Appl. Math. Inf. Sci. 1, 1-10 (2013)

Applied Mathematics & Information Sciences
*An International Journal*



# DCFRS：A Distributed Collaborative Filtering Recommender System based on Cloud Computing for Mobile Commerce

**Yaming Zhang1 and Haiou Liu[2]**

[1]School of Economics and Management, Yanshan University, Qinhuangdao 066004 P. R. China
[2]School of Economics and Management, Yanshan University, Qinhuangdao 066004 P. R. China

*Email: haiou2009@163.com*

Received: Jan 14, 2012; Revised Mar 19, 2012; Accepted Aug. 19, 2012
Published online: 1 Oct 2012

**Abstract:** As the state-of-the-art method in recommender systems, collaborative filtering (CF) has proved to be one of the most successful algorithms in various applications. However, traditional centralized collaborative filtering recommender systems (CFRS) suffers from sparse data problem and a lack of scalability as their calculation complexity increases quickly both in time and space when the number of the records in the user database increases. As a result, distributed collaborative filtering (DCF) is attracting increasing attention as an alternative implementation scheme. In this paper, we proposed a distributed collaborative filtering recommender system (DCFRS) for mobile commerce based on cloud computing because of its advantage of scalability as an alternative architecture. The experimental results demonstrate that the proposed algorithm improves the accuracy of a centralized system containing the same ratings and proves the feasibility and advantages of the proposed cloud-based DCF scenario.

**Keywords:** Collaborative filtering, distributed collaborative filtering, recommender system, mobile commence, cloud computing.

## 1 Introduction

The amount of information today is growing explosively especially in large scale mobile environment, which is commonly referred to the “information overload” problem. Therefore, it is time to create technologies that can help us sift through all the available information to find useful contents and reliable sources [1]. One solution to reduce the negative impact of information overload is the use of recommender system, which has become a classic tool that interlinks users with information content and sources. Recommender systems apply data analysis techniques to the problem of helping users find the items they would like to purchase by producing a predicted likeliness score or a list of top-N recommended items for a given user. Collaborative filtering (CF), based on the assumption that people with similar tastes prefer the same items, is such a personalized recommendation technique widely used in recommender systems. In order to generate a recommendation, CF initially creates a neighborhood of users with the highest similarity to the user whose preferences are to be predicted. It then generates a prediction by calculating the normalized and weighted average of the ratings of the users in the neighborhood. Nowadays, CF technology has gradually been implemented in various applications and proved to be very promising both in research and industry. However, there remain several challenges for collaborative filtering recommender systems (CFRS) and related algorithms. In this paper, we mainly discussed the improved CF algorithm from the perspective of cloud computing.

### 1.1. Related work

In this section, we briefly present some of research literature related to this paper such as CFRS, DCFRS and cloud computing.

1) Collaborative Filtering Recommender Systems

Since the mid-1990s, recommender systems have emerged to help people address the “information overload” problem and emerged as an independent research area. From then on, various approach such as CF, Bayesian networks, clustering, and Horting have been applied to recommender systems. Among all these approaches proposed, CF [2,3,4,5,6] has proved to be the most familiar, most widely implemented, and most mature recommendation technique. Generally speaking, CF algorithms can be normally categorized into two general classes: memory-based and model-based approaches. The memory-based recommender systems use the whole rating matrix to calculate the similarities between the active users and other users or the similarities between the active items and other items [7]. According to the similarities, the memory-based recommender systems can offer recommendations for users. In model-based algorithms, however, an explicit model of the relationships between users or items is first constructed. This model is then used to recommend the product or service to the users.

However, there are two limitations for CFRS regardless of its success in many application settings, thus influencing its recommendation accuracy.

The first problem is the high-dimensional sparse data problem. The sparsity problem, referring to the situation where insufficient historical transactions are available for inferring reliable consumer similarities [8], remains to pose a great challenge for traditional CF algorithm. In many cases, there is little user's rating data when the system is in its early stages, and the user ratings may not exceed 1% of the total number of projects in large systems, causing poor recommendation quality. Recently, many studies have been carried out to alleviate the sparsity problem and many useful methods were proposed. For example, Billsus et al. [9] incorporated the method of Singular Value Decomposition (SVD) into the CFRS for removing unrepresentative users or items and reducing the dimensions of rating data. In [10], taxonomic information was introduced to facilitate the inference of profile similarity in the sparse ratings data. Songjie Gong [11] presented a recommendation algorithm on integration of item semantic similarity and item rating similarity. The presented approach can effectively deal with the sparsity problem. By calculating the direct similarity and indirect similarity between users, YiBo Chen [12] proposed a new CF approach based on the user similarity matrix to alleviate the sparsity problem.

