International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

User Based Personalized Search with Big Data

M. C. Sabitha

Rajalakshmi Engineering College, Chennai, Anna University, India

Abstract: ***To develop a Keyword-Aware Service Recommendation method, named KASR, to address scalability and inefficiency problem in Big Data with traditional service recommender systems, which fails to meet users' personalized requirements and diverse Preferences. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences, and therefore fails to meet users' personalized requirements. Current recommendation methods usually can be classified into three main categories: content-based, collaborative, and hybrid recommendation approaches. Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly, yielding the big data analysis problem for service recommender systems. Consequently, traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analysing such large-scale data.***

Keywords: KASR, Big Data, ratings and rankings, personalized requirements, recommendation method.

1. Introduction

In recent years the amount of data in our world has been increasing explosively and analyzing large data sets so called Big Data. Big Data refers to datasets whose size is beyond the ability of current technology, method and theory to capture, manage, and process the data within a tolerable elapsed time.

Big Data management stands out as a challenge for IT companies. The solution to such a challenge is shifting increasingly from providing hardware to provisioning more manageable software solutions. Big Data also brings new opportunities and critical challenges.

In the last decade, the amount of customers, services and online information has grown rapidly yielding the big data analysis problem for service recommender systems. Consequently traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analyzing such large-scale data. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences, and therefore fails to meet users' personalized requirements.

- **Content-based Approaches** recommend services similar to those the user preferred in the past.
- **Collaborative filtering (CF) Approaches** recommend services to the user that users with similar tastes preferred in the past

In CF based systems users receive recommendations based on people who have similar tastes and preferences, which can be further classified into item-based CF and user-based CF.

Item-Based System: In item-based systems the predicted rating depends on the ratings of other similar items by the same user.

User-Based System: User based systems the prediction of the rating of an item for a user depends upon the ratings of the same item rated by similar users. And in this work, we will take advantage of a user-based CF algorithm to deal with our problem.

1.1 Cloud Computing

Cloud computing is a successful paradigm of service oriented computing and has revolutionized the way computing infrastructure is abstracted and used. The major goal of cloud computing is to share resources, such as infrastructure, platform, software, and business process.

Cloud computing can provide effective platforms to facilitate parallel computing, which has gained significant attention in recent years to process large volume of data. There are several cloud computing tools available such as Hadoop, Mahout, MapReduce of Google, the dynamo of Amazon, the dryad of Microsoft and Neptune of ask, etc.

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 and mass data storage with good fault tolerance. MapReduce is a popular distributed implementation model proposed by Google, which is inspired by map and reduce operations in the Lisp programming language.

Nowadays, the trend “everything as a service” has been creating a Big Services due to the foundational architecture of services computing. And “servicelization” is the way of offering social networking services, big data analytics, and Internet services. Thus the cloud computing tools aforementioned can be used to improve the scalability and efficiency of service recommendation methods in the “Big Data” environment.

