TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/10 pp353-359
Volume 18, Number 4, August 2013

Community-Based User Domain Model Collaborative Recommendation Algorithm

Fulan Qian, Yanping Zhang*, Yuan Zhang, and Zhen Duan

Abstract: Collaborative Filtering (CF) is a commonly used technique in recommendation systems. It can promote items of interest to a target user from a large selection of available items. It is divided into two broad classes: memory-based algorithms and model-based algorithms. The latter requires some time to build a model but recommends online items quickly, while the former is time-consuming but does not require pre-building time. Considering the shortcomings of the two types of algorithms, we propose a novel Community-based User domain Collaborative Recommendation Algorithm (CUCRA). The idea comes from the fact that recommendations are usually made by users with similar preferences. The first step is to build a user-user social network based on users' preference data. The second step is to find communities with similar user preferences using a community detective algorithm. Finally, items are recommended to users by applying collaborative filtering on communities. Because we recommend items to users in communities instead of to an entire social network, the method has perfect online performance. Applying this method to a collaborative tagging system, experimental results show that the recommendation accuracy of CUCRA is relatively good, and the online time-complexity reduces to $O(n)$.

Key words: collaborative recommendation; tagging system; community-based; recommendation system

1 Introduction

The amount of Internet information has increased sharply over the years. How to effectively filter overload information and find useful data and interrelations is a hot issue in network data mining. The study is of great significance, especially when considering big data.

Recommendation systems are a type of effective tool used to solve the problem of information overload[1]. They can automatically recommend items to a target user according to observed user preference information, purchase behavior, evaluation behavior, and so on.

Recommendation systems have been successful in business applications. Amazon.com (http://www.amazon.com) uses a recommendation system to predict users' purchase behaviors, recommends goods of potential interest, and improves Amazon's commercial profit. Some websites use recommendation systems to increase customer satisfaction. For example, Movielens (http://movielens.umn.edu) is a movie recommendation website that can recommend movies of most interest to a target user according to the user's historical rating records.

The Collaborative Filtering (CF) approach is probably the most familiar, most widely implemented, and most mature of the recommendation approaches[2]. Its core concept is to utilize a collective intelligence to collect answers from crowd behavior

• Fulan Qian, Yanping Zhang, Yuan Zhang, and Zhen Duan are with the School of Computer Science and Technology, Anhui University, Hefei 230601, China. E-mail: zhangyp2@gmail.com.
• Yuan Zhang is also with the School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei 230601, China.
∗ To whom correspondence should be addressed.
Manuscript received: 2013-06-17; revised: 2013-06-27; accepted: 2013-06-27

and data. Breese et al.[3] introduced a classification of CF algorithms that divides them into two broad classes: memory-based algorithms and model-based algorithms. A memory-based algorithm, such as the user-based K-Nearest Neighbor (KNN)[4] algorithm utilizes an entire database of user preferences to compute recommendations. These algorithms tend to be simple to implement and require no training (offline) cost. But as the size of user and item sets increase, the online performance of memory-based algorithms tends to decrease. A model-based algorithm, for example, the Bayesian Network[5], SVD[6], pLSA[7], LDA[8], PMF[9, 10], and other modified algorithms, builds a model of the preference data and uses it to produce recommendations. Usually, the model-building process is time-consuming and only done periodically. The online performance of model-based algorithms is better than memory-based algorithms. However, many model-based algorithms lack interpretability; for example how and why to select user (item) vector dimensions. In addition, the primary shortcoming of model-based algorithms is that they need to regenerate to create a new model if users (items) change their behaviors.

According to the characteristics of both types of algorithms, researchers have tried various hybrids of the model- and memory-based approaches[11–13]. The basic idea is to improve the online memory-based recommendation efficiency through an increase in the offline time to construct a user domain model.

These approaches generally establish the relevant user domain by a model-based method in the offline state to exchange offline work for online recommendation efficiency of memory-based algorithms. In Ref. [12], the user domain model is built by bisecting k-means clustering algorithms on user preference data. Some data sets contain a user trust list such as the lists in Epinions (http://www.epinions.com). Every member of Epinions maintains a trust list which represents a network of trust relationships among users. In Ref. [14], the user domain model is built as a social network which is established based on user trust lists in a dataset. There are other more direct methods that use KNN[15] to calculate the top K similarities with a target user and use these to represent the user's domain. On one hand, these methods have a strong dependence on datasets; for example, they require a user trust list. Selecting the number of clusters can lead to a lack of universality in the KNN algorithm. On the other hand, these methods do not use historical data to mine additional knowledge of the user social behavior domain model.

