A Time Efficient Multi-Node Clustering Approach in Recommendation System

Sabbir Hossain

2015-2-60-111

Faisal Ahmed

2015-2-60-019

Asifuzzaman Miah

2015-2-60-012

A thesis submitted in partial fulfillment of requirements for the degree of
Bachelor of Science and Engineering

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh
December, 2019

Declaration

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by us under the supervision of **Mahamudul Hasan Munna**, Lecturer, Department of Computer Science and Engineering, East West University. We also declare that no part of this Thesis/project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

....

(Mahamudul Hasan Munna)

Supervisor

Signature

....

(Sabbir Hossain)

2015-2-60-075

....

(Faisal Ahmed)

2015-2-60-019

....

(Asifuzzaman Miah)

2015-2-60-012

Letter of acceptance

This project entitled" A Time Efficient Multi-Node Clustering Approach in Recommendation System " submitted by Sabbir Hossain, ID: 2015-2-60-111, Faisal Ahmed, ID: 2015-2-60-019 and Asifuzzaman Miah, ID: 2015-2-60-012 to the Computer Science and Engineering Department, East West University, Dhaka-1212, Bangladesh is accepted as satisfactory for partial fulfillment of requirements for the Award of Degree of Bachelors of Science (B. Sc.) in Computer Science and Engineering on December, 2019.

Supervisor

....

(Mahamudul Hasan Munna)

Lecturer,

Department of Computer Science and Engineering,

East West University

Chairperson

....

(Dr.Taskeed Jabid)

Chairperson and Professor,

Department of Computer Science and Engineering, East West University

Acknowledgement

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions with and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express my thankfulness other than simply listing those people who have contributed to this thesis itself in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express our deepest gratitude to the almighty for His blessings on us. Next, our special thanks go to our supervisor, Mahamudul Hasan Munna, who gave us this opportunity, initiated us into the field of Machine Learning, and without whom this work would not have been possible. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our B.Sc. study were simply appreciating and essential. His ability to muddle us enough to finally answer our own question correctly is something valuable what we have learned and we would try to emulate, if ever we get the opportunity.

We would like to thank Sir for his overall support and his valuable suggestion during our working session. Last but not the least, we would like to thank our parents for their unending support.

There are numerous other people too who have shown me their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our heart and hope to find a more appropriate place to acknowledge them in the future.

Sabbir Hossain
2015-2-60-111

Faisal Ahmed
2015-2-60-019

Asifuzzaman Miah
2015-2-60-012

Abstract

In the era of Information and communications technology where data is fueling the growth of organizations, where companies ingest raw data in massive volumes from from countless sources. But the question is how can they examine the data which both insightful, useful meaningful. This is where Big Data comes to play. Apache Spark is an open-source framework that is used process Big Data. Apache Spark is the leading platform for large- scale SQL, batch processing, stream processing , and machine learning. But one of the major drawbacks is that the time taken for processing traditional algorithms is much longer and it is also difficult to process large volume of data. Here Apache Spark multi-node clustering comes into big rescue. Which is a collection of independent machines connected through a dedicated network to work as a single centralized data processing resource. Collaborative Filtering is becoming so popular now a-days. To handle huge data sets traditional recommender systems often face challenges. In order to overcome some difficulties, some restrictions we have implemented distributed approach to do parallel computing so that we can deal with big datasets. We used Apache Spark Multi Node Clustering to do this. We have used several clustering algorithms find the similarity between users. Finally, we compare the overall permanence for a single machine vs multi-node clustering machines. In terms of scalability Apache Spark maintains great position. We consider improving scalability, Robustness of the system and evaluation Parameters. We also implemented Clustering algorithm using PySpark. The work of Apache PySpark is used to elaborate efficient parallel Implementation of our recommendation system.

Content

Chapter 3 Literature Review

Chapter 4 proposed Method

Chapter 1

Introduction

1.1 Introduction

Now-a-Days Collaboration Filtering is becoming popular. Recommender system is being used in many tech industries to solve numerous problems. In this Tech world vast amount of data is generated in every tech-related platform. So dataset are becoming large to handle data and recommendations scheme we have to use distributed approach. Distributed approach nothing but divided the work in multiple computers that we can get faster outputs and less computation time. For this we have used Apache spark. We have simulated our work on pyspark environment. Recommendation can come handy because it deals with user-user collaboration which is identified by rating or similarities. Our proposed model is desired to solve basic problems in recommendation system which is run time problem. The problem is scalability which happened because of large dataset. As we are implementing larger data set and distributed approach so we ensure solution of scalability problem of recommendation system.

