International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 03 | Mar-2016 www.irjet.net p-ISSN: 2395-0072

Implementation of Recommender Engine with Phoenix

Abhishek Srivastava[1], Murtaza Zaveri[2], Ankita Bargale[3]

Mumbai University, Information Technology Dept, K.J.S.I.E.I.T,

Mumbai, India

abhishekkumar.s@somaiya.edu[1], murtaza.z@somaiya.edu[2], ankita.b@somaiya.edu[3]

Abstract - *Recommender systems are now spreading widely and trying to make profit out of customers and successfully meet their needs. The motivation to do the project comes from absence of recommender system on discussion forums. This idea is targeted to suggest a list of highly skilled-users to the users who post question based on a particular skill. This list can enable the learners to interact with skilled users and get their questions answered from a more reliable person. Distributed hybrid recommendation algorithm will be used to maximize the usage of main memory, and devise a framework that invokes Phoenix as a sub-module to achieve high performance.*

Key Words: **Recommendation engine; Hadoop; hybrid filtering; Phoenix; discussion forum**

1.INTRODUCTION

Recommender systems are now pervasive in consumer's lives. Their aim is to help users in finding items that they would like to buy based upon huge amounts of data collected. Social networking websites such as Facebook, LinkedIn and other commercial websites use these systems. The core of a recommender system is parsing a huge amount of data to predict a user's preference and his or her similarity with other group of users. Collaborative Filtering, Content-based Filtering, and Hybrid filtering are the approaches that are applied to build a recommender system. Our goal is to apply a collaborative filtering algorithm in a discussion forum that collects users' information, such as name, skill, previously answered question, up votes on the answer and recent activity.

There are many algorithms that could be applied on data to produce a recommended list of users. User-based, Item-based, and Model-based methods are ways of predicting a user preference [1]. The number of users, items, or clusters in each one will respectively determine the function performance. The most well known is User-based Collaborative Filtering. This algorithm predicts an item's rate for a user by collecting data about a particular user and similar users.

Recommendation systems can be divided into two broad categories: content-based and collaborative filtering systems. Content-based systems recommend items that contain elements (e.g., text, image, movie star) similar to those the user liked in the past. Collaborative filtering systems, on the other hand, recommend items based on "closeness" between users or between items. Although many other recommendation systems exist, the collaborative filtering systems so far have been thought of as the best recommendation systems. Our work builds upon item-based collaborative filtering recommendation algorithms and supports both shared memory multicore systems and distributed-memory clusters.

2. RELATED WORK

This section specifies the various conventional approaches that are used by some of the most top rated recommendation based websites. Recommendation systems are often implemented with the MapReduce programming model due to the model's simplicity and robustness. Users can simply write some MapReduce jobs having a map and a reduce function, then combine those jobs together and implement a recommendation system quickly[2]. The most widely used MapReduce framework is Apache Hadoop, which supports distributed memory clusters. In order to realize a high performance recommendation system, first we need to choose a faster MapReduce framework.

Most recommendation algorithms strive to alleviate information overload by identifying items which a user may find worthwhile. Content-based (CB) filtering uses the characteristics of items and collaborative filtering (CF) depends on the belief of similar customers to recommend items[3]. Hybrid methods help to improve the performance of

recommendation algorithms. Hybrid methods have avoided certain limitations of CB and CF, scalability and sparsity but there are still major problems in large-scale recommendation systems.

HBase is a noSQL distributed database which is developed on top of Hadoop Distributed File System (HDFS). HDFS contains the non-textual data like images. Location of such data is stored in HBase is presented[4]. This hybrid architecture enables faster search and retrieval of the data which is a growing need in any organization who are having very large data volumes.

Existing system uses Apache Mahout with MapReduce programming model. These systems are based on collaborating filtering or content based filtering methods. Content-based techniques are limited by the features that are explicitly associated with the objects that these systems recommend. To have a sufficient set of features, the content must be in a form that can be parsed automatically by a computer. For extracting features from text documents, information retrieval techniques work well. While some other domains have an inherent problem with automatic feature extraction.

The main disadvantage is that collaborative filtering system cannot produce recommendations if there are no ratings available. There is a problem of poor accuracy if very less amount of data is available about users' ratings. These two previous disadvantages are called as Cold-Start problem. It is also seen that many existing collaborating filtering algorithms work slow on a huge amount of data

3. PROPOSED METHODOLOGY

As mentioned before there is no existing recommendation system used on discussion forums. The system we propose simulates an environment of discussion forum which enable users to post their questions and answer to previously asked question. This system also implements a recommendation engine which will recommend a list of expert users so that answer seeking users can directly contact with experts on particular skill. The questions posted on website are visible to all the users. Anyone can give answer to these questions. So the solution seeker will get answer in less time. Also up votes and down votes are given to the answers by other readers. Hence solution seeker can easily get to know which is the most correct and relevant answer based on up votes and down votes.