In this paper, we propose a novel algorithm: the Community-based User domain model Collaborative Recommendation Algorithm (CUCRA). This algorithm maps a user-item data set to a user-user social network based only on user-item preference data. It then finds user similar preference communities to define a user domain model by detecting communities on a user-user social network. Finally it makes memory-based recommendations in the community-based user domain model. The algorithm does not depend on additional dataset information, and uses communities within a social network as user domain models that include detailed behavioral interpretability.

2 Method Description

2.1 Community-based user domain model generation method

Real-world social networks often exhibit a strong community structure[16]. A community is usually considered as a group of users whose members interact with others more frequently than with those outside the group. Users in the same communities are frequently connected, and thus probably have similar tastes and interests. In a social network, these communities often yield better recommendation results.

Based on the above motivation, we propose the CUCRA algorithm implemented in two parts: the first part is building the offline user domain model; the second part is recommending items to target users in the model.

The more important part is the former, that is, the community-based user domain model generation method. The generation method follows three steps: (1) Calculate user similarities using a user-item preference dataset; (2) map the similarity relationships to user-user social network relationships; (3) apply an existing community detection method to generate communities as user domains.

In the second step, we define a K-nearest neighbor graph $G_K = (V; E_K)$ as a user-user social network where V is the set of users, and E_K is the set of edges. According to the KNN strategy, for each user, compute the top-K largest similarity users, and establish edges between each two users. In cases where there are many edges between two users, only a single edge is kept. We map the user-item data relationship to the user-user network relationship through the above method. The intrinsic mechanism is mining user social behavior through user-item rating behavior. This mining process is described in Fig. 1.

Building the model consumes offline time; next, we analyze the time complexity of Algorithm 1. We calculate the similarity between any two users according to the rating of items in Phase 1, so the minimum time cost is $O(m^2n)$. We then determine the top-K largest similarity users of the target user in Phase 2. The essence of the algorithm is sorting, and the time-complexity of sort algorithms is $O(m\log n)$. There are various community detection algorithms and the most well-known and mature algorithm is the Gauss-Newton (GN) algorithm, the detection algorithm used in this version of CUCRA. Other algorithms include the Newman fast algorithm, PLA, and modifications of these, algorithms. The maximum time complexity

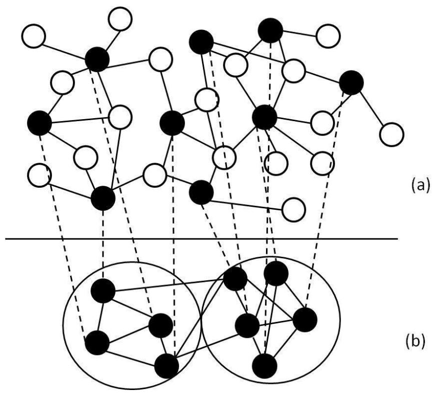

Fig. 1 (a) User-item relationship; (b) User-user social network. The black spots represent users and the white spots represent items. The figure shows how the user-item relationship is mapped to the user-user social network.

Algorithm 1 CUCRA Part 1: Community-based user domain model generation method (offline phase)

```
Input: $m$ users-$n$ items preference data set
Output: a set of communities
Phase 1: Calculate the similarities of users using user-item
         preference data set
Phase 2: Build a KNN social network
For $i \in V$
    Compute the top-$K$ largest similarity
    users Establish edges between each two users
End for
Phase 3: Apply an existing community detection method to
         generate communities.
```

of community detection algorithms is $O(m^3)$, such as GN, while minimum time complexity is $O(m^2)$ such as PLA. Because the whole time complexity of an algorithm is determined by the maximum value of every phase. Thus, time complexity in building a user domain model is $O(m^3)$

2.2 Produce recommendation

After building the user domain model, we recommend items of interest to a target user in a community. The formula to compute prediction is defined by Eq. (1).

$$\tilde{r}_{i\alpha} = \bar{r}_i + \frac{1}{\sum_v |S_{ij}|} \sum_{j\in \mathrm{Ucom}(i)} S_{ij}(r_{j\alpha} - \bar{r}_j) \qquad (1)$$

$\mathrm{Ucom}(i)$ denotes the set of users in the same community as user i, S_{ij} represents the similarity between user i and user j, and $r_{j\alpha}$ is the rating of user j for item α .