Another limitation is that most existing recommender systems based on CF are not scalable enough in large scale mobile environment. This is because most of the CFRS is centralized, which is suitable for only single website but can not adapt to large-scale distributed commerce. So the poor scalability of the CF algorithm makes it difficult to meet the needs of large-scale commercial recommendation especially for mobile commence [13]. Therefore, how to combine modern distributed technology to enhance the scalability of CFRS has attracted widely public concern. Several methods have been proposed to deal with the scalability problems. By using the folding-in projection technique, an incremental SVD CF algorithm was proposed by Sarwar et al. [14] to derive the SVD decomposition and alleviate the scalability problem when a set of new ratings is added to the database. Focusing on the scalability issue of traditional CFRS, Bo Xie et al. [15] developed a distributed user-profile management scheme based on distributed hash table (DHT) algorithms. Experiment results indicated that the proposed approach can effectively alleviate the scalability problem, have good coverage rate and recommendation quality.

2) Distributed Collaborative Filtering (DCF)

As an alternative implementation scheme for centralized recommender systems, DCF is attracting increasing attention in recent years because of its advantage in terms of scalability. Tveit firstly proposed a distributed recommender algorithm which is constructed on P2P network. The target user had the information of searching similar neighbors flooding on the network and waited for feedback from other nodes. The calculation of similarity was performed by the nodes which matched the query information. However, this method is passive and has low efficiency and accuracy in searching similar neighbors as they only make the strict match between ratings. Yuan et al. proposed a scalable distributed mechanism DROCF in pure P2P networks. In DROCF, the text features of document are represented by a vector. For increasing the quality of recommendation, the lexical chain method is employed to deal with the semantic problem of representing text. Then a peer’s preference is represented by a feature space consisting of all the vectors of saved documents. For acquiring better quality of recommendation, DROCF makes document recommendation by searching for nearest peers with similar preference through local information of recent ratings. Shlomo [16] proposed a novel approach to overcome the sparsity of data and cold start problems of CF by exploiting multiple distributed information repositories. They employed CF to generate user-personalized recommendations over different data distribution policies. Experimental results demonstrated that topical distribution outperforms geographical distribution. Han [17] proposed a DCF algorithm called PipeCF together with two novel approaches: significance refinement and unanimous amplification, to further improve the scalability and prediction accuracy. Xie et al.[15] introduced a DHT-based DCF algorithm which used the advantage of DHT technology in storing distributed data and location to deal with the problem of locating neighbors in distributed collaborative filtering recommender systems (DCFRS). However, this method of searching neighbors fails to retrieve partial similarity neighbors and the calculation of similarity only focuses on the target user while the appropinquity degree between the users' ratings is neglected. Therefore, there still has much work to do in locating neighbors exactly and improving the accuracy of DCFRS especially in large mobile environment.

3) Cloud-based Recommender System

Recently, several researchers [18,19] have claimed the effectiveness of incorporating cloud computing into the recommender system to address the sparse data and scalability problem of traditional CF. As a new business computing method, cloud computing is the further development and commercial implementation of distributed and parallel processing, virtualization and grid computing. It will provide large data "cloud" for mobile commence, promote the integration of mobile communications and the Internet, and achieve the business process management of mobile commerce. Besides, cloud computing can also improve the depth and scale of data mining, and resolve the real-time and quality problem of mobile commerce recommender system[18,19]. Therefore, the improved CF approach based on cloud computing can addresses the sparsity problems, allowing recommender systems to scale to large data sets and produce high-quality recommendations.

So far, few studies mentioned the recommender system for mobile commence based on cloud computing. To solve the scalability problem, Liang [20] proposed a parallel user profiling approach and a scalable recommender system. The current advanced cloud computing techniques including Hadoop, Map Reduce and Cascading are employed to implement the proposed approaches. The experiments were conducted on Amazon EC2 Elastic Map Reduce and S3 with a real world large scaled dataset from Del.icio.us website. Zhao [21] pointed out that CF algorithms were widely used in a lot of recommender systems. However, the computational complexity of CF is high thus hindering their use in large scale systems. In this paper, they implemented user-based CF algorithm on a cloud computing platform, namely Hadoop, to solve the scalability problem of CF. Experimental results showed that a simple method that partition users into groups according to two basic principles,i.e., tidy arrangement of mapper number to overcome the initiation of mapper and partition task equally such that all processors finish task at the same time, can achieve linear speedup. However, both of the two studies propose the recommender system based on Hadoop or Map Reduce, which are two kinds of platform of cloud computing. Therefore, the results are not universal and the system scalability is limited.

