Implementing and Evaluating Collaborative Filtering (CF) Using Clustering

Sachin S. Agrawal and Ganjendra R. Bamnote

Abstract A tremendous increase has taken place in the amount of online content. As a result, by using traditional approaches, service-relevant data becomes too big to be effectively processed. In order to solve this problem, an approach called clustering based collaborative filtering (CF) is proposed in this paper. Its objective is to recommend services collaboratively in the same clusters. It is a very successful approach in such settings where interaction can be done between data analysis and querying. However the large systems which have large data and users, the collaboration are many times delayed due to unrealistic runtimes. The proposed approach works in two stages. First, the services which are available are divided into small clusters for processing and then collaborative filtering algorithm is used in second stage on one of the clusters. It is estimated to decrease the online execution time of collaborative filtering algorithm because the number of the services in a cluster is much less than the entire services available on the web.

Keywords Collaborative filtering · Clustering · Big data

1 Introduction

A large number of applications and websites are built on Internet which target on data and user interaction. It is assumed that the users of these systems will be introduced to the new content by recommendations given by their friends and can also submit feedback to make these recommendations better. Data collection has grown extremely and is in front of the ability of frequently used tools to capture, manage, and process by software's [1]. For large data sets, the term known as Big

S.S. Agrawal
College of Engineering and Technology, Akola, India
e-mail: sachin.s.agrawal@gmail.com

G.R. Bamnote (✉)
Prof Ram Meghe Institute of Technology and Research, Badnera, Amravati, India
e-mail: grbamnote@rediffmail.com

S.C. Satapathy and S. Das (eds.), *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*,
Smart Innovation, Systems and Technologies 51, DOI 10.1007/978-3-319-30927-9_16

data is used. These data sets are very large or difficult that old data processing applications are insufficient. The main challenges that are accompanied with such data sets are storage, data creation, analysis, sharing, capture, search, transfer, information privacy and visualization. Thus to find the very large volumes of data and find useful information for future actions is the main challenge for Big Data applications [2]. These demands are met by using user data or preferences and algorithms to advise novel items that will help them in decision making process through the techniques called Recommender systems (RS).

A RS has many components like items, preferences, users, neighborhoods and ratings. Collaborative filtering (CF) such as item and user-based methods are the major techniques applied in RSs [3]. The objects or things that are recommended to a user are called items. Items, for example can be news articles, reviews, games, movies or songs. The characterization of these items can be by their specific metadata which comprise of relevant titles, tags or keywords. Users are the people who are being recommended. Many times they require guidance or assistance in preferring an item in an application [4].

The large number of services that need to processed in real time need to be decreased. The techniques which can decrease the volume of data by a huge factor by grouping similar services together are known as clustering. Thus, we propose a collaborative filtering method by using clustering. The proposed approach consists of two stages, i.e., clustering and collaborative filtering. The first step i.e., clustering, is a preprocessing step to separate big data into usable parts [5]. A cluster which contains some similar services like a movie contains some like-minded actors. The computation time of CF algorithm can be reduced significantly because in a cluster the total number of services is less than the overall number of services. Also the ratings of similar services within a cluster are more related than that of dissimilar services [6], thus the recommendation accuracy based on users ratings can be enhanced. Clustering technique is a promising way to improve the scalability of collaborative filtering by reducing the quest for neighborhoods between clusters instead of using complete data set. It recommends accurate and better recommendations to users. So for each specific user a user model can be made for enhanced recommendations. These user model works as profiles where actions and preferences are encoded. It also represents the history of a user along with their interactions with items in the recommender system and such interactions are called as preferences. Many a times preferences are classified as ratings if a recommender system gives a media to rate items [7]. The user view of an item in a recommender system can be understood as preferences and it can be both implicit/explicit. A group of similar users will be represented by a neighborhood that reports users and their preferences [8]. Classification of Collaborative filtering can be done in two classes. These classes are model-based methods in which the total user-item rating dataset is used to make predictions and neighborhood-based (memory based) methods.

