A Study On Movie Recommendation System Using Parallel MapReduce Technology

Goral Godhani, Maulik Dhamecha
V.V.P.Engineering College -Rajkot, Gujarat

***Abstract* - Recommendation systems have gained tremendous popularity over the past few years. As we all know recommendation systems have provided a boon in the field of online shopping and other browsing portals. Recommendation systems use machine learning techniques to predict what the user may like based on his history of interaction with a system full of items. At present there are many approaches known to implement a recommendation system. These approaches have several algorithms with various efficiencies. The efficiency and the accuracy of the recommendation system solely depend on the algorithm used. Hence, we've designed, implemented, and successfully deployed a system that uses an ensemble of item-, and content-based recommendation systems. In this paper we are going to explain how the results from two algorithms which are run on Hadoop are combined to get more accurate movie recommendations.**

***Keywords*— Recommendation system algorithms, collaborative filtering, content based filtering, Hadoop, Apache**

I. INTRODUCTION

The rapid development of the Internet have brought about great changes in the world, which cause information overload[1-3]. Information overload has two sides .For information users, searching for what they need from the vast amounts of information accurately is becoming more difficult .For information manufacturers, making information they produce conspicuous in the vast amounts of information is also a problem need to be solved urgently. Recommendation system is based on the analysis of used behavior to meet users' need and interest.

Due to the large amount of information present in the World Wide Web, we observe a difficulty for users to deal with this huge quantity of content available. This problem is known as information overload, and a tool that helps individuals to manage such content is recommender systems. There are a number of ways to build recommender systems; basically they are classified as content-based filtering, collaborative filtering and the combination of both of them [1], [2].

In 2003 Deng Ailin put forward a collaborative filtering recommendation algorithm based on project score predicts [4] in order to solve the sharply fallen recommendation quality rating caused by extremely thin data .In 2004, Gao Fengrong adopt the method of clustering and classification respectively by divided sparse score matrix, reducing the scope of neighbor and the need to predict the number of resources, improving the efficiency and scalability of the collaborative filtering recommendation algorithm [5] With the growing amounts of information, Research focusing on the recommendation system has changed from stand-alone mode to server cluster [6-8] In 2011, Jing Jiang and others further divided collaborative filtering recommendation algorithm based on the project into three stages of the main computing [7].After the segmentation of each Map Reduce phase can be run in parallel in each node of the cluster. Meanwhile they put forward the data partitioning strategy based on Hadoop platform to maximize the local data storage, reducing the communication overhead between system and improving the recommendation efficiency of the recommendation system. In 2012, Juan Yang and others further used mongo as auxiliary storage database [8].

At present, collaborative filtering recommendation methods are mainly based on matrix decomposition [9][10]. However, the current method in scalability performance is poor, and unable to adapt to the massive data processing. In this paper, aiming to the recommendation system's error range,. We analyzed deeply the traditional collaborative filtering recommendation algorithm based on user, run the improved the algorithmon on MapReduce in the platform of Hadoop ,which has made the effect more accuracy.

II. LITERATURE SURVEY

One disadvantage of collaborative filtering is the computational effort spent to calculate similarity between users and/or items in a vectorial space composed of user ratings in a useritem matrix. Similarity metrics (Pearson correlation, cosine similarity, etc.) must be applied to this matrix in order to infer clusters of similar users or items. However, such vectorial space makes this a large dimensionality matrix. Besides, the vectors are redundant because there will be users with similar ratings for the same items [2]. Such limitations have inspired researchers to use dimensionality reduction techniques, such as Singular Value Decomposition (SVD), in order to extract latent semantic relationships between users and items, transforming the vectorial space into a feature space containing topics of interest [5], [6], [10], [12]. Nevertheless, other challenges have to be dealt with, such as sparsity, overfitting and data distortion caused by imputation methods [5].

Considering the limitations and challenges depicted above, hybrid recommenders play an important role because they group together the benefits of content based and collaborative filtering. It is known that limitations of both approaches, such as the cold start problem, overspecialization and limited content analysis, can be reduced when combining both strategies into a unified model [1]. However, most recent systems which exploit latent factor models do not consider the metadata associated to the content, which could provide significant and meaningful information about the user's interests.

2.1 Collaborative filtering

This is the most widely used approach in recommendation systems. It is used to provide recommendations using known attributes of users with similar interests. It is mainly based on analysing user history, user behaviour, user likes and dislikes. This analysis is used to find the degree of similarity between users. Generally, two algorithms are used to identify similar users, namely K-Nearest Neighbour (k-NN) algorithm and the Pearson Correlation. We are reviewing K-nearest neighbour approach.

2.1.1 Item-Based K-Nearest Neighbour algorithm

This is similar to User Based K-Nearest Neighbour approach, except, the nearest neighbour similarity is based upon item to item similarity instead of user to user similarity. This approach is explained in the following steps. This algorithm is written in hadoop's mapreduce framework in our system for efficient processing.

