

Recommendation System using Keyword Base Approach

Mr.Nilesh Sherla¹, Prof.Kiran Joshi², Prof. Sowmiya Raksha³

Department of Computer Engineering, Veermata Jijababi Technological Institute, Mumbai, India^{1,2,3}

Abstract: Big data refers to datasets that aren't solely massive, however additionally high in variety and velocity that makes them troublesome to handle using tradition tools and techniques. Due to the rise of such Data, solutions got to be studied and provided so as to handle and extract worth and information from these datasets. Nowadays Web services are very widespread .Recommender systems represent user preferences for the aim of suggesting things to get or examine. They are many basic applications in electronic commerce and data access, providing suggestions that effectively prune massive data areas so users are directed toward those things that best meet their wants and preferences. A variety of techniques are projected for activity recommendation, including content-based, collaborative, knowledge-based and different techniques. In this paper, we are presenting "Keyword-Aware Service Recommendation Method", to deal with the above challenges. It aims at presenting a customized service recommendation list and recommending the foremost applicable services to the users effectively. Specifically, keywords area unit wont to indicate users' preferences, and a user-based cooperative Filtering algorithm is adopted to get applicable recommendations. To improve the scalability and efficiency of KASR in "Big Data" environment, the proposed system proposes techniques that have been implemented it on a Map Reduce framework in Hadoop platform.

Keywords: Recommender system, Preference, Keyword, Big data, Hadoop, MapReduce.

INTRODUCTION

Data is growing at an enormous speed creating it tough to handle such large amount of data. The main problem in handling such large amount of data is as a result of that the result of that the amount is increasing quickly as compared to the computing resources. The Big data term that is getting used currently a days is quite name because it points out solely the dimensions of the data not putting an excessive amount of attention to its different existing properties.

Today, Big Data management stands out as a challenge for IT corporations. the answer to such a challenge is shifting more and more from providing hardware to provisioning more manageable package solutions[2] .Big Data additionally brings new opportunities and significant challenges to trade and domain.[3]. Similar to most big data applications, the large data tendency conjointly poses significant impacts on service recommender systems. With the growing range of different services, effectively recommending services that users most well-liked has become a vital analysis issue. Service recommender systems are shown as valuable tools to assist users modify services over load and supply acceptable recommendations to them. Examples of such sensible applications include CDs, book, web content and numerous alternative product currently use recommender systems [5]. Over the last decade, there has been a lot of analysis done each in business and world on developing new approaches for service recommender systems. Many major e-commerce Websites are used recommendation systems to produce relevant suggestions to their customers. The recommendations may be supported numerous parameters, like item popular on the company's Website; user characteristics like geographical location or different demographic information; or past shopping for behavior of prime customers.

CLOUD COMPUTING AND MAP REDUCE

Cloud computing could be a successful paradigm of service oriented computing and has revolutionized the approach computing infrastructure is abstracted and used. the main goal of cloud computing is to share resources, like infrastructure, platform, software, and business method.

Cloud computing will offer effective platforms to facilitate parallel computing, that has gained vital attention in recent years to method large volume of data. There are many cloud computing tools out there, such as Hadoop driver, MapReduce of Google [8], the generator of Amazon.com, the dryad of Microsoft and Neptune of Ask.com, etc. Among these tools, Hadoop is that the most popular open supply cloud computing platform galvanized by MapReduce and Google classification system papers, that supports MapReduce programming framework and mass Data storage with smart fault tolerance. MapReduce could be a widespread distributed implementation model planned by Google, which is galvanized by map and scale back operations within the Lisp programming language. Nowadays, the trend "everything as a service" has been making an enormous Services era as a result of the foundational design of services computing. And "servicelization" is that the method of giving social networking services, huge information analytics, and net services [9] [2]. so the cloud computing tools may be used to improve the scalability and efficiency of service recommendation ways within the "Big Data" atmosphere.

To make recommendation a lot of helpful to user, personalize recommendations are offered as hierarchical list of things. In performing this ranking, suggested systems attempt to predict what the most appropriate products or services area unit, supported the user's preferences and constraints. According to the definition of recommender system, recommender system is made public as system that

produces individualized recommendations as output or has the result of guiding the user in personalized interest or useful services during a very large space of potential decisions.

The recommendation strategies are classified into three main categories: content primarily based, collaborative and hybrid recommendation approaches. Content based approaches recommend services just like those the user most popular within the past. Cooperative filtering (CF) approaches advocate services to the user that users with similar tastes most popular in the past. Hybrid approaches mix the content based and CF methods in many alternative ways.

