An Efficient Recommender System based on Collaborative Filtering

V. Anbarasu[1], X. Linda[2] and S. Mahalakshmi[3*]

[1] Associate Professor, Department of Information Technology,
Jeppiaar Engineering College, Chennai.

[2] Department of Information Technology,
Jeppiaar Engineering College, Chennai.

[3]Department of Information Technology
Jeppiaar Engineering College, Chennai.

Corresponding author's email: suganmaha2k10 [AT] gmail.com

ABSTRACT— ***In general, Big Data enterprise large-volume of complex, growing data sets with multiple, autonomous sources. The utmost underlying challenge for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions. In view of this challenge, we propose a method called Clustering based Collaborative Filtering approach. It consists of two stages: clustering and Collaborative Filtering. Clustering is an initial step to separate big data into manageable parts. A cluster contains some similar services. In the second stage, a Collaborative Filtering algorithm is applied on one of the clusters. As the number of services in a cluster is much less than the total number of services, the computation time of collaborative filtering algorithm can be reduced significantly. Besides, since the ratings of similar services within a cluster are more relevant than that of dissimilar services, the recommendation accuracy based on user ratings may be enhanced.***

Keywords— Clustering, Collaborative Filtering, Mashup.

1. INTRODUCTION

Generally big data is composed of huge volume of data and services. Nowadays end users encounter many difficulties in finding ideal services among the daunting services. Recommender systems (RSs) are techniques and intelligent applications which helps the users in a decision making process where they want to choose some items from the overwhelming set of alternative products or services. Collaborative Filtering (CF) such as item and user-based methods are the majestic techniques applied in Recommender systems. The item-based collaborative filtering algorithm recommends a user the items that are similar to what he/she has preferred before. Though traditional collaborative filtering techniques are good and have been successfully applied in many e-commerce RSs, they face two main challenges for big data application they are 1) to make decision within acceptable time and 2) to generate ideal recommendations from so many services. Imperatively, a critical step in traditional collaborative filtering algorithms is to compute similarity between every set of users or services which may take too much time, even exceed the processing capability of current RSs. As a result, service recommendation based on the similar users or similar services would either lose its timeliness or couldn't be done at all.

