User Personalized Search on Big Data Application

Indumathi V
PG Scholar
Department of Computer science and engineering
Prathyusha Institute of Technology and Management
India
Email:

Meena R
Assistant Professor
Department of Computer science and Engineering
Prathyusha Institute of Technology and Management
India

Veerapandian N
Assistant Professor
Department of Computer science and Engineering
Prathyusha Institute of Technology and Management
India
Email:

***Abstract*--Nowadays everyone would have experienced the recommendation system on web. When we login to YouTube, Amazon or Flip kart, we are presented with a list of items recommended for us. Facebook provides us with a list of Friend recommendations. When we search on Google, it throws up a suggested search text. Success of different types of recommendation sites depends on the volume and quality of data available. Not like earlier days now we have Bigdata and hadoop which processing a large volume set of data. Inearlier days without this Bigdata, the recommendation systems are suffered a lot by scalability and inefficiency problem. In this paper we are concentrating about how to bring the recommendation system with accuracy and high scalable with help of Bigdata.**

***Keyword*--Bigdata,Hadoop,MapReduce,Recommendation, collaborative systems.**

I.INTRODUCTION

Big data is a buzzword, used to describe animmense volume of both structured and unstructured data that is so colossal that it's difficult to process using traditional database and softwaretechniques fig:1. The challenges include capture, cu-ration, storage, search, sharing, transfer, analysis and visualization. Big data sizes are constantly moving target as 2012 ranging from a few dozen terabytes to petabytes of data in a single data set.[1] The big data tendency also poses heavy impacts on service recommender systems. With the growing number of alternative services, effectively recommending services that users preferred have become an important research issue. Over the last decade there has been several research are gone through both in industry and academic a different approaches for service recommendation system[11].

II.MOST PRIMITIVE FACTS

A.Collaborative Filtering in Recommender Systems: **R**ecommender systems and collaborative
filtering became a topic of increasing interest among human–computer interaction, machinelearning, and information retrieval researchers. This interest produced a number of recommender systems for various domains, such as Ringo for music, the Bell Core Video Recommender for movies, and Jester for jokes. Outside of computer science, the marketing literature has analysed recommendation for its ability to increase sales and improve customer experience.

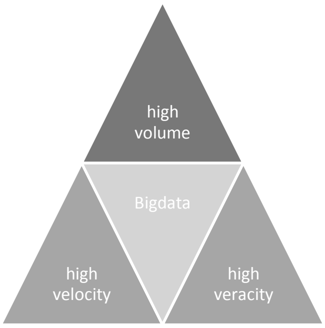

Fig 1:Big data as3V's

In the late 1990s, commercial deployments of recommender technology began to emerge. Perhaps the most widely-known application of recommender system technologies is Amazon.com. Recommendation systems use a number of different technologies. We can classify these systems into two broad groups.[5]

- Content-based systems examine properties of the items recommended.
- Collaborative filtering systems recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.

A pure content-based system has several shortcomings. Generally, only a very shallow analysis of certain kinds of content can be supplied. In some domains the items are not amenable to any useful feature extraction methods with current technology (such as movies, music, restaurants). A second problem, which has been studied extensively both in this domain and in others, is that of over-specialization. When the system can only recommend items scoring highly against a user's profile, the user is restricted to seeing items similar to those already rated.[5]
Pure collaborative recommendation solves all of the shortcomings given for pure content-based systems. By using other users' recommendations, we can deal with any kind of content and receive items with dissimilar content to

those seen in the past. Since other users' feedback influences what is recommended, there is the potential to maintain effective performance given fewer ratings from any individual user[6].The only one main problem in collaborative algorithm is scalability, it doesn't espouse large data sets.

B.Map Reduce in Bigdata:

Processing large volume of data in recent years are done by the cloud computing .The cloud computing is to share the resources such as infrastructure, platform, software and business process. Cloud computing is becoming a reality for many businesses, with private cloud deployments often leading the way. Organizations continue to store more and more data in cloud environments, which represent an immense, valuable source of information to mine. Plus, clouds offer business users scalable resources on demand in collaborating with Bigdata processing tools.