A. Motivation:

Over the last decade, there has been a lot of analysis done both in business and world on developing new approaches for service recommender systems [10]. A lot of firms capture large scale data regarding their customers, providers and operations. The ascent of the amount of consumers, services and different on-line data yields service recommender systems in "Big Data" setting, which poses crucial challenges for service recommender systems. Moreover, in most existing service recommender systems such as hotel reservation systems and Restaurant place guides the ratings of services and therefore the service recommendation lists presented to users are an equivalent. They need not thought-about users totally different preferences, while not meeting users personalized necessities and plenty of times such recommendations aren't helpful for the user. [11] Then there is a requirement of personalized recommendation system which will facilitate the user with the choice of product.

B. Problem Statement and Objective:

Handling of enormous scale data, to supply recommendation, consequently, ancient service recommender systems usually suffer from scalability and inefficiency issues once process or analyzing such large-scale knowledge. Moreover most of existing service recommender systems present identical ratings and rankings of services to completely different users while not considering various users' preferences, and thus fails to satisfy users' personalized necessities. The proposed system provides technique to individualize suggested services. The project proposes a novel method of personalized recommendation system. In which Keyword-candidate List and Domain wordbook are maintained for explicit system. Preferences are taken from user. And similar users are searched out by keyword extraction methodology and similarity calculations. Then the keywords are classified, and weights of reviews of comparable users are calculated. Then finally, recommendation list of top-k things is generated.

LITERATURE SURVEY

There are several recommender systems developed in both academe and trade. An analysis of variety of techniques accustomed incorporate trust into recommender systems as compared to plain recommendation algorithms. During this work trust illusion formula is used. By measuring Mean Absolute Error, Root Mean Sq. Error and the prediction coverage and therefore

analyze the results of these techniques on 3 totally different datasets with increasing numbers of ratings removed that showed that trust based mostly systems is additional correct and economical than ancient recommender rule of ratings.

1. Travel Recommendation by Mining People Attributes and Travel Group Types From Community-Contributed Photos:

Leveraging community-contributed data (e.g., blogs, GPS logs, and geo-tagged photos) for personalized recommendation is one of the active research problems since there are rich contexts and human activities in such explosively growing data. In this work, we focus on personalized travel recommendation and show promising applications by leveraging the freely available community-contributed photos. In literature it has been proposed to conduct personalized travel recommendation by further considering specific user profiles or attributes (e.g., gender, age, race) as well as travel group types (e.g., family, friends, couple). Instead of mining photo logs only, we exploit the automatically detected people attributes and travel group types in the photo contents. By information-theoretic measures, it has been demonstrated that such detected user profiles are informative and effective for travel recommendation—especially providing a promising aspect for personalization. It has been effectively mine the demographics of individual and group travelers for different locations (or landmarks) and their travel paths. A probabilistic Bayesian learning framework which further entails mobile recommendation on the spot is introduced as well. It has been confirmed that people attributes of individuals and groups are promising and orthogonal to prior works using travel logs only and can further improve prior travel recommendation methods especially for difficult predictions by further leveraging user contexts via mobile devices.

2. User-based Collaborative-Filtering Recommendation Algorithms on Hadoop:

Collaborative Filtering (CF) algorithm is a widely used personalized recommendation technique in commercial recommendation systems, and many works have been down in this field to improve the performance. However, a big problem of CF is its scalability, i.e., when the volume of the dataset is very large, the computation cost of CF would be very high. Recently, cloud computing have been the focus to overcome the problem of large scale computation task. Cloud computing is the provision of dynamically scalable and often virtualized resources as a service over the Internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing services often provide common business applications online that are accessed from a web browser, while the software and data are stored on the servers. Collaborative Filtering (CF) algorithms are widely used in a lot of recommender systems; however, the computational complexity of CF is high thus hinder their use in large scale systems. In literature it has been implemented that user-based CF algorithm on a cloud

computing platform, namely Hadoop, to solve the scalability problem of CF. It has been resulted that that a simple method that partition users into groups according to two basic principles, i.e., tidy arrangement of mapper number to overcome the initiation of mapper and partition task equally such that all processors finish task at the same time, can achieve linear speedup.