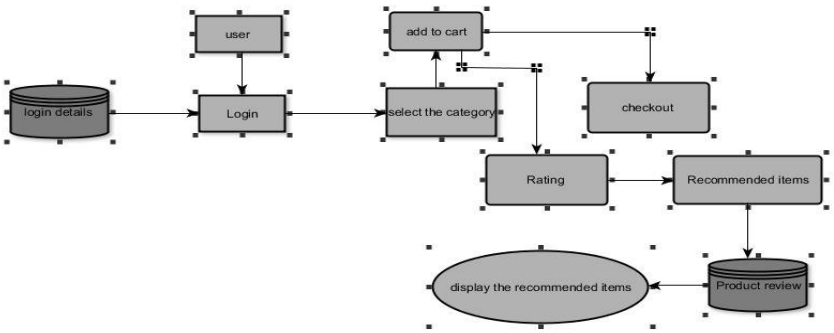

Figure 1: ARCHITECTURE of Recommender System

2. LITERATURE SURVEY

Clustering methods for collaborative filtering have been widely studied by some researchers. Hao Ma [1] deals with recommendation techniques. It has two stages, first it proposes a diffusion method that propagates similarities between different nodes and generates a recommend. Then it illustrates how to generate different recommendations on problem into our graph diffusion framework. Recommender systems are based on Collaborative Filtering .This technique automatically predicts the interest of an active user by collecting rating information from similar users or others. However in most of the cases, rating data are always unavailable since information on the web is less structured and more diverse. Sonia Ben [2] aims at defining a new user model called user semantic model, to perform user semantic preferences based on item features and user ratings. This model is built from the user item model by using fuzzy clustering algorithm the Fuzzy-C Mean (FCM) algorithm. This paper aims at combining two techniques they are Collaborative Filtering and content based filtering. However, these techniques must face many challenges like data sparsity problem due to missing data in the user item matrix. Z. Zheng [3] use Collaborative Filtering approach for predicting QOS values of web services. Song Jie [4] proposes a personalized recommendation approach that joins the user clustering technology to solve the problems like scalability and sparsity in the Collaborative Filtering. The recommendation joining user clustering and item clustering Collaborative Filtering is more scalable and more accurate than the traditional one. Guibing Guo [5] develops a multiview clustering method through which users are iteratively clustered from the views of both rating patterns and social trust relationships. However, a critical drawback is that the newly-issued ratings cannot be quickly involved for predictions.

3. USAGE OF BIGTABLE IN CLUB COLLABORATIVE FILTERING

A Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. It is used for storage purpose in club collaborative filtering. It is capable of storing big data in distributed and scalable manner. First the characteristic similarities among the services are calculated by weighted sum of description similarities and functionality similarities. Then the services are blended into the clusters based on their characteristic similarities. Next, an item based collaborative filtering algorithm is applied within the cluster where the target service belongs to. Bigtable is also called as service Bigtable because it stores all services. A service Bigtable is defined as a table expressed in the format of

<*Service_ID*> <*Timestamp*> {<*Description*>: [<*d*1>,<*d*2>,…]; <*Functionality*>: <f1>,<f2>,… ; <*Rating*>: <*u*1>,<*u*2>,… }.The elements in the expression are as follows:

1. $Service\_ID$ is the row key for uniquely identifying a service.
2. $Timestamp$ is used to identify time when the record is written in service Bigtable.
3. $Description$, $Functionality$ and $Rating$ are three column families.

Row key	Rating	Description	Functionality
S_1	U_1=5 → t_6 U_2=4 → t_5	d_1="driving" → t_9	f_1= "Google maps" ↓ t_8

Table 1: STRUCTURE of bigtable

4. ALGORITHMS PROPOSED

4.1. Clustering Stage

4.1.1 Stem words

In the collaborative filtering approach, the words in D_t and D_j are gotten from service bigtable which are stemmed by Porter Stemmer and put into D_t' and D_j'.

4.1.2 Compute Description Similarity and Functionality Similarity

Description similarity and functionality similarity are both computed by Jaccard similarity coefficient (JSC)

$$D\_sim(s_t, s_j) = \frac{|D_t' \cap D_j'|}{|D_t' \cup D_j'|} \quad (1) \qquad F\_sim(s_t, s_j) = \frac{|F_t \cap F_j|}{|F_t \cup F_j|} \quad (2)$$

4.1.3 Compute Characteristic Similarity

Characteristic similarity is computed by weighted sum of description similarity and functionality similarity, where $\alpha, \beta \in [0,1]$ which is computed as follow:

$$C\_sim(s_t, s_j) = \alpha \times D\_sim(s_t, s_j) + \beta \times F\_sim(s_t, s_j) \quad (3)$$

4.1.4 Cluster Services

Clustering methods partition a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria. Clustering algorithms can be either hierarchical or partitioned.

4.2. Deployment of Collaborative Filtering Stage

4.2.1 Compute Rating Similarity

(Pearson Correlation Coefficient)PCC-based rating similarity between s_t and s_j is computed by formula (4):

$$R\_sim\ (s_t, s_j) = \frac{\sum_{u_i \in U_t \cap U_j} \left(r_{u_i,s_t} - \bar{r}_{s_t}\right)\left(r_{u_i,s_j} - \bar{r}_{s_j}\right)}{\sqrt{\sum_{u_i \in U_t \cap U_j} \left(r_{u_i,s_t} - \bar{r}_{s_t}\right)^2} \sqrt{\sum_{u_i \in U_t \cap U_j} \left(r_{u_i,s_j} - \bar{r}_{s_j}\right)^2}} \quad \textbf{(4)}$$

the enhanced rating similarity between st and sj is computed by formula (5):

$$R\_sim'\ (s_t, s_j) = 2 \times \frac{|U_t \cap U_j|}{|U_t| + |U_j|} \times R\_sim(s_t, s_j) \quad \textbf{(5)}$$