2 Basic Knowledge

The metadata from the web service description language files was investigated by Liu [9] to compute the similarity amongst web services and defined a web service as =(P, Q, R, S), where P is the web service name, Q is the group of messages exchanged, R is the group of data type, and S is the group of operations provided by the web service. Li [10] defined a web service for evaluating reputation of service as $WS(A, R, B, C, D, E)$ where A is identity, B is classification, R is description of text, C is transaction volume, E is degree of reputation and D is review group. A service proposed by Zielinnsk [11] can be defined as a conceptual specification of information technology functions that are business-aligned. While the definitions of service are different and application-specific, they have common elements which essentially include service descriptions and service functionalities. Another important user activity that reflects their opinions on services is rating. Service rating is an important element mainly in application of service recommendation. As a number of services are emerging on the Internet, such large volume of service-related elements are generated and distributed across the network, which cannot be effectively accessed by common database management system. To tackle this problem, Hadoop is used to store services. It uses a disseminated user level file system across the cluster to handle storage sources. Hadoop [12] is an open source achievement of Map-Reduce technique for large datasets analysis. A file system called HDFS (Hadoop distributed file system) is used in Hadoop. The HDFS provides the foundation in storing files in a storage node. Task trackers and job trackers are provided by Mapreduce. It has been successfully used by Google to process big datasets. A distributed file system is needed by Mapreduce and also an engine which monitors, coordinate, collect and distribute the consequences. HDFS contains nodes and name node and is a master and slaver framework. The name node manages namespace in the file system and is a center server and data node handles the data stored [13].

2.1 Architecture of HDFS

On the compute nodes, the HDFS stores data and also provides large average bandwidth amongst the cluster. The mechanism consists of master and slave nodes. The master node consists of single name node and the slave node consists of number of data nodes. In the cluster, the nodes are distributed one data node/machine, whose purpose is to handle data block attached to the machines. The operations on file system namespace are executed by the name node. It also maps data blocks to data nodes. For serving write and read requests data node are responsible from client and execute block procedures upon instructions [14]. For higher performance, resiliency and load-balancing, HDFS divide data into large pieces and on the server it makes multiple copies. By means of data copies at

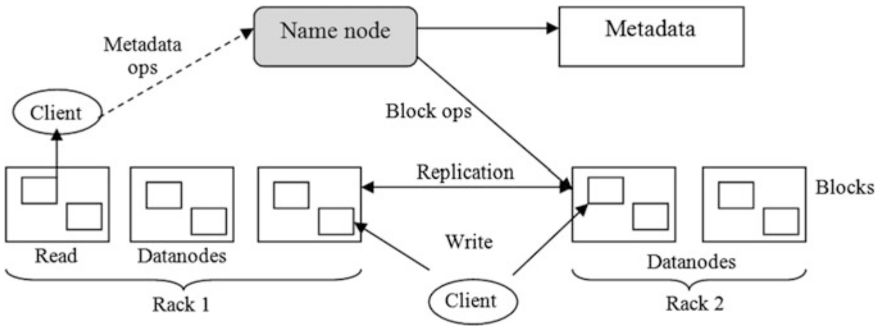


Fig. 1 HDFS architecture

multiple servers, whichever server at any time can access in computation, writing and reading of a data. Figure 1 shows the HDFS architecture. All decisions concerning block replication are made by name node. For a large cluster it cannot be realistic to join the nodes in flat topology.

The general method is dividing the nodes in many racks. The switches in a rack are joined by core switches and a switch is shared by a rack share. Nodes in multiple racks undergo number of switches.

2.2 Mapreduce

Its main objective is to give facilities which permit execution and development of jobs which are processed form big scale data. Its goal is to use the processing capacity in a much more efficient way which is given by processing cluster. Simultaneously it gives a logical model which makes the advancement of distributed applications simpler. It is made in such a way that it is flexible to machine crash failures. Map-reduce is used by Google to handle large data sets whose size can exceed in terabytes. It is motivated to achieve this through the thought of functions which are of high order.

3 Deployment of Clustering and Collaborative Filtering

The proposed approach works in two stages. First, the services which are available are divided into small clusters for processing and collaborative filtering algorithm is used in second stage on one of the clusters.