1.2 Recommender System

A recommender system is an information system that is used to predict the best items or products to the users according to their past behavior and possibly using other kinds of data. Generally, users assign to their ratings to an item that he or she has used or explored and it is the salient source of data for recommender system. By utilizing that rating data recommender system produces a rating prediction model, and then by using that model recommendation has been made to the users for further items according to the user preference. Fig 1.1 represent

Recommender system model.

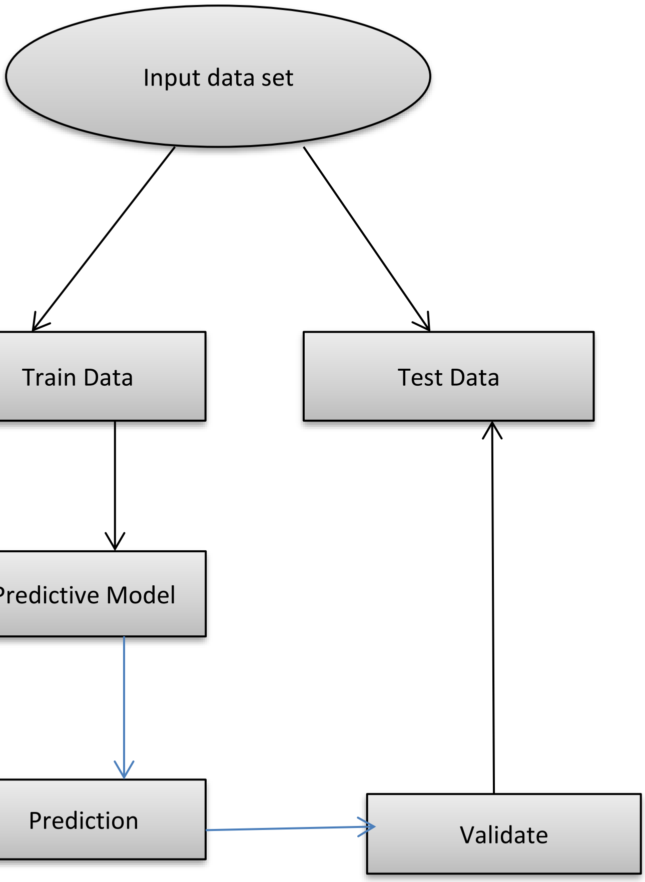


Figure 1.1: Recommender System

1.3 Motivation

The motivation about our time efficient recommender approach are described below.

The novelty of this work lies in designing multi node clustering system for the purpose of reducing run time problem of recommender system. One of the problem of recommender system is run time problem. For vast data set late run time problem is accrue.

The chapter of performance evaluation we showed we can reduce run time problem by multi node clustering system using apache spark.

1.4 Problem Statement

From a survey it is reported that more than 1.17 billion people in the world is using internet now a days. From this people great portion is using online community, eCommerce site, video streaming site. All those sites are using recommender system. These system usually take the users profiling approach by allowing a user to select his interested subjects. One of the problem of this system is run time problem. If recommender system algorithm apply in a very vast amount of data set then run time is not very efficient . so we are working on multi node clustering system, which is very efficient and can reduce the run time problem. We are connected multiple computer to create a multi node clustering recommender system.

1.5 Contribution

We are using apache spark for create multi node and handle the data set. Spark is an open source big data analytics framework which solves algorithms and parallel computing. Comparing to another big data analytic giant Hadoop it is much more wider range of feature and functionality. Hadoop's main functionality which is Hadoop MapReduce has some lacking such as dealing with very large dataset and also computation time is big[]. The reason for spark faster execution is the use of resilient distributed dataset(RDD). It is an distributed collection of nodes of the clusters in parallel. Another factor involved in spark execution is lazy evaluation. In this method, the actions are evaluated and transformations are stored for future extractions.

We are also using anaconda distribution for using python language to implement some algorithm. Anaconda distribution is open source platform[] to implement python and machine learning on different types of operating system like Windows, Linux, Mac os etc.

1.6 Thesis outline

Chapter 1: This chapter represent the introduction of our work, recommender system, motivation, problem statement, and contribution

Chapter 2 : This chapter represent the background study of our thesis

Chapter 3 : This chapter represent the literature review or related word of recommender system

Chapter 4: This chapter represents proposed method of our thesis.

Chapter 5: This chapter represents performance measurement.

Chapter 6: This chapter represent conclusion, summary and future work

Chapter 2

Background study

2.1 Introduction

Recommendation is a social process through people close to us suggest movies, songs, sports songs, food etc. This process has become a useful component in today's world because of the massive growth Information in internet. Recommendation in internet world is not explicit rather its implicit. The people Are not likely get suggestion directly from their review and peers. Recommendation is computational process which helps users to generate recommendation for them by identifying many clustersautomatically who are same category like them. The system takes users in a same cluster when their choices or preferences fall in same characteristics. But now-a-days everyone have access to everyone's Preferences because of rapid growth of technology and new kind of technology. So recommendations Provides a service by measuring similarity of a particular user and group of users computationally.

