Single-criteria Collaborative Filter Implementation using Apache Mahout in Big data

Manu M N1*, Ramesh B2
1Department of Computer Science & Engineering, SJBIT, Bengaluru
2Department of Computer Science & Engineering, MCE, Hassan Visvesvaraya Technological University, Belagavi
Karnataka, India

Available online at: www.ijcseonline.org

Received: 21/12/2016 Revised: 30/02/2016 Accepted: 18/01/2017 Published: 31/01/2017

Abstract—In everyday life recommendation system plays an important role and collaborative filtering (CF) is used widely in many e-commerce applications for online product recommendation. A recommender system is mainly used for better predictions as better decision making by using preferences during searching, shopping etc. The preferences of other users, user's past preferences and big data is the driving force behind Recommendation systems. In this paper, we present collaborative filter types and their main challenges. Using the open source library Apache Mahout, we implemented collaborative filter using single-criteria to recommend items to particular users. Also, we showed the flow of Apache Mahout command’s execution to process the huge data using LogLikelihood similarity algorithm in big data scenario.

Keywords—Recommendation system; Collaborative filtering; Apache Mahout; Big data; Hive

1. INTRODUCTION

Recommender systems are called as suggestion systems for the users. Varieties of algorithms are used on huge data which would help in giving suggestions for new items for the users [1]. Recommender systems are applied in different types of applications which depict their popularity. The most popular ones are probably entertainment, information, research articles, social tags and websites, and products in general [2]. Recommender systems have changed the perception of people finding product information and their details and even other people. Recommender system suggests a new collection of things that a user has never experienced before by studying the behavior pattern. Over the years, recommender systems evolved with different collection of tools and it welcomes many researchers to develop effective recommenders.

A recommender system consists of user data, item preferences/ratings and neighborhood. Items are objects or products that are used for recommending to a user. In example [31], items are films, music albums, sports and articles such as news articles, politics news, products details, technologies articles and weather reports. All the above said items are categorized under respective information. Users are used to give preferences to an item and use recommendation for better decision. Neighborhood is used to make better decisions to produce more accurate, efficient recommendations.

A. Types of recommendation systems

We present different types of recommendation systems that varying in terms of different domains. Based on Paper [3], author differentiates between six classes of recommendation systems. They are:

- Content-based filtering: also called as cognitive filtering. It recommends items based on a user past history and the detail of the items [4][32]. In a content-based recommendation system, the user history (profile) refers to builda user profile to describe what type of item user likes. Text documents are used as source in content-based filtering systems.
- Collaborative filtering: Popular and widely used technique [5]. It works based on analyzing a huge history data on user activities, behavior, preferences and predicting the users like or dislikes based on similarity of other users. Neighborhood methods focus on relationships between items and users alternatively.
- Demographic recommendation system: It provides item based recommendations on user's demographic profile [5]. For an example, recommend the items or products based on the age of the user, user's habits etc.
- Knowledge-based recommendation system: It recommends based on how item useful for user, how certain features meets user's preferences and needs based on specific domain knowledge [5]. Here the utility of the recommendation for the user directly interpreted to similarity score.
- Community-based recommendation system: In this, system recommends an item based on preferences of users community or friends. Example Social networks.
- Hybrid recommendation system: Combining content-based filtering and collaborative filtering could be more effective in some scenario [6]. For that Hybrid recommendation system demonstrated as most useful and powerful approach can be implemented.

B. Key challenges

In this paper, we discuss few important key challenges of recommendation systems and they are:

- Cold start: If the user has no previous history or behavior. In this, user has to put more efforts to construction of their user profile and preferences.
Scalability: Large amount of datasets with users could need large performance power. And also how it deal with big data and stream data produced by active users with items [7][8][9].

- Sparsity: Many a time’s ratings on items by users are very low (less), Because of this problem; popular items will have less rating [10].

C. Applications recommendation system

Here we are listed few popular applications of recommendation system. Well-known applications are:

- LinkedIn: Most popular business-oriented social networking site. It is used to forms recommendations for users, job, organization or company and group etc [11]. LinkedIn uses open source Apache Hadoop technology recommendation. LinkedIn uses item-based collaborative filtering, which is used to find relationships between pairs of items.

- Amazon: Most popular e-commerce site uses Amazon item-to-item collaborative filtering technique. Whenever customers buy an item in this site to purchase, it recommends items of similar purchased by others users [12].

The research work in this paper presents a novel architecture to recommend items to users using apache mahout, hive in big data platform. We implemented item based collaborative filtering algorithm using clustering distributed frameworks. As mention in the paper [33], for better and accurate prediction, we are using recommendation system to recommend an items to users based on their past history and preferences.

This rest of paper has been structured as follows, section 2 describes big data recommendation system, related works are in section 3. Section 4 describes apache mahout and hive, Section 5 describes algorithm for similarity. Section 6 describes Experimental studies Concluding remarks are at the end.

II. BIG DATA RECOMMENDATION SYSTEM

Recommendation system cannot do better without the proper input supply of huge data. The traditional technologies are not able to process large amount of data so quickly [13]. So, new technologies are needed to handle big data in order to provide strong and accurate recommendations [33]. Huge data should be processed quickly and meaningfully in less time. Market legends like Google, IBM, Facebook, Twitter etc are moving into big data technologies. However, it is clear that the recommendation system also require new technologies. To realize this, in this paper, we used Apache Hadoop based map-reduce platform and on top Mahout could be used to identify right algorithms, similarity methods and other tasks to recommend an items to users.

A. Recommendation algorithms

In Collaborative filtering, recommendations are completely based on the active users to rate an items [30][31]. For each item ‘i’ or user ‘u’, a neighborhood is formed with similar related items or users.

Collaborative filtering consists following techniques:

- Item based recommendation
- User based recommendation

Item-based recommendation: Similarities are calculated between items and make recommendations [14]. This can be computed off line because items usually don’t change.

User-based recommendation: Recommend items by finding similar users [15]. Because of the dynamic nature of users it is facing scaling problem.

III. RELATED WORKS

Numbers of researchers have addressed various issues of Big Data and recommendation system. This section gives the work and major contributions contributed by the researchers under various domains.

The nature of big data makes it necessary for us to store the data in distributed file system architecture and Hadoop (HDFS) is a widely used tool for doing so. Map Reduce concept is widely used for analysis of Big data. The author [16] gives a brief knowledge on Both Hadoop (HDFS) architecture and Map Reduce Technique. The working of Map Reduce architecture is clearly explained using the four components namely Name node, Data node, Job Tracker, Task Tracker.

In real time, Analysis of streaming data for low cost is becoming very important. The data streams arrive at a high speed; therefore algorithms must work under very strict space and time constraints. Sampling and using distributed systems are two techniques are used to deal with big data. The author [17] also mentions few tools which are handy for these tasks namely Scribe, Cassandra, S4, Storm, Mahout, Moa, R, Pegasus, Graphlab. The author also states that sentimental analysis is a classification problem, where the classification is between positive or negative groups.

In E-Commerce era RS has played very important role in practice and research [18]. Collaborative filtering has proved that one of utmost important for its ease in both theoretical and practical. Collaborative filtering [19] predicts the active customers interest based on accumulated rating information of the on the same wave length customers in a chronological database.

The data scalability is considered as one of the major problem in collaborative filtering, which leaders to the bad performance. In this papers [20][21], Scalability is one of most difficult challenges in recommendation system. To solve this, author introduced k-means algorithm for item clustering prediction in item based collaborative filtering (CF). K-means algorithm was used in item clustering prediction and it was more scalable and accurate technique than any other approaches used earlier.

In this paper [22], multi-criteria collaborative filtering was considered to measure similarity between two users.
Authors estimated the users overall ratings by using nonadditive indifference indices.
To achieve better performance in recommender system, authors [23] designed new technique called item-based collaborative filtering framework. In this, users were rated items directly without any biasing by using Pearson’s method.
With respect to data stream mining, J. Chandrika and K.R AnandaKumar [24] discussed the importance of resources adoption and quality awareness in streaming technology. Newly proposed novel framework was designed to generalize streaming techniques. They proposed resource awareness and quality awareness system.
Most of the recommendation systems depend on similarity measure and this similarity measure depends on user ratings for neighbor’s selections. Authors [34] analyzed that cosine and Pearson’s similarity measures are not improving the recommender system performance. Different genres of movies and average ratings of users are considered to compare the performance of recommendation systems.
In this paper [26], authors joined the two approaches by combined the advantages of model based collaborative filtering and memory based collaborative filtering to gain more accurate recommendation than traditional collaborative filtering. Initially it employed memory based collaborative filtering to solve the sparsity problem by filling the empty ratings of the matrix (user-item) and finally item based collaborative filtering technique was used to form the nearest neighbors of each item.
This paper [27] explains how to reduce the power consumption in event recognition system using big data analytics to handle huge data in real time. The proposed framework provides system scalability to perform under parallel, remote, and distributed in computing environment.

IV. APACHE MAHOUT AND HIVE
Apache Mahout is Machine Learning library of set of algorithms for open source Apache Hadoop [33]. It is Apache Software Foundation product with the goal of scalable performance and building scalable algorithms in the areas of machine learning. To produce distributed, scalable algorithms mahout’s implemented MapReduce paradigm in Apache hadoop. In this work we used hadoop based MapReduce implementations to recommend items to users.
Hive is used for SQL queries over petabytes of data in hadoop [35]. Data summarization, querying and analysis of big data can do using hive tools and also using this developer’s access hadoop data.