In the online phase, which is described by Algorithm 2, because the similarity relation has already been calculated for the target users, we only need to find common rating items within the community of m' users. Online time complexity is $O(m'n)$, and m' is far less than n , so online time complexity for a community-based algorithm is $O(m'n) \simeq O(n)$.

Traditional memory-based collaborative filtering algorithms directly calculate similarities between target users and all other m users. The algorithms have no offline time and only have online time. In order to calculate the similarity between the target user and another user, they first need to find common rating items between users. Thus calculating similarities between the target user and all m users requires community $O(mn)$ time complexity. Thus a community-based user domain algorithm has obvious advantages in online time complexity from the above analysis.

3 Experimental Analysis

We evaluate our algorithm on the MovieLens dataset (http://www.datatang.com/data/14772), a publicly available dataset containing both tagging and rating information. In order to strengthen the similarity accuracy, we use linear hybrid similarity metrics

Algorithm 2 CUCRA Part 2: Recommending items to target users in the communities (online phase)

```
Input: users $i$, items $\alpha$, the community that user $i$ belongs to
Output: the prediction of user $i$ rating item $\alpha$
Compute the prediction by the Eq. (1)
```

(rating-based cosine similarity and tag-based cosine similarity) in selecting similarity metrics in Phase 1 of CUCRA Part 1 (Algorithm 1).

We first prune the dataset for our analysis. For the tagging information, we only keep those tags which are added to at least three distinct movies. As for the users, we only keep those users who have at least 3 distinct tags in their tagging history. For movies, we only keep those movies that are annotated by at least 3 distinct tags.

We obtain two types of records after pruning, the tagging records and the rating records. Some statistics of these two types of records are presented in Table 1.

3.1 Similarity measure metrics

3.1.1 Rating-based cosine similarity

The rating-based cosine similarity of user i and user j is defined as follows.

$$\text{sim}_{ij}^{\text{rate}} = \frac{\sum_{\alpha \in U_{ij}} r_{i\alpha} r_{j\alpha}}{\sqrt{\sum_{\alpha \in U_{ij}} r_{i\alpha}^2 \sum_{\alpha \in U_{ij}} r_{j\alpha}^2}} \tag{2}$$

U_{ij} represents the set of items which are rated by both user i and user j.

3.1.2 Tag-based cosine similarity

Let $\boldsymbol{Z}$ be the tagging matrix, and each of its elements z_{ik} is the tf $\cdot$ idf value of user i and tag k[17]. z_{ik} can be computed by Eq. (3).

$$z_{ik} = \text{tf}(i,k) \cdot \log_2 \left(\frac{N}{\text{df}(k)} \right) \tag{3}$$

In Eq. (3) $\text{tf}(i,k)$ is the normalized frequency of appearance of tag k in user i's tagging history and $\text{df}(k)$ is the number of users who have used tag k. idf is the inverse document frequency measured by the log of the total number of users N divided by the number of users who have used tag k which is represented by $\text{df}(k)$.

The tag-based cosine similarity is defined as follows.

$$\text{sim}_{ij}^{\text{tag}} = \frac{\sum_{k \in T^{ij}} z_{ik} z_{jk}}{\sqrt{\sum_{k \in T^{ij}} z_{ik}^2} \sqrt{\sum_{k \in T^{ij}} z_{jk}^2}} \tag{4}$$

where T^{ij} denotes the index set of tags which are used by both user i and user j.

Table 1 Properties of the datasets.

Record	Property	Number of sets
Rating	Total users	805
	Total movies	9519
	Total ratings	360 346
Tagging	Total users	805
	Total movies	10 512
	Total tags	54 087

Then we use the linear hybrid of the two similarity metrics, as defined by Eq. (5).

$$S_{ij} = \lambda \cdot \text{sim}_{ij}^{\text{rate}} + (1-\lambda) \cdot \text{sim}_{ij}^{\text{tag}} \tag{5}$$

We choose the hybridization parameter $\lambda = 0.4$ because we obtain the minimum Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values near the $\lambda = 0.4$ by randomly dividing the dataset 10 times as shown in Fig. 2.

3.2 Evaluation metrics

We use two popular metrics, the MAE and the RMSE, to measure the prediction quality of our proposed approach in comparison with other collaborative filtering methods.

