ITEM BASED COLLABORATIVE FILTERING USING CLOUD COMPUTING (HADOOP)

1. M.Divya,
(mdivyamdu@gmail.com)
Ph.No:8056893790
Third Year, Department of Information Technology,
Thiagarajar College of Engineering,
Madurai-625015.

2. K.Divya krishnaveni,
(divyakrishnan2695@gmail.com)
Ph.No:8220991909
Third Year, Department of Information Technology,
Thiagarajar College of Engineering,
Madurai-625015.

ABSTRACT:

Recommender systems generate meaningful recommendations to a collection of users for items or products that might interest them. In short, It is a Information filtering technology commonly used on e-commerce websites that uses collaborative filtering approach. Recommender systems differ in the way they analyze data sources. One such major technique used in analyzing data sources is Collaborative filtering. Collaborative filtering filters information by using the recommendations of other people. One of its category is **Item-To-Item based Approach** which analyzes the user-item matrix to identify the relationships between different items, and then use these relationships to indirectly compute recommendations for users. Amazon, a popular e-commerce site, uses Item-To-Item based Collaborative filtering recommendation. Being a most popular e-commerce website, Amazon needs to support terabytes and petabytes of data, for which a big data analystic tool, Hadoop is used. **Hadoop** is a **Cloud Computing** platform which is used by most of the companies, to provide cloud computing services. It is a software framework for distributed processing of large data sets. With Cloud Computing, we could process a huge amount of data through **Map-Reduce framework** without worrying about the constraints of the resources. The aim of our project is to develop and compare recommendations for purchasing mobiles, which uses the item-based collaborative filtering algorithm, based on Hadoop. We gathered data for the research from a real social portal, where the users can express their preferences regarding the applications. We have also worked based on the information available on the web in the form of ratings, reviews, opinions, complaints, remarks, feedback, and comments about any item (in specific

mobiles). We implemented Hadoop version with the use of the Mahout library which is an element of the Hadoop ecosystem.

Keywords: Recommender Systems, Collaborative Filtering, Item based Approach, Cloud Computing, Hadoop, Mahout.

I. INTRODUCTION

The growth of the websites increases the information and items rapidly. Users are unable to find the relevant information they want. Recommender systems are **mechanisms** that can be used to help **users** to make **purchase decisions**. A recommender system is actually a program that utilizes **algorithms** to predict customers purchase interests by filtering their shopping patterns. Different approaches and models have been proposed and applied to real world industrial applications. The most popular recommendation technique is the **Collaborative Filtering (CF) model**. In CF, previous transactions are analyzed in order to establish connections between users and products. When recommending items to a user, the CF-based recommender systems try to find information related to the current user to **compute ratings** for every possible item. Items with the **highest rating** scores will be presented to the user.

The rating information is very important for obtaining good prediction accuracy, because it precisely indicates user'spreferences and the degree of their interest oncertain items. However, the rating information is not always available . Some websites do not have a rating mechanism and thus their users cannot leave any rating feedback on the products. This situation requires evaluating implicit information which results in a lower prediction accuracy of the recommender systems.

II. LITERATURE SURVEY:

Existing recommendation system recommends mobiles to the user based on the mobile model and the ratings given by that user to the mobile or based on the number ofviews for that mobile. Fuzhi Zhang et al (2010), proposed a **two-stage algorithm** that uses location of the users to predict the interest. V.Mohanraj et al (2012) used the concept of **ontology** to predict the interest of the user. The system was self adaptive and predicted the future browsing pattern of the user. Ozgur Cakir et al (2012) developed a recommendation system using **association rules**. Apriori algorithm is used to generate the rules for recommendation. The basket ratio which is the ratio between the number of items viewed to the number of items added to the shopping cart is increased in this method.

Emmanouil Vozalis et al made an analysis on the types of recommendation algorithms that are in existence. Item-based recommendation is a method in which two users who have rated a item are separated and the **similarity index** is computed among them. When the similarity indexis greater

than the threshold, then similar items are recommended to them. A model which uses Collaborative filtering algorithm for supervised learning was developed.This model classifies even the new unseen item. According to this model, there are only two classes **C1: Like C2: Dislike**. Content-Boosted Collaborative Filtering utilizes Content based Filtering to fill in the missing ratings from the initial user-item matrix. It then employs classicCollaborative Filtering techniques to reach a final prediction.