3.1 Calculate Description and Functionality Similarity

By Jaccard similarity coefficient, similarities are calculated which measure similarity amongst sample sets statistically. The coefficient of similarity can be defined as the ratio of the cardinality of their conjunction to the cardinality of the union for 2 sets. Mathematically, similarity of description between A and B is computed by

$$D\_Similarity(A,B) = \frac{\left|D'_A \cap D'_B\right|}{\left|D'_A \cup D'_B\right|} \tag{1}$$

From the above formula, the larger $\left|D'_A \cap D'_B\right|$ is, the more similar the two services are. Dividing by $\left|D'_A \cup D'_B\right|$ is the scaling factor which ensures that description similarity is between 0 and 1. Mathematically, functionality similarity between A and B is computed by

$$F\_Similarity(A,B) = \frac{\left|F'_A \cap F'_B\right|}{\left|F'_A \cup F'_B\right|} \tag{2}$$

3.2 Calculate Characteristic Similarity

The characteristic similarity between A and B is computed by weighted sum of description similarity and functionality similarity. Mathematically,

$$C\_Similarity(A,B) = \alpha \times D\_Similarity(A,B) + \beta \times F\_Similarity(A,B) \tag{3}$$

In above formula,$\alpha \in [0, 1]$ is the weight of description similarity, $\beta \in [0, 1]$ is the weight of functionality similarity and $\alpha + \beta = 1$. The relative importance between these two can be expressed by the weights.

3.3 Cluster Services

Clustering techniques are widely used in RS to divide object sets into clusters. The objects in the identical cluster are similar to each other than objects in diverse clusters. Usually, algorithms of cluster analysis are being employed where data storage is large. The algorithms used for clustering may be partitional and hierarchical. Some standard partitional approaches (e.g., K-means) suffer from several limitations such as their result depends on the accurate value of K which is originally unidentified, number of clusters K choices and size of cluster is not considered while executing the algorithm K-means, a number of clusters can turn out to be unfilled and can result is early completion of the algorithm. Also algorithms converge to a local minimum [15]. The hierarchical clustering methods are grouped into agglomerative or divisive on the basic of whether the hierarchy formation is from top-down fashion or bottom-up [16]. Many current modern clustering systems use a clustering strategy as agglomerative hierarchical clustering, because of its simple structure of processing and adequate performance level [17]. The AHC algorithm for service clustering is as follows:

```
Input: A set of items $S = \{I_1 ,............I_n\}$, a characteristic
       similarity matrix $D = [d_{i,j}]_{n \times n}$ the number of required
       clusters $k$.
Output: Dendrogram for $k$ = 1 to | $I$ |
Algorithm:
       1. $c_i = \{I_i\}, \forall i$.
       2. $d_{Ci,Cj} = d_{i,j}$ $\forall$i,j
       3. for $k$ = |$I$| down to $K$
       4. $\text{Dendogram}_k = \{c_1,.......c_k\}$;
       5. l,m = $\text{argmax}_{i,j}$ $d_{Ci,Cj}$
       6. $c_l$ = Join ($c_l$ , $c_m$);
       7. for each $c_h \in$ I
       8. if $c_h \neq c_l$ and $c_h \neq c_m$.
       9. $d_{cl,ch}$ = Average($d_{Cl,Ch}$, $d_{Cm,Ck}$);
       10. end if
       11. end for
       12. I = I − {$c_m$};
       13.end for
```

Algorithm: AHC algorithm

3.4 Compute Rating Similarity

The rating similarity computation between items is time consuming but important step in item-based CF algorithms. The cosine similarity between ratings vectors and the Pearson correlation coefficient (PCC) are the common rating similarity measures included [18, 19]. By measuring the resemblance amongst two users or items or measuring the character of two series to move together in a linear or proportional manner the PCC calculates the similarity between two items or users. Preferences calculation is given by:

$$PCC(a,b) = \frac{\sum_i \left(w_{a,i} - \bar{w}_a\right)\left(w_{b,i} - \bar{w}_b\right)}{\sqrt{\sum_i \left(w_{a,i} - \bar{w}_a\right)^2 \sum_i \left(w_{b,i} - \bar{w}_b\right)^2}} \quad (4)$$

where a and b are users/items, i is item, $w_{a,i}$ and $w_{b,i}$ are ratings from a and b for i, $\bar{w}_x$ and $\bar{w}_y$ are mean ratings for user a and b.

3.5 Select Neighbors

Based on the rating similarities between services, the neighbors of a target service A are determined as

$$N(A) = \{B|R\_Similarity(A, B) > \gamma, A \neq B\} \tag{5}$$

here $R\_similarity(A, B)$ is the enhanced rating similarity between service A and B,γ is a rating similarity threshold. The larger value of γ is, the chosen number of neighbors will moderately less but they can be more similar to the target service, hence the coverage of collaborative filtering will reduce but the accuracy can increase [20]. In contrast, the smaller value of γ is, the more neighbors are selected but some of them can be only somewhat similar to the target service, thus the coverage of CF will increase but the accuracy would decrease.