In this paper we present a cloud-based DCFRS which opens up the possibilities of addressing the sparse data and scalability problem of traditional centralized CF algorithms.

### 1.2. Contributions

This paper has three primary research contributions:

The first contribution focuses on alleviating the sparse data problem of traditional CF algorithm. We present an interest map model to divide the rating data in different matrixes and find the target user's nearest interest neighbors, thus alleviating the sparse problem issue.

The second contribution is that a cloud-based DCF recommendation mechanism is presented to address the scalability problem. We pre-classify the score clusters based on interest map model and the distributed structure of cloud computing, that is, selecting the most similar users in the same cluster and store them in cloud serves. Then we apply an improved clustering algorithm based on database division strategy and DCF algorithm within the similar users instead of a whole database.

The third contribution is the Movie-Lens dataset is used for an experimental comparison of the quality of traditional CF and the proposed cloud-based CF algorithm. Test results demonstrate that our experiment results illustrate that the recommender accuracy is acceptable and reasonable.

### 1.3. Organization

The remainder of this paper is structured as follows. In the section 2 and 3, we design a cloud-based DCFRS for mobile commence and present a novel distributed recommender algorithm based on cloud computing to address the two problems above for sake of broader development space and more accurate decision support. The results obtained in this simulation are discussed in Section 4. The conclusion and future works are concluded in Section 5.

## 2 Recommendation Mechanism for cloud-based DCFRS for Mobile Commence

The framework of cloud-based approach for DCFRS is depicted in Fig. 2.1. As is shown in Figure 1, the capability to learn users' interest and preferences is at the heart of the proposed DCFRS. In order to provide proper recommendations to mobile users, DCFRS requires user models of characteristics, preferences, and needs, which is typically referred to in the literature as “user interest modeling”. A new conception of “interest map” is proposed here to illustrate the interest modeling phase of DCFRS. Interest map is a graph associated with the common interest for persons, and many websites such as Hunch and GetGlue focus on interest map as the "taste" engine to deliver personalized recommendations. This paper proposed the interest map evaluation matrix for mobile users and a distributed recommendation algorithm based on cloud computing for sake of real-time personalized recommendations to meet the sensitivity of positions, emergency and portability for mobile commerce.

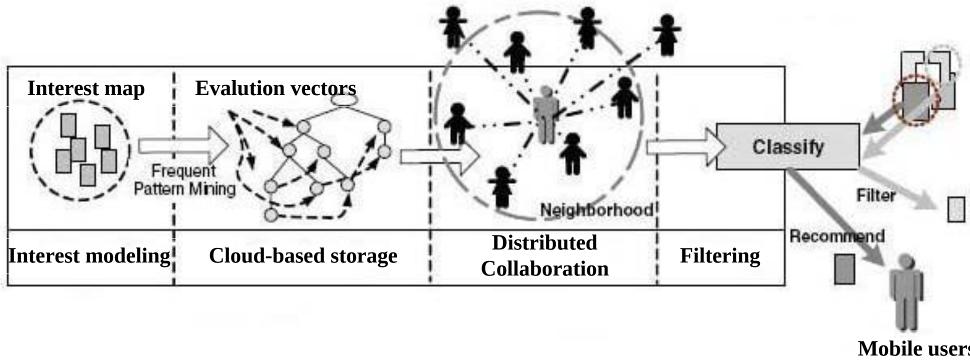


**Figure 2.1.** An overview of cloud-based DCFR

### 2.1. Interest Map Evaluation Matrix and Evaluation Vectors

For most mobile commence users, they may concentrate on one or several interest points. Therefore, the rating data for users can also be concentrated. If we can divide the rating data in different matrixes and find the target user's nearest interest neighbors, the calculation amount of recommender system can be greatly reduced and the problem of data sparseness can also be improved [18,19]. Besides, the scalability of cloud computing could expand the calculation scale while reducing the response time. Therefore, we can improve the recommendation accuracy theoretically.

To provide effective recommendation for new users, traditional recommender systems can collect the evaluation information from different users for different commodities. And all these information can be saved as a database in the mobile servers. The evaluation information from N users for M commodities can be seen in Tab. 2.1. In this table, the evaluation information from a user for each commodity can be listed in the set EV={A,B,C,D,E,F,…}. A, B, C… can be the scores or the rate of each commodity.