A. Recommendation System using APACHE Mahout
Using big data technology tools, we stored data into HDFS (Hadoop Distributed File System). Mahout provides the place to find meaningful processing of big data. The Apache Mahout aims to make it process easier and faster to turn big data could be converted to big information. Preferences are produced by user ‘u’ for an item ‘i’ by combining the preferences of other user’s ‘u’*. Similarly here, the two user's preferences have a high Cosine similarity, Pearson correlation or Log Likelihood similarities or other similarities. In the item-based approach, we produce a preferences ‘r’ for an items by looking at the set of other items ‘i’ that are similar to i that user ‘u’ has rated and then combines the ratings by ‘u’ of i into a predicted rating by ‘u’ for i [28][31][32].

V. ALGORITHMS FOR SIMILARITY
List of similarity algorithms in Mahout [29]

- Pearson Correlation Coefficient
- Cosine-Based Similarity
- LogLikelihood ratio Similarity

Pearson Correlation Coefficient: It measures the relations between two variables a and b, where a and b could be any parameters that considered under the study. Here, we consider user ‘u’ and item ‘i’ as the parameters as they have huge influence on recommendation [29].

The algorithm considers preferences on which both users or items overlap. It attempts to find each user’s ‘u’ or items ‘i’ deviations from their average rating while identifying linear dependencies between two users or items. The equation uses actual preference values to find correlation between users or items and gives larger weights to users or items that agree extreme cases. In pearson correlation, where ‘a’ and ‘b’ represent the two users or items for which the coefficient is being calculated, ‘i’ is an item, \(r_{ai}\) and \(r_{bi}\) are individual ratings from ‘a’ and ‘b’ for item ‘i’, and \(\bar{r}_a\) and \(\bar{r}_b\) are average ratings for user a and b.

\[
sim(a,b) = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2 \sum_i (r_{bi} - \bar{r}_b)^2}}
\]

Cosine-based Similarity: In this case, two items are thought of as two vectors in the m-dimensional user space. The similarities between the items are measured by computing the cosine of the angle between these two vectors. Formally, in the \(m \times n\) ratings matrix, similarity between items a and b, denoted by \(sim(a, b)\) is given by

\[
sim(a,b) = \frac{\sum_i r_{ai}r_{bj}}{\sqrt{\sum_i r_{ai}^2} \sqrt{\sum_i r_{bj}^2}}
\]

LogLikelihood Similarity: The likelihood similarity is a statistical approach employed to calculate the similarity between two variables based on their occurrence related to them. When two or more items or events have same degree of preference then we go for likelihood similarity to predict the possible event to occur. This also includes initial conditions where both variables have zero preferences [29][30]. In Mahout, similarities between items and users are calculated using LogLikelihood similarity. Rather considering the actual preferences, the ratio of events...
occurred recently is opted as a parameter of estimate choice to the user. In our experiments, we are using LogLikelihood similarity for recommend a movies to the users.

VI. EXPERIMENTAL STUDIES

The main goal was to preprocess big data for implement and produce recommendations based on amount of input data and also check the time efficiency of the implemented system. We are going to achieve this with the use of distributed cluster environment, big data platforms like Hadoop, Hive and Mahout by computing the item similarity matrices.

A. Data set description

The considered input data set was list of movies called MovieLens data sets. These data sets are available at GroupLens Research Project, University of Minnesota as last updated till Jan/2016. This data set consists of ratings |R|=22884377, users |U|=247753 and movies |M|=34208. The description of data set contains user’s ID, age, sex (gender), occupation, area, zip code etc. The data set consists of three tables: (1) Users (userid, name, etc): This table contains all the information of users. (2) Movies (movieid, name, genre, etc): Information about movies. (3) Ratings (userid, itemid and rating). Each column in the ratings table 1 represents how much a user liked a movie after watching the movie.

<table>
<thead>
<tr>
<th>User (u)</th>
<th>Items (i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>i2</td>
</tr>
<tr>
<td>u1</td>
<td>2</td>
</tr>
<tr>
<td>u2</td>
<td>5</td>
</tr>
<tr>
<td>u3</td>
<td>4</td>
</tr>
<tr>
<td>u4</td>
<td>1</td>
</tr>
<tr>
<td>u5</td>
<td>2</td>
</tr>
</tbody>
</table>

B. Experimental Hadoop setup description

Experimental Hadoop setup was performed on a hadoop and HBase clusters with configuration of three nodes, one node as HDFS MasterNode and other two nodes as HDFS DataNodes. The configuration set up of each nodes was Processor- Intel Core i5 with Turbo Boost up to 3.1GHz, Cache memory- 3MB L3 cache of 2.5GHz dual-core, Memory- 8GB of 1600MHz DDR3 memory of RAM and Storage- 500GB hard disk drive. Nodes are connected with 100Mbps Ethernet network. The experiments run over Hortonworks Data Platform (HDP): Horton sandbox version HDP 2.4 runs on about 2.5 GB of disk space and Apache Mahout Distribution Version 0.10.2.

