

Study of Collaborative Filtering Recommendation Algorithm - Scalability Issue

Reena Pagare.
Dept. of Computer Engineering,
MIT College of Engg,
Pune University, India

Shalmali A. Patil.
Dept. of Computer Engineering,
MIT College of Engg,
Pune University, India

ABSTRACT

Recommender systems provide an important response to the information overload problem as it presents users more practical and personalized information services. Collaborative Filtering technique is the most successful in the recommender systems field. Collaborative filtering creates suggestions for users based on their neighbors preferences. But it suffers from poor accuracy, scalability and cold start problems. The tremendous growth of the number of customers and products in recent years poses some key challenges for recommender systems in which high quality recommendations are required and more recommendations per second for millions of customers and products need to be performed. Thus, the enhancement of scalability and efficiency of collaborative filtering (CF) algorithms become progressively more important and difficult. This paper focuses on study of different collaborative filtering algorithms taking into consideration the scalability issue. The different algorithms studied are cluster based, item based and context based.

Keywords:

Recommender System, Collaborative Filtering, Scalability.ifx

1. INTRODUCTION

Day by day the popularity of the Internet and information technology is increasing rapidly. Ever-expanding network information, information on individual people has turn out to be increasingly demanding. In this era of information overload, recommender systems play a dominant role in discovering valuable and interesting information for users searching among massively large databases. It is not easy to obtain what users want, because users sometimes do not know what they want to search for. Recommender systems provide personalized recommendations to users, directing them to ideal, needed and fascinating items.

Initially different recommendation systems constructed with different methods implementing to provide recommendations to users. In that, first was the GroupLens system built in 1994 on the observation that every time a user read Usenet News article which then gives out a valuable and helpful opinion. Such opinions were captured as "ratings" and used to produce personal predictions for like-minded readers. This was put on view as part of the article header. Then in 1995, Ringo system came into existence. By using social information filtering technique, this had provided recommendations for music artists. In 1995, Video Recommender Hill also used similar method to support recommendations. For that, they had used e-mail and the web among a virtual community of movie fans. Due to attracting features of Recommender system, it quickly became well-liked in research as well as in commercial practice. By 1996, popularity of recommendation system was taken into consideration by several companies, and they had started

marketing recommender engines [8].

Recommender systems can be divided in three categories, based on how user profile information is used :

- (1) **Content-based (CB):** The recommender will retrieve items whose content is similar to those of the profile.
- (2) **Collaborative Filtering (CF):** The recommender will retrieve items based on connections or similarities between user profiles.
- (3) **Hybrid approach:** It combines CB and CF [10].

2. BACKGROUND

Collaborative filtering is the most incredible method used by recommender systems to estimate predictions based on preferences of similar users while overcoming the information overload problem. CF in particular, attempts to automatically find users similar to the one asking recommendations based on their past preferences.

Typical CF process includes two main steps:

- (1) Formation of a neighborhood consisting of k most similar users
- (2) Producing a prediction on the information provided by such neighbors.

There are two major approaches for collaborative filtering algorithms:

- (1) model-based approaches
- (2) memory-based approaches

Model-based approaches use training data to generate a model. These models have been used to predict the ratings for the items that a user has not been rated before. In this approach the raw data is usually processed offline. For example, decision trees, aspect models, latent factor models and clustering methods are model-based approaches for collaborative filtering.

Memory-based approaches look at similar users or items based on their previous rating and combine their ratings in order to make new predictions. In this approach, the raw data is kept and processed in memory. Examples of memory-based collaborative filtering algorithms are user-based and item based methods. In user-based methods, similar users are the users who give similar ratings to items.

Accuracy and scalability are two important issues in recommender systems. Accuracy is the ability of the system to precisely predict how users rate items. A scalable recommender algorithm can cope with large numbers of users or items. Model-based approaches usually have better scalability, but lower accuracy, compared with memory-based approaches which have a better accuracy, but lower scalability [2].