In traditional recommender systems, the serves save the evaluation information in form of centralized storage to realize effective recommendation for new users. However, centralized storage can not adapt to large-scale applications of mobile commence and reduce the recommendation quality. Therefore, we divide the evaluation information database into partitioned matrixes according to the same evaluation for each commodity. The partitioned matrixes can be saved in different sub tables such as shown in Tab.2.2 and Tab.2.3.

**Table 2.1.** Evaluation information from different users for different commodities

| M / U | Commodity 1 (M1) | M 2 | M 3 | … | M n |
|---|---|---|---|---|---|
| User 1 | A | C | B | … | D |
| User 2 | E | A | C | … | B |
| User 3 | A | E | A | … | G |
| User 4 | F | C | B | … | D |
| … | … | … | … | … | … |
| User N | E | B | D | … | C |

**Table 2.2.** Users with same evaluation for commodity 1

| (M 1,Ev A) | (M 1,Ev E) | (M 1,Ev F) |
|---|---|---|
| User 1 | User 2 | User 4 |
| User 3 | User N | … |
| … | … | … |

**Table 2.3.**Users with same evaluation for commodity 2

| (M2,Ev A) | (M 2,Ev B | (M 2,Ev C) | (M 2,Ev E) |
|---|---|---|---|
| User 2 | User N | User 1 | User 3 |
| … | … | User 4 | … |
| … | … | … | … |

In the cloud-based DCFRS for mobile commence, the mobile users with the same interest may provide the same evaluation information. Divide the total evaluation information database into different sub-database with the same evaluation information, so each sub-database is a set of users with the same evaluation for a commodity (homogeneous users with same $ev$ for commodity $m$ ). Then we save the same evaluation information from different users of a commodity in a cloud server.

## 2.2. Cloud-based Distributed Recommendation Mechanism

In the cloud computing environment, the architecture of cloud servers is distributed. To realize distributed management of the evaluation information database, we divide the database into sub-database and save different evaluation set in each cloud server. Same evaluation of one commodity saved in each cloud server can be seen in Fig.2.2.

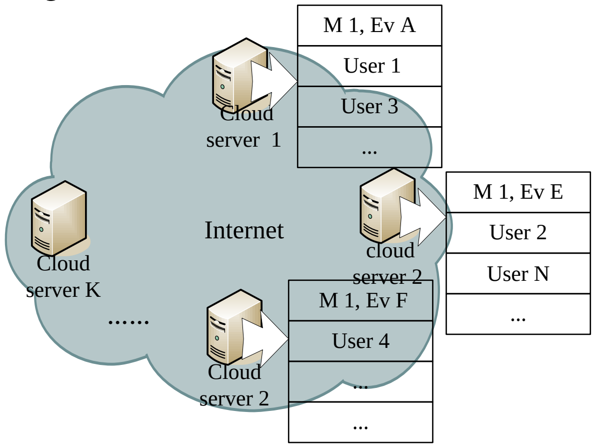


**Figure 2.2.** Same evaluation of one commodity saved in each cloud server

To manage the evaluation information effectively, we can also save all the evaluation information in a cloud server, just as shown in Fig.2.3. At this time, all the evaluation of commodity 1 are saved in cloud server 2. In the same way, we can also save all the evaluation information in other cloud servers.

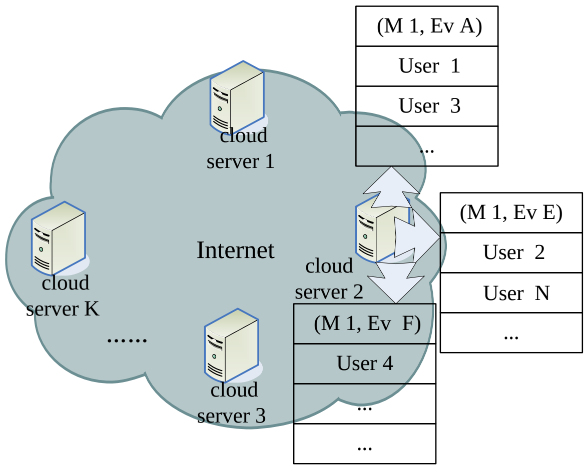


**Figure 2.3.** All the evaluation of one commodity saved in one cloud server

If a user is interested in many commodities, it may lead to collaborative query and the update of many cloud servers. Therefore, we propose a distributed architecture of cloud computing based on P2P. A structured P2P network model (such as Chord, CAN, Pastry) is given as shown in the Fig.2.4, and we can realize collaborative query and update among different cloud servers by distributed hash table technology (DHT). In the cloud computing environment, the cloud-based DCFRS can query related evaluation information from different cloud servers according to the structured P2P network model, and then provide more accurate decision support for mobile users.