3.6 Compute Predicted Rating

For an active user U_A for whom predictions are being made, whether a target service A is worth recommending depends on its predicted rating. If $N(A) \neq \phi$, similar to the computation formula proposed by Wu et al. [20], the predicted rating $P(U_A, A)$ in an item-based CF is computed as:

$$P(U_A, A) = \bar{r}_A + \frac{\sum B \in N(A)(r(U_A, B) - \bar{r}_B) \times R\_Similarity(A, B)}{\sum B \in N(A) R\_Similarity(A, B)} \tag{6}$$

here, $\bar{r}_B$ is the mean rating of B, $N(A)$ is neighbor set of B, $A \in N(B)$ denotes A is a neighbor of the target service B, $r(U_A, B)$ is rating that an active user U_A give to B, $\bar{r}_B$ is average rating of B, and $R\_similarity(A, B)$, is enhanced rating similarity between service A and B [21].

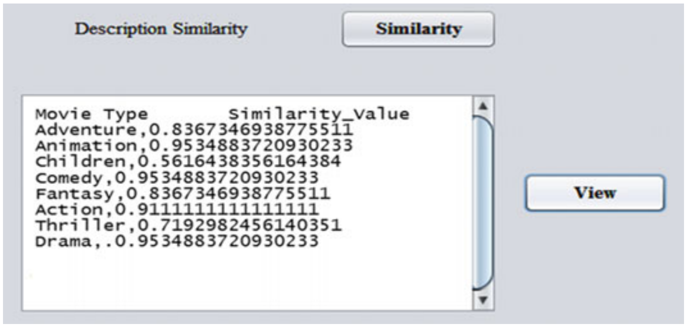
4 Experiments and Results

4.1 Data Sets Used

To verify the proposed approach, the Movie-Lens and the Group-Lens data sets are used. The data sets include data about movies, movie ratings and users. Different sizes of data sets are used in experiments. Some of these are the Movie-Lens ML-100-K which has around one lakh ratings obtained from about one thousand users for about one thousand seven hundred movies, the Movie-Lens ML-1M data set which has about ten lakhs obtained from six thousand users for four thousand movies. The Movie-Lens ML-10-M data set which has about 1 crore ratings and one lakh tags obtained from ten thousand movies rated by about seventy two thousand users [22].

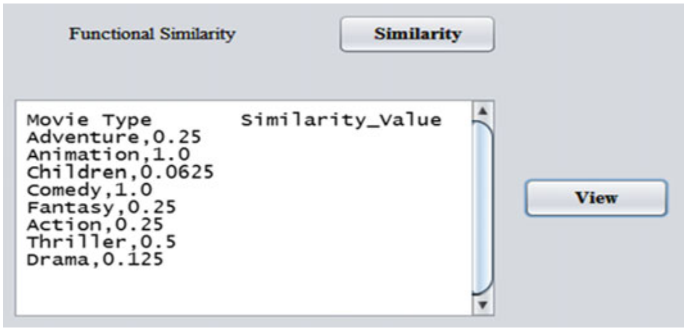
4.2 *Evaluation of Similarity Algorithms*

The main objective of this approach is to calculate the accurateness of a recommender system when there is increase in number of users, user preferences and items. By using the Movie-Lens data sets with different users preferences and feedback, we have evaluated description similarity and functional similarity by using the clustering algorithm. The results are shown in Figs. 2 and 3.



```
Movie Type        Similarity_Value
Adventure,0.8367346938775511
Animation,0.9534883720930233
Children,0.5616438356164384
Comedy,0.9534883720930233
Fantasy,0.8367346938775511
Action,0.9111111111111111
Thriller,0.7192982456140351
Drama,.0.9534883720930233
```

Fig. 2 Description similarity



```
Movie Type        Similarity_Value
Adventure,0.25
Animation,1.0
Children,0.0625
Comedy,1.0
Fantasy,0.25
Action,0.25
Thriller,0.5
Drama,0.125
```