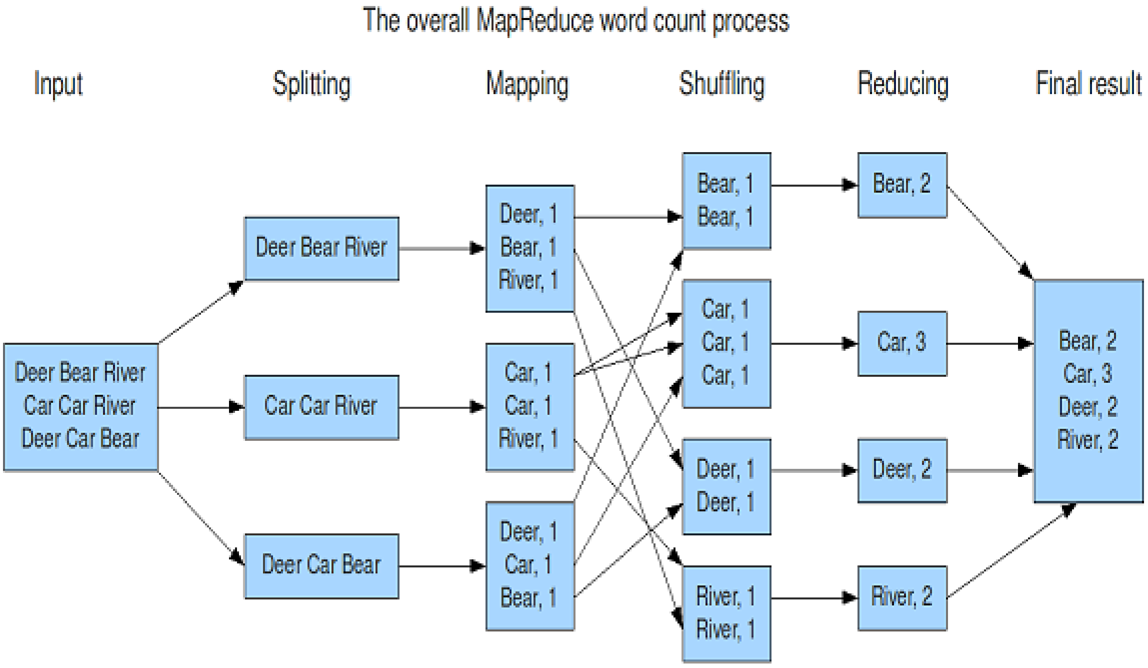

Figure 1: Map and Reduce Count Process

1.2 Map Reduce

Mapreduce work flows have to create two scripts map scripts, reduce scripts. The rest will be handled by Amazon Elastic MapReduce (EMR) framework. When we start a map/reduce workflow the framework will split the input into segments, passing each segment to different machine. Each machine then runs the map scripts on the portion of data attribute

The map scripts take some input data and maps it to key, value pair according to specification. The reduce script take the collection of key, value pair and reduces them according to the user specified reduced script

1.3 Hadoop

Hadoop distributed file system (HDFS) is based on Google's GFS (Google file system).It provides a redundant storage of massive amount of data. Data is distributed across all nodes at load time. Hadoop splits files into large blocks (default 64MB or 128MB) and distributes the blocks amongst the nodes in the cluster. To process the data, Hadoop Map/Reduce transfers code (specifically Jar files) to nodes that have the required data, which the nodes then process in parallel. This approach takes advantage of data locality[3] to allow the data to be processed faster and more efficiently via distributed processing than by using a more conventional supercomputer architecture that relies on a parallel file system where computation and data are connected via high-speed networking.

Hadoop Components:

- **HDFS**: Distributed File System
- **MAPREDUCE:** Distributed Data Processing Model
- **Hive:** Distributed Data Warehouse, provides SQL based query language
- **HBase:** Distributed Column based database
- **Pig:** Data Flow Language and execution environment

2. Literature Survey

The survey contains various information about a service recommendation list and recommending the most appropriate service to the user effectively.

Fay chang, et al [1] proposed a distributed, persistent multidimensional sorted map. The map is indexed by a row key, column key, and a timestamp each value in the map is an uninterrupted array of bytes. In Web table, we would use URLs as row keys, various aspects of web pages as column names, and store the contents of the web pages in the contents. The row keys in a table are arbitrary strings .Every read or write of data under a single row key is atomic, a design decision that makes it easier for clients to reason about the system's behavior in the presence of concurrent updates to the same row. Bigtable maintains data in lexicographic order by row key. The row range for a table is dynamically partitioned. Each row range is called a tablet, which is the unit of distribution and load balancing.

Rafael sotelo et al [2] presents a television scheduling system that aims to aid developing countries in digital television transition by providing content interesting for the people while reducing content acquisition costs. It is based on recommender systems for audiovisual content with special considerations for groups of viewers. The system proposed is fed with content uploaded by the people, properly classified. In a first and basic scenario the system generates four different thematic signals with predefined genres. In the second one, the profile for each user is known and audience is modeled as a group of viewers with known user profiles, allowing its segmentation and channel scheduling according to actual preferences.

Greg linden at el[3] recommendation algorithms are best known for their use on e-commerce Websites, where they use input about a customer's interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favourite artists. At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother. The click-through and conversion rates two important measures of Web-based and email advertising effectiveness vastly exceed those of untargeted content such as banner advertisements and top-seller lists.

Saranya et al [4] propose a personalized travel recommendation model considering users' attributes as well as their group types and the knowledge mined from travel logs .We investigate the association of people attributes such as time, popular landmarks, etc., We also recommend the nearby location suggestions in mobile using android.

Liliana ardissono et al [5] The PPG (personal program guide) offers advanced facilities for browsing TV content. Moreover, the user may ask for details about a program (e.g., cast, content description and parental rating), she can record it, ask to be advised when the transmission of the program starts (memo function), and so forth. The user can also retrieve the list of programs she has asked to be alerted about (Memo TV events), she has recorded (Recorded TV Events button), or she has bought (Bought TV Events). Although the system acquires the information about the user's interests in an unobtrusive way, it also accepts explicit feedback about programs that may be rated by clicking on the "thumb up/down" buttons located in the bottom-right area of the User Interface.