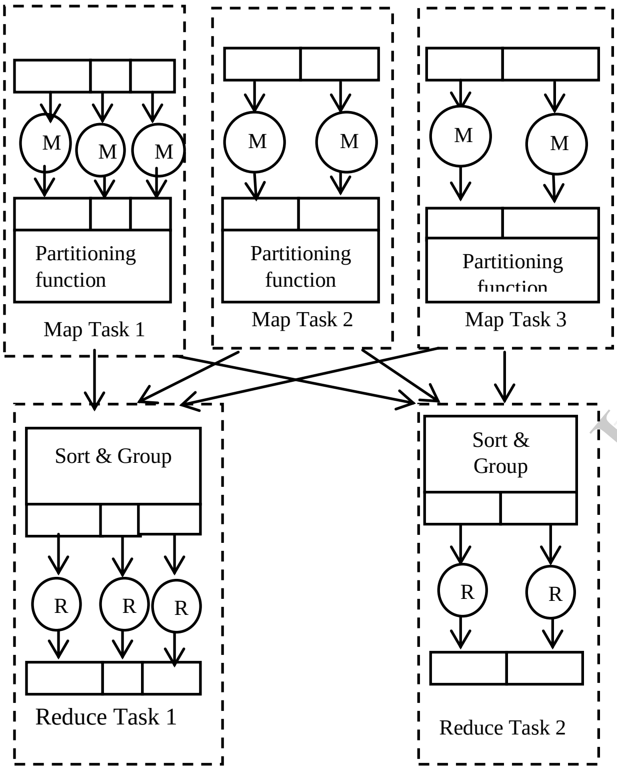

Fig 2: The implementation of map and reduce task.

There are several cloud computing tools available, such as Hadoop (http://hadoop.apache.org/), Mahout, the Dynamo of Amazon.com, the Dryad of Microsoft and Neptune of Ask.com. Among these tools, Hadoop is the most popular open source cloud computing platform inspired by MapReduce and Google File System papers which supports MapReduce programming framework.[12]

Map-Reduce is a programming model and associated implementation for processing and generating huge data sets.MapReduceframework has large number of cluster nodes, for each cluster node a single job tracker per master will be responsible for scheduling and monitors .The slave process and then re-executes the tasks when it fails and a single task tracker per slave will execute the task as directed by the masters.[2]

The MapReduce programming paradigm executes a job in two phases: *Map* and *Reduce*. In map step the master node takes large task as an input and sliced it into smaller sub task;distributes these to worker nodes. The worker node will do the same as master node and creates a multi-level tree structure. The worker processes minor task and hands back to master. In reduce step the master node takes the answer to the sub task and combines them in a predefined way to get output as (key-value) pair to the original task [7].

Key MapReduce Features:

- Scale-out Architecture - Add servers to increase processing power.
- Security & Authentication - Works with HDFS and HBase security to make sure that only approved users can operate against the data in the system
- Resource Manager - Employs data locality and server resources to determine optimal computing operations
- Optimized Scheduling - Completes jobs according to prioritization
- Flexibility – Procedures can be written in virtually any programming language
- Resiliency & High Availability - Multiple job and task trackers ensu.re that jobs fail independently and restart automatically[7].

This paper solves the scalability problem of collaborative filtering algorithm by implement that on MapReduce framework and making more accurate prediction in recommendation system in evolving the emotional aspects (arousal and valence) among the keywords.

III.USER PERSONALIZED SEARCH ON RECOMMENDATION SYSTEM METHOD (UPSR)

Definition 1: Data assortment &NLP process: Huge Collection of data are retrieved from open source datasets that are publicly available from major Travel Recommendation Applications. Big Data Schemas were analysed and a Working Rule of the Schema is determined. The CSV (Comma separated values) files were read and manipulated using Java API and implementing NLP over the Comma Separated Values files. A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb.

*Definition 2: Tagging and chunking files:*The CSV Files in distributed Systems are invoked through Web Service Running in the Server Machine of the Host Process through a Web Service Client Process in the Recommendation System. The data that Retrieved to the Recommendation Systems are provided with part of speech tagging and chunking process. Each and Every process on the Recommendation Application invokes Web Service which uses light weighted traversal of data using XML. The Users can Review each hotel and can post comments also. The Reviews gets updated to the CSV Files as it get retrieved.