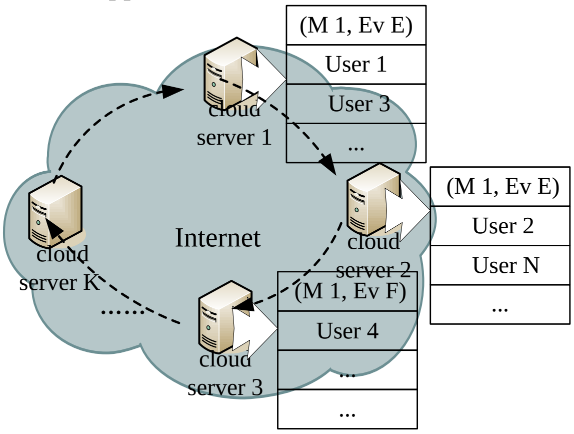


**Figure 2.4.** Collaboration of the cloud server based on P2P

## 3 Cloud-based DCF Algorithms for Mobile Commerce

Based on the distributed recommendation principles above, this article proposed cloud-based DCF algorithms as follows.

### 3.1. Cloud-based distributed storage strategy

In the proposed distributed recommendation system, Hadoop distributed file system (HDFS) was implemented to store the files and data. HDFS, composed of a Namenode and a number of Datanode, is a distributed file system with a characteristic of high fault tolerance. It has very high data throughput, and achieved a very good fault tolerance mechanism. HDFS provides multiple access interfaces, including API and various operation commands. HDFS can provide original data set with storage space for temporary files; offer data input and output service during the data pretreatment and data mining process. Therefore, with the aid of Hadoop HDFS design, user interest parameter data can be divided into fixed size distribution module and stored in the SuperDataNodes of Hadoop HDFS. The high reliable distributed data file storage function can be achieved by copy storage strategy, specific flow diagram as shown in Fig.3.1.

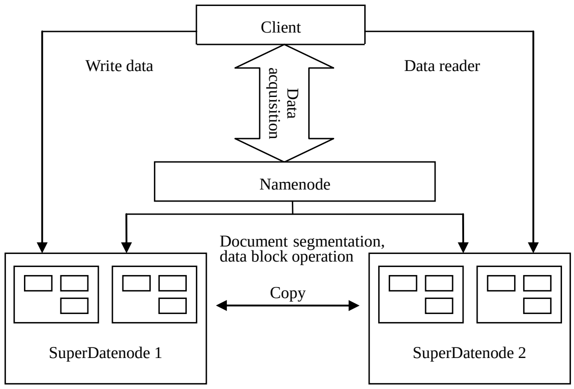


**Figure 3.1.** Hadoop replication storage strategy

### 3.2. Improved Clustering Algorithm in Cloud

The key connection of DCF is whether the original centralized user database can be divided and stored into the distribution of nodes in each local database. In other words, the original centralized user database should be divided according to the same evaluation information, and stored in the cloud servers. In this paper, we use data clustering method to divide each user's data to a peer node to realize database division strategy.

Generally speaking, user scores data with the same interest focus on specific areas [22]. Therefore, if the resources for users interested in the same class can be clustered into a class, then the sparsity of the matrix will be greatly reduced. The user scores classification can be changed into resource classification. Here, we use K-means clustering algorithm.

K-means clustering algorithm is one of the most commonly used method in collaborative filtering. The basic principle is that the data objects contain $n$ resource items, and each resource represented as a vector consisting of $m$ variables. As a resource variable, each user gives scores to the user resources. Through K-means algorithm, the $n$ resource items can be divided into $k$ classes. Each class belongs and only belongs to a cluster, and each cluster contains at least one resource item.

By the K-means clustering algorithm, the original matrix can be divided into several sub-matrix for CF. Assume that the original user-resources score matrix is divided into $K$ sub-matrices, each sub-matrix contains the number of users is $m_k\left(k=1,\cdots,K\right)$, the resource item is $n_k\left(k=1,\cdots,K\right)$. Generally, we can get $m_k \prec m$, $n_k \prec n$.

### 3.3. Nearest-neighbor Rating Matrix in Interest Map

In this case, the similarity between two items $i$ and $j$ in the interest map is measured by computing the Pearson-r correction $corr_{ij}$ and specific process is shown as follows.

1) Find the cloud server matrix $E^k_{m_i \times n_k}$ with the evaluation set of user " $a$ "for resources (commodities), and the interest description of user " $a$ " for each matrix is marked with $U^{(a)}$.

$$U^{(a)} = \{E_{ak} | k = 1, L, K\} \quad (1)$$

Where $E_{ak}$ means the <commodities, score> set from user " $a$ " for the commodities in matrix $k$. If user " $a$ " gives no comment for all the commodities in matrix $k$, then $E_{ak} = \varnothing$.