The Fig. 1 shows block diagram of proposed system:

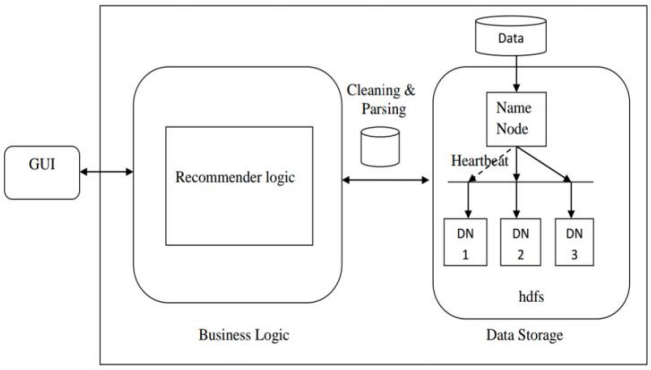

Fig. 1 Block Diagram

The system transforms users actions on the forum into a source of recommendation based on domain or tag concepts. In order to achieve this goal we had to deal with the extracted row data and transform this data into a utility matrix consisting of users, tags, number of previous answers, last activities etc.

The proposed system uses hybrid filtering approach. Hybrid recommendation systems are mixture of single recommendation systems as sub-components. This approach is introduced to deal with a problem of standard recommendation systems. Two main problems have been identified by researchers in the same field such as cold-start problem and stability versus plasticity problem. The cold-start problem takes place when some learning based techniques like collaborative, content-based and demographic recommendation algorithms are used. Stability problem means that it is difficult to change established users' profile which is formed after a given period of time using the systems[5].

The following two algorithms will be used to implement the proposed system:

1. K-means Algorithm**:** It is a partition-based classical clustering algorithm. It is widely used in many areas of data mining. The reciprocal of Euclidean distance is used to compute the similarity between the input vector and the clustering center. The advantages of the K-Means algorithm are that it can classify large data sets efficiently, reliable in theory, simple and fast[7].
2. Cosine Similarity Algorithm**:** It is also known as vector-based similarity. The formulation considers two items and their ratings as vectors. Also defines the similarity

between them as the angle between these vectors[6]:

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

4. PSEUDO CODE

Step 1: Clusters the data into predefined k groups.

Step 2: Select random points 'k' as cluster centers.

Step 3: According to the Euclidean distance function, assign objects to their closest cluster center.

Step 4: Now, calculate the centroid or mean of all objects in each cluster.

Step 5: Repeat steps 2, 3 and 4 till the same points are assigned to each cluster in consecutive rounds.

5. CONCLUSION AND FUTURE WORK

Compared to some of the other recent recommendation systems our proposed system is comparatively better in the sense of accuracy and speed. It evaluates the possibility to combine the various recommendations techniques and form proposed hybrid algorithms to improve recommendation performance. The proposed system will be a great web application that can club with today's high demanding E-learning web sites. With the help of the proposed system, users will not be only getting answers of their questions easily, but they will get an opportunity to invite expert users on the forum to answer their questions. We have proposed a new approach for utilizing Phoenix to build a high-performance recommendation system. The same approach can be used to build other types of big data applications. Our future work is to make the framework more general so that different application developers can use the framework as a library.

ACKNOWLEDGEMENT

On the very outset of this report, we would like to thank all the personages who have helped us in successful completion of this project. Without their active guidance, help, cooperation and encouragement, we would not have made headway in the project. We are obliged to the Principal Dr. Dilip Pangavhane, Head of Information Technology department Prof. Uday Rote and other faculty members of K. J. S. I. E. I. T. for providing valuable information in their respective fields.

REFERENCES

[1] Recommender System Using Collaborative Filtering Algorithm by Ala Alluhaidan, Grand Valley State University

[2] J. Talbot, R. M. Yoo, and C. Kozyrakis, "Phoenix: Modular mapreduce for shared-memory systems," in Proceedings of the second international workshop on MapReduce and its applications, 2011, pp. 9–16.

[3] A Hybrid Recommendation Method with Reduced Data For Large-Scale Application by Sang Hyun Choi, Young-Seon Jeong & Jeong, M. K. in Proceedings of The 10th international conference on World Wide Web, 2001, pp. 285–295.

[4] Hadoop-HBase for large-scale data by Vora, M. N. India in Computer Science and Network Technology (ICCSNT), 2011 International Conference.

[5] Hybrid Web Recommendation Systems (Core Presentation Summary with Discussions) by Jae-wook Ahn

[6] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl. Item -based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference, pages 285-295, 2001.

[7] An Efficient K- Means Clustering Algorithm Using Simple Partitioning* Ming-Chuan Hung, Jungpin Wu+, Jin-Hua Chang And Don-Lin Yang. Journal Of Information Science And Engineering 21, 1157-1177 (2005) 1157

[8] https://hadoop.apache.org/