3. Bayesian-inference Based Recommendation in Online Social Networks:

Recommendation is playing an increasingly important role in our life. Accurate recommendations enable users quickly locate desirable items without being overwhelmed by irrelevant information. It is of great interest for vendors to recommend to their potential customers products matching their interests, and hopefully turn them into committed buyers. No wonder, in the Netflix contest, an improvement of 10% recommendation accuracy is awarded with 1 million dollars. In real life, people often resort to friends in their social networks for advices before purchasing a product or consuming a service. Findings in sociology and psychology fields indicate that human beings tend to associate and bond with similar others, so called homophily. Due to the stable and long-lasting social bindings, people are more willing to share their personal opinions with their friends, and typically trust recommendations from their friends more than those from strangers and vendors. The phenomenally popular online social networks, such as Facebook, twitter, and YouTube. In literature it has been proposed that a Bayesian-inference based recommendation system for online social networks. In the system, users share their content ratings with friends. The rating similarity between a pair of friends is measured by a set of conditional probabilities derived from their mutual rating history. A user propagates a content rating query along the social network to his direct and indirect friends. Based on the query responses, a Bayesian network is constructed to infer the rating of the querying user. In literature it has been developed distributed protocols that can be easily implemented in online social networks. In literature it has been proposed to use Prior distribution to cope with cold start and rating sparseness. The proposed algorithm is evaluated using two different online rating data sets of real users. It has been shown that the proposed Bayesian-inference based recommendation is more accurate than the traditional Collaborative Filtering (CF) recommendation and the existing trust-based recommendations. It allows the flexible trade-offs between recommendation quality and recommendation quantity.

II. KASR MODEL

The main steps of KASR are

1. Capture user preferences by a keyword-aware approach: In this step, 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.

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.

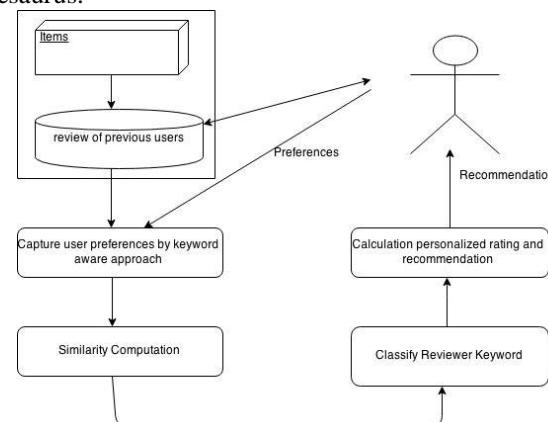


Fig.1Proposed Model

The keyword extraction process is described as follows:

A. Preprocess: 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. The keyword stripping algorithm is used to remove the commoner morphological and inflexional endings from words in English.

B. 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. If a keyword appears more than once in a review, the times of repetitions will be recorded. In this paper, it is regarded that keywords appearing multiple times are more important. The times of repetitions will be used to calculate the weight of the keyword in preference keyword set in the next step.

2. Similarity computation:

The second step is to identify the reviews of previous users who have similar tastes to an active user by finding neighborhoods of the active user based on the similarity of their preferences. 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. Two similarity computation methods are introduced in our recommendation method: an approximate similarity computation method and an exact similarity computation method. The approximate similarity computation method is for the case that the weights of the keywords in the preference keyword set are unavailable, while the exact similarity computation method is for the case that the weight of the keywords are available.

3. Calculate personalized ratings and generate recommendations:

Based on the similarity of the active user and previous users, further filtering will be conducted. Given a threshold similarity, the preference keyword set of a previous user will be filtered out, otherwise it will be retained. The thresholds given in two similarity computation methods are different, which are both empirical values. Once the set of most similar users is found, the personalized ratings of each candidate service for the active user can be calculated. Finally, a personalized service recommendation list will be presented to the user and the service(s) with the highest rating(s) will be recommended to him/her. Here, we use a weighted average approach to calculate the personalized rating of a service for the active user.

V EXECUTION OF MAPREDUCE MODEL

The Map invocations are distributed across multiple machines by automatically partitioning the input data into a set of M splits. The input splits can be processed in parallel by different machines. Reduce invocations are distributed by partitioning the intermediate key space into R pieces using a partitioning function (e.g., hash(key) mod R). The number of partitions (R) and the partitioning function are specified by the user. Figure 2 shows the overall flow of a MapReduce operation in our implementation, when the user program calls the MapReduce function. The following sequence of actions occurs.

