International Journal of Technology and Engineering System (IJTES)
Vol 7. No.4 2015 Pp. 342-348
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

RECOMMENDER SYSTEM WITH COLLABORATIVE FILTERING APPROACH FOR BIGDATA APPLICATION

BHARATHI S
PG Scholar, Department of IT,
M.A.M College of Engineering Anna University, Trichy.
bharathikalai153 @gmail.com

ABSTRACT

Growth of services on the internet is increasing day to day as a result services related data become too large to process by traditional data processing systems. To address this problem Clustering based collaborative filtering approach is proposed it deals with recruiting similar services in the same clusters to recommend services collaboratively. The approach possess two stages in first stage the services are divided into small scale clusters, in second stage the collaborative filtering algorithm is imposed over one of the clusters. Thus the total number of services in the cluster is much less than the total number of services available on the web. It is expected to reduce the online execution time of the collaborative filtering.

Key Terms: Big Data , Clustering, Collaborative filtering, Mash up

I.INTRODUCTION

BigData is a massive set of complex data an obviously it is difficult to processed by traditional data processing system [1]. Bigdata applications where data collection has growing tremendously and is beyond the ability of commonly used software tools to capture, manage and process within a tolerable elapsed time[2].The fundamental challenge of bigdata application is to extract useful information from the complex data[3].collaborative filtering (CF) approach posses two dominant techniques applied in Recommender systems such as user based CF and item based CF. user based CF is that people who agree in the past tend to agree in future. Item based CF algorithm recommends a service which is similar that what they preferred before [4]. The primary barriers of traditional CF techniques over Big Data applications are such as to make decision within acceptable time and to generate ideal recommendations from so many services. Remedy to this issue is to decrease the total number of services that need to be processed in real time. Thus clustering based collaborative filtering approach is proposed with two stages: clustering and collaborative filtering. Clustering is a method of separating the data which possess similar character (like minded user's) and dissimilar charactеritics data are kept in different cluster[5]. Rest of the paper speaks about such as follow.

In section II,BigTable is designed for storage requirement of ClubCF. section III, ClubCF algorithm is described briefly. In section IV experimental background.In section Analysis of related works. In section VI Conclusion and future work.

II PRELIMINARY KNOWLEDGE

Definition 1:
A service Bigtable is defined as a table expressed in the format of
< _ >< > {< >: [<d1>,<d2>,…]; < >: <f1>,<f2>,... ;
< >:<u1>,<u2>,... }
The elements in the expression are specified as follows:
1. _ is the row key for uniquely identifying a service.

2. is used to identify time when the record is written in service Bigtable.

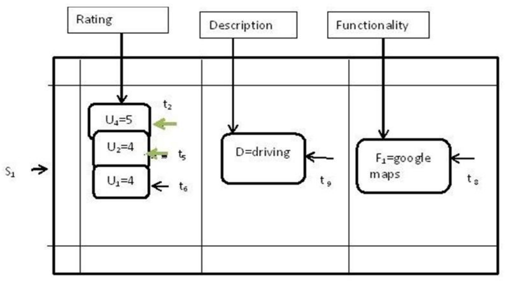
3. , and are three column families. 4. The identifier of a description word, e.g., $d1$ and $d2$, is used as a qualifier of *Description*.

5. The identifier of a functionality, e.g., $f1$ and $f2$, is used as a qualifier of *Functionality*.

6. The identifier of a user, e.g., $u1$ and $u2$, is used as a qualifier of *Rating*.A slice of service Bigtable is illustrated in TABLE I. The row key is s 1. The *Description* column family contains the words for describing s 1, e.g., "driving". The *Functionality* column family contains the service functionalities, e.g., "Google maps". And the *Rating* column family contains the ratings given by some users at different time, e.g., "4" is a rating that "u1" gave to "s1" at timestamp "t6".

III A CLUSTERING BASED COLLABORATIVE FILTERING APPROACH

PART OF BIGTABLE



According to Definition 1, a service could be expressed as a triple $s = D, F, R$, where D is a set of words for describing s, F is a set of functionalities of s, R is a set of ratings some users gave to s. Five kinds of service similarities are computed based on D, F and R during the process of ClubCF, which are defined as follow.

Definition 2: Suppose = , , and = , , are two services. The similarity between and is considered in five dimensions which are description similarity $D\_$, , functionality similarity $F\_$, , characteristic similarity $C\_$, , rating similarity $R\_$, and enhanced rating similarity $R\_$ ′ , , respectively.