4.2.2 Select Neighbors

Based on the enhanced rating similarities between services, the neighbors of a target service st are determined according to constraint formula (6):

$$(s_t) = \{s_j |\ R\_sim'\ (s_t, s_j) > \gamma\ ,\ s_t \neq s_j\} \quad \textbf{(6)}$$

4.2.3 Compute Predicted Rating

The predicted rating $(u_{a,t})$ in an item-based collaborative filtering is computed as follow:

$$Pu_{a,t} = \bar{r}_{s_t} + \frac{\sum_{s_j \in N(s_t)} \left(r_{u_a,s_j} - \bar{r}_{s_j}\right) \times R\_sim'(s_t, s_j)}{\sum_{s_j \in N(s_t)} R\_sim'(s_t, s_j)} \quad \textbf{(7)}$$

5. COMPARISON AND TEST RESULTS

5.1 Experimental Background

To verify collaborative filtering, a mashup dataset is used in the experiments. Mashup is an ad hoc composition technology of Web applications that allows users to draw upon content retrieved. Mashup provides a flexible and easy-of-use way for service composition on web. The data for experimental testing was collected from ProgrammableWeb which is built around user-generated mashup service. This extracted data produces datasets. It includes mashup service name, tags and APIs used.

No	Name	APIs(Fi)	Tags(Di)	Stemmed Tags(Di')
S1	4WheelzRouteMate	Google Maps	driving,google,map,streetview	drive,google,map,streetview
S2	GuruLib	Amazon Product Advertising	books,library,videos	book,library,video
S3	100 Destinations	Google Maps+Twitter	fun,mapping,photo,social,travel	fun,map,photo,social,travel
S4	Anuncios Total	Google Maps+Twitter	ads,deadpool,shopping	ads,deadpool,shop
S5	22books	Amazon Product Advertising	books,list,shopping,social	book,list,shop,social
S6	Favmvs	Google Search+MTV	deadpool,MTV,music,video	deadpool,MTV,music,video
S7	FlickrCash	Flickr	photos,shopping	photo,shop

Table 2: INPUT DATA of Mashup Services

5.2 Experimental Case Study

According to recommender system, the experimental process is promoted by two stages: Clustering stage and Collaborative Filtering stage. Steps are carried as specified in part 3.

5.3 Experimental Evaluation

To evaluate the accuracy of collaborative filtering, the measure of the deviation of recommendations called Mean Absolute Error (MAE) is used. The collaborative filtering is a revised version of traditional item-based collaborative filtering approach. Therefore we compare the MAE of collaborative filtering with traditional item-based collaborative filtering approach (IbCF) to check its accuracy. The value of *K*, which is the third input parameter of Algorithm 1, is set

to 3, 4, 5, and 6, respectively. Furthermore, rating similarity threshold γ is set to 0.1, 0.2, 0.3 and 0.4. Under these parameter conditions, the predicted ratings of test services are calculated by collaborative filtering and Item based collaborative filtering. Then the average MAEs of Collaborative Filtering and Item based collaborative filtering can be computed using formula (8).

$$MAE = \frac{\sum_{i=1}^{n} | r_{a,t} - P(u_a, s_t) |}{n} \quad \textbf{(8)}$$

The comparison results are shown in Fig. 2 (a), (b), (c) and (d), respectively. There are several discoveries as follows.