Types of Recommender Systems

The Collaborative filtering was first coined by Goldberg for email filtering system called Tapestry4 [9]. **Tapestry** was an electronic messaging system that allowed users to rate messages. Tapestry provided good recommendations, but it had drawbacks: The user was required to write **complicated queries.**

The **GroupLens5** generates the automated recommendation system, which provides the users with recommendation on Usenet6 postings. It recommended articles to the users similar to the target user.

Ringo recommender system was developed by Shardanand and Maes and it is used as recommendations for music albums and artists. Here recommendations are done using e-mails. Also video recommendation systems are there, they are also using recommendations through emails.

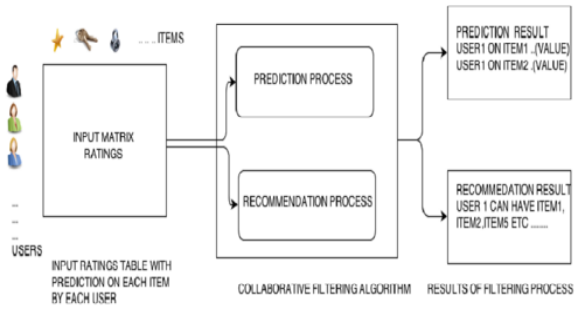
Bayesian network based recommendersystem is represented by decision tree. Information of uses is represented by the nodes and edges. The size of the trained model is very small so that it is very fast to deploy it. It gives accurate as nearest neighbor methods, it does not provide accurate prediction for the frequent changing situation.

Horting is a technique based on graph. Here, the node represents user and the edges represent the similarity degree between two users. Here, the recommendation is made by searching for the nearest neighbor nodes and then combining the scores of the neighbors.

Collaborative Filtering

Collaborative Filtering is the one of the most **successful technologies** of the recommender system. It finds the **relationships** among the **new individual** and the **existing data** in order to further determine the similarity and provide recommendations. It uses **user ratings** of products in order to identify additional products that the new user may like as well. Collaborative filtering techniques are being applied to larger and larger sets of items. The work-flow of collaborative systems is

evident from the figure below[5].



A user expresses the opinions by rating items of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.

1. The system matches this users rating against other users and finds the people with most similar tastes.
2. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user.
3. There are two methods in the collaborative filtering, they are **User-based Collaborative Filtering** andthe **Item-based Collaborative Filtering.** Let's discuss in brief about Item-Based Collaborative Filtering Algorithm.

Item-based Collaborative Filtering Algorithm

To overcome the **problems of** the **user-based**, item-based recommender systems were developed. Item-based recommender is a type of collaborative filtering algorithm that look at the **similarity** between items to make a prediction. The Item-based similarities are computed using column-wise. The work-flow of the item based collaborative filter is as shown in Figure 2 below:

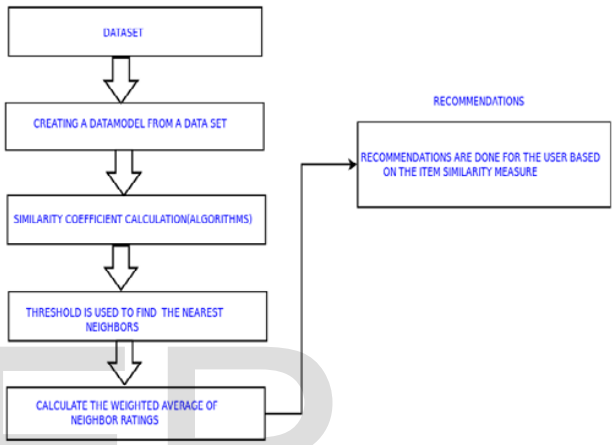


Fig. 2: Item-based Collaborative Filtering Work-flow

Item-based collaborative filtering algorithm is calculated using the **Item-user ratingmatrix** and the **Utility Matrix**. User-item matrix is described as an *m, n* ratings matrix $Rm,,$ where row represents *m* users and column represents *n* items. The element of matrix $ri,,$ means the score rated to the user *i*on the item *j*, which commonly is acquired with the rate of user interest.