2) Compute the similarity between the items in matrix $k$ with user " $a$ ". To make the correction computation accurate, we use Pearson-r correction method to compute the similarity between the user " $i$ " and user " $a$ " in matrix $k$, we can get:

$$Sim^k(a,i) = \frac{\sum_j (r_{aj} - \overline{r_{ak}})(r_{ij} - \overline{r_{ik}})}{\sqrt{\sum_j (r_{aj} - \overline{r_{ak}})^2 (r_{ij} - \overline{r_{ik}})^2}} \quad (2)$$

Where $j \in I_{ak} \cap I_{ik}$, $I_{ak}$ is the number of the commodity scored in matrix $k$ from user " $a$ ", $I_{ik}$ is the number of the commodity scored in matrix $k$ from user " $i$ ".

3) Get the sequence of $Sim^k(a,i)$ from user " $a$ ", and the top $m_{ak}$ will be taken as the nearest-neighbors for user " $a$ ".

### 3.4. A Distributed Collaborative Filtering Algorithm based on Cloud Computing

For user " $a$ ", calculate the predicted value of the commodities he has not evaluated, and then generate a list of commodities recommended according to the predicted values for the users timely.

1) Find the sub-matrix $E^k_{m_i \times n_k}$ with mobile user " $a$ ",

2) For the commodity $j$ not evaluated by user " $a$ ", calculate its predicted evaluation value $P^k_{aj}$ form user " $a$ ".

$$p^k_{aj} = \overline{r_{ak}} + k \sum_{i=1}^{m_{ak}} \omega^k(a,i)(r_{ij} - \overline{r_{ik}}) \quad (3)$$

Where $\overline{r_{ak}}$ is the average score from the users in sub-matrix $k$, $m_{ak}$ is the number of nearest-neighbors, $r_{ij}$ is the score of commodity $j$ from the nearest-neighbor $i$, and $\overline{r_{ik}}$ is the average score from $i$ in sub-matrix $k$. $k$ is the normalization factor.

3) If the commodity $j$ belongs to several categories, the max predicted value will be selected as the final prediction.

4) Place the $P^k_{ak}$ of each categories in descending order. If the interest of user " $a$ " scatter in $C$ categories (C<K), the total recommender number is $N$, then the recommender commodities are top $N/C$.

## 4 Design of the experiment

With the aim of demonstrating the superiority of the proposed DCFRS compared to the traditional CF, we have performed a set of experiments that enable the comparison of both the quantitative and qualitative results obtained with the methods when applied to the MovieLens data set [23], which is a typical benchmark for collaborative recommendation. The MovieLens 10M data set here contains 10,000,054 ratings and 95,580 tags applied to 10,681 movies by 71,567 users.

### 4.1. Metrics and Methodology

To evaluate the accuracy and effectiveness of the recommendations, we report our results using the mean absolute error (MAE) evaluation metric, which is easy to understand and can evaluate recommending quality intuitively. MAE is an average of the absolute errors $p_i - q_i$, where $p_i$ is the prediction set and $q_i$ is the true value set. For all test datasets, the definition of MAE is given by

$$MAE = \frac{\sum_{i=1}^{N} |pi - qi|}{N} \quad (4)$$

### 4.2. Experimental Results

In this section, we compared the recommendation prediction influencing of our proposed DCF with traditional CF in two aspects: one is the sparsity level of datasets, and the other is the number of neighbors. The experimental result is shown in Fig.4.1 and Fig.4.2.

To evaluate the sensitivity of traditional CF and cloud-based CF algorithms under different sparse rating levels, our experiment was implemented in different levels of 0.65, 0.71, 0.75, 0.81, 0.86 and 0.90. From Fig.4.1, we can see that both of the algorithms are declining with the increase of rating density. In addition, the cloud-based CF algorithm has less MAE values than traditional CF. However; the gap between the two algorithms becomes larger with the increase of the sparsity. That means, when the rating metric becomes sparser, the cloud-based CF algorithm will performance better MAE.