Faustino Sanchez et al [6] recommender system for sport videos, transmitted over the Internet and/or broadcast, in the context of large-scale events, which has been tested for the Olympic Games. The recommender is based on audiovisual consumption and does not depend on the number of users, running only on the client side. This avoids the concurrence, computation and privacy problems of central server approaches in scenarios with a large number of users, such as the Olympic Games. The system has been designed to take advantage of the information available in the videos, which is used along with the implicit information of the user and the modeling of his audiovisual content consumption. The system is thus transparent to the user, who does not need to take any specific action. Another important characteristic is that the system can produce recommendations for both live and recorded events. Testing has showed advantages compared to previous systems.

Zibin zheng et al [7] QoS rankings provide valuable information for making optimal cloud service selection from a set of functionally equivalent service candidates. To obtain QoS values, real-world invocations on the service candidates are usually required. To avoid the time-consuming and expensive real-world service invocations, this paper proposes a QoS ranking prediction framework for cloud services by taking advantage of the past service usage experiences of other consumers. Our proposed framework requires no additional invocations of cloud services when making QoS ranking prediction. Two personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly. Comprehensive experiments are conducted employing real world QoS data. Quality-of-service can be measured at the server side or at the client side. While server-side QoS properties provide good indications of the cloud service capacities, client-side QoS properties provide more realistic measurements of the user usage experience. The commonly used client-side QoS properties include response time, throughput, failure probability, etc.

3. Proposed Approach

The user can login to the process through the user name and password. The user name and password are entered in admin process. Then authentication process begin after login to the admin process then preprocess begin. In preprocess technique two servers are created and hotel details are given. By using this information taggering is done and output is been found. Then reviews of user is been collected and thus process is been analysed. Later the best choice is been recommended.

Huge Collection of data is retrieved from open source datasets that are publicly available from major Travel Recommendation Applications. Big Data Schemas were analyzed and a Working Rule of the Schema is determined. The CSV (Comma separated values) files were read and manipulated using Java API that itself developed by us which is developer friendly light weighted and easily modifiable.

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 a clean GUI and can be queried on Demand. 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 gets retrieved.

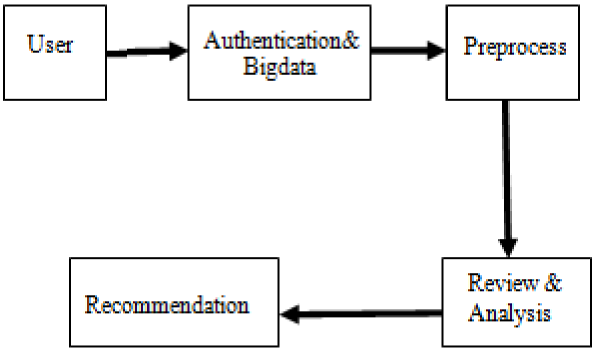

Figure 2: System Architecture

The Traditional View of Service Recommender Systems that shows Top-K Results are displayed ith 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.

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. 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. 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.

3.1 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 (keyword strip-ping) is used to remove the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems.

3.2 Keyword Extraction

In this phase, 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. For example, if a review of a previous user for a hotel has the word "spa", which is corresponding to the keyword Fitness in the domain thesaurus, then the keyword Fitness should be contained in the preference keyword set of the previous user. If a keyword appears more than once in a review, the times of repetitions will be recorded.

4. Implementation

To improve the scalability and efficiency of KASR in "Big Data" environment, we implement it in a MapReduce framework on Hadoop platform. Then the user login to process through the authorised user name and password. After login to process the authentication process begins. The pre-processor is been done. After pre-processor output is been taggered.

4.1 Experiment Setup and Datasets

Technically, our experiments are conducted in a Hadoop platform. And to evaluate the accuracy and scalability of KASR, two kinds of dataset are adopted in the experiments: a real dataset and a synthetic dataset. Due to the space limit, more details about the experiment settings and dataset can be found respectively.

4.2 Experiment Evaluation

Two groups of experiments conducted to evaluate the accuracy and scalability of KASR. In the first one, we compare KASR with UPCC and IPCC in MAE, MAP and DCG to evaluate the accuracy of KASR.