*Definition 3: Service Recommender Application:*The Traditional View of Service Recommender Systems that shows Top-K Results are displayed with Paginations with which a user can navigate Back and Forth of the Result sets. All Services Ratings and Reviews of Each Hotels are listed. A User can Plan or Schedule a Travel highlighting his requirements in a detailed way that shows the Preference Keywords Set of the Active User. A Domain Thesaurus is built depending on the Keyword Candidate List and Candidate Services List. The Domain Thesaurus can be Updated Regularly to get accurate Results of the Recommendation System.

Definition 4: Map Reduce and Hadoop:

(1) Capture user preferences:

The preferences of active users and previous users are formalized into their corresponding preference keyword sets respectively. An active user refers to a current user needs recommendation.

a) Preferences of an active user.

An active user can give his/her preferences about candidate services by selecting keywords from a keyword-candidate list, which reflect the quality criteria of the services he/she is concerned about. Besides, the active user should also select the importance degree of the keywords. The importance degree of the keywords is shown in Table 3: “1” represents the general, “3” represents important and “5” represents very important.

b) Preferences of previous users.

The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword-candidate list and domain thesaurus. And a review of the previous user will be formalized into the preference key-word set of User.

(2) The keyword extraction process:

a) Pre-process:

Firstly, HTML tags and stop words in the reviews snippet collection should be removed to avoid affecting the quality of the keyword extraction in the next stage. And the Porter Stemmer algorithm is used to remove the commoner morphological and inflexional endings from words in English.

b) Keyword extraction:

Each review will be transformed into a corresponding keyword set according to the keyword-candidate list and domain thesaurus. If the review contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the preference keyword set of the user.

(3) Similarity computation:

The similarity computation is to identify the reviews of previous users who have similar tastes to an active user by finding neighbourhoods of the active user based on the similarity of their preferences [6]. Before similarity computation, the reviews unrelated to the active user's preferences will be filtered out by the intersection concept in set theory. If the intersection of the preference keyword sets of the active user and a previous user is an empty set, then the preference keyword set of the previous user will be filtered out.

a) Approximate similarity computation:

A frequently used method for comparing the similarity and diversity of sample sets, Jaccard coefficient, is applied in the approximate similarity computation.

$$sim(AUK, PUK) = jaccard(AUK, PUK) = \frac{|AUK \cap PUK|}{|AUK \cup PUK|}$$

b) Exact similarity computation

A cosine-based approach is applied in the exact similarity computation, which is similar to the Vector Space Model (VSM) in information
Retrieval

<u>Algorithm 1: Basic Algorithm of UPSR</u>

Input: The Active user’s preference keyword set (*AUK)*
The candidate services *WS*
Output: The Top-K highest ratings

```
for each service $WS$
    $\hat{R}$=Null, $sum$ =0, $r$= 0
    for each review $R$of service $WS$
        process the review into a preference keyword set $PUK$
        if $PUK \cap AUK$ = Null then
            insert $PUK$ into$\hat{R}$
        end if
    end for
    for each keyword set $PUK \in \hat{R}$
        $sim(AUK,PUK) = SIM(AUK,PUK)$
        if $sim(AUK,PUK)<$ thresholdthen
            remove $PUK$ from $\hat{R}$
        else $sum=sum+1$, $r=r+rj$
        end if
    end for
    $r = r / sum$
    get $pr$
    //*pr=personalized rating of service*//
end for
sort the services according to the personalized ratings $pr$
return the services with the Top-$K$ highest ratings.
```

Symbols	Definition
K	The keyword-candidate list, *K*={*k1*, *k2*, …,*kn*}
AUK	The preference keyword set of the active user.
PUK	The preference keyword set of a previous user
sim(AUK ,PUK)	The similarity between *AUK* and *PUK*
W_{AK}	The preference weight vector of active user
W_{PK}	The preference weight vector of previous user