- In fig 2(a), the$\gamma < 0.4$, the MAE of Collaborative Filtering decreases as K increases. The threshold value plays no role in Item based collaborative filtering.
- In fig 2(b), When γ<0.4, MAE values of Collaborative Filtering and Item based collaborative filtering both decrease as the value of increases.
- In fig 2(c) When γ<0.4, MAE values of Collaborative Filtering are lower than Item based collaborative filtering. Consequently, the predicted ratings of the target services will be more precise than that of IbCF.
- In fig 2(d) While γ=0.4, MAE values of Collaborative Filtering and item based collaborative filtering both increase. When k=5 and k=6, MAE values of Collaborative Filtering are even more than that of IbCF.

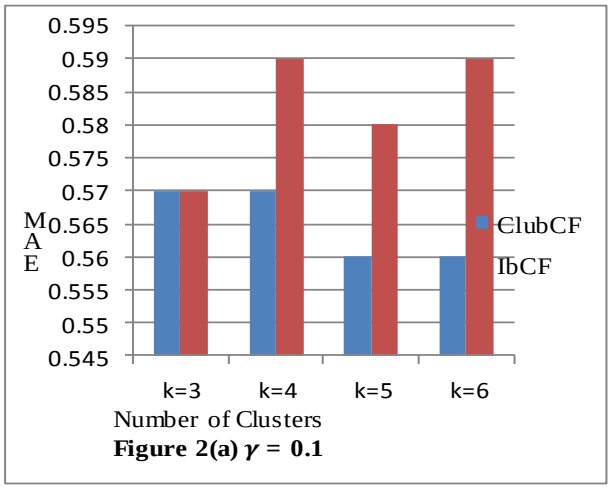

Figure 2(a) γ = 0.1

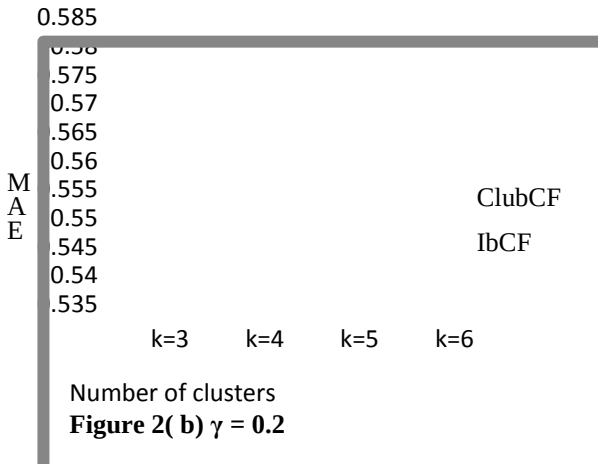

Figure 2(b) γ = 0.2

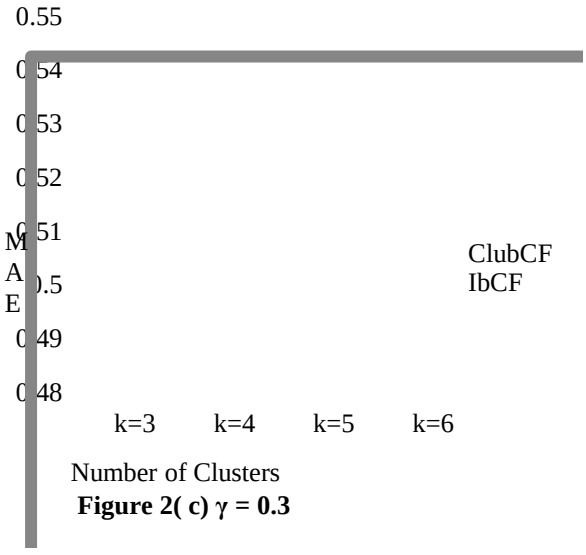

Figure 2(c) γ = 0.3

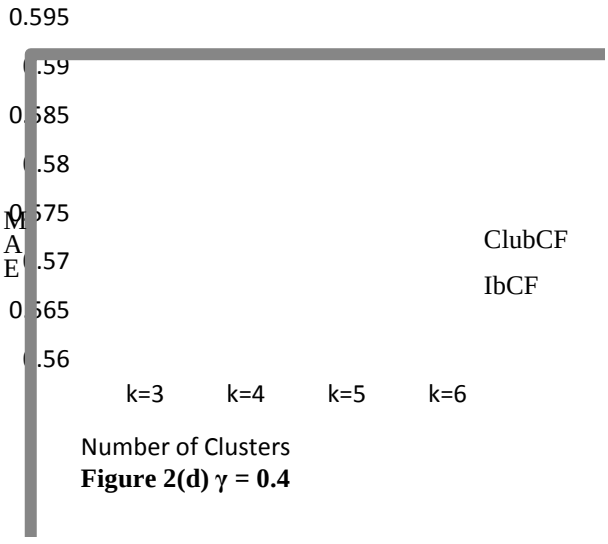

Figure 2(d) γ = 0.4

Additionally, to evaluate the efficiency of Collaborative Filtering, the online computation time of Collaborative Filtering is compared with that of Item based collaborative filtering, as shown in Fig. 3 (a), (b), (c) and (d). There are several discoveries as follows.

- In all, Collaborative Filtering spends less computation time than Item-based collaborative filtering. Since the number of services in a cluster is less, the computation time is also reduced.

- As the rating similarity threshold γ increase, the computation time of Collaborative Filtering decrease. However, only when γ=0.4, the decrease of computation time of Item based collaborative filtering is visible.
- When γ=0.4, as *K* increase, the computation time of Collaborative Filtering decrease obviously.