2.2 Ratings and User profiles

The main point of recommender systems is the data of user-rating which the system depends on and operates. Before introducing the desired algorithms of recommender systems , we define the terms related to this data will be used throughout this thesis.

- **User** : The user of the system whom we will give recommendations
- **Item** : an item is defined by the content of the system, contents can be books, movie, product,web pages etc.
- **Rating :** Rating can be a numerical representation which is given by a user's preferences for a particular item.
- **User-Item Matrix** : we can represent the ratings of the users by a matrix. Where column are the items and users in the rows. Recommender systems will use this matrix to evaluate recommendation to users. Collection of ratings Can be done implicitly or explicitly.

2.3 Information Gathering From User

Rapid growth of Internet provides users the capability to choose among a diversity of information The information can be their profession, lifestyle, status etc. These information can come from different sources. For example web pages, articles, journals, emails, news or many multimedia sites. The users get many benefit from the sources written above but eventually the failed to handle it. There are many techniques to gather information from users. Recommender system is ingenerally connected with this information gathering.

2.4 Recommender System

In recent years Recommender systems have started provide a technological proxy for social recommendation process. This System tells whether a user will like a particular product. So the systems actually predicts the outcome. Recommender system has been introduced in Web stores, online groups, media journals etc. Now a days people more focusing on e-commerce sites for business purpose or communications, where recommender systems are explicitly used to predict items to the users or customers.

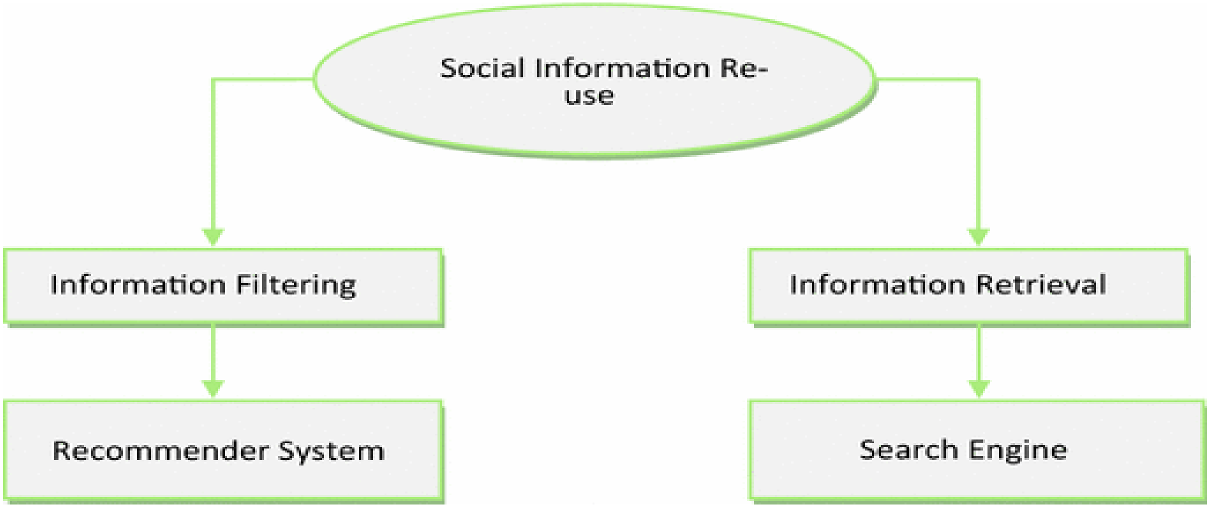


Figure 2.1: Recommender System for Social Information

2.5 Requirements For Recommender Systems

To make recommendations there are some things need to be considered, those are given below

- **Background Information:** Before Recommendation process starts system will have the information.
- **Input:** when user provides information to system recommendation will trigger.
- **Algorithm:** this will combine above two procedures and embed on recommendation system

2.6 Recommender System Classification

Recommender systems can be classified as following category:

- **Content Based Recommender System**
- **Collaborative Filtering based Recommender System**
- **Knowledge based Recommender System**
- **Hybrid Recommender System**
- **Demographic Recommender System**

2.7 Content-Based Recommender System

Content based recommender system recommends a product or item to a user based upon the features or properties of that item. For example if a recommender system predicts a movie for a user then recommender system will predict based upon movie category like action, thriller or drama, then movie duration etc. Users will give information in the system about the description of an item and recommender system will take that as an input and start processing recommendations for that user. The recommender system will compare the inputs from various users and then make predictions to that user. Users preferences can be changed so recommender system is built such a way that it will updated automatically based on user's feedback.