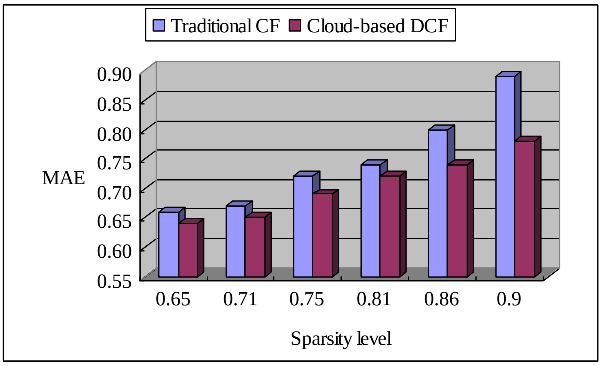


**Figure 4.1.** MAE impact of traditional CF and cloud-based CF for different sparsity levels.

Another experiment was implemented to measure the effects of different number of neighbors on MAE performance. Here the sparsity level was set to 0.75 and the number of neighbors was set from 5 to 50 scaling with 5 intervals. The experiment results can be shown in Fig.4.2. From figure7 we can safely draw the conclusion that both of the two algorithms are linearly proportioned to the number of neighbors. Additionally, the cloud-based CF have a smaller MAE than traditional CF from 5 to 50 scaling, indicating that the proposed CF algorithm in this paper has better accuracy than traditional CF under the same sparsity level conditions.

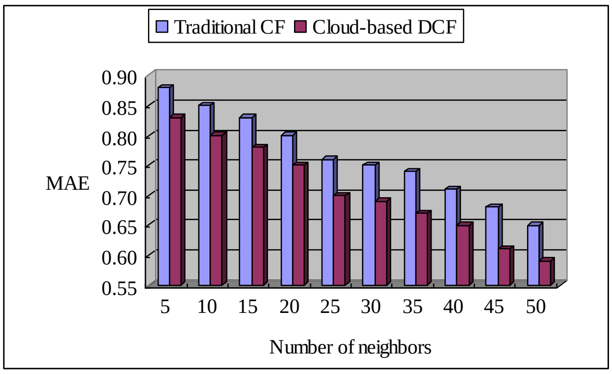


**Figure 4.2.** MAE impact of traditional CF and cloud-based CF for different number of neighbors.

To further illustrate the sensitivity of the neighbor parameter regarding the recommendation performance of traditional CF, K-means based CF and cloud-based CF algorithms; we depict the number of neighbors against the MAE measurement. Just as shown in Fig.4.3, When the number of neighbors is from 5 to 50 with 5 intervals, the phenomena of performance is similar with Fig.4.3. As the neighbors increase, the MAE decreases because more information is provided for prediction. Another conclusion is that the cloud-based CF has less MAE than traditional CF, indicating that the proposed algorithm in this paper provide dramatically better performance and better quality than traditional CF algorithms.

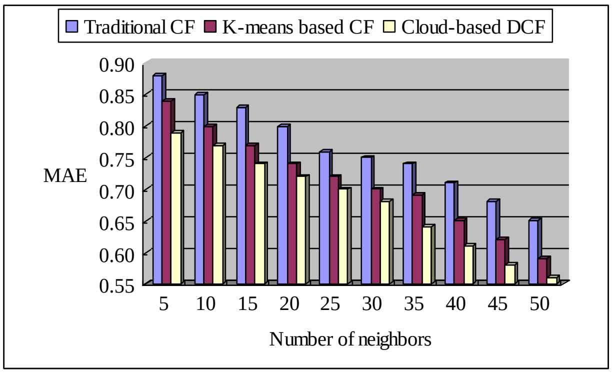


**Figure 4.3.** MAE comparison of traditional CF, K-means based CF and cloud-based CF for different number of neighbors.

### 5 Conclusion and future work

With the development of wireless networks and GPS technology, recommender system for mobile commerce has attracted much attention from business and academic in today's "Mobile World". To resolve the data sparsity problem of traditional CF, we presented a novel mechanism based on cloud user interest map to fill in the unrated rating in sparse matrix. Moreover, a DCFRS framework for the cloud-based mechanism is proposed to improve the predictive accuracy and the scalability of traditional CF algorithm. Case studies and experimental results illustrate that our approach hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations.

Although the approach presented in this study has shown promising results, it has also opened several tasks for future work. For example, as pointed out by a report from Symantec, security and privacy are the two main problems preventing cloud adoption. To deal with fraudulent behavior, anonymity and privacy problems under cloud computing conditions, we intend to investigate the possible usages of our model for credible recommendation, which is one of the emerging topics of trusted secure cloud.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos.71271186 and 71101124), the Ministry of Education of Humanities and Social Science Fund(Nos. 12YJA630191 and 12YJC790101) and Scientific Research Project of Hebei Education Department(Nos. ZH2011117 and 2011120).

## References

[1] Pavel Pesout, Ondrej Matustik, *The influence of the expansion of the smart mobile phones on the insurance industry, Applied Mathematics & Information Sciences,* 6, 437-440(2012).