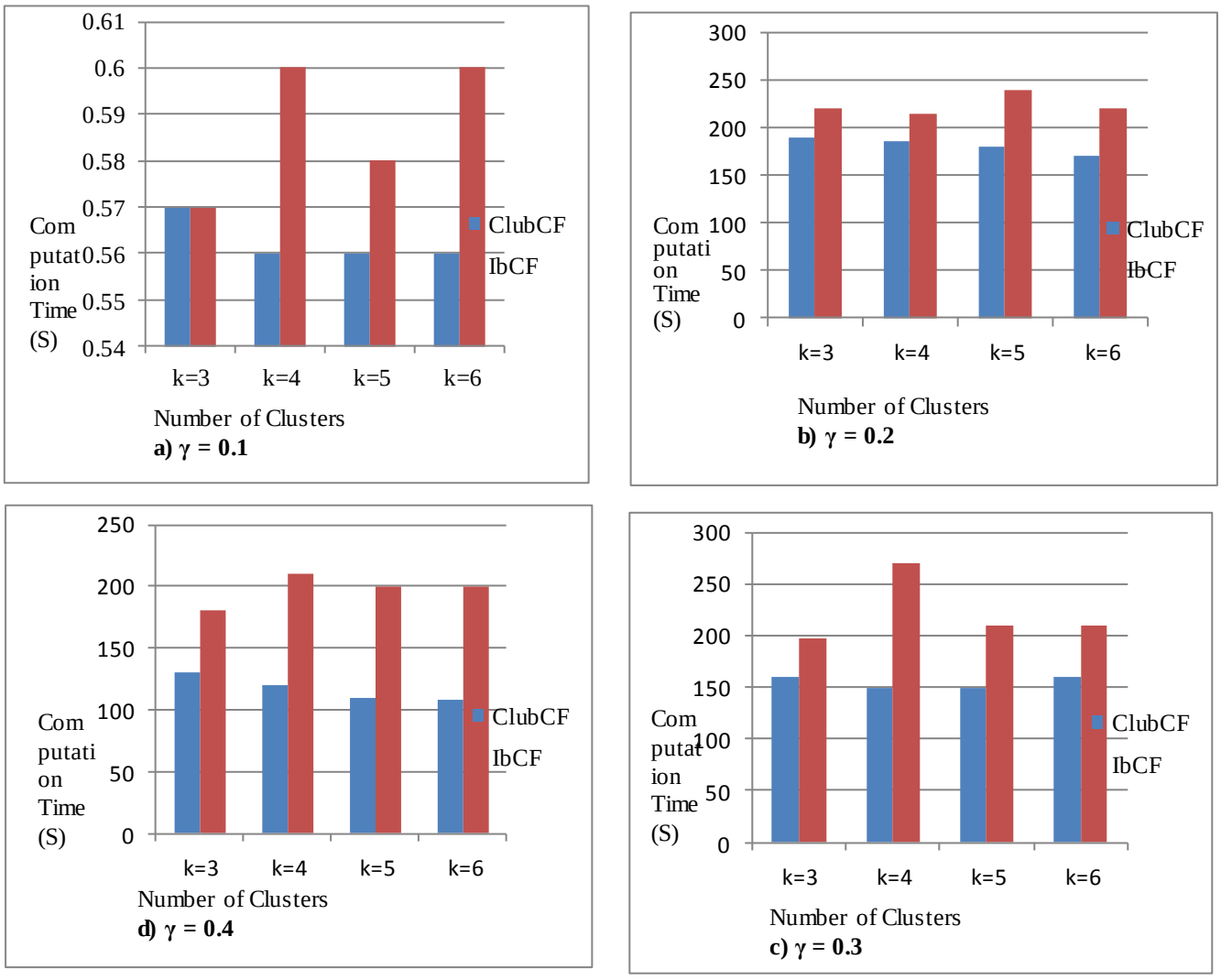

Figure 3: COMPARISON of Computation Time with Collaborative Filtering and Item based collaborative filtering

According to the computation analysis, we come to know that Collaborative Filtering may gain good scalability via increase the parameter *K* appropriately. Along with adjustment of γ, recommendation precision is also improved.

6. CONCLUSION

In this paper, we proposed a Collaborative Filtering approach for big data applications relevant to service recommendation. First the services are grouped into clusters using AHC algorithm then collaborative filtering is applied on the clusters so that the ratings for similar services within the same cluster are computed. The main benefit of Collaborative Filtering is to reduce the cost of online computation time because the number of services in a cluster is much less than that of in the whole system. The ratings of services in the same cluster are more relevant with each other than the one in other clusters; also the prediction based on the ratings of the services in the same cluster will be more accurate than the ratings of similar or dissimilar services in all clusters.

7. FUTURE WORK

Future research can be done in two areas. First, it is done with the respect to service similarity; here semantic analysis may be performed on the description text of service. In this way, more semantic-similar services may be clustered together, which will increase the coverage of recommendations. Second, with respect to users, by mining their implicit

interests from usage records or reviews may be a complement to the explicit interests (ratings). Hence these, recommendations can be generated even if there are only few ratings. This will solve the sparsity problem to some extent.

8. REFERENCES

[1] Hao Ma, Irwin King, Senior Member, IEEE, and Michael Rung-Tsong Lyu, Fellow, IEEE "**Mining Web Graphs for Recommendations"** IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 6, JUNE 2012.
[2]Sonia Ben Tunisia & Nancy **"User Semantic Model for Hybrid Recommender Systems"** Boyer KIWI Team, LORIA laboratory.
[3] Z. Zheng, H. Ma, M. R. Lyu, et al., **"QoS-aware Web service recommendation by Collaborative Filtering,"** IEEE Trans. on Services Computing, vol. 4, no. 2, pp. 140-152, February 2011.
[4] Song Jie Gong Zhejiang "**A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering**" Business Technology Institute, Ningbo 315012, China
[5] Guibing Guo, Jie Zhang "**Leveraging Multiviews of Trust and Similarity to Enhance Clustering-based Recommender Systems"** Neil Yorke-Smith_School of Computer Engineering, Nanyang Technological University, Lebanon; and University of Cambridge, UK fgguo1.