose the coefficients that achieve the best recommendation effectiveness in [22]. Finally, we get a curve of recalls and an average recall.

4.3 Experimental results and discussion

The experimental results are shown in [Table I](#) and [Figure 1](#):

The average recalls of the CF, the UNCF and the MAWA are:

$$\overline{Recall}_{CF} = 0.3469$$

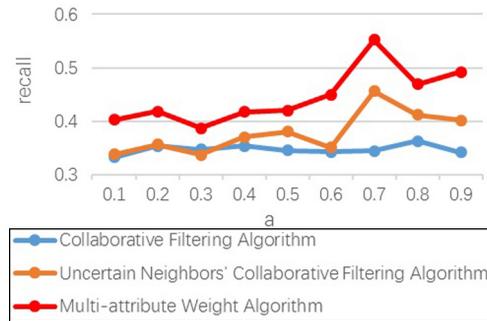
$$\overline{Recall}_{UNCF} = 0.3778$$

$$\overline{Recall}_{MAWA} = 0.4453$$

a	CF algorithm	UNCF algorithm	MAWA
0.1	0.3319	0.3374	0.4026
0.2	0.3537	0.3562	0.4185
0.3	0.3470	0.3364	0.3865
0.4	0.3531	0.3700	0.4173
0.5	0.3455	0.3800	0.4205
0.6	0.3420	0.3506	0.4494
0.7	0.3443	0.4558	0.5519
0.8	0.3630	0.4120	0.4691
0.9	0.3416	0.4015	0.4921

Table I.
Table of three algorithms' recalls

Figure 1.
Curve of three
algorithms' recalls



In [Figure 1](#), it is obvious that the MAWA's recall is higher than the CF and the UNCF. The average recalls show precisely that the MAWA increases the recalls of the CF and the UNCF, respectively, by 28.3 and 17.9 per cent. Because the MAWA and the CF are calculated on the same data set, the result proves that the MAWA can release the influence of the sparse matrix problem and improve the accuracy of recommendation.

5. Conclusion and future work

In this paper, we propose the MAWA to make up for the shortcomings of the CF algorithm. The MAWA uses attribute weight and attribute value weight to calculate the resources' recommendation value. Compared with the CF, the MAWA uses matrices with far less dimensions, so the MAWA does not have sparse matrix problem. We take the popular MOOC videos, for example, and conduct an experiment on the true data set and the result shows that the MAWA has higher efficiency than the CF and the UNCF.

It should be emphasized that the MAWA uses attribute weight and attribute value weight respectively, which can reflect the users' preferences in the coarse granularity and fine granularity.

In the future, we will focus on adding the evolutionary mechanism to make the MAWA suitable for users' changing preferences.

References

- Adomavicius, G. and Tuzhilin, A. (2013), "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", *Multimedia Services in Intelligent Environments*, Springer International Publishing, pp. 734-749.
- Al-Atabi, M. and Deboer, J. (2014), "Teaching entrepreneurship using massive open online course (moooc)", *Technovation*, Vol. 34 No. 4, pp. 261-264.
- Bobadilla, J., Ortega, F. and Hernando, A. (2012), "A collaborative filtering similarity measure based on singularities", *Information Processing & Management*, Vol. 48 No. 2, pp. 204-217.
- De Campos, L.M., Fernández-Luna, J.M., Huete, J.F. and Rueda-Morales, M.A. (2010), "Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks", *International Journal of Approximate Reasoning*, Vol. 51 No. 7, pp. 785-799.
- Felfernig, A., Friedrich, G., Jannach, D. and Zanker, M. (2006), "An integrated environment for the development of knowledge-based recommender applications", *International Journal of Electronic Commerce*, Vol. 11 No. 2, pp. 11-34.

- Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S. and Stettinger, M. (2014), "Basic approaches in recommendation systems", *Recommendation Systems in Software Engineering*, Springer, Berlin, pp. 15-37.
- Feng, Z.J., Xian, T.A.N.G.A. and Feng, G.J. (2004), "An optimized collaborative filtering recommendation algorithm", *Journal of Computer Research and Development*, Vol. 41, pp. 1842-1847.
- Guo, Y. and Deng, G. (2007), "An Improved Personalized Collaborative Filtering Algorithm in E-Commerce Recommender System" *International Conference on Service Systems and Service Management, IEEE*, pp. 1582-1586.
- Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J. (1999), "An algorithmic framework for performing collaborative filtering" *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA*, ACM, pp. 230-237.
- Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004), "Evaluating collaborative filtering recommender systems", *ACM Transactions on Information Systems (Systems)*, Vol. 22 No. 1, pp. 5-53.
- Huang, C.G., Yin, J., Wang, J., Liu, Y.B. and Wang, J.H. (2010), "Uncertain neighbors' collaborative filtering recommendation algorithm", *Chinese Journal of Computers*, Vol. 33 No. 8, pp. 1369-1377.
- Jeong, B., Lee, J. and Cho, H. (2010), "Improving memory-based collaborative filtering via similarity updating and prediction modulation", *Information Sciences*, Vol. 180 No. 5, pp. 602-612.
- Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedl, J. (1997), "GroupLens: applying collaborative filtering to Usenet news", *Communications of the Acm*, Vol. 40 No. 3, pp. 77-87.
- Ortega, F., Sánchez, J.L., Bobadilla, J. and Gutiérrez, A. (2013), "Improving collaborative filtering-based recommender systems results using Pareto dominance", *Information Sciences*, Vol. 239, pp. 50-61.
- Papagelis, M., Plexousakis, D. and Kutsuras, T. (2005), "Alleviating the sparsity problem of collaborative filtering using trust inferences", *Trust Management*, pp. 125-140.
- Pazzani, M. and Billsus, D. (1997), "Learning and revising user profiles: the identification of interesting web sites", *Machine Learning*, Vol. 27, pp. 313-331.
- Sarwar, B.M., Karypis, G., Konstan, J. and Riedl, J. (2002), "In-cremental SVD-based algorithms for highly scalable recommender systems", *Proceeding of the 5th International Conference on Computer and Information Technology, Dhaka*.
- Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J.A., Cerezo, A. and Romero, F.P. (2011), "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0", *Information Sciences*, Vol. 181 No. 9, pp. 1503-1516.
- Zhang, J., Lin, Z., Xiao, B. and Zhang, C. (2009), "An optimized item-based collaborative filtering recommendation algorithm", *IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC, IEEE*, pp. 414-418.

About the authors

Yuqin Wang is currently pursuing an MS degree at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. Her current research interests include big data, distributed machine learning and recommendation system.

Bing Liang is currently pursuing a PhD degree at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. His current research interests include data mining, machine learning and pervasive computing.

Wen Ji received the MS and PhD degrees in communication and information systems from Northwestern Polytechnical University, China, in 2003 and 2006, respectively. She is an Associate Professor in the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS),

Beijing, China. From 2014 to 2015, she was a Visiting Scholar at the Department of Electrical Engineering, Princeton University, USA. Her research areas include video communication and networking, video coding, channel coding, information theory, optimization, network economics and ubiquitous computing. Wen Ji is the corresponding author and can be contacted at: jiwen@ict.ac.cn

Shiwei Wang received the PhD degree in Control Systems from the Liverpool John Moores University in 2007. He is currently the R&D Director of Weihai Yuanhang Technology Development Co., Ltd., Weihai, China, as an awardee of the Thousand Talents Plan of China. His current research interests include artificial intelligent, machine vision, robotics and the corresponding application in food and pharmaceutical industry.

Yiqiang Chen received the BSc and MS degrees from the University of Xiangtan, Xiangtan, China, in 1996 and 1999, respectively, and the PhD degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2002. In 2004, he was a Visiting Scholar Researcher with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Hong Kong. He is currently a Professor and Director with the Pervasive Computing Research Center, ICT, CAS. His research interests include artificial intelligence, pervasive computing and human-computer interface.