Implementation process consists of five major steps. Considering data source as Movie set data with attributes <userid, itemid, ratings> and a novel architecture to recommend new movies to the customer or users as shown in the below figure 1. After Apache Mahout is properly installed, first we need to get Mahout up and running, below figure 1 shows the flow of Apache Mahout Command’s execution.

![Diagram](image)

**Step 1: Data Collection**

MovieLens data set was available in .zip format in below mentioned link. Using Wget command, first we downloaded the data set into our experimental set up system. The downloaded file has to unzip and stored in fileml-latest. We can view this available data set using Apache Hive commands.

To get MovieLens data:

1. `wget http://grouplens.org/datasets/movielens/latest`
2. `unzip ml-latest.zip`

**Step 2: Data Preparation using Hive**

After data collected, put it in a .CSV file using Apache Hive. In this stage, we are using Apache Hive for preparing the data as required. Here input file 00000_0 contains comma separated formatted data like <UserId, ItemId, Rating>

```
hadoop fs -tail /manu/userdemo/mctab1input/00000_0
```

929,44,5 930,28,5 931,60,2 932,58,5 933,28,4 934,613 935,42,5

**Step 3: Similarity Measures**

Choosing a similarity measure for produce best recommendation and requires careful testing, evaluation and research. In this paper, we are considered a Mahout
item-based similarity called SIMILARITY_LOGLIKELIHOOD to produce recommendations.

**Step 4: Recommender engine using Apache mahout**
Build Recommender engine using Configure the Mahout commands. After Mahout was properly installed, configure the syntax. Enter the following command are used to run the recommender job

```
$ mahout recommenditembased -s SIMILARITY_LOGLIKELIHOOD -i /inputpathfile (where input file stored in file)
-o /Outputpathfile (Where we want to store output) --numRecommendations10 (Number of recommendation is depends on Requirements)
```

The above Mahout command was executed. The series of Jobs were run to get final result using input file path manu/mahout/input.

**Step 5: Recommendations result**

The output file path manu/mahout/output/part-r-00000 contains recommendations as result. It was generated in this step can be transformed using Apache Hive to produce Mahout’s item based recommender to recommend movies and its preferences to the users in the form moviesId [movieId:rating, movieId:rating,........etc]

```
hadoop fs -tail /manu/mahout/output/part-r-00000
```

Time taken:: 15/10/19 18:27:06 INFO

```
VII. CONCLUSION AND FUTURE ENHANCEMENT
```

Big data and recommender engines have already proved an important useful combination for big corporation in the matter of profit. Big data tools and technologies are relatively more affordable in present scenario. In recommender system techniques, Collaborative filtering used in many e-commerce filed and it has been proved as most successful techniques in present scenario. Movie recommendations are extremely important to provide a good user experience from the user’s viewpoint. In our paper, we considered open source Apache Mahout and hive to recommend movies to users or similar movies based on past experiences.

The above result shows big data tools have a better scaling capability on huge datasets compare to conventional data mining tools. Item based recommendation is achieved through single-criteria collaborative filter which is implemented on Apache mahout a big data tool. The recommendation set put forth seems to show convincing levels of acceptance.

Further, the achieved output set if serves as input for hybrid collaborative filters. We get more precise and high weightage recommendation set. Also, in further work to achieve high accuracy multi-criteria based approach could be involved.

**REFERENCES**


Authors Profile

Mr.Manu M N received his Bachelor’s Degree in Computer Science and Engineering from SIT, Tumakuru, during the year 2007 and Master of Technology in year 2010 in Computer Science and Engineering from PESCE, Mandya, Karnataka, India. Currently he is pursuing his Ph.D in VTU. He is working in SJB Institute of Technology from the past 7 years. His area of interests is Big Data and Artificial Intelligence. He has published
and presented papers in journals, national and international level conferences.

*Dr. Ramesh B* completed his B.E degree in Computer Science and Engineering from Mysore University, Karnataka, India in 1991, M.Tech degree in Computer Science from DAVV, Indore, Madhya Pradesh, India, in 1995 and Ph.D degree from Anna University in 2009. Currently he is working as a Professor in the Department of Computer Science and Engineering at Malnad College of Engineering, Hassan, India. His current research interests lie in the areas of Big Data, Artificial Intelligence, congestion control, QoS-aware routing algorithms in ad hoc networks and multimedia networks.