Fig. 3 Functional similarity

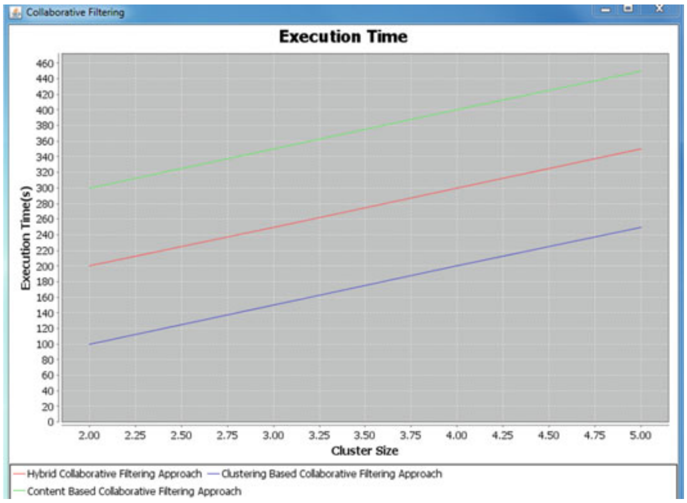


Fig. 4 Execution time

Figure 4 and 5 shows the graph of execution time (s) versus cluster size and graph of efficiency (similarity level) versus the cluster size for three different approaches i.e. hybrid, content and clustering based. From the graph we can see that the execution time is minimum by using the proposed approach and the efficiency of the clustering based approach is more as compared to other approaches.

The accuracy comparison of different approaches are shown in Fig. 6, from the results we can observe that the accuracy of clustering based approach is 80 % more

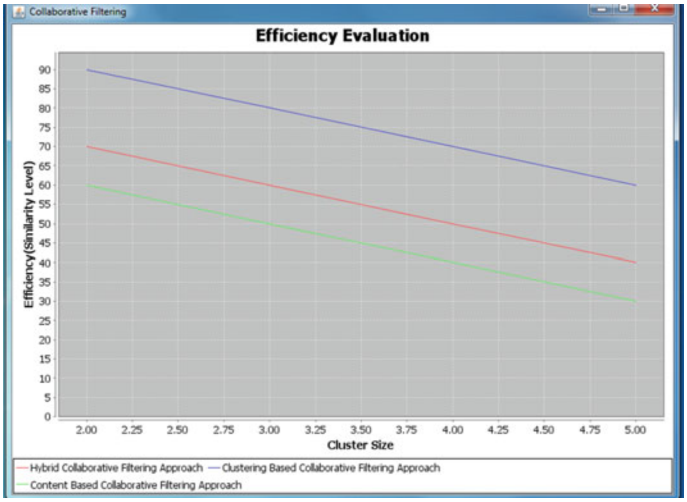


Fig. 5 Efficiency evaluation

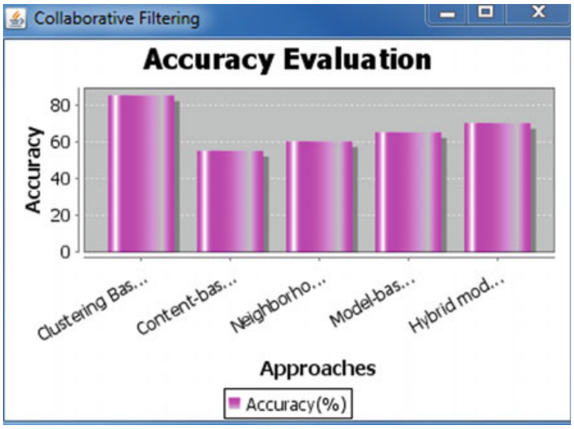


Fig. 6 Accuracy evaluation

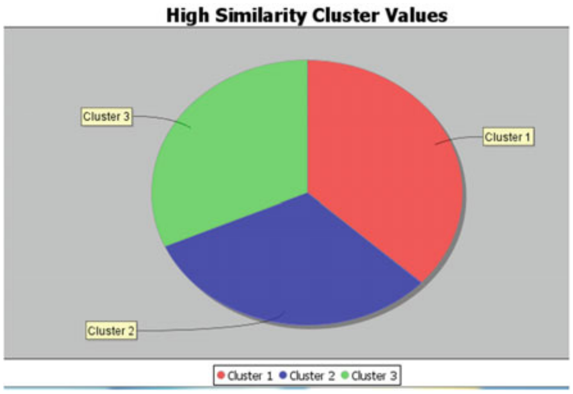


Fig. 7 High similarity cluster values

as compared to content, neighborhood, model and hybrid based approaches. Figure 7 shows the high similarity cluster values for three different clusters.