A. The aim is to predict rating of an item *'i'* by a user *'u'*. This can be achieved by finding similarity weights between items using cosine similarity formula.

$$\text{Similarity (Item1, Item2)} = \text{Cos } \theta = \frac{\text{Item1. Item 2}}{||\text{Item1}|||| \text{ Item2}||}$$

B. The similarity vector is then formulated by arranging the weights in descending order and thus K Nearest Neighbour of item *'i'* can be easily found.
C. Now, another vector *RU* that consists of all items already rated by user is formed.
D. Finally, the similarity vector of every item is combined with vector *RU* to predict rating of items by user. These ratings are then used to recommend an item to user.
In real time the number of users is vast compared to number of item, thus Item-Based K-Nearest Neighbour algorithm proves to be efficient in practice. Also, the ratings data is vast; hence Hadoop is used for big data management.
The top-5n results from this algorithm are taken for final ensemble.

2.2 Content Based Filtering

This is another common approach which is based on features of movie Meta data and history of user which in turn builds up user and movie profile. These profiles play the key role in recommending movies to the user.
The movie which shares the maximum features in a user's history is more likely to be recommended. We have implemented this algorithm in the following three simple steps in hadoop's mapreduce framework to achieve parallelism.
a. Firstly, a set of all features or attributes of movies present in the dataset is created.
b. The system maintains for each attribute value the number of movies in the user's history with that attribute value plus the rating the user has given for such movies.
c. The final score of a movie for a particular user is then a non- negative combination of the different scores each of the movie's attribute values have for the particular user.
Content-based filtering is efficient at calculating the predicted ratings for movies. It considers the contents of movies instead of item similarity.
The top-5n results are taken from this algorithm for final ensemble.

III. FINAL PARALLEL HYBRID RECOMMENDATION SYSTEM ENSEMBLE

The final top-*n* recommendations for a particular user *u* are computed by first asking each of the two recommenders (in parallel) to compute the top-*5n* recommendations for *u* and then computing for each recommended movie (by any of the individual recommenders), a linear weighted combination of the recommendation values of the two recommenders. We optimized the weights *w(i), w(c)* of the item-based, and content-based recommenders. The resulting values are sorted in descending order, and the top-*n* movies are returned.

IV. IMPLEMENTATION DETAILS

The implementation of our project takes place in two stages:
1. Offline Part.
2. Online Part.

4.1 Offline Part

In the offline part, the training is done after a regular period of time. For example after a week or after a month. All the user interactions are recorded within this period. This recorded data is then processed in order to train the system. These processes are carried out in the server. In our project we have used already recorded data from MovieLens.org.

4.1.1 Hadoop Multi-node cluster

In real-time, the Movie data generated is in petabytes and hence it is not feasible to process data using a single machine. Hadoop helps in distributing the processing.
In our system, we have set up a Hadoop Multi-Node cluster which automatically divides the input data into multiple chunks and processes it in a distributed environment. Hence, it increases the efficiency drastically.

V. EVALUATING THE MOVIE RECOMMENDATION SYSTEM

In our system, we wish to predict the rating a user would give to a movie (e.g. 1-star through 5-stars). In such cases, we wish to measure the accuracy of the system's predicted ratings.

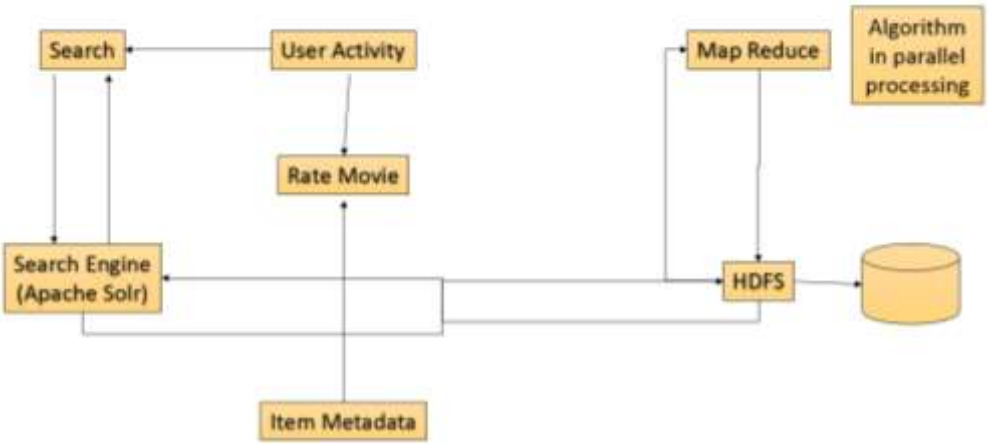

Figure 1.Proposed Framework

5.1 Online Part

The online part deals with indexing the output file using Apache Solr Search engine tool. The indexed file can then be easily accessed from the client side. The client activities also come in the online part.