3. CHALLENGES IN RECOMMENDER SYSTEM

Even though CF has been proven to be successful and widely accepted, it has vulnerabilities related to the characteristics of data sets. With ever-increasing popularity of the Internet, number of users accessing the Internet and the number of products offered online rapidly increases, which causes scalability problem. So it is becoming a challenge to put forward some predictions to many customers online in a limited time span. CF systems are supposed to return recommendations to many customers during an online dealings or interactions. Similarly, Users generally rate the small amount of products, hence sparsity problem is observed with a large set of items. So it becomes difficult to find neighbors and to provide accurate predictions when inadequate amount of ratings are available. Besides performance, accuracy is imperative for the success of CF schemes. Another problem of CF systems is that their vulnerable architecture. Due to which, it cannot preserve individuals privacy. Due to privacy concerns, it is not an easy task to collect truthful and an adequate amount of data for CF purposes. Many people do not want to give data about themselves due to privacy related issues. If it is assured that users privacy will not be spoiled, customers feel more comfortable and happy to give their data. Thus, besides performance and accuracy, protecting privacy is also important. In recommender systems, Cold start problem refers to the situation where an item cannot be recommended unless it has been rated by a substantial number of users. When a new node joins the system with a very low reputation score or no reputation score at all, its chance of being selected for interaction is generally rare. Hence, it is hard for a new user to raise his or her reputation score [3, 10].

Hence, the 4 main challenges are specially found in Collaborative filtering method in recommendation system are:

- (1) Scalability
- (2) Sparsity
- (3) Cold Start
- (4) Privacy

4. RELATED WORK

Different techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendation. Such techniques are discussed below.

4.1 Scalable Collaborative Recommender Algorithm Based on User Density-Based Clustering [2]

In this paper, besides using the traditional methods, the researchers have proposed a hybrid recommender system combining density-based user clustering method based on users demographic information with the user-based collaborative filtering. Their results show that the proposed method improves accuracy and scalability.

In their proposed method, the cascade hybridization of user-based collaborative filtering over density-based clustering algorithm is used. The result of first one gets enhanced by the second algorithm. It performs 3 steps, which are as follows.

- (1) Preprocessing
- (2) Clustering
- (3) User-based collaborative filtering

In order to generate recommendation, after preprocessing, the clustering technique is applied on it and clusters are produced. And finally, the user-based collaborative filtering is applied on that clusters. These all steps are described below.

(1) Preprocessing:

In this step, data normalization and page smoothing takes place. Here, in the research, age and gender are converted into numerical values for further similarity measurements, such as, 0 and 1 values are used to represent the users sex. To smooth the age of the user, the normalization technique is used by which different age ranges are formed.

(2) Clustering:

In this step, the density-based clustering algorithm named as DBSCAN is used to find the arbitrary shaped clusters by defining the density of the data points within the specific radius. In this algorithm, two parameters are used, which are epsilon and minimum number of neighborhoods. Also there are two types of data points, core points and border points. If the data point is having at least a minimum specified number of neighborhoods with distance less than or equal to Epsilon, then it is called as core point. These are also called as density-reachable points. If the data point is not having minimum specified number of neighborhoods with distance less than or equal to Epsilon, then it is called as border point. Initially, DBSCAN algorithm considers a random point P and finds all density-reachable points from it. Then by calculating number of neighborhoods, it decides whether the point P is a core point or border point. Then this algorithm visits the next point in the database. DBSCAN also merges two clusters into one cluster, if they are density-reachable.

(3) User-based collaborative filtering:

User-based collaborative filtering is a memory based collaborative filtering algorithm. In which, unseen rate for an item is predicted with the help of previous rating of the users and their similarities. Initially, in this algorithm, the similarity between any two users by comparing their ratings of co-rated items is calculated. For that, researchers have used Pearson Correlation, which is illustrated in the below formula (1).

$$W_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

Where,

I - set of items for which both user u and v have been given ratings

\bar{r}_u - average rating of the co-rated items of the u^{th} user

Then by finding the groups of users similar to the active user, the weighted average of rates is found out. The group of similar users is known as nearest neighborhoods. By using it, the algorithm makes prediction for an active user to an item.