The metric MAE is defined as

$$\text{MAE} = \frac{\sum_{(i,\alpha) \in \Gamma} |r_{i\alpha} - \tilde{r}_{i\alpha}|}{|\Gamma|} \tag{6}$$

where Γ is the set of ratings in test data, $|\Gamma|$ is the size of Γ and $\tilde{r}_{i\alpha}$ is the predicted rating from user i to item

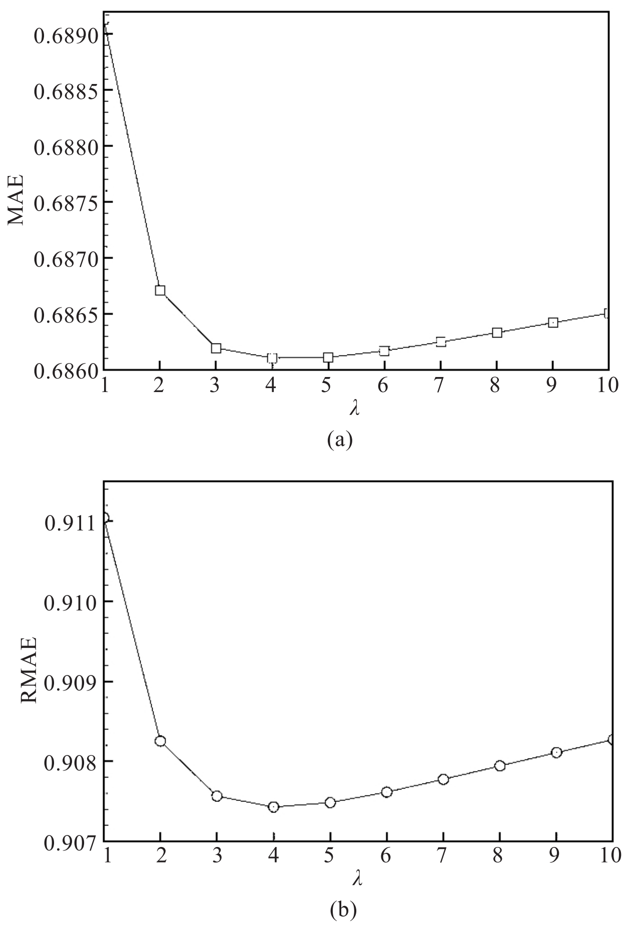

Fig. 2 The value of the hybridization parameter $\boldsymbol{\lambda}$. (a) The MAE with the different values of $\boldsymbol{\lambda}$, and (b) the RMSE with the different values of $\boldsymbol{\lambda}$. MAE and RMSE are minimized when $\boldsymbol{\lambda}$=0.4.

j.

Because the MAE in different communities has different values, we normalize the values using Eq. (7).

$$\text{MAE}_{\text{norm}} = \frac{|\text{Com}_i|}{|m|} \cdot \text{MAE}_i \tag{7}$$

In Eq. (7) $|m|$ is the number of all users, and $|\text{Com}_i|$ is the number of users in the i-th community. MAE_i is the MAE value in the i-th community.

The metric RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,\alpha)\in\Gamma}(r_{i\alpha} - \tilde{r}_{i\alpha})^2}{|\Gamma|}} \tag{8}$$

Likewise, we define the normalized RMSE.

$$\text{RMSE}_{\text{norm}} = \frac{|\text{Com}_i|}{|m|} \cdot \text{RMSE}_i \tag{9}$$

where RMSE_i is the RMSE value in the i-th community.

From their definitions, we can see that smaller MAE or RMSE or their normalized values indicate better performance.

3.3 Comparisons

In this section, in order to show the effectiveness of our proposed approach, we compare the proposed method to the following methods:

(1) KNN algorithm: this algorithm directly recommends items to a target user in the user's domain based on K most similar users in the user model.

(2) User-based collaborative filtering: this method directly recommends items to a target user in a set of all other users.

In our experiments, we randomly split the rating records into two parts. One part is 80% of ratings used as training data to predict the remaining 20% of ratings as test data. Training data and test data contain all users, and the items in the two datasets have no intersection.

To predict the rating in the training data of every community, and to make the comparison of actual ratings in the test data, we evaluate the recommendation accuracy using normalized MAE and RMSE.

We choose the parameter $K = 13$ for the dataset. By the experimental results it can be seen that when the neighbor number $K \geqslant 13$, the number and structure of communities which are obtained by the GN algorithm become steady. In contrast, the selection of K in the KNN algorithm often requires a great deal of experimental verification.