With this assumption, a ClubCF approach for Big Data application is presented, which aims at recommending services from overwhelming candidates within an acceptable time. Technically, ClubCF focuses on two inter dependable stages, i.e., clustering stage and collaborative filtering stage. In the first stage, services are clustered according to their characteristic similarities. In the second stage, a collaborative filtering algorithm is applied within a cluster that a target service belongs to. Concretely,Table depicts the specification of the ClubCF approach step by step.

Step 1.1: Stem Words

Different developers may use differentform words to describe similar services. Using these words directly may influence the measurement of description similarity. Therefore, description words should be uniformed before further usage. In fact, morphological similar words are clubbed together under the assumption that they are also semantically similar. For example, „map‟ , „maps‟ , and „mapping‟ are forms of the equivalent lexeme, with „map ‟ as the morphological root form. To transform variant word forms to their common root called stem, various kinds of stemming algorithms, such as Lovins stemmer, Dawson Stemmer, Paice/Husk Stemmer, and Porter Stemmer, have been proposed [6]. Among them, Porter Stemmer (http://tartarus.org/martin/PorterStemmer/) is one of the most widely used stemming algorithms. It applies cascaded rewrite rules that can be run very quickly and do not require the use of a lexicon [7]. In ClubCF approach, the words in are gotten from service Bigtable where row key = " " and column family = " ". The words in are gotten from service Bigtable where row key = " " and column family = " ". Then these words are stemmed by Porter Stemmer and put into ′ and ′, respectively.

STEP 1.2: COMPUTE DESCRIPTION SIMILARITY AND FUNCTIONALITY SIMILARITY

Description similarity and functionality similarity are both computed by Jaccard similarity coefficient (JSC) which is a statistical measure of similarity between samples sets [8]. For two sets, JSC is defined as the cardinality of their intersection divided by the cardinality of their union. Concretely, description similarity between and is computed by formula (1):

$D\_$, = ′∩ ′ / ′∪ ′ (1) It can be inferred from this formula that the larger ′∩ ′ is, the more similar the two services are. Dividing by ′∪ ′ is the scaling factor which ensures that description similarity is between 0 and 1.

The functionalities in are gotten from service Bigtable where row key = " " and column family = " ". The functionalities in are gotten from service Bigtable where row key = " " and column family = " ". Then, functionality similarity between and is computed using JSC as follow: $F\_simst, = Ft\cap FjFt\cup Fj$ (2)

STEP 1.3: COMPUTE CHARACTERISTIC SIMILARITY

Characteristic similarity between st and sj is computed by weighted sum of description similarity and functionality similarity, which is computed as follow: $C\_simst, =\alpha\times D\_sim\ st, +\beta\times F\_simst,$ (3) In this 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 weights express relative importance between these two. Provided the number of services in the recommender system is n , characteristic similarities of every pair of services are calculated and form a $n\times n$ characteristic similarity matrix D . An entry dt, in D represents the characteristic similarity between st and sj.

STEP 1.4: CLUSTER SERVICES

Clustering is a critical step in our approach. 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.
Generally, cluster analysis algorithms have been utilized where the huge data are stored [9]. Clustering algorithms can be either hierarchical or partitional. Some standard partitional approaches (e.g., *K*-means) suffer from several limitations: 1) results depend strongly on the choice of number of clusters *K*, and the correct value of *K* is initially unknown; 2) cluster size is not monitored during execution of the *K*-means algorithm, some clusters may become empty ("collapse"), and this will cause premature termination of the algorithm; 3) algorithms converge to a local minimum [10].

Hierarchical clustering methods can be further classified into agglomerative or divisive, depending on whether the clustering hierarchy is formed in a bottom-up or top-down fashion. Many current state-of-the-art clustering systems exploit agglomerative hierarchical clustering (AHC) as their clustering strategy, due to its simple processing structure and acceptable level of performance. Furthermore, it does not require the number of clusters as input. Therefore, we use an AHC algorithm [11][12] clustering.