(4) Hybridization technique:

Finally, partitioning of the rates based on the clusters they belong to is done and then the user-based collaborative filtering is applied on those partitions separately. This method contains the advantages of both model-based and memory-based collaborative filtering approaches simultaneously.

4.2 A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering [11]

Some collaborative filtering algorithms are based on user based clustering and some are based on item based clustering. But in this paper, the researchers have proposed the combined method

of both user-based clustering and item-based clustering, which is more scalable and accurate than traditional ones. In traditional algorithms, the aim is to predicting the rating for target item for the target user. For that, the User-Item matrix is used as input. This is represented in terms of table (1), as following.

User \ Item	I_1	I_2	I_n
U_1	R_{11}	R_{12}	R_{1n}
U_2	R_{21}	R_{22}	R_{2n}
....
U_m	R_{m1}	R_{m2}	R_{mn}

Table 1. : Training and test set

Where,

I - Item

U - User

R_{ij} - Rating of i_{th} user for j_{th} item

m - Total number of users

n - Total number of items

In the proposed approach, initially, the User clustering technique is used. In which, the clusters are build up based on the similar ratings of the different users. So based on similarity threshold, the fixed size partitions get generated and requested numbers of clusters are created. For this, the k-means clustering algorithm is used. Every cluster center finds out its average rating. To cover up the sparsity problem, the vacant rating values in the User-item matrix are filled with the average rating of the user center for the specific item. This provides the smoothening effect to the algorithm and produces the dense users ratings.

In the next step, the item clustering technique is used for producing recommendations. The Pearsons correlation is used to calculate the similarity between the items that have similar ratings. And then the k-means clustering algorithm is used over dense user-item matrix. Hence, it produces the item centers for item clusters. After that, the similarity between target item and item centers are calculated by using the Pearsons correlation formula. By that the most similar item clusters for target item is found out. Then it is a need to find out the similarity between the target item and items in the selected clusters. Over that, the Top- K most similar items are found out. Finally, by using the weighted average method the rating for target user for target item is produced.

4.3 Scaling-Up Item-based Collaborative Filtering Recommendation Algorithm Based on Hadoop [5]

In this paper, the researchers have used the popular computing frame work of cloud computing to solve scalability problem in collaborative filtering. For this, item based collaborative filtering is applied on Map-Reduce on Hadoop cluster. The collaborative filtering uses 3 costly computations to perform the algorithm, which are converted into 4 parallel phases of Map-Reduce to reduce the communication cost.

As per the name indicates, Map-Reduce contains 2 functional programming phases, which are Map and Reduce phases. For which, input and output of the computation is $\langle key, value \rangle$ set. As per the phases, the functions are:

$Mapper : \{key1, value1\} \rightarrow list\{key2, value2\}$

$Reducer : \{key2, list \langle value2 \rangle\} \rightarrow list\{key3, value3\}$

A developer can add a new phase between the above two phases, which can be operated by the local pairs to reduce the networks

communication load within their cluster. So for this, jobtrackers and tasktrackers are made available to implement parallelism within the clusters. Jobtrackers provides input to map phase by splitting provided data into number of pieces. Tasktracker collects the intermediate result from map functions of local distributed file system and store it within the Hadoop distributed file system.

In item-based collaborative filtering, over hadoop, Pearson correlation technique is used to find out the similarity between the two items and weighted average approach is used to find the prediction of rate of item for target user.

When actually applying item-based collaborative filtering algorithm on MapReduce, it follows the following 4 phases of computation:

Phase I:

The system is initially provided with the tuple $R \langle i, j, R_{(i,j)} \rangle$, where $R_{(i,j)}$ is the rating value of user j for item i . This produces the output of $\langle j, R_{(i,j)} \rangle$ tuple from Map-I. In this phase, the tuples with same i are shuffled on same machine. This output is given as input to reduce-I, which produces 2 outputs where is an average rating of item i and $I_{all} = \langle i, R_i \rangle$ which consists of all items and their average rating.