[2] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry, *Using collaborative filtering to weave an information tapestry, Communications of the ACM,* 12, 61-70 (1992).

[3] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John T. Riedl, *Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems,* 22, 5-53 (2004).

[4] Greg Linden, Brent Smith, Jeremy York, *Amazon.com recommendations item-to-item collaborative filtering, IEEE Internet Computing,* 1, 76-80(2003).

[5] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, *Grouplens: an open architecture for collaborative filtering of netnews, Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work,* 1,175-186(1994).

[6] Upendra Shardanand, Pattie Maes, *Social information filtering: algorithms for automating word of mouth, Proc. of the SIGCHI conference on Human factors in computing systems,* 1,210-217(1995).

[7] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, *GroupLens: Applying collaborative filtering to usenet news, Communications of the ACM,* 3,77-87 (1997).

[8] Thanh D. X. Duong, Vu N. Duong, *Principal Component Analysis with Weighted Sparsity Constraint, Applied Mathematics & Information Sciences*, 4, 79-91 (2010).

[9] D. Billsus and M. Pazzani, *Learning collaborative information filters, Proc. 15th International Conference on Machine Learning,* 1, 46-54(1998).

[10] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme, *Taxonomy-driven computation of product recommendations, Proc. 13th International Conference on Information and Knowledge Management,* 1,406–415 (2004).

[11] Songjie Gong, *A Personalized Recommendation Algorithm on Integration of Item Semantic Similarity and Item Rating Similarity, Journal of Computers,* 5, 1047-1054 (2011).

[12] YiBo Chen, *Solving the Sparsity Problem in Recommender Systems Using Association Retrieval, Journal of Computers,* 5, 1047-1054 (2011).

[13] Zhang Yaming, Liu Haiou, *The Development of Cloud Computing—based on the Aspect of Technology and Business Value, Forum on Science and Technology in China,* 8, 126-133(2010).

[14] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *Incremental SVD-based algorithms for highly scaleable recommender systems, Proc. 5th International Conference on Computer and Information Technology,* 1,399-404(2002).

[15] Bo Xie, Peng Han, Fan Yang, *DCFLA: A distributed collaborative-filtering neighbor-locating algorithm, Information Sciences,* 17, 1349–1363(2007).

[16] Berkovsky, Shlomo, Eytani, Yaniv; Manevitz, Larry, *Efficient collaborative filtering in content-addressable spaces, International Journal of Pattern Recognition and Artificial Intelligence,* 21, 265-289(2007).

[17] Peng Han, Bo Xie, Fan Yang, Ruimin Shen, *A scalable P2P recommender system based on distributed collaborative filtering, Expert Systems with Applications,* 27, 203–210(2004).

[18] Zhang Yaming, Liu Haiou, Li Shiyong, *Research on the Cloud Computing Oriented Recommender System Model for Mobile Commenc, Proc. of the Second International conference on Artificial Intelligence, Management Science and Electronic Commerce,*1, 5124-5127 (2011).

[19] Zhang Yaming, Liu Haiou, *A Distributed Collaborative Filtering Recommendation Mechanism for Mobile Commerce Based on Cloud Computing, Journal of Information and Computational Science,* 8,3883–3891(2011).

[20] Huizhi Liang, Hogan, J., Yue Xu, *Parallel User Profiling Based on Folksonomy for Large Scaled Recommender Systems: An Implimentation of Cascading MapReduce, Proc. of the IEEE International Conference on Data Mining Workshops,* 1, 156-161(2010).

[21] Zhi-Dan Zhao, Ming-Sheng Shang, *User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop, Proc. of the Third International Conference on Knowledge Discovery and Data Mining,* 1, 478-481(2010).

[22] Li Xiao, Lixue Chen, Jingzhong Xiao, *A new algorithm*

*for shortest path problem in large-scale graph, Applied Mathematics & Information Sciences,* 6, 657-663 (2012).

[23] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, *An algorithmic framework for performing collaborative filtering, Proc. 1999 Conference on Research and Development in Information Retrieval,* 1,230-237(1999).

Yaming Zhang received the MS degree in Computer science from Yanshan University of China in 1996, and the PhD degree in Business management from Renmin University of China in 2002. He is currently a professor and PhD supervisor in School of Economics and Management, Yanshan University, Qinhuangdao, China. His research interests include mobile commence, and cloud computing.

Haiou Liu received the MS degree in Yanshan University in 2009. He is currently a PhD student in School of Economics and Management, Yanshan University, Qinhuangdao, China. His research interests include cloud computing and consumer behaviors.