STEP 2.1: COMPUTE RATING SIMILARITY

Rating similarity computation between items is a time-consuming but critical step in item-based CF algorithms. Common rating similarity measures include the Pearson correlation coefficient (PCC) [13] and the cosine similarity between ratings vectors. The basic intuition behind PCC measure is to give a high similarity score for two items that tend to be rated the same by many users. PCC which is the preferred choice in most major systems was found to perform better than cosine vector similarity [14]. Therefore, PCC is applied to compute rating similarity between each pair of services in ClubCF. Provided that service st and sj are both belong to the same cluster, PCCbased rating similarity [15] between st and sj is computed by formula (4):
$R\_simst,sj = rui,st-rstrui,sj-rsj$
$ui\in Ut\cap Ujrui,t-rst\ 2ui\in Ut\cap Ujrui,sj-rsj$

$2ui\in Ut\cap Uj$ (4) Here, Ut is a set of users who rated st while Uj is a set of users who rated sj, ui is a user who both rated st and sj, rui,st is the rating of st given by ui which is gotten from service Bigtable where row key = "st" and column key = "$Rating{:}ui$", rui,sj is the rating of sj given by ui which is gotten from service
Bigtable where row key = "sj" and column key
= "$Rating{:}ui$", rst is the average rating of st, and rsj is the average rating of sj. It should be noted that if the denominator of formula (4) is zero, we make 0, in order to avoid division by 0. Although PCC can provide accurate similarity computation, it may overestimate the rating similarities when there are a small amount of corated services. To address this problem, the enhanced rating similarity [16] between st and sj is computed by formula (5): $R\_sim'\ st,sj$ =2 $\times\ Ut\cap UjUt+Uj\times R\_sim(st,sj)$ (5) In this formula, $Ut\cap Uj$ is the number of users who rated both service st and sj , Ut and Uj are the number of users who rated service st and sj, respectively. When the number of co-rated services is small, for example, the weight $2\times\ Ut\cap UjUt+Uj$ will decrease the rating similarity estimation between these two users. Since the value of $2\times\ Ut\cap UjUt+Uj$ is between the interval of [0,1] and the value of
R_simst,i is in the interval of [-1,1], the value of $R\_\ '\ ,$ is also in the interval of [-1,1].

IV EXPERIMENTS AND EVALUATION

Experimental Background

To verify ClubCF, 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 from external data sources to create value-added services [17]. Compared to traditional "developer-centric" composition technologies, e.g., BPEL (Business Process Execution Language) and WSCI (Web Service Choreography Interface), mashup provides a flexible and easy-of-use way for service composition on web [18]. Recently, "mashup" has become one of the hottest buzzwords in the area of web applications, and many companies and institutions provide various mashup solutions or re-label existing integration solutions as mashup tools. For eg.,Housing Maps (http://www.housingmaps.com) combines property listings from Craigslist (http://www.craigslist.org/) with map data from Google Maps (http://maps.google.com/) in order to assist people moving from one city to another and searching for housing offers. More interesting mashup services include Zillow (http://www.zillow.com/) and SkiBonk (http://www.skibonk.com/).

Manual mashup development requires programming skills and remains an intricate and time consuming task, which prevents the average user from programming own mashup services. To enable even inexperienced end users to mashup their web services, many mashup-specific development tools and frameworks have emerged [29]. The representative approaches of end user mashup tools include Google Mashup Editor, Yahoo Pipes, Microsoft Popfly, Intel Mash Maker, and IBM‟s QEDWiki [19]. These tools speed up the overall mashup development process, resulting in an explosion in the amount of mashup services available on the Internet. Meanwhile, a large number of mashup services are similar to each other, in their components and in the logic [20]. Over mashup-oriented big data, ClubCF is a suitable approach for recommending ideal mashup services for users.

The data for our experiments was collected from Programmable Web, a popular online community built around user-generated mashup services. It provides the most characteristic collection [21]. The extracted data was used to produce datasets for the population of mashup services. The dataset included mashup service name, tags, and APIs used. As of Dec 2012, 6,225 mashup services and related information are crawled from this site, which are labeled with 20,936 tags among which 1,822 tags are different.

And, 15,450 APIs are used by these mashup services among which 1,499 APIs are different in name. The tags are stemmed using Porter Stemmer algorithm and 1,608 different stems of tags are obtained. Since there are very few ratings available by now, we generate pseudorandom integers in the range 0 to 5 as the ratings of mashup services. Assume there are 500 users that have rated some mashup services published on the website. Then the user-item matrix consists of 500 rows and 6,225 columns. In total, 50,000 non-zero ratings are generated. The sparsity level of the matrix is 98.39% (sparsity level =1-50000/500*6226=0.9839). We add an empirical evaluation based on a well known statistical test, namely the l-fold cross validation [22]. The ratings records are split into l mutually exclusive subsets (the folds) of equal size. During each step, it is tested on fold and trained on the rest. The cross-validation process is then repeated l times, with each of the l subsets used exactly once as the validation data. In this paper, 5-fold cross validation is applied (i.e., l=5). In order to distribute test data and training data over all clusters, 20% services of each cluster was included in test data and 80% of it was included in training data for each data split.

V RELATED WORK

Clustering methods for CF have been extensively studied by some researchers. Mai et al. [23] designed a neural networks-based clustering collaborative filtering algorithm in ecommerce recommendation system. The cluster analysis gathers users with similar characteristics according to the web visiting message data. However, it is hard to say that a user's preference on web visiting is relevant to preference on purchasing. Mittal et al. [24] proposed to achieve the predictions for a user by first minimizing the size of item set the user needed to explore. K-means clustering algorithm was applied to partition movies based on the genre requested by the user. However, it requires users to provide some extra information. Li et al. [25] proposed to incorporate multidimensional clustering into a collaborative filtering recommendation model. Background data in the form of user and item profiles was collected and clustered using the proposed algorithm in the first stage. Then the poor clusters with similar features were deleted while the appropriate clusters were further selected based on cluster pruning. At the third stage, an item prediction was made by performing a weighted average of deviations from the neighbor's mean. Such an approach was likely to trade-off on increasing the diversity of recommendations while maintaining the accuracy of recommendations. Zhou et al. [26] represented Data-Providing (DP) service in terms of vectors by considering the composite relation between input, output, and semantic relations between them. The vectors were clustered using a refined fuzzy Cmeans algorithm. Through merging similar services into a same cluster, the capability of service search engine was improved significantly, especially in large Internet-based service repositories. However, in this approach, it is assumed that domain ontology exists for facilitating semantic interoperability. Besides, this approach is not suitable for some services which are lack of parameters. Pham et al. [27] proposed to use network clustering technique on social network of users to identify their neighborhood, and then use the traditional CF algorithms to generate the recommendations.

This work depends on social relationships between users. Simon et al. [28] used a highdimensional parameter-free, divisive hierarchical clustering algorithm that requires only implicit feedback on past user purchases to discover the relationships within the users. Based on the clustering results, products of high interest were recommended to the users. However, implicit feedback does not always provide sure information about the user‟ s preference. In ClubCF approach, the description and functionality information is considered as metadata to measure the characteristic similarities between services. According to such similarities, all services are merged into smallersize clusters. Then CF algorithm is applied on the services within the same cluster. Compared with the above approaches, this approach does not require extra inputs of users and suits different types of services. Moreover, the clustering algorithm used in ClubCF need not consider the dependence of nodes.

VI CONCLUSION AND FUTURE

In this paper, we present a ClubCF approach for big data applications relevant to service recommendation. Before applying CF technique, services are merged into some clusters via an AHC algorithm. Then the rating similarities between services within the same cluster are computed. As the number of services in a cluster is much less than that of in the whole system, ClubCF costs less online computation time. Moreover, as the ratings of services in the same cluster are more relevant with each other than with the ones in other clusters, prediction based on the ratings of the services in the same cluster will be more accurate than based on the ratings of all similar or dissimilar services in all clusters. These two advantageous of ClubCF have been verified by experiments on real-world data set. Future research can be done in two areas. First, in the respect of service similarity, 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, mining their implicit interests from usage records or reviews may be a complement to the explicit interests (ratings). By this means, recommendations can be generated even if there are only few ratings. This will solve the sparsity problem to some extent.

VI REFERENCES

[1] M. A. Beyer and D. Laney, “The importance of ‟ big data‟ : A definition,” Gartner, Tech. Rep., 2012.

[2] X. Wu, X. Zhu, G. Q. Wu, et al., “Data mining with big data,” IEEE Trans. on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, January 2014.

[3] A. Rajaraman and J. D. Ullman, “Mining of massive datasets,” Cambridge University Press, 2012.

[4] W. Zeng, M. S. Shang, Q. M. Zhang, et al., “Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?,” *International Journal of Modern Physics C*, vol. 21, no. 10, pp. 12171227, June 2010.

[5] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, “Fuzzy c-Means Algorithms for Very Large Data,” *IEEE Trans. on Fuzzy Systems*, vol. 20, no. 6, pp. 1130-1146, December 2012.

[6] R. S. Sandeep, C. Vinay, S. M. Hemant, “Strength and Accuracy Analysis of Affix Removal Stemming Algorithms,” *International Journal of Computer Science and Information Technologies*, vol. 4, no. 2, pp. 265-269, April 2013.