Table 1: symbols and its definition in UPSR

IV.RELATED WORK

There have been many recommender systems developed in both academia and industry. In [3], the authors propose a Bayesian-inference-based recommendation system for on-line social networks. They show that the proposed Bayesian-inference-based recommendation is better than the existing trust-based recommendations and is comparable to Collaborative Filtering recommendation. In [4], Adomavicius and Tuzhilin give an overview of the field of recommender systems and describe the current generation of re-commendation methods. They also describe various limitations of current service recommendation methods, and dis-cuss possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. Most existing service recommender systems are only based on a single numerical rating to represent a service's utility as a whole [9]. In fact, evaluating a service through multiple criteria and taking into account of user feedback can help to make more effective recommendations for the users.

With the development of cloud computing software tools such as Apache Hadoop, Map-Reduce, and Mahout, it becomes possible to design and implement scalable recommender systems in "Big Data" environment. The authors of [6] implement a CF algorithm on Hadoop. They solve the scalability problem by dividing dataset. But their method doesn't have favourable scalability and efficiency if the amount of data grows. [7] Presents a parallel user profiling approach based on folksonomy information and implements a scalable recommender system by using Map-Reduce and Cascading techniques. Jin et al. [8] propose a large-scale video recommendation system based on an item-based CF algorithm. They implement their proposed approach in Qizmt, which is a .Net Map-Reduce framework, thus their system can work for large-scale video sites. Generally speaking, comparing with existing methods, UPSR utilizes reviews of previous users to get both of user preferences and the quality of multiple criteria of candidate services, which makes recommendations more accurate, and thereby implementing UPSR on MapReduce has favourable scalability and efficiency.

V.CONCLUSION

This paper proposed a user personalized search in recommendation system with Big data application. This paperwill elaborate how the key-words are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. More specifically, a keyword-candidate list and domain thesaurus are provided to help obtain users' preferences. The active user gives his/her preferences by selecting the keywords from the keyword-candidate list, and the preferences of the previous users can be extracted from their reviews for services according to the keyword-candidate list and domain thesaurus. The method aims at presenting a Top K rating listand recommending the most appropriate service(s) to the users. Moreover, to improve efficiency and to make highly scalable, "Big Data" environment is used here and that one is implemented it on a MapReduce framework in Hadoop platform.

REFERENCE

[1] J. Manyika, M. Chui, B. Brown, et al, "Big Data: The next frontier for innovation, competition, and productivity," 2011.

[2] J. Dean, and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, Vol. 51, No.1, pp. 107-113, 2005

[3] X. Yang, Y. Guo, Y. Liu, "Bayesian-inference based recommendation in online social networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 4, pp. 642-651, 2013.

[4] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State of- the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6 pp. 734-749, 2005.

[5] Greg Linden, Brent Smith, and Jeremy York "Item-to-Item Collaborative Filtering", amazon.com

[6] Z. D. Zhao, and M. S. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," In the third Internation-al Workshop on Knowledge Discovery and Data Mining, pp. 478-481, 2010.

[7] H. Liang, J. Hogan, and Y. Xu, "Parallel User Profiling Based on Folksonomy for Large Scaled Recommender Systems: An Impli-mentation of Cascading MapReduce," In Proceedings of the IEEE International Conference on Data Mining Workshops, pp. 156-161, 2010.

[8] Y. Jin, M. Hu, H. Singh, D. Rule, M. Berlyant, and Z. Xie, "MySpace Video Recommendation with Map-Reduce on Qizmt," Proceedings of the 2010 IEEE Fourth International Conference on Semantic Computing, pp.126-133, 2010.

[9] G. Adomavicius, and Y. Kwon, "New Recommendation Techniques for Multicriteria Rating Systems," IEEE Intelligent Systems, Vol. 22, No. 3, pp. 48-55, 2007

[10] Rafael Sotelo, Jose Joskowicz, Alberto Gil Solla,"An affordable and inclusive system to provide interesting contents to DTV using Recommender Systems", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting,2013.

[11] www.saa.com/en-us/insight/bigdata.

[12] http://hadoop.apache.org/.

[13]Suphakit, Jatsada, Ekkachai and Supachanun," Using of Jaccard Coefficient for Keywords Similarity", Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong.