Efficient Recommender System using Collaborative Filtering Technique and Distributed Framework

Sonali B. Ghodake[1], Ratnamala S. Paswan[2]

[1]*Pune Institute of Computer Technology, Pune, India.*
[2]*Pune Institute of Computer Technology, Pune, India.*

---***---

Abstract - *In recent years there has been a dramatic increase in the amount of online content. Recommender systems form a specific type of Information Filtering (IF) technique. To date a number of recommendation algorithms have been proposed, where collaborative filtering is one of the most famous and adopted recommendation technique. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation. In the last decade, the amount of customers and online information has grown rapidly, yielding the big data analysis problem for recommender systems. Consequently, traditional recommender systems often suffer from scalability and inefficiency problems when processing or analysing such large-scale data. Due to this, the implementation of these algorithms on single node machine is time consuming and fail to meet the computing requirement of large data sets. Distributed processing of big data across multiple clusters of nodes can help to improve the performance in such cases. In this paper, the former collaborative filtering recommendation algorithm is designed to parallel on MapReduce framework and uses Pearson correlation as similarity metric. Apache Hadoop is parallel distributed framework. Hadoop distributed file system(HDFS) allows distributed processing of big data across multiple clusters of nodes.*

***Key Words*: Recommendation, Collaborative filtering, Pearson correlation, Apache Mahout, Hadoop**

1.INTRODUCTION

On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services. It makes use of content based, collaborative filtering or hybrid approach. Collaborative filtering technique is most prominent approach to generate recommendations. We require a robust system to handle large amount of data. It was defined as,” Recommender systems form a specific type of information filtering (IF) technique that attempts to present information items (e.g. Books, news, movies, music, images, web pages, etc.) that are likely of user’s interest”. For example, we can recommend any product, locations and services (books, videos, music, TV programs, documents, research resources and website). We require a robust system to handle these large amount of data (also called Big Data). Distributed processing of big data across multiple clusters of nodes can help to improve the performance in such cases. Hadoop is an open source framework. To process very huge amount of data it develops and executes distributed applications.

” Big data is a term that describes large volumes data-both structured and unstructured that require advanced techniques to enable the capture, storage, distribution, management, and analysis of the information” [3].

The 3 Vs of Big Data management:

1.Volume: There is large amount of data than ever before, size of data is continuously increasing.

2. Variety: Variety refers to the different forms of data, such as text data, sensor data, audio data, video data, graph, and many more.

3.Velocity: Velocity refers to the data processing speed. Data is continuously coming and our interest is in obtaining useful information from streams of data in real time [7].

2. RELATED WORK

Prem Melville and Vikas Sindhwani [1] in their research paper present the basic concept of recommender system and its different approaches. It also gives the Pros and Cons of different approaches of recommender system.

Satya Ranjan Dash and Satchidananda Dehuri [2] in their research paper present comparison of different classification techniques such as Bayesian classification, Classification by decision tree. They use different parameters for comparison of different algorithms such as RMSE, ROC Area, MAE, Kappa Statistics, time taken to build the model, Relative Absolute Error, Root Relative Squared Error, and the percentage value of classifying instances.

Mustansar Ali Ghazanfar and Adam Prugel-Bennett [3] in their research paper proposed a switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. The algorithm proposed in this paper is scalable and provide better performance in terms of accuracy and coverage than other algorithms. It also eliminates some recorded problems with the recommender systems.

Dr. G. R. Bamnote, Prof. S. S. Agrawal [4] in their research paper explore different recommender system algorithms such as user based collaborative and item based collaborative filtering using Apache Mahout.

Mr. Sachin Walunj, Mr.Kishor Sadafale [5] in their research explains why selecting a foundational platform is an important step in developing recommender systems for personal, research, or commercial purposes. They present implementation of a mahout framework, which provide flexibility in using pre-existing algorithms. As it is built on the Hadoop framework, it solved the problem of scalability.

Nitesh V. Chawla and Darcy A. Davis [7] in their research paper present a Big Data driven approach towards personalized health care and demonstrate its applicability to patient centred outcomes and its meaning full use. They use collaborative filtering method for future recommendation.