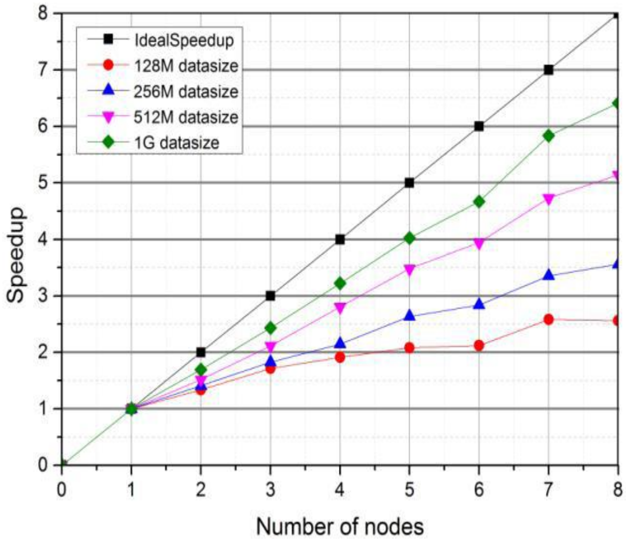

Figure 3: Speedup of KASR

5. Conclusion

In this paper we concentrated on generating the pre-processor technique. Thus the result is tagged. The chunk method is applied to process. The NLP techniques are used in Kasr process. Then positive and negative reviews are analysed and best recommendation are suggested to people. Our method aims at presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users. Moreover, to improve the scalability and efficiency of KASR in "Big Data" environment, we have implemented it on a MapReduce framework in Hadoop platform. Finally, the experimental results demonstrate that KASR significantly improves the accuracy and scalability of service recommender systems over existing approaches.

6. Acknowledgement

Our completion of this paper could not have been accomplished without the support of staff members those who are working with us. We are very much thankful to them. For the reference, we refer many articles and research papers of many authors and institutions those are available in online and offline. We offer our sincere appreciation for the learning opportunities provided by those authors and institutions. At last but not least, our heartfelt thanks to all our family members for caring us and for the huge support.

References

[1] Fay Chang “Bigtable: A distributed Storage for Structured Data” ACM Transactions on Computer Systems, 2008.

[2] Rafael Sotelo “An affordable and inclusive system to provide interesting contents to DTV using Recommender Systems” IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2012.

[3] G. Linden, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” IEEE Internet Computing, Vol. 7, 2003.

[4] S.Saranya “Individualized Travel Recommendation by Mining People Ascribes and Travel Logs Types from Community Imparted Pictures” International Journal of Computer Science and Information Technologies, 2014.

[5] Maria Alduan “Recommender System for Sport Videos Based on User Audiovisual Consumption,” IEEE Transactions on Multimedia, 2013.

[6] Zibin Zheng “QoS Ranking Pre-diction for Cloud Services,” IEEE Transactions on Parallel and Distributed Systems, 2013

[7] G.Adomavicius “New Recommendation Techniques for Multicriteria Rating Systems,” IEEE Intelligent Systems, 2007.

[8] G. Kang “AWSR: Active Web Service Recommendation Based on Usage History,” IEEE International Conference on Web Services, 2012.

[9] Z. D. Zhao “User Based Collaborative Filtering Recommendation Algorithms on Hadoop," In the third International Workshop on Knowledge Discovery and Data Mining, 2010.

[10] A. Iosup “Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing,” IEEE Transaction on Parallel Distributed System, June 2011.

[11] H. Khazaei “Performance Analysis of Cloud Computing Centres Using m/g/m/m+r Queuing Systems,” IEEE Transaction on Parallel Distributed System, May 2012.

[12] Z.Yu.X.Zhou “A hybrid similarity measure of contents for TV personalization,” Conference on Multimedia System, May 2010.

[13] X. Yang, “Bayesian-inference based recommendation in online social networks,” IEEE Transactions on Parallel and Distributed Systems, 2013.

[14] G.Adomavicius “New Recommendation Techniques for Multicriteria Rating Systems,” IEEE Intelligent Systems, 2007.

[15] G. Adomavicius “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, 2005.

[16] W. Hill, “Recommending and Evaluating Choices in a Virtual Community of Use,” In CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing System, 1995.

Author Profile

M.C.Sabitha is currently a PG scholar in Computer Science and Engineering from the Department of Computer Science at Rajalakshmi Engineering College, Chennai. She received his Bachelor Degree in Computer Science from Bhajarang Engineering College, Chennai and Tamilnadu. Her Research areas include Cloud Computing, Grid Computing and Distributed System.