The Utility Matrix

In a recommendation-system application there are two classes of entities, which we shall refer to as users and items. Users have preferences for certain items,and

these preferences must be teased out of the data. The data itself is represented as a utility matrix, giving for each user-item pair, a value that representswhat is known about the degree of preference of that user for that item. Valuescome from an ordered set, e.g., integers 1–5 representing the number of starsthat the user gave as a rating for that item. We assume that the matrix issparse, meaning that most entries are "**unknown**." An unknown rating implies that we have no explicit information about the user's preference for the item.

Example :In the below diagram, we see an example utility matrix, representing users' ratings of mobiles on a 1–5 scale, with 5 the highest rating. Blanks representthe situation where the user has not rated the mobile. The mobile model names areSG, SGSD, and SGS4 for Samsung Galaxy Grand , Samsung GalaxySDuos, and Samsung Galaxy S4, NL for Nokia Lumia 520, and MG2,MG3, and Motorola for MotorolaGen G2, MotorolaGen G3 and Motorola. The users are representedby capital letters A through D.

	SG	SGSD	SGS4	NL	MG2	MG3	MG
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

A utility matrix representing ratings of mobiles on a 1–5 scale. Notice that most user-mobile pairs have blanks, meaning the user has not rated the mobile.The goal of a recommendation system is to predict the blanks in the utility

matrix. For example, would user A like MG3? There is little evidence fromthe tiny matrix in the above figure. We can note the similaritybetween MG3 and MG, and then conclude that since A did not like MG2, theywere unlikely to enjoy MG3 either. Alternatively, with much more data, wemight observe that the people who rated both MG2 and MG3 tended to givethem similar ratings. Thus, we could conclude that A would also give MG3 alow rating, similar to A's rating of MG2. This algorithm generates recommendations based on a few customers who are most similar to

the user. It can measure the similarity of two customers, A and B, in various ways; a common method is to measure the cosine of the angle between the two vectors:

$$\text{Similarity}(\bar{A}, \bar{B}) = \cos(\bar{A}, \bar{B}) = (\bar{A} \cdot \bar{B}) / (\|\bar{A}\| . \|\bar{B}\|)$$

The above equation finds customers who are similar to the user, and the cluster models divides the customer base into many segments. The algorithm's goal is to assign the user to the segment containing the most similar customers.

III. METHODOLOGY:

Item-based algorithms are two steps algorithms; 1. The algorithms scan the past information of the users, the ratings they gave to items are collected during this step. From these ratings, similarities between items are built and inserted into an item-to-

item matrix *M*. The elements of the matrix *M* represents the similarity between the items in row *i* and the item in column *j*. The algorithms select items that are most similar to the particular item a user is rating. Similarity values are calculated using the different measures, Pearson, Cosine Coefficient and so on. The next step is to identify the target item neighbors; this is calculated using the threshold-based selection and the top-n technique. Then final step is the prediction from the top-n results. To compute the calculation of the similarity measurement, would consume intensive computing time and computer resources. When the data set is large, the calculation process would continue for several hours. This can be solved by using collaborative filtering algorithm on the **Hadoop platform**.