[7] V. Gupta, G. S. Lehal, “A Survey of Common Stemming Techniques and Existing Stemmers for Indian Languages,” *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 2, pp. 157-161, May 2013.

[8] A. Rodriguez, W. A. Chaovalitwongse, L. Zhe L, et al., “Master defect record retrieval using network-based feature association,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 3, pp. 319-329, October 2010.

[9] T. Niknam, E. TaherianFard, N. Pourjafarian, et al., “An efficient algorithm based on modified imperialist competitive algorithm and K-means for data clustering,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 306-317, March 2011.

[10] M. J. Li, M. K. Ng, Y. M. Cheung, et al. “Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters,”

IEEE Trans. on Knowledge and Data Engineering, vol. 20, no. 11, pp. 1519-1534, November 2008.

[11] G. Thilagavathi, D. Srivaishnavi, N. Aparna, et al., "A Survey on Efficient Hierarchical Algorithm used in Clustering," *International Journal of Engineering*, vol. 2, no. 9, September 2013.

[12] C. Platzer, F. Rosenberg, and S. Dustdar, "Web service clustering using multidimensional angles as proximity measures," *ACM Trans. on Internet Technology*, vol. 9, no. 3, pp. 11:111:26, July, 2009.

[13] G. Adomavicius, and J. Zhang, "Stability of Recommendation Algorithms," *ACM Trans. on Information Systems*, vol. 30, no. 4, pp. 23:123:31, August 2012.

[14] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information retrieval*, vol. 5, no. 4, pp. 287-310, October 2002.

[15] A. Yamashita, H. Kawamura, and K. Suzuki, "Adaptive Fusion Method for Userbased and Item-based Collaborative Filtering," *Advances in Complex Systems*, vol. 14, no. 2, pp. 133-149, May 2011.

[16] D. Julie, and K. A. Kumar, "Optimal Web Service Selection Scheme With Dynamic QoS Property Assignment," *International Journal of Advanced Research In Technology*, vol. 2, no. 2, pp. 69-75, May 2012.

[17] J. Wu, L. Chen, Y. Feng, et al., "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 428439, March 2013.

[18] M. R. Catherine, and E. B. Edwin, "A Survey on Recent Trends in Cloud Computing and its Application for Multimedia," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 1, pp. 304-309, January-February 2013.

[19] X. Liu, Y. Hui, W. Sun, et al., "Towards service composition based on mashup," in *Proc. of IEEE Congress on Services*, pp. 332339, July 2007.

[20] X. Z. Liu, G. Huang, Q. Zhao, et al., "iMashup: a mashup-based framework for service composition," *Science China Information Sciences*, vol. 57, no. 1, pp. 1-20, January 2013.

[21] H. Elmeleegy, A. Ivan, R. Akkiraju, et al., "Mashup advisor: A recommendation tool for mashup development," in *Proc. of IEEE Int'l Conf. on Web Services*, pp. 337-344, October 2008

[22] S. An, W. Liu, S. Venkatesh, et al., "Unified formulation of linear discriminant analysis methods and optimal parameter selection," *Pattern Recognition*, vol. 44, no. 2, pp. 307-319, February 2011.

[23] J. Mai, Y. Fan, and Y. Shen, "A Neural Networks-Based Clustering Collaborative Filtering Algorithm in E-Commerce Recommendation System," in *Proc. 2009 Int'l Conf. on Web Information Systems and Mining*, pp. 616-619, June 2009.

[24] N. Mittal, R. Nayak, M. C. Govil, et al., "Recommender System Framework using Clustering and Collaborative Filtering," in *Proc. 3rd Int'l Conf. on Emerging Trends in Engineering and Technology*, pp. 555-558, November 2010.

[25] X. Li, and T. Murata. "Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity," in *Proc. 2012 IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence and Intelligent Agent Technology*, pp. 169-174, December 2012.

[26] Z. Zhou, M. Sellami, W. Gaaloul, et al., "Data Providing Services Clustering and Management for Facilitating Service Discovery and Replacement," *IEEE Trans. on Automation Science and Engineering*, vol. 10, no. 4, pp. 1-16, October 2013.

[27] M. C. Pham, Y. Cao, R. Klamma, et al., "A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis," *Journal of Universal Computer Science*, vol. 17, no. 4, pp. 583604, April 2011.

[28] R. D. Simon, X. Tengke, and W. Shengrui, "Combining collaborative filtering and clustering for implicit recommender system," in *Proc. 2013 IEEE 27th Int'l Conf. on. Advanced Information Networking and Applications*, pp. 748-755, March 2013.