5.1.1 Search Engine

Solr enables powerful matching capabilities including phrases, wildcards, joins, grouping across any data type. The dataset is a Pipe Separated File and is posted to the Solr Server for providing quick search result. It is able to do so because it indexes each document in XML like tags and when a query is fired to the Solr Server, it parses through the indexed file instead of the actual document. This speeds up the search and the entire row is returned. Each field can be separately accessed as required. The working steps are as follows:

i) When a user searches for a Movie name, a query will be fired to the server that will return a set of relevant documents.

ii) User will select the required movie

iii) Another query will be fired to retrieve set of all movies that are recommended for the selected movie as "*People who liked this also liked:"*

Root Mean Squared Error (RMSE) is perhaps the most popular metric used in evaluating accuracy of predicted ratings. The system generates predicted ratings rui for a test set T of user-item pairs (u, i) for which the true ratings rui are known. Typically, rui are known because they are hidden in an offline experiment, or because they were obtained through a user study or online experiment. The RMSE between the predicted and actual ratings is given by:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}.$$

Mean Absolute Error (MAE) is a popular alternative, given by

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i|.$$

The MAE for Item Based collaborative filtering alone, Content Based Filtering alone and combination of both were calculated and compared as follows.

As observed from the above table that the accuracy of the recommendation system increases drastically with the combination of Item based collaborative filtering and Content based filtering. Hence, combination of the results predicts more accurate ratings.

VI. CONCLUSIONS

We have developed a system for real time recommendation generation. The primary goal of the project is to create a system that will provide near relevant recommendations has been successfully achieved. Our system is aimed to ease the users woes of having to select from a huge variety of options available at hand. Also, accessing the system environment simpler so that the person who is unaware about the system can handle it with ease. Our system has also dealt with the issues related to design a real-time recommendation system like handling big data.

The ability for a recommendation system to bubble up activity in real time is a huge advantage because the system is always on.

Recommendation systems are popping up everywhere, from movies to news to travel and leisure. They provide valuable, personalized information that can greatly influence the way we use the Web. They might even be the beginnings of an artificial intelligence for each of us. Recommendation systems are out there: watching, and learning.

VII. REFERENCES

[1] Yuanzhuo Wang, Xiaolong Jin. Journal of Computer, **36**,4(2013)
[2] Jianguo Liu, Tao Zhou, Binghong Wang. Progress in natural science, **19**,7(2009)
[3] Shahabi C,Chen Yishin. Distributed and Parallel Databases,**14**,9(2003)
[4] Ailin Deng, Yangyong Zhu, Bole Shi. Journal of software. **14**,8(2003)
[5] R. Shankar, Prof. Ateshkumar Singh, Ms. Ila Naresh Patil (2013), Overview Of Recommendation System & A Speedy Approach In Collaborative Filtering Recommendations Algorithms With Mapreduce, International Journal of Computational Engineering Research, Vol, 03, Issue, 9.
[6] NilayNarlawar, IlaNaresh Patil (2014), A Speedy Approach:User-Based Collaborative Filtering With Mapreduce, International Journal of Computer Engineering &Technology(IJCET), Vol 5, Issue 5.
[7] Zhi-Dan Zhao, Ming-Sheng Shang (2010), User-based Collaborative-Filtering Recommendation Algorithms on Hadoop, Third International Conference on Knowledge Discovery a Data Mining.
[8] Mohamed Sarwat, Justin J. Levandoski, Ahmed Eldawy, and Mohamed F. Mokbel (2014), LARS*: An Efficient and ScalableLocation-Aware Recommendation System, IEEE Transactions On Knowledgeand Data Engineering, vol. 26, no. 6.
[9] Emmanouil Amolochitis, Ioannis T. Christou, Zheng-Hua Tan (2014), Implementing a Commercial-Strength Parallel Hybrid Movie Recommendation Engine, IEEE.
[10] Pedro G. Campos, Ignacio Fernández-Tobías, Iván Cantador, Fernando Díez (2011), Context-Aware Movie Recommendations: An Empirical Comparison of Pre-Filtering, Post-Filtering and Contextual Modeling Approaches, Escuela Politécnica Superior Universidad Autónoma de Madrid.
[11] Shichao Zhang (2010), KNN-CF Approach: Incorporating Certainty Factor to kNN Classification, IEEE, Vol.11 No.1.
[12] Mustansar Ali Ghazanfar and Adam Pr¨ugel-Bennett, An Improved Switching Hybrid Recommendation System Using Naive Bayes Classifier and Collaborative Filtering, School of Electronics and Computer Science, University of Southampton, Highfield Campus, SO17 1BJ, United Kingdom.
[13] Incheon Paik and Hiroshi Mizugai (2010), Recommendation System Using Weighted TF-IDF and Naïve Bayes Classifiers on RSS Contents, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.14 No.6.
[14] 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, 17(6):734–749, 2005.
[15] M. D. Ekstrand, J. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. Foundations and Trends in Human-Computer
    a. Interaction, 4(2):175–243, 2011.