Phase II:

Map-II takes input \bar{R} from Reduce-I and produces the output $N(k, l)$. N defines a set of users who have rated items k and l . Reduce-II takes this output as input and finds out the similarity $S(k, l)$ of items k and l .

Phase III:

In Map-III and Reduce-III, the similarity matrix is found out. The resulted tuples $\langle i, j, \bar{R}_i \rangle$ of Map-III with same user j get shuffled to the same machine. After that, the Reduce-III emits $B = \langle i, j, R_{(i,j)}, \bar{R}_i, S_i \rangle$ for each $\langle i, j, R_{(i,j)} \rangle$.

Phase IV:

In Map-IV, $B = \langle i, j, R_{(i,j)}, \bar{R}_i, S_i \rangle$ tuples with same j get shuffled to the same machine in the form of $\langle i, R_{(i,j)}, \bar{R}_i, S_i \rangle$. In Reduce-IV, by taking inputs $\langle i, R_{(i,j)}, \bar{R}_i, S_i \rangle$ and $I_{(all)}$ emits the final output as Prediction $= \langle i, j, P_{(i,j)}, \bar{R}_i \rangle$ for all users to all the items.

4.4 Collaborative Filtering Recommendation Algorithm Based on Cluster [4]

In this paper, to improve the scalability, the researchers have proposed on improved collaborative filtering approach, which is cluster-based collaborative filtering recommendation algorithm.

In collaborative filtering, normally, there are 3 steps involved in recommendation generation, which are:

- (1) Rate data input
- (2) Formation of neighbors
- (3) Generating recommendation

Then, for finding out the neighbors, to calculate the similarity between them, different methods of measurement are used, such as, cosine similarity, adjusted cosine similarity and relevant similarity. With the nearest neighbors set, the recommendation is generated for the target user.

Over this collaborative filtering algorithm, the researchers have applied clustering method to make the groups of data objects of several categories. The data objects with high similarity find in same class. And the data objects with low similarity do not fit in to same class. This concept helps to improve accuracy and efficiency.

Cluster analysis is described as follows: Set of data is provided as $V\{v_1, v_2, \dots, v_n\}$. Where, v_i are called data points. Based on the degree of similarity between data points of data collection is divided into k categories, as follows: $\{C_1, C_2, \dots, C_k\}$ and $\bigcup_{i=1}^k C_i = V$, the process is known as clustering, $C_i (i = 1, 2, \dots, k)$ called clusters.

This technique of clustering is used to generate the personalized recommendation which requires a session file and transaction file for pattern clustering. For this, the following steps are implemented:

Step 1: The information access is mainly through site files, log files and user information, then which is processed using clustering technique.

Step 2: The similarity between patterns is calculated.

Step 3: Clustering algorithm to the transaction mode cluster is used to form the recommendation.

The researchers have used the K-means clustering algorithm. After clustering, the neighbors are found out by the center-based method. And finally, the recommendation is generated for unseen rate to an item.

4.5 Hybrid Recommender System with Temporal Information [1]

In this paper, the researchers have proposed to combine the memory based approach with model based approach in collaborative filtering to build a hybrid recommender system. It considers the demographic information, temporal information, rating similarity and item attributes similarity to construct user models. Basically, this method divides into two steps. First is constructing user models offline, and then, other is selecting neighbors and finding out the recommendation based on temporal information. The following figure (1) explains the offline working of hybrid recommender system with temporal information.