2.8 Collaborative Filtering Recommender System

Collaborative filtering recommends something based on user to user ratings. There are some lackings in content based recommender system which is automatic information processing, collaborative filtering developed to overcome this issue. Collaborative filtering works with community of users who gives ratings or feedback to a particular item and Recommendations are done for the current user by matching similar ratings from other users. By this way the users with similar rating can form a community. CF is different than other methods because its directly focused on information evaluation instead of analysis. It categorizes the information from users by their opinion instead of information itself.

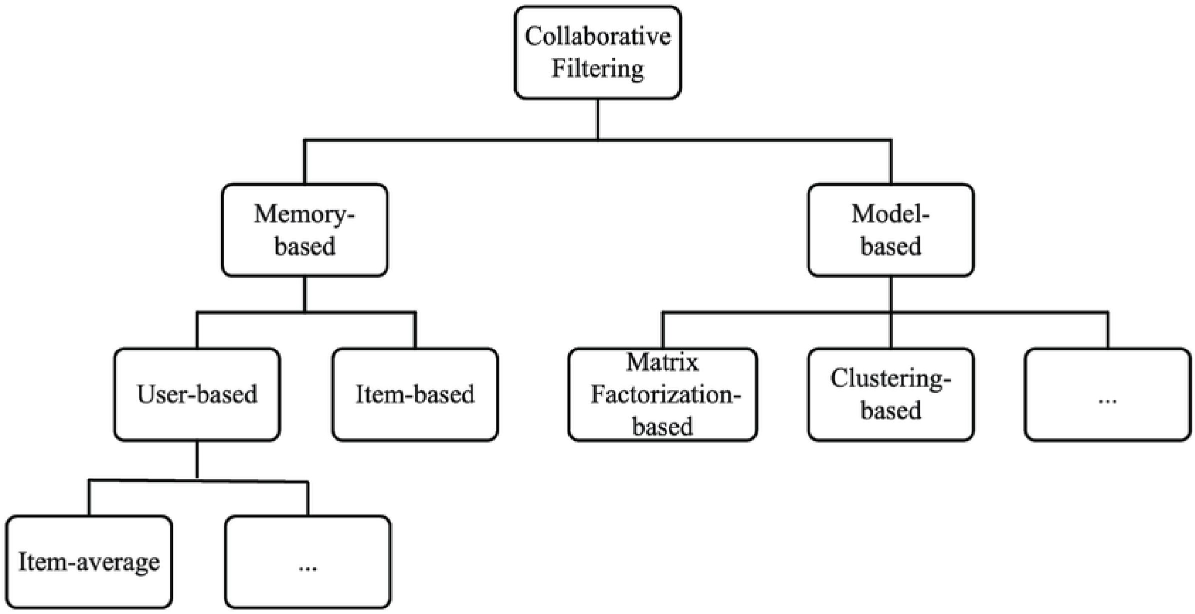


Figure 2.2 : Collaborative Filtering Recommender System

2.8.1 Memory- Based Collaborative Filtering

Memory based CF have reached different level of popularity because they are intuitive and simple. In conceptual level it avoids the complications for a expensive mdoel- developing stage. There are some lackings:

- **Scalability :** Most memory based CF require high level of computations according to the growth of users and items data. So for the increase of data millions of users and items will affected with problems such as scalability.
- **Sparsity :** Memory based CF needs to evaluate huge datasets. Active users can get recommendations for an item which for other user. As a result accuracy of recommendation can be poor.

Memory based CF has two types of approach:

- **Item-Item Based Collaborative Filtering**
- **User-User Based Collaborative Filtering**

2.8.1.1 Item-Item Based Collaborative Filtering

When handling datasets of million users for different domains, there arises some challenges such as scanning vast number of e-commerce sites makes the system slower and also the system unable to predict in real time. So many sites have implemented a new technique called item based recommendation, which focuses pre-processing in offline. The main thing which differs this technique from others is this algorithm evaluates predictions based on the similarity of items not the similarity of users.

2.8.1.2 Model Based Collaborative Filtering

This method recommends item by developing model from user ratings. This method uses a probabilistic approach for computing expected value of a user prediction. Model building approach can be done by some machine learning algorithms such as clustering, rule based and Bayesian network. Clustering model do classification problem by clustering same users under same class and predicting probability.

2.9 Hybrid Collaborative Filtering

Hybrid recommender system is nothing but combination of content based and collaborative based filtering techniques. This method overcomes the shortcomings of content based and collaborative based techniques.

2.10 Knowledge-Based Recommender System

Knowledge-based recommender system implies the knowledge about items and users and finds out what items will fulfill their requirements. Users have freedom to explore and investigate the information space, by doing that they keep updating their needs.

2.11 Demographic-Based Recommender System

Demographic Based Recommender System classifies the user depending on their personal information or attributes and makes predictions according to demographic classes. This approach is indirectly related to statistical models. This method shows the same advantages and disadvantages like knowledge based recommender system which discussed previously.

2.12 Challenging Issues

There are many challenging issues found on recommendation systems such as:

- **Lack of Data:** Not all users can be available for a desired system. Many users can skip entering information, their preferences, likes and dislikes.

- **Unpredictable Results :** Sometimes results may be irrelevant for the system or unexpected for the system.
- **Updating Users Preferences : -** there are some data which are mandatory from users but also need to be updated over time.
- **Data Changing :** recommendation system often find trouble to keep pace with user's tastes and their opinions.
- **Overhead :** whole system might look simple to implement but core computations are lot for the recommender system.

2.13 Cold Start Problem

Cold start problem is considered as the prime problem for a recommender system. More Precisely, it tells about the fact that the system fails to give any conclusion or prediction when an item or user doesn't provide efficient information.

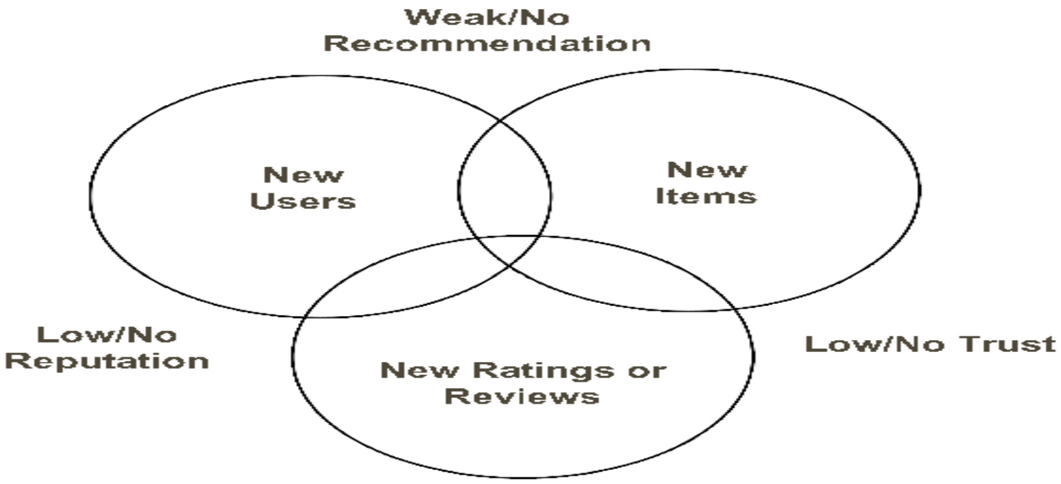


Figure 2.3 Cold Start Problem

This system involves with automated data modelling. System doesn’t have sufficient information about users so system start up with slow preprocessing which is heinous for the system.

There are three cases at cold start algorithm, those are given below.

- **New Item :** The item cold-start problem refers to when items added to the catalogue have either none or very little interactions. Content based algorithm are less prone to this problem because recommenders choose which items to recommend based on the features that item possess so if there are no interactions of new items still recommender can made predictions

- **New Community:** New community problem or called as systematic bootstrapping refers to startup of the system. Recommender have no information to rely on present
- **New User :** When a new users enrolls in the system the for a certain period of time the recommender has to provide recommendation without depending on the users past interactions. Since the users have face recommendations with poor quality he might stop using the system so that recommender cant recommend him anything.

2.14 Apache Spark Architecture

Apache Spark frame work architecture uses master slave which consists of a driver. This driver run as master node and many executors run as worker node in the cluster. Apache spark can use batch processing and real time processing. Driver program calls the main program of applications help by the Sparkcontext object. The Spark Driver also contains Different types of other components like DAG Scheduler, Task Scheduler, Backend scheduler, and Block Manager which are working for translating the user written code into jobs which are actually executed on the cluster. The spark driver working with the cluster manager to clustering the job into various other jobs . . Cluster manager does the allocation work and split the work into multiple worker node.

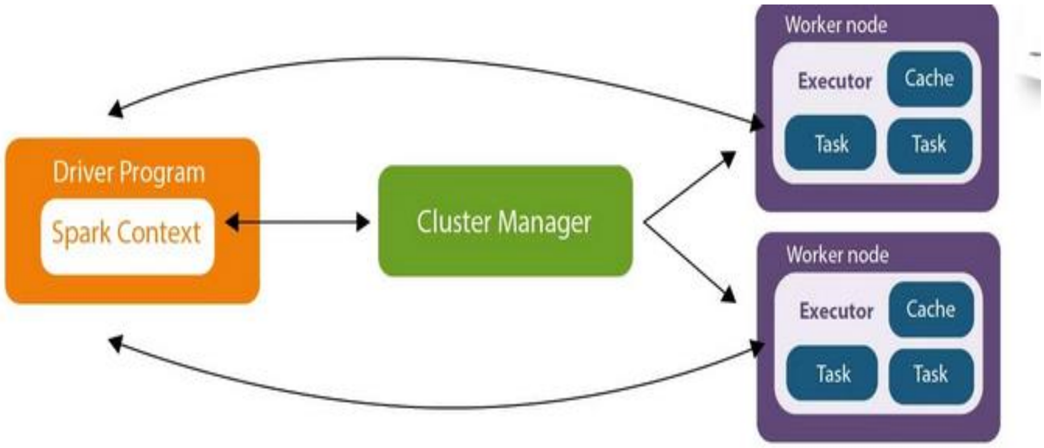


Figure 2.4 : Apache Spark architecture

2.15 Resilient Distributed Dataset (RDD)

RDD is immutable, collection of items that can executed in multiple devices at the same time. Each dataset in the RDD can divided into logical part then executed into different cluster nodes.

2.16 cluster Manager

The SparkContext work with different cluster manager those are given below.

- **Standalone Cluster**
- **Hadoop YARN**
- **Apache Mesos**

2.16.1 Standalone Cluster

Standalone cluster is there is one executer to run the applications on each worker node.

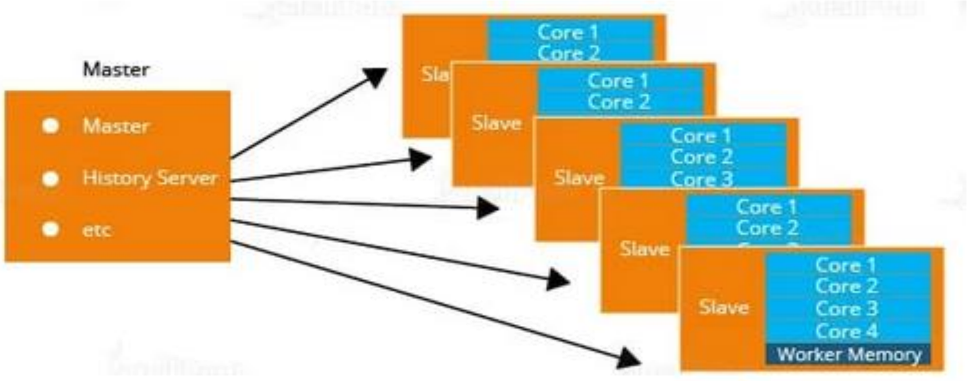


Figure 2.5 : Standalone Cluster

2.16.2 Hadoop YARN

This takes care of resource management for hadoop system. It contain two component which is **resource manager** and **node manager**. **resource manager** manage resource of an application. Node manager have application manager and container. Map reducing task run in to the Container.

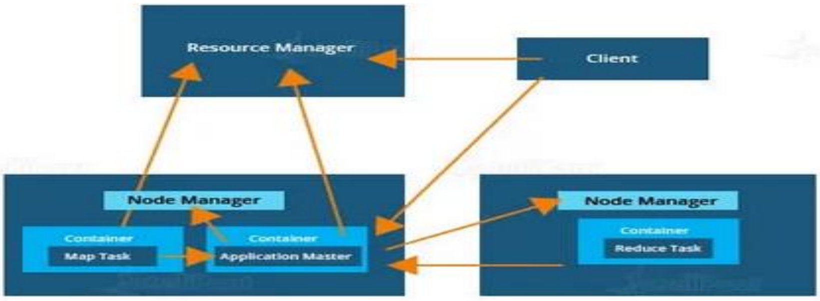


Figure 2.6 : Hadoop YARN

2.16.3 Apache Mesos

Mesos frame work send a request to the resource from the cluster. The application perform the task based on the request.