IV. IMPLEMENTATION:

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 computinghave been the focus to overcome the problem of large scale computation task. Cloud computing is the provisionof dynamically scalable and often virtualized resources as a service over the Internet[5]. Users need not have knowledgeof,expertise in, or control over the technology infrastructurein the cloud that supports them. **Cloud computing services** often provide **common business applications** online that area accessed from a web browser, while the software and dataare stored on the servers.In order to solve scalability problem of recommendersystem, we implement the Collaborative Filtering algorithmon the cloud-computing platform. There are several cloudcomputingplatforms available, for example, the Dryad [6] of Microsoft, the Dynamo [7] of amazon.comand Nettune[8] of Ask.com etc. In this paper, we choose the **Hadoop**platform as the base of our implement. Because the Hadoopplatform [9], [10]is an **open source cloud computing platform**,it implements the **MapReduce Framework** that havebeen successfully evaluated by Google.com. The Hadoopplatform uses a distributed file system, **Hadoop Distributed File System**(HDFS)[2], to provide high throughput accessto application data. Using the Hadoop platform, we caneasily make the program execute in parallel, and the MapReduce framework allows the user to break a big problem formany small problems, then the small problems could be handled by the Hadoop platform, thus improve the speedof computing.As for the implement of Collaborative Filtering algorithmon cloud computing platform, there are few work have beendone. Abhinandan Das and Mayur Datar[2] proposed a Collaborative Filtering algorithm for news recommendation,which is a combining of memory based and modelbasedCollaborative Filtering algorithm. There is also a very closerated project, **Mahout**[4] , which implements the Collaborative Filtering recommendation

system base on Taste[5],a flexible, fast collaborative filtering engine for Java.

V. OVERVIEW OF MAPREDUCE

The MapReduce model is proposed by **Google**. The MapReduce is inspired by the **Lisp** programming language. MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers, collectively referred to as a cluster or a grid. Computational processing can occur on data stored either in a file-system (unstructured) or in a database (structured). MapReduce can take advantage of locality of data, processing data on or near the storage assets todecrease transmission of accumulated data as a part of the reduction.

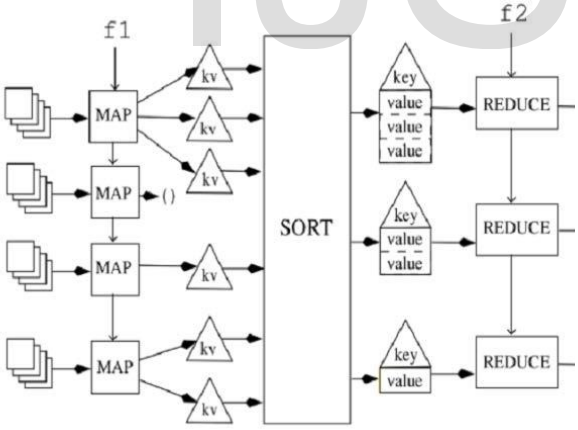


Fig. 3: MapReduce Work-flow

The calculation process of the MapReduce model into two parts, Map and the reduce phase. In the Map, written by the user, it takes a set of input key/value pairs, and produces a set of output key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key and passes them to the Reduce phase. In the Reduce phase, the function also written by the user, accepts an intermediate key and a set of values for that key. It merges together these values to form a possibly smaller set of values [3].

In the Hadoop platform, default input data set size of one mapper is less than 64MB, when the file is larger than 64MB, the platform would split it into a number of small files which size less than 64MB automatically. For the inputfile, the Hadoop platform initializes a mapper to deal with it, the files line number as the key and the content of that line as the value. In the map stage, the user defined process deal with the input key/value and pass the intermediate key/value to the reduce phase, so the reduce phase would implement them [4]. When the files block are computed completely, The Hadoop platform would kill the corresponding mapper, if the documents are not finish, the platform would chooses one file and initializes a new mapper to deal with it. The Hadoop platform should be circulate the above process until the map task is completed. This section explains the working of the collaborative filtering within the MapReduce framework. For making recommendations, the first step is store the txt or .csv files in the Hadoop Distributed File System (HDFS).

Mapping Phase in Collaborative Filtering

The ratings.csv file is stored in the HDFS. The HDFS splits the data and gives that data to each Datanode, and initializes the mappers to each node. The mappers build the ratings matrix between the users and the items at first, the mapper read the item ID file by line number, take the line number as the input key and this line corresponding item ID as the value.

Reduce Phase in Collaborative Filtering

In the reduce phase, the Hadoop platform would generates reducers automatically. The reducers collect the users ID and its corresponding recommend-list, sort them according to user ID.

VI. SCALABILITY: A COMPARISON

Amazon.com has more than 29 million customersand several million catalog items. Other majorretailers have comparably large data sources.While all this data offers opportunity, it's also acurse, breaking the backs of algorithms designedfor data sets three orders of magnitude smaller.Almost all existing algorithms were evaluated oversmall data sets. For very large data sets, a scalable recommendationalgorithm must perform the most expensivecalculations offline. As a brief comparison shows,existing methods fall short:

• **Traditional collaborative filtering** does little orno offline computation, and its online computation scales with the number of customers andcatalog items. The algorithm is impractical onlarge data sets, unless it uses dimensionalityreduction, sampling, or partitioning — all ofwhich reduce recommendation quality.

• **Cluster models** can perform much of the computationoffline, but recommendation qualityis relatively poor. To improve it, it's possible toincrease the number of segments, but thismakes the online user–segment classificationexpensive.

• **Search-based models** build keyword, category,and author indexes offline, but fail to providerecommendations with interesting, targetedtitles. They also scale poorly for customers withnumerous purchases and ratings.

VII. CONCLUSION

Along over two decades of research and commercial development, recommender systems have proved to be a successful technology to overcome the information overload that burdens users in modern online media. According to a survey, 62% of the customers who notice the recommendations purchase the recommended products. The key driver for this success is to provide more relevant recommendation by incorporating customer interest. These recommendations can be provided more accurately by analyzing the features of the product to be recommended and matching it with the interest of the user accordingly.

Recommendation engines are a natural fit for analytics platforms. They involve processing large amounts of consumer data that collected online, and the results of the analysis feed real-time online applications. Hadoop is being increasingly used for building out the recommendation platforms. This paper mainly focuses on evaluating the classification accuracy metrics using the Apache Hadoop and Mahout. By experiments we conclude that, Mahout can handle large amount of structured data, using the machine learning algorithms. Now days, the data size is increasing with the unstructured format, it is not possible to handle with the Mahout. When we combine Apache Hadoop and Mahout for the recommendation, it can recommend large amount of structured and the unstructured data efficiently and fastly

VIII. FUTURE WORK

By using the **Apache Hadoop** and **Mahout**, c large amounts of data can be recommended efficiently. But when it comes to real time, random access is not possible by using Apache Hadoop. Hence, instead of storing the Hadoop sequence file in the HDFS, we can use **Hbase** or **Sqoop** for retrieving the data real time. Also, by combining both the Item-based and the User-based collaborative filtering, recommender system can predict accurate recommendations to user.

[1] G.D. Linden, J.A. Jacobi, and E.A. Benson, Collaborative Recommendations Using Item-to-Item Similarity Mappings, US Patent 6,266,649 (to Amazon.com), Patent and Trademark Office, Washington, D.C., 2001.

[2] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5-53.

[3] Shang Ming-Sheng, Zhang Zi-ke. Diffusion-Based Recommendation in Collaborative Tagging Systems.*Chin. Phys. Lett.*,2009,26(11): 118903

[4] Shang Ming-Sheng, Jin Ci-Hang, Zhou Tao, Zhang Yi-

[5] Vakali, P. Mehra: Guest Editors. Introduction: ”Cloud Computing, IEEE Internet Computing”, 12(5), Sep. 2009.

[6] Isard M, Budiu M, Yu Y, Birrell A, Fetterly D. Dryad: Distributed data-parallel programs from sequential building. blocks. In: Proc. of the 2nd European Conf. on Computer Systems *(EuroSys).*, 2007. 59 72.

[7] DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W. Dynamo: Amazons highly available key-value store. In: Proc. of the 21st ACM Symp. on Operating Systems Principles. New bioff H, Leung ST. The Google file system.York: ACM Press, 2007. 205 220. ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming. New York: ACM Press, 2003. 119 130.

[8] Dean J, Ghemawat S. Distributed programming with Mapreduce. In: Oram A, Wilson G, eds. Beautiful Code.
Sebastopol: OReilly Media, Inc., 2007. 371 384.
[9] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2005,51(1):107 113.