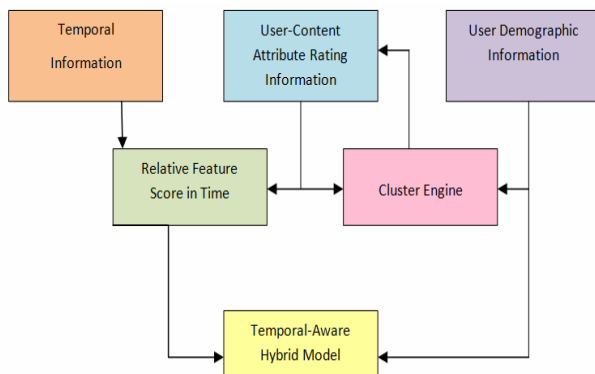


Fig. 1: Offline working of hybrid recommender system with temporal information

In first step, to fill up the vacant spaces in user rating matrix and to classify the contents, the clustering engine uses attribute similarity and demographic information. Then, based on that, by using Relative Feature-Score-Time and demographic information the temporal aware hybrid model is generated. Due to reduction in the size of temporal aware hybrid model than user-item rating matrix, the scalability of system increases.

In figure (2), the online working of hybrid recommender system with temporal information is shown.

For this step, the input is provided as temporal aware hybrid model. Then the set of neighbors is found out by using model-based collaborative filtering. It takes less time to compute, only because it considers the contents that user like at that specific

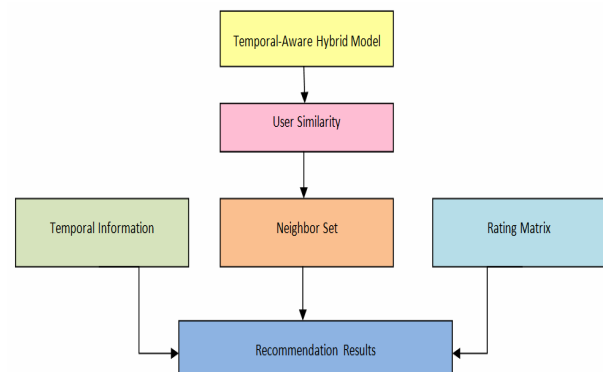


Fig. 2: Online working of hybrid recommender system with temporal information

time. For finding the similarity, one of the following techniques is used: the Pearson correlation, cosine vector similarity, adjusted cosine vector similarity and relevant similarity. Then, the recommendation result is calculated based on temporal information, neighbor set and rating matrix.

4.6 An Improved Profile-based CF Scheme with Privacy [3]

In this paper, the researchers have projected a pre-processing scheme to overcome the scalability problem.

Normally, in collaborative filtering, it takes much of the online time to calculate similarities between users or items. But, as the number of users and items in the system are escalating very rapidly, it is reducing the performance of the system. So, to surmount the scalability issue, the researchers have used content based profiling method. In this, the user profiles are constructed upon content of the item. These profiles work as supplementary profiles. This helps to reduce the dimension space for similarity calculation.

In this approach, initially, the items need to be classified base on the criteria among items. Considering, engineering field, it has various streams (Computer, Automobile, Civil, etc.), electronic devices (Smart-phones, tablets, PCs, etc.). Then, the kNN algorithm is used to compute similarities based on content based profiles. And, for the prediction calculation weighted sum technique is used.

To find the content based profiles, according to users purchase and users rating, the two approaches are used, which are as follows:

- (1) **Purchased-based profiles:** To produce purchased-based profiles, the item is rated or not is essential. When a user buys (or purchases or rates) a item, the value of the each category in which that item belongs gets risen by 1.
- (2) **Rating-based profiles:** To generate rating-based profiles, the rating of item plays important role. Hence, it is initially checked whether that item is rated or not. For rated item, the rating based profile gets increased by that specific rating value.

By processing offline and reducing the amount of time to calculate the prediction, the profiling scheme contributes well to overcome the scalability issue.

4.7 F-REC: Fast and Accurate Computation of Recommendations based on Rating Frequencies [9]

In this paper, the researchers have used new scheme named RF-REC for recommendation over collaborative filtering in

Title of the Paper	Conclusion	Drawbacks	Performance Measurement	Future Work
Clustering Methods				
1. A Scalable Collaborative Recommender Algorithm Based on User Density-Based Clustering [2011]	<ul style="list-style-type: none"> Contains advantages of both memory and model based CF Scalable Accurate 	<ul style="list-style-type: none"> Sparsity problem Complexity is greater $[O(n^2)]$ compared to complexity of K-means $[O(n)]$ 	<ul style="list-style-type: none"> Mean Absolute Error 	<ul style="list-style-type: none"> Inter-cluster rates smoothening can be implemented in future
2. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering [2010]	<ul style="list-style-type: none"> Low MAE than traditional CF Quality improvement More accurate and more scalable than traditional CF 	<ul style="list-style-type: none"> Cold start problem 	<ul style="list-style-type: none"> Mean Absolute Error 	<ul style="list-style-type: none"> Cold start problem is to be handled
3. Scaling-up Item-based Collaborative Filtering Recommendation Algorithm based on Hadoop [2011]	<ul style="list-style-type: none"> Minimize communication cost Scalable Efficient performance 	<ul style="list-style-type: none"> Scalability to be handled more effectively 	<ul style="list-style-type: none"> Speedup Isoefficiency 	<ul style="list-style-type: none"> On algorithm side, optimize allocation and execution process to be included On application side, scalability is to be improved
4. Collaborative Filtering Recommendation Algorithm Based on Cluster [2011]	<ul style="list-style-type: none"> Combination of CF and clustering to improve scalability and sparsity problem 	<ul style="list-style-type: none"> Cold start 	<ul style="list-style-type: none"> (Only algorithm is proposed) 	<ul style="list-style-type: none"> For better performance, the cold start and privacy is to be taken into consideration
Other Methods				
5. Hybrid Recommender System with Temporal Information [2012]	<ul style="list-style-type: none"> Scalable Sparsity and cold start problems are covered 	<ul style="list-style-type: none"> Privacy issue is not handled 	<ul style="list-style-type: none"> Mean Absolute Error Recommendation time 	<ul style="list-style-type: none"> Geographic information can be considered Improving time stamping for better performance RS for N-screen service
6. An Improved Profile-based CF Scheme with Privacy [2011]	<ul style="list-style-type: none"> Improved performance Improved accuracy More scalable 	<ul style="list-style-type: none"> Sparsity problem Accuracy and quality are not so good compared to CF scheme 	<ul style="list-style-type: none"> Mean Absolute Error 	<ul style="list-style-type: none"> For more accurate results combining the proposed schemes, such as clustering or data reduction techniques combine for enhancement
7. RF-REC: Fast and Accurate Computation of Recommendations based on Rating Frequencies [2010]	<ul style="list-style-type: none"> Highly scalable More accurate Easy to implement High prediction Prediction at constant time 	<ul style="list-style-type: none"> Poor performance Higher model building time 	<ul style="list-style-type: none"> Root Mean Squared Error 	<ul style="list-style-type: none"> Improve quality of diversity and novelty

Table 2. : Comparative Study of Different Algorithms Implemented to Reduce Scalability Issue in Recommender System

recommender system to provide personalized recommendations to overcome the problems such as scalability, efficiency.

This scheme has some advantages, as

- (1) Its computational cost is very low.
- (2) Easy to implement.
- (3) It takes constant time for prediction.
- (4) New ratings get easily absorbed in the existing database.

In collaborative filtering method, there is data sparsity problem in the input data matrix. So RF-REC scheme works to predict such rating and fill up it into the matrix. In user-item matrix, the RF-REC scheme checks out the frequencies of both user and item rating to the respective sparse block and foretell the rating for that block. For such calculation, it uses the prediction function, which is as follows:

$$\hat{r}_{u,i} = \arg \max_{v \in \mathcal{R}} (freq_{user}(u, v) + 1 + 1_{user}(u, v)) / (freq_{item}(i, v) + 1 + 1_{item}(i, v)) \quad (2)$$

Where,

$freq_{user}$ - Frequency rating of target user u with value v
 $freq_{item}$ - Frequency rating of target item i with value v
 $1_{user}(u, v)$ and $1_{item}(i, v)$ - Indicator functions

But, this RF-REC scheme gives only inter value as output, so 'weighted RF-REC scheme' is developed which results into floating value for more accurate prediction. In weighted RF-REC scheme rather than considering one maximizing rating of product, the rating frequencies of all ratings is considered.

5. CONCLUSION

The study shows that different techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendation. They all are using the similar input dataset- Movie Lens work group (www.grouplens.org), except "Collaborative Filtering Recommendation Algorithm Based on Cluster". Also, for comparing them the similar performance measurement factor named 'Mean Absolute Error' (MAE) is taken into consideration for most of the methods. Here we have studied 4 cluster based CF algorithms, in which a collaborative filtering recommendation algorithm which is based on user-based clustering and item-based clustering is best method, because it shows lower MAE than traditional CF. Also it is more accurate and scalable than traditional CF with a quality improvement.

But the hybrid recommender system with temporal information is best method from all methods which we have studied. Because it constructs offline to make the recommendation system to recommend item for a user within user bearable time which will also reduce the computational time. Also, it solves scalability, sparsity and cold start issue and provides final recommendation quickly and accurately.

6. REFERENCES

- [1] Farman Ullah, Ghulam Sarwar, Sung Chang Lee, "Hybrid Recommender System with Temporal Information", (ICOIN), International Conference on Information Networking, IEEE 2012, pp. 421-425
- [2] Siavash Ghodsi Moghaddam, Ali Selamat, "A Scalable Collaborative Recommender Algorithm Based on User Density-Based Clustering", 3rd international conference on Data Mining and Intelligent Information Technology Applications (ICMiA), IEEE 2011, pp. 246-249
- [3] Alper Bilge and Huseyin Polat, "An Improved Profile-based CF Scheme with Privacy", Fifth IEEE in International Conference on Semantic Computing, IEEE 2011, pp. 133-140
- [4] Xingyuan Li, "Collaborative Filtering Recommendation Algorithm Based on Cluster", International Conference on Business Computing and Global Information (BCGIN), IEEE 2011, pp. 645-648
- [5] Jing Jiang, Jie Lu, Guangquan Zhang, Guodong Long, "Scaling-up Item-based Collaborative Filtering Recommendation Algorithm based on Hadoop", IEEE World Congress on SERVICES, 2011, pp. 490-497
- [6] Gilda Moradi Dakhel, Mehregan Mahdavi, "A New Collaborative Filtering Algorithm Using K-means Clustering and Neighbors Voting", International Conference on Hybrid Intelligent Systems (HIS), IEEE 2011, pp. 179-184
- [7] Kyung-Yong Chung, Daesung Lee and Kuinam J. Kim, "Categorization for grouping associative items using data mining in item-based collaborative filtering", International Conference on Information Science and Applications (ICISA), IEEE 2011, pp. 1-6
- [8] Joseph A. Konstan John Riedl, "Recommender systems: from algorithms to user experience", User Modeling and User-Adapted Interaction, 2012, Vol. 22, pp. 101-123
- [9] Fatih Gedikli, Faruk Bagdat, Mouzhi Ge, and Dietmar Jannach, "RF-REC: Fast and Accurate Computation of Recommendations based on Rating Frequencies", IEEE 13th Conference on Commerce and Enterprise Computing (CEC), 2011, pp. 50-57
- [10] Mozhgan Tavakolifard, Kevin C. Almeroth, "Social Computing: An Intersection of Recommender Systems, Trust/Reputation Systems, and Social Networks", IEEE Network, July/August 2012, Vol. 26, No. 4, pp. 53-58
- [11] SongJie Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering", Journal of Software, , July 2010, Vol. 5, No. 7