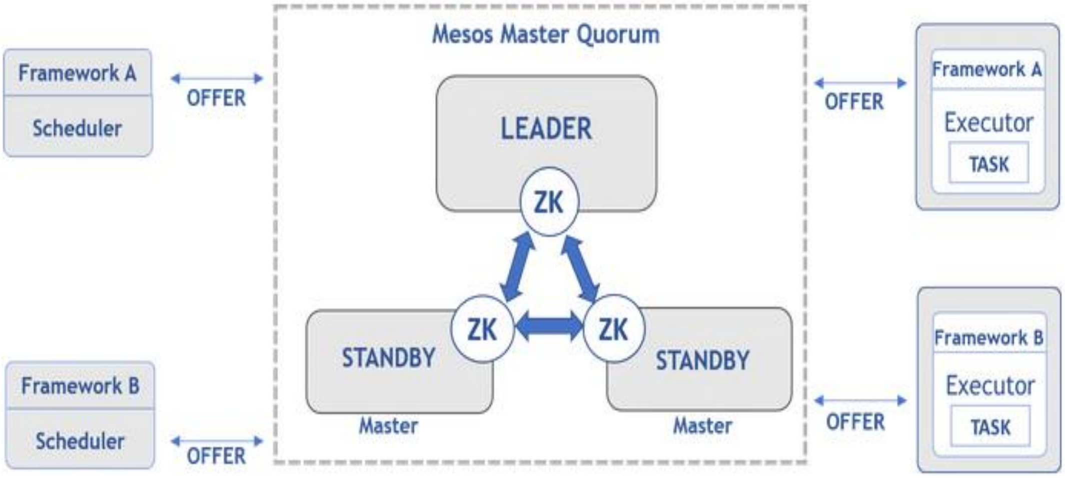


Figure 2.7 : Apache Mesos

Chapter 3

Literature Review

3.1 Hybrid Distributed Collaborative Filtering using PySpark

Hybrid Collaborative Filtering is the mixed concept of Content Based Filtering and collaborative Filtering. The task of using hybrid filtering by using ALS, Dimensionality reduction algorithm and Clustering approach Discussed by Ananya Stitipragyan [1]. Because of the huge growth of Data in The world parallel computing has become unnecessary part of a a tech Enthusiasts and organizations. Use of Hybrid filtering can come handy dealing with large datasets. Dimensionality reduction can be used to reduce dimension by matrix factorization Doing some basic clustering Algorithms such as k-means, k-medoids in multiple computer can handle Large data parallel so that the workloads can be distributed among one or more computers connected within a network. By using spark clustering running time can be reduced with every addition of nodes in clusters. Disadvantages of using spark clusters are requirement of high RAM and Storage.

3.2 Collaboration Filtering For Model Parallelism and Bulk Synchronous System

Collaborative Filtering for model parallelism and for bulk synchronous Systems is discussed by Ariyam Das, Xiangrui Meng, Ameet Talwalker [2], To improve high computation and scaling datasets to reduce workloads of multiple computers there is bulk synchronous parallel systems implementing of distributed ALS algorithm can reduce computation time

Spark MLLib is industrially efficient framework which used in parallel computing. MLlib can outperform many asynchronous systems in terms of Computation speed, accuracy and scalability. If the numbers of parameter increases it can still perform as good as it is. MLlib is very fast and its more efficient than mapreduce algorithm. To solve large scaled collaborative filtering this algorithm can be implemented. In bulk systems there are no occurrence of race conditions so individual process can run independently.

3.3 Hybrid Distributed Collaborative Filtering Model For Recommender Engine

In the present world we need an efficient analysis for scalability of data. We also have technology like MapReduce, Hive etc. When the discussion is about scalability then apache spark takes a stepping stone. Scalability Solving using distributed collaborative filtering is discussed by Rakesh K Lenka, Rabindra K. Barik, Sasmita Panigrahi and S.Panda [3]. In today’s Era Recommender systems has been growing very popular in ecommerce Marketplace. In recommender systems there are many known problems Such as Cold Start problems, Scalability and sparsity. Apache spark has introduced a hybrid solution combining dimensionality reduction and k-Means clustering algorithm. Hybrid Collaborative filtering algorithm can be boosted with addition of multiple clusters adding to spark. Hybrid collaborative Filtering can deal with large scale of parameters so there are many advantages using this approach over un-hybrid one.

3.4 Scalable Collaborative Filtering Recommender algorithm

Scalable or dealing with large scale of data entries into a system is a tough job for recommenders systems. Users always want from a Recommender system that the system can give him relevant info about his likings. Scalable Neighborhood based Collaborative Filtering has been discussed by Sebastian, Christoph Boden [6]. Large scale Dataset computations demands good GPU power. As we already know In recommender systems there are plenty of matrix factorization here And there, so this matrix calculation requires powerful GPU on the System. Increasing number of operations . In recommender systems will increase CPU overhead. So works have to distribute on parallel environment. Without these approach Recommender Systems will find difficulties and will grow non-linearly. Making scalable approach on collaborative filtering time computation can be and there will be less overhead on CPU and GPU as well.

3.5 Clustering Method for Collaborative Filtering

Clustering means grouping. In field of machine learning clustering can Be described as grouping users based on their choices or context. In Recommender system clustering can be done in many ways. K-means k- medoids, k-nearest neighbors are the main basic algorithms. But now data are becoming sparse, there are lots of anomalies or noisy data in datasets. Presenting formal statistical model and comparing different algorithms for modeling parameters including variations of different types of clustering approach is discussed by Ungar LH, Foster DP [9] for effective clustering accurate classifier is needed. Clustering can be a good example of reducing work load. In distributed field of machine learning clustering methods plays a vital role. In apache spark multi node clustering clusters can be divided into multiple computers so that algorithms can run parallel. Collaborative filtering methods use precision recall, accuracy, f1 score to determine clustering algorithm performance.