5 Conclusion

The CF approach is used in various recommendation systems, which has been proved to be one of the most successful methods in recommender systems. Services are merged into some clusters via an AHC algorithm before applying CF technique. Then the rating similarities between services within the same cluster are calculated. As the number of services in a cluster is much less than that of the whole system, costs of computation time is low. Also, as the ratings of services in the same cluster are related with each other, prediction based on the ratings of the services in the same cluster will be accurate than based on the ratings of all similar or different services in all clusters. These two advantages have been verified by experiments.

References

1. Wu, X., Wu, G., Zhu, X.: Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**(1), 97–107 (January 2014)
2. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. University Press of Cambridge
3. Bellogín, A., Díez, F., Cantador, I.: An empirical comparison of social, collaborative filtering (CF), hybrid recommender. ACM Trans. Intell. Syst. Tech. **4**(1), 1–37 (January 2013)
4. Zeng, W., Zhang, Q., Shang, M.: Can dissimilar users contribute to accuracy and diversity of personalized recommendation? IJMPC **21**(10), 1217–1227 (June 2010)
5. Havens, T.C., Hall, L.O., Leckie, C., Palaniswami, M.: Fuzzy c-means algorithm for very large data. IEEE Trans. Fuzzy Syst. **20**(6), 1130–1146 (December 2012)
6. Liu, Z., Zheng, Y., Li, P.: Clustering to find exemplar terms for key-phrase extraction. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 257–266, May 2009
7. Rodriguez, A., Chaovalitwongse, W., Zhe, L.: Master defect record retrieval using network based feature ass. IEEE Trans. Syst. Man Cybern. App. Rev. **40**(3), 319–329 (October 2010)
8. Adomavicis, G., Zhang, J.: Stability of recommendation algorithms. ACM Trans. Inf. Syst. **30** (4), 23:1–23:31 (August 2012)
9. Liu, X., Mei, H., Huang, G.: Discovering homogeneous web services community in the user centric web environment. IEEE Trans. Serv. Comput. **2**(2), 167–181
10. Li, H.H., Tian, X., Du, X.Y.: A review based reputation evaluation approach for web services. Int. J. Comput. Sci. Tech. **249**(5), 893–900 (Sep 2009)
11. Zielinnski, K., Szydlo, T., Szymacha, R.: Adaptive soa solution stack. IEEE Trans. Serv. Comput. **5**(2), 149–163 (April-June 2012)
12. Shafer, J., Rixner, S.T., Cox, A.: The hadoop distributed file system (HDFS): balancing portability and performance. IEEE Int. Symp. Perform. Anal. Syst. S\W. doi: 10.1109/ISPASS.2010.5452045. pp. 122–133, 28–30 March 2010
13. Kirankumar, R., Vijayakumari, R., Gangadhara, R.K.: Comparative analysis of google file system and hadoop distributed file system. IJAT CSE. **3**(1), 24–25 (Feb 2014)
14. HDFS Guide [Online]. http://hadoop.apache.org/common/doc/current/hdfs_user_guide
15. Li, M.J., Cheung, Y., Ng, M.: Agglomerative fuzzy k means clustering algorithm with selection of number of clusters. IEEE Trans. Knowl. Data Eng. **20**(11), 1519–1534 (November 2008)
16. Zhao, Y., Fayyad, U., Karypis, G.: Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Discov. **10**(2), 141–168 (November 2005)
17. Platzer, C., Dustdar, S., Rosenberg, F.: Web service clustering using multi-dimensional angle as proximity measures. ACM Trans. Internet Tech. **9**(3), 11:1–11:26 (July 2009)
18. Taherian, T.F., Niknam, T., Pourjafarian, N.: An efficient algorithm based on modified imperialist competitive algorithm & K means for data clustering. Eng. App. Artif. Intell. **24**(2), 306–317 (March 2011)
19. Thilagavathi, G., Aparna, N., Srivaishnavi, D.: A survey on efficient hierarchical algorithm used in clustering. Int. J. Eng. **2**(9), 306–317 (Sep 2013)
20. Julie, D., Kumar, K.: Optimal web service selection scheme with dynamic QoS property assignment. IJART. **2**(2), 69–75 (May 2012)
21. Wu, J., Chen, L., Feng, Y.: Predicting quality of service for selection by neighborhood based collaborative filtering (CF). IEEE Trans. Syst. Man Cybern. Syst. **43**(2), 428–439 (March 2013)
22. Lens G MovieLens. Available [Online] http://grouplens.org/datasets/movielens.html