Xingyuan Li. [8] in their research proposed an improved collaborative filtering approach - Cluster based collaborative filtering recommendation algorithms which improve scalability of collaborative filtering algorithms and reduce data sets sparse of the recommended system.

Kala Karun. A, Chitharanjan K. [11] in their research paper suggest some of the major enhancements to Hadoop especially in data storage, processing and placement. They also explain how system becomes fault tolerant and scalable.

Marios et al [12] in their research paper proposed a collaboration based recommender in an Internet of things environment. Their approach relies on user to object space-time interaction patterns.

3. MOTIVATION

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. Information retrieval systems, such as Google, Devil Finder and Altavista have partially solved this problem but prioritization and personalization of information were absent. This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user's preferences, interest, or observed behaviour about item [2]. Collaborative filtering algorithm is the most widely used algorithm in recommender system. As scale of recommender system continues to expand, the number of users and items of recommender system is growing exponentially. As a result, the single-node machine implementing these algorithms is time consuming and unable to meet the computing needs of large data sets. To improve the performance distributed processing of big data across multiple clusters of nodes is needed.

The present recommender system works well on small data sets. The difference in the implementation of recommender depends upon how they analyse the big input data to recognize the similarity between users and items that indicates the relevant preferences for that user. Generic user based recommender works with data model encapsulating recommender input data in Apache Mahout which is extensible data mining library. The recommender system framework can use any similarity metric. The reason for choosing Pearson correlation is that the computation would be fast. Hadoop software library allows distributed processing of big data across multiple clusters of nodes.

4. RECOMMENDER SYSTEM

It was defined as, "Recommender systems form a specific type of information filtering (IF) technique that attempts to present information items (e.g. movies, music, books, news, images, web pages, etc.) that are likely of interest to the user" [1].

To accomplish the task of recommendations, the recommender systems usually employ any of the following recommendation approaches:

Content-Based Filtering:

The particular user's ratings of other similar resources to item.

Collaborative Filtering:

Similar users' rating of that resource or item.

Hybrid Approaches:

It is a combination of different approaches and techniques, basically combining collaborative and content based filtering.

1. Collaborative Filtering

Collaborative filtering methods are based on collecting and analysing a large amount of information on users' behaviours, activities or preferences and predicting what user will like based on their similarity to other users'.

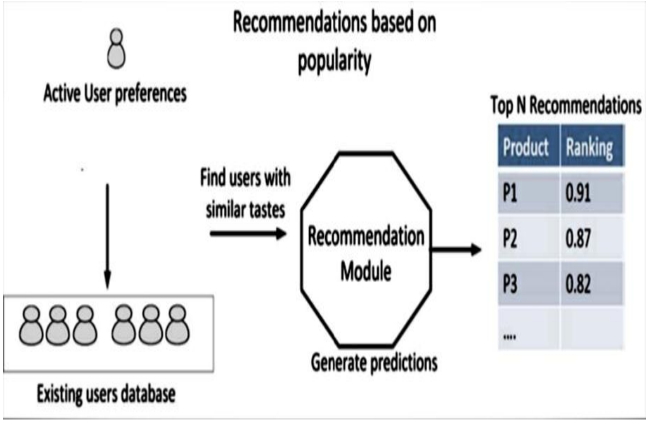


Fig -1: Recommendations based on popularity

Recommendations can be built using any of the following ways consisting of user, item and rating as inputs [1]:

- **User Based Recommendations:**

  Here, users with identical characteristics and terminologies are used as inputs to build user based recommendations.

- **Item Based Recommendations:**

  In this recommendation system, similar items together institute the computation of item based recommendation.

- **Slope-one:**

This recommendation system does not consider similarity metric as standard component. It is fast and simple approach for item recommendation.

The proposed system makes use of user based collaborative filtering technique for recommendations.

5. DISTRIBUTED FRAMEWORK

Apache Hadoop is an open-source parallel distributed framework. It allows to store and process big data in a distributed environment across clusters of computers. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provide file system and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS):

A distributed file system that provides high-throughput access to application data.

Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. HDFS uses master slave architecture's file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write opera-tion with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode[5].

The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

The Map Task: This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).

The Reduce Task: This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task [11].

Typically, both the input and the output are stored in a file-system.

6. APPACHE MHOUT

Apache Mahout is an open source project. It provides free implementation of scalable ma-chine learning algorithms. Three major ma-chine learning techniques are [4]:

Recommendation.

Classification.

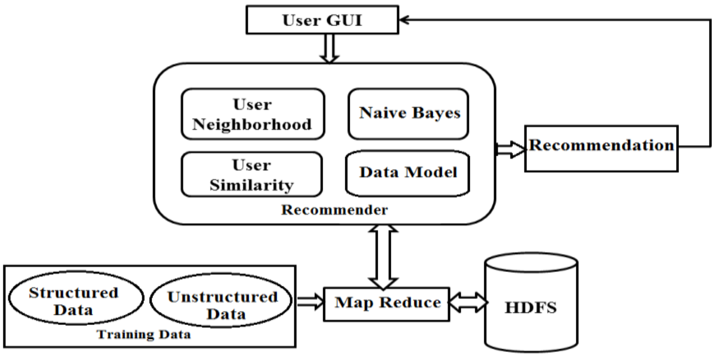
Clustering.

Apache mahout is not restricted to Hadoop-based implementations but it can be implemented on top of Apache Hadoop using the map/reduce paradigm. The extensible collective intelligence library helps us to implement various recommendation algorithms. It sup-ports Distributed Naive Bayes and Complementary Naive Bayes classification implementations. Various collaborative frameworks are also provided to generate recommendations. It enables to start exploring machine learning algorithms easily [5]. Fast prototyping and evaluation can be considered as two main strengths of mahout.

7. PROPOSED FRAMEWORK

In this proposed system, collaborative filtering technique is implemented on Hadoop to improve the performance of recommender system.

FIG 2: Architecture of implementation of Recommender system using Hadoop.



The figure shows architecture of implementation of Recommender system using Hadoop. It utilizes open source platform of apache mahout with Hadoop. This implementation is plat-form independent and performs distributed map reduce computation. Apache Hadoop with Mahout is most suitable for implementation of large scale and distributed generic recommender systems. Classification has one of the major role in Recommendation system. This System is developed for healthcare application. It Recommend best hospitals and doctors to the user/patient based on user/patients' information [6]. Input to the system is users' data i.e. Age, Gender, Location, Budget, Disease, Urgency. Input file is stored on HDFS. Based on input data naive bayes classifier gives the probability of users' cureness, if it is 'YES' then user will proceed for getting recommendation. Collaborative filtering is performed on user input, then top rated hospitals and doctors are recommended to the user.

This system basically works on two main algorithms which are as follows:

1. **Naive Bayes Classifier.**
2. **User Based Collaborative Filtering Technique.**

7.1. Naive Bayes Classifier

The concept of classification plays a significant role in recommendation systems. Naive bayes classifier is the most suitable algorithm as it maximizes the predictive accuracy.

Let D be a training set of tuples and their associated class labels. Each Tuple is an 'n' dimensional attribute vector.
Given features X = X1, X2, X3. Xn
Predict a class label C.

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

record is classify into particular class without knowing any initial conditions.

Calculate likelihood i.e. probability that given record will classify into particular class based on input conditions.

Flike= P(X/Ci)

Calculate evidence i.e. probability that given record is from input dataset.

Fevei =P(X)

Calculate posterior probability of each feature for each class.

Fpost =P(Ci/X)

P(Ci/X) =P(X/Ci) * P(Ci) /P(X)
Maximize P(X/Ci) * P(Ci) as P(X) is constant.

F_class = predicts X belongs to Class Ci iff

P(Ci/X) > P(Cj/X) for i != j .

7.2. User Based Collaborative Filtering technique

It generates recommendations on the basis of similarity among users. It consists of five primary components that work with Users, Items and Preferences (Rating): \\

- **Data Model Creation:**
The recommender input data is encapsulated in mahout and the storage is done as resulting preferences. All the data the recommender systems need access to is provided by data model implementation.

- **User Similarity:**
There are many implementations to define similarity within Mahout [6]. Pearson correlation similarity metric is used in the proposed recommender system framework.

The Pearson correlation has an integer value between –1 and +1. It computes the tendency of the numbers to move together proportionally. The values in a particular series and another series undergo a linear relationship. The Pearson correlation is approximately 1 when the computed tendency is high. The Pearson correlation similarity is more reliable if the result is based on big data.

Let a, b : Users for which the coefficient is being calculated.
P: Set of items, rated both by a and b.

$$sim(a,b)=\frac{\sum_{p\in P}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum_{p\subset P}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum_{p\subset P}(r_{b,p}-\bar{r}_b)^2}}$$

Ra,p and rb,p are individual ratings from a and b for p, and ra and $\bar{ r_b }$ are
Possible similarity values between -1 and 1.

User Neighbourhood:

Compute a neighbourhood of similar users near a given user that can then be used by the Recommenders.

Recommendation:

It is responsible for actually making the recommendations to the user by determining users with similar test. All these components are dragged together for developing recommendations for users in the pro-posed system. The input file is transformed into comma separated file(.csv) using Hive in the ETL step which helps enhance the performance of the recommendation sys-tem.

8. EXPERIMENTAL SETUP

The experimental setup for the proposed sys-tem will consider Hadoop setup and environment build, installation and setup of Hive Module, MLA module for running algorithm and extracting information, GUI Module for taking input from user and displaying output.

The particulars about platform and technology are as follows which are used to build proposed system:

OS: Ubuntu 15.10.
Build Environment: Java.
Data/File Storage: HDFS.
External database: MySQL.

9. IMPLEMENTATION RESULTS

Each time you have to write following commands and run Hadoop on terminal: start-all.sh: This command is used to start working of all the daemons.
jps: It is java virtual machine process status tool. It is used to see the number of daemons running on your local machine.

stop-all.sh: This command is used to stop the working of all Hadoop daemons.

10. IMPLEMENTATION ANALYSIS

10.1 Comparison Between Execution Time Required for Hadoop and MySQL Database

This section provides the performance and results of the developed system. Experiment has been carried out for dataset size ranging from 50 MB-200 MB. For each data size processing time is recorded in seconds. Table 10.1 shows a comparison between the time required by Hadoop and MySQL. Speedup is adopted to measure the performance of the system. Speedup refers to how much a parallel system (Hadoop) is faster than a corresponding sequential system (MySQL), which can be defined as follows:

$$S= T_1/T_p$$

where, T_1 is the sequential execution time (MySQL), T_p is the parallel execution time (Hadoop). As shown below, Table 10.1 present CPU execution time required by Hadoop and MySQL with different data size. The last column of the table shows speedup obtained by Hadoop over MySQL database.

From Table 10.1, it is observed that the speedup of the system on Hadoop increases relative linearly with the increase in data size.

Sr. No.	Data Size(MB)	MySQL Time(sec)	Hadoop Time(sec)	Speedup
1	50	10	14	0.76
2	70	25	20	1.2
3	90	36	22	1.6
4	110	26	13	2.8
5	130	120	40	3
6	150	210	55	3.8
7	170	250	64	3.9
8	190	270	66	4

TABLE 1: Comparison between Hadoop and MySQL

10.2 Performance Testing with Graph

Graphs in figure 10.1 show comparison between the time required by Hadoop and MySQL. The graph is drawn for data size ranging from 50MB-200MB. The blue graph shows results obtained by MySQL. Red graph shows results obtained by Hadoop.

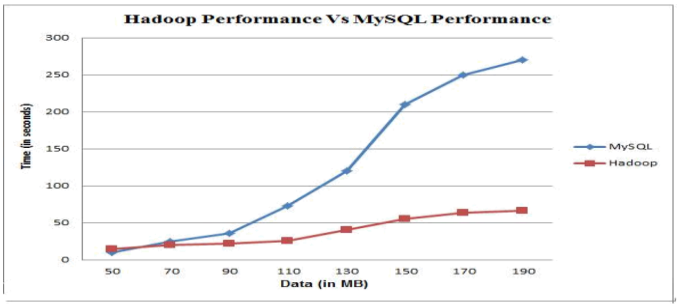


Chart 1: Hadoop Performance vs MSQL Performance

From this graph it is observed that for small to medium data size Hadoop require almost the same time. Hadoop performance is much lower than MySQL for small data size. So, to get Hadoop performance, dataset size must be larger. It is observed that with the increase in data size, time taken by Hadoop decreases and in case of MySQL, as the data size increases, the time taken also increases.

11. CONCLUSION

Data in the form of reviews, opinions, feed-back, remarks, and complaint treated as Big Data cannot be used directly for recommendation system. This data first need to be filter/transform as per requirement. Thus, through this paper overview of need of recommendation using distributed framework is done. Proposed system makes use of user based collaborative filtering technique for recommendation, naive bayes classifier for classification and Hadoop is used as distributed framework. Apache Mahout framework provide flexibility in using pre-existing algorithms. As it is built on the Hadoop framework it solves the problem of scalability. Because of Hadoop, system is highly scalable, fault tolerant and it can handle data set of million ratings on single node. The proposed implementation is platform independent. Experimental results demonstrate that proposed system significantly improves the performance and scalability of recommender system over existing approach.

REFERENCES

[1] Prem Melville and Vikas Sindhwani, Recommender Systems, IBM T.J. Watson Research Center.

[2] Satya Ranjan Dash, Satchidananda Dehuri2, Comparative Study of Different Classification Techniques for Post Oper-ative Patient Dataset, International Journal of Innovative Research in Computer and Communication Engineering ,Vol. 1, Issue 5, July 2013.

[3] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering, IEEE 2013.

[4] Dr. G. R. Bamnote Prof. S. S. Agrawal, Evaluating and Imple-menting Collaborative Filtering Systems Using Apache Mahout, International Conference on Computing Communication Control and Automation,2015.

[5] Mr. Sachin Walunj,Mr.Kishor Sadafale, An Online Recom-mendation System for E-commerce Based on Apache Mahout Framework, June 1, 2013.

[6] Lavannya Bhatia, S.S.Prasad, Building a Distributed Generic Recommender Using Scalable Data Mining Library , IEEE International Conference on Computational Intelligence Communication Technology,2015.

[7] Nitesh V. Chawla, PhD1 and Darcy A. Davis, Bringing Big Data to Personalized Healthcare: A Patient-Centered Framework
, June 25, 2013.

[8] Xingyuan Li , Collaborative Filtering Recommendation Algo-rithm Based on Cluster, 2011 International Conference on Computer Science and Network Technology .

[9] Zhangguang Qian, Liuji Qing, Zhangrui Xue A , Collabora-tive Filtering Recommendation Algorithm based on Correlation and Improved Weighted Prediction, 2011 IEEE.

[10] Hao Ma, Irwin King and Michael R. Lyu, Effective Missing Data Prediction for Collaborative Filtering, ACM July 2327, 2007.

[11] Xingyuan Li , A Review on Hadoop HDFS Infrastructure Extensions, Proceedings of 2013 IEEE Conference on Infor-mation and Communication Technologies (ICT 2013).

[12] Mario Muoz-Organero,Gustavo A. Ramrez-Gonzlez, Pe-dro J. Muoz-Merino and Carlos Delgado Kloos, A Collabo-rative Recommender System Based on Space-Time Similarities , IEEE CS 2010.

[13] Bankim Patel,Atul Patel,Atul Patel, Big Data Analysis: Recommendation System with Hadoop Framework, 2015 IEEE International Conference on Computational Intelligence Communication Technology .

[14] Martin Wiesner and Daniel Pfeifer, Health Recommender Systems: Concepts, Requirements, Technical Basics and Chal-lenges , Int. J. Environ. Res.Public Health, 2014.

[15]Ratnamala Mantri, Rajesh Ingle and Prachi Patil, SCDP: Scalable, Cost–Effective, Distributed and Parallel Computing Model for Academics, ICNCS, Volume:5, 77-80, ISBN:978-1-4244-8677-9, IEEE 2011.

[16]Ratnamala Mantri, Ashwini Jawalikar,\textit{Implementation and Performance Analysis of Academic$_$MapReduce Algorithm (AcdMR) , International Journal of Computer Applications, Volume:121, Issue:19, July 2015.

[17]Sonali Ghodake, Ratnamala Mantri, Survey on Recommender System
using Distributed Framework, International Journal of Science and Research (IJSR), Volume:5, Issue:1, January 2016.