In essence, we first use the KNN method to cluster the users to build a social network, and then we use the GN algorithm to cluster the users again. Compared to single clustering, double clustering can further mine the intrinsic relationships among users. The relationships of people with similar preferences are extracted by hierarchical feature granulating. Such inherent relationships are relatively steady, and do not change with an increase in the number of neighbors. This allows better performance than the KNN algorithm (Table 2). The process is shown in Fig. 3.

Table 2 compares prediction qualities of ratings produced by the selected CF algorithms. From the experimental results, the proposed version of CUCRA exhibits good performance in recommendation accuracy with reduced online time complexity, although the user-based CF method exhibits best overall recommendation accuracy. CUCRA demonstrates an advantage in real-time response systems and can improve the speed of online recommendations, and future modifications to CUCRA may produce improved results.

Figure 4 displays the degree distributions of the user-user social network graph generated by the neighbors' number $K = 13$. The distribution follows the power-

Table 2 Comparison of rating-prediction quality of the selected CF algorithms (*K*=13).

CF algorithm	Time complexity (offline)	Time complexity (online)	MAE	RMSE
KNN	-	$O(mn)$	0.7296	0.9679
User-based CF	-	$O(mn)$	0.6861	0.9074
Community-based	$O(m^3)$	$O(n)$	0.6919	0.9602

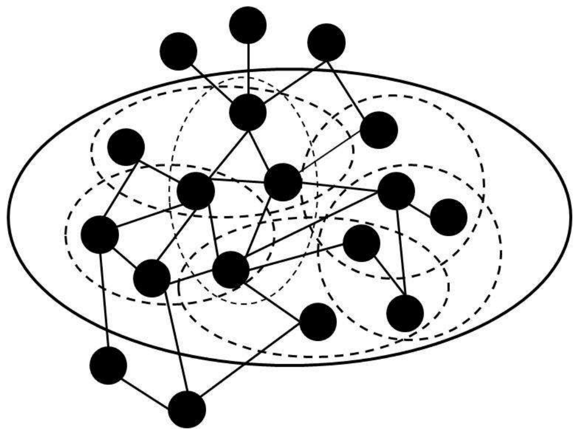

Fig. 3 The relationship between the KNN algorithm and CUCRA. KNN clustering is presented by the dashed lines, while GN clustering is presented by the solid lines. CUCRA produces twice the number of clusters, including both the KNN clusters and GN clusters.

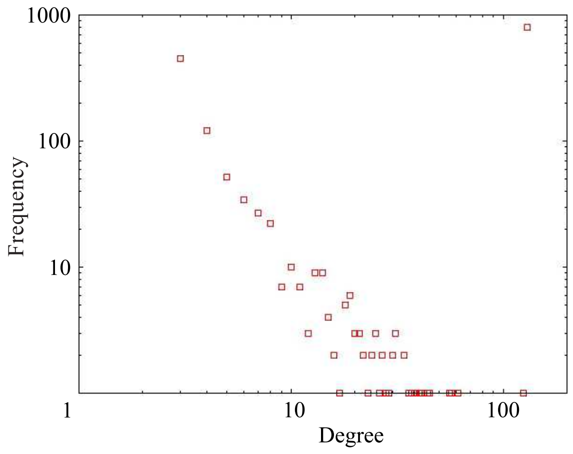

Fig. 4 The degree distribution of the user-user social network when *K*=13.

law, as is often observed in social networks[18]. Most of the nodes have small degrees, while there are also some nodes with very high degrees.

4 Conclusions

CUCRA transforms a user-item dataset into user-user social networks with the KNN method, and then mines further inner relationships between users using community detective algorithms. The nature of the transformation is a type of two-layer-cluster, with a better performance in extracting stable user domains of similar preferences in contrast to one-layer-cluster transformations.

CUCRA has the following advantages: (1) Based on community, the selection of a user domain has interpretability in user behavior. (2) The algorithm has strong real-time performance with low online time complexity. (3) Avoid the problems caused by many tuning parameters in the learning process commonly encountered in various model-based CF algorithms. (4) Preference similarity relations of users within the community are relatively stable so that the need for model regeneration is reduced. (5) It can directly use the related algorithms in social networks or other similar complex networks.

The method proposed in this paper transforms a user-item dataset into user-user social networks, applies a community detection algorithm on social networks to evaluate user behavior patterns, and provides recommendations tailored to users' interests. Many social networks are complex networks[19–21] so there are many related research studies of complex networks that may apply to recommendation techniques. We have only investigated use of the basic community detection algorithm GN in this paper, and analysis related to target selection and improvement in community detection methods may result in better performance of CUCRA.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (No. 61175046), the Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province (No. KJ2013A016), the Academic Innovative Research Projects of Anhui University Graduate Students (No. 10117700146), Youth Science Fund of Anhui University (No. KJQN1116).