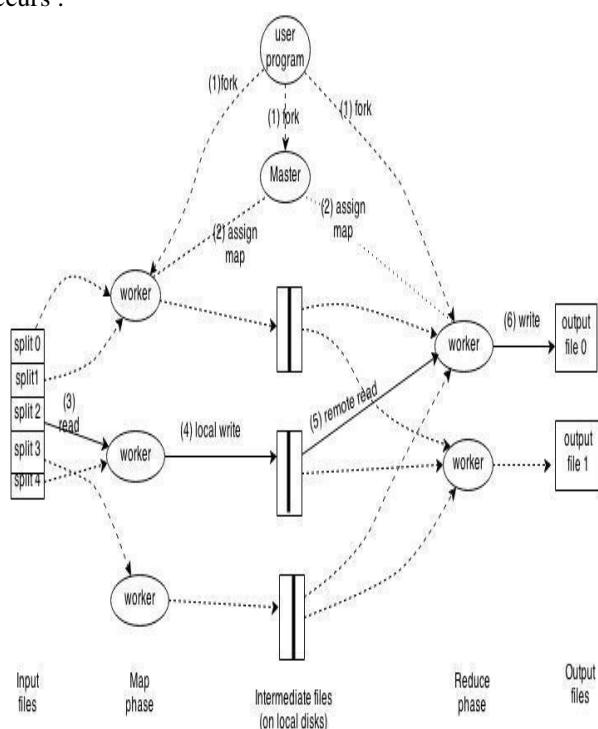


fig 2.MapReduce Execution

- 1).The MapReduce library in the user program first splits the input files into M pieces of typically 16 megabytes to 64 megabytes (MB) per piece (controllable by the user via an optional parameter). It then starts up many copies of the program on a cluster of machines.
- 2).One of the copies of the program is special—the master. The rest are workers

that are assigned work by the master. There are M map tasks and R reduce tasks to assign. The master picks idle workers and assigns each one a map task or a reduce task.
 3). A worker who is assigned a map task reads the contents of the corresponding input split. It parses key/value pairs out of the input data and passes each pair to the user-defined *Map* function. The intermediate key/value pairs produced by the *Map* function are buffered in memory.
 4).Periodically, the buffered pairs are written to local disk, partitioned into R regions by the partitioning function. The locations of these buffered pairs on the local disk are passed back to the master, who is responsible for forwarding these locations to the reduce workers.
 5)When a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of the map workers. When a reduce worker has read all intermediate data, it sorts it by the intermediate keys so that all occurrences of the same key are grouped together. The sorting is needed because typically many different keys map to the same reduce task. If the amount of intermediate data is too large to fit in memory, an external sort is used.
 6)The reduce worker iterates over the sorted intermediate data and for each unique intermediate key encountered, it passes the key and the corresponding set of intermediate values to the user's *Reduce* function. The output of the *Reduce* function is appended to a final output file for this reduces partition.
 7) When all map tasks and reduce tasks have been completed, the master wakes up the user program. At this point, the MapReduce call in the user program returns back to the user code. After successful completion, the output of the MapReduce execution is available in the R output files (one per reduce task, with file names as specified by the user). Typically, users do not need to combine these R output files into one file. they often pass these files as input to another MapReduce call, or use them from another distributed application that is able to deal with input that is partitioned into multiple files.

VI CONCLUSIONS

we have proposed a keyword-aware service recommendation method, named KASR. In KASR, keywords are used to indicate users' preferences, and a userbased 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. 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

REFERENCES

- [1]. G. Adomavicius, and Y. Kwon, "New Recommendation Techniques for Multi criteria Rating Systems," IEEE Intelligent Systems, Vol. 22, No. 3, pp. 48-55, 2007.
- [2]. Z. D. Zhao, and M. S. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," In the third International Workshop on Knowledge Discovery and Data Mining, pp. 478-481, 2010.
- [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,
- [4]. Y. Chen, A. Cheng and W. Hsu, "Travel Recommendation by Mining People Attributes and Travel Group Types From Community-Contributed Photos". IEEE Transactions on Multimedia, Vol. 25, No.6, pp. 1283-1295, 2012.
- [5]. J. Dean, and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, Vol. 51, No.1, pp. 107-113, 2005.
- [6]. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proceedings of the 10th international conference on World Wide Web, pp. 285-295, 2001
- [7]. Z. Zheng, X Wu, Y Zhang, M Lyu, and J Wang, "QoS Ranking Prediction for Cloud Services," IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 6, pp. 1213
- [8]. Wanchun Dou, Xuyun Zhang, Jinjun Chen Shunmei Meng, "KASR: A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, vol. 99, no. 2, 2014.
- [9]. Y. Guo, Y. Liu X. Yang, "Bayesian-inference based recommendation in online social networks," IEEE Transactions on Parallel and Distributed Systems, vol.24, no. 4, pp. 642-651, 2013.
- [10] J. Dean, S. Ghemawat, and W. C. Hsieh F. Chang, "Bigtable: A distributed storage system for structured data," ACM Transactions on Computer Systems, vol. 26, no. 2