3.6 Big Data Analysis with Apache Spark

Implementing big data distributed system over a cluster is one of the big challenges which most of the current Tech companies face [15]. Data is growing day by day so advanced system to handling this type of data is needed. As dataset is increasing so reducing computation and implementing parallel computing is mandatory. Without parallel system or distributed systems data will become sparse. Apache spark is a framework for performing general data analytics on a parallel pattern. Apache spark is a framework which used in big data management which builds around easiness of use, speed and analytics of high degree parameters. Its processing power is diverse among the various kinds of data. Performance evaluation between Spark and Cassandra set is discussed by Pallavi Singh, Saurabh Anand [15] Data set can be extracted without parallel computing but with increasing size of data process can become slow. To handle processing capability cloud computing can be implemented. Fast and Scalable system leads to understanding of spark. Big Data Analytics with Spark is a clustering computing which supports Java, Scala and Python programming. Spark environment is multi-functional and efficient than hadoop's mapreduce. Spark is diverse in interactive Queries, Cloud Computing and Stream Processing.

3.7 Parallel and Distributed Clustering Framework for Spatial Data Mining

Clustering techniques are useful to identify and extract different pattern from desired datasets. In large datasets algorithms often find challenges for high dimensionality, heterogeneity, high Complexity. Developing Dynamic Parallel and Distributed Clustering approach to analyse big data within efficient response time has been discussed by Malika Bendechache, A-Kamel Tari M-Tahar Kechadi [18] . Clustering can be done with two major ways one is Partitioning and hierarchical. Dynamic parallel and distributed clustering technique is flexible, efficient and scalable. Hadoop for Big Data Mining designed for distributed processing for large datasets including many features such as Scalability among Big datasets, Fault Tolerance is another key features which mainly describes when data sent to a particular node then the same data is replicated to other nodes as well, which ensures data safety because if one copy is lost then another will be there. Also Hadoop is fast and simple to use because of simple API.

Chapter 4

Proposed Method

4.1 Introduction

Mainly in this chapter we going to discuss about our working procedures .In this chapter we will discuss about multiple clustering methods and their pseudo code or steps of algorithm. We try to generate sample matrix for our model and try to implement recommender system.

4.2 KNN Basic

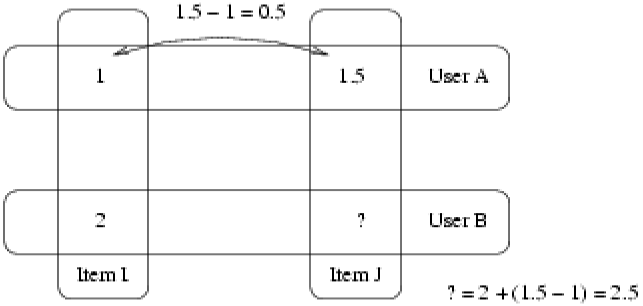
KNN or K-nearest neighbor algorithm used for Regression and classification problems.It’s a supervised learning method. It assumes that similar things exists in close proximity or similar things are near to each other. First we load the data then choose random k neighbors.It calculates the distance between query example and current example of the data then sort the distances in correct order. Then we get the K entries from the sorted list and label them. For regression we need the mean of entries and for classification we need the mode of the k entries.

4.3 Singular Value Decomposition

SVD or Singular value decomposition is a matrix factorization approach for our model. To reduce higher dimension for a dataset we need to imply such a formula which reduces dimensionality by creating multiple matrices. Its an efficient approach to factorize a matrix into singular value or vectors. So we can get rid of scalability problems for recommendations.

4.4 Slope-One Algorithm

Slope one is used for item based collaborative filtering based on ratings. It calculates the ratings difference between users and produces mean of that differences. We can predict a unrated slot with this algorithm.



4.5 Baseline Algorithm

Baseline algorithm is a method which is heuristic, summary statistics and randomness measurement algorithm. We can use it to predict baseline performance of a model. Two mostly common used baseline algorithm is:

1. Random Prediction algorithm

2. Zero Rule algorithm

4.6 Co-Clustering

Co-Clustering or biclustering is a set of techniques in cluster analysis. If a given matrix A and if we want to cluster its rows and cloumns simultaneously this method will come handy . We use it for our model to speed up the process for handling large datasets.

4.7 Alternating Least Square

ALS is matrix factorization algorithm and it runs in a parallel fashion. We use it in apache spark to deal with large scale datasets. Its doing a good job in solving scalability and sparseness of ratings data .There are many advantages of this algorithms which are:

1. Its object function is