References

[1] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, Recommender systems, *Physics Reports*, vol. 519, no. 1, pp.1-49, Oct. 2012.

[2] R. Burke, Hybrid recommender systems: Survey and experiments, *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, Nov. 2002.

[3] J. S. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc, 1998, pp. 43-52.

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: An open architecture for collaborative filtering of net news, in *CSCW 1994: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina, United States: ACM Press, 1994, pp. 175-186.

[5] L. Ungar and D. Foster, A formal statistical approach to collaborative filtering, in *Proceedings of the Conference of Automated Learning and Discovery*, Pittsburg, PA, USA, 1998.

[6] G. Takács, I. Pilászy, B. Németh, and D. Tikk, On the gravity recommendation system, in *Proceedings of KDD Cup Workshop at SIGKDD 2007,13th ACM International Conference on Knowledge Discovery and DataMining*, SanJose, CA, USA, 2007, pp. 22-30.

[7] T. Hofmann, Latent semantic models for collaborative filtering, *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 89-115, Jan. 2004.

[8] D. L. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet allocation, *The Journal of Machine Learning Research*, vol. 3, pp. 993-1022, Mar. 2003.

[9] R. Salakhutdinov and A. Mnih, Probabilistic matrix factorization, *Advances in Neural Information Processing Systems*, vol. 20, pp. 1257-1264, 2008.

[10] R. Salakhutdinov and A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in *Proceedings of the 25th International Conference on Machine Learning*, ACM, July 2008, pp. 880-887.

[11] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach, in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., June 2000, pp. 473-480.

[12] S. K. L. Al Mamunur Rashid, G. Karypis, and J. Riedl, ClustKNN: A highly scalable hybrid model-& memory-based CF algorithm, in *Proceeding of WebKDD*, Philadelphia, PA, USA, 2006.

[13] G. R. Xue, C. Lin, Q. Yang, W. Xi, H. J. Zeng, Y. Yu, and Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Aug. 2005, pp. 114-121.

[14] E. Viennet, Collaborative filtering in social networks: A community-based approach, in *Computing, Management and Telecommunications* (*ComManTel*), 2013 *International Conference on*, IEEE, Jan. 2013, pp. 128-133.

[15] R. Pan, P. Dolog, and G. Xu, KNN-based clustering for improving social recommender systems, in *Agents and Data Mining Interaction*, Springer, 2013, pp. 115-125.

[16] S. Fortunato, Community detection in graphs, *Physics Reports*, vol. 486, no. 3, pp. 75-174, 2010.

[17] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, Personalized recommendation in social tagging systems using hierarchical clustering, in *Proceedings of the 2nd ACM Conference on Recommender Systems*, New York, NY, USA, 2008, pp. 259-266.

[18] L. Tang and H. Liu, Community detection and mining in social media, in *Synthesis Lectures on Data Mining and Knowledge Discovery*, Morgan-Claypool, 2010.

[19] H. Ebel, L.-I. Mielsch, and S. Bornholdt, *Phys. Rev. E*, vol. 66, no. 3, pp. 035103, 2002.

[20] B. Albert-László and A. Réka, Emergence of scaling in random networks, *Science*, vol. 286, pp. 509-512, 1999.

[21] S. Redner, How popular is your paper? An empirical study of the citation distribution, *Eur. Phys. J. B*, vol. 4, no. 2, pp. 131-134, 1998.

Fulan Qian is currently a lecturer in Anhui University and a PhD candidate in the School of Computer Science and Technology at Anhui University. She received her MS degree from Anhui University in 2005. Her main research interests include quotient space theory and social network.

Yuan Zhang is currently a PhD candidate in Anhui University and a lecturer in the School of Electronic and Information Engineering at Anhui Jianzhu University. She received her MS degree from Anhui University in 2003. Her research interests are in the area of quotient space, subspace clustering algorithm, and computer network.

Yanping Zhang is currently a professor in Anhui University. She received her PhD from Anhui University in 2005. Her main research interests include computational intelligence, quotient space theory, artificial neural networks and intelligent information processing, and machine learning.

Zhen Duan is currently a lecturer in Anhui University. He received his PhD from Anhi University in 2010. His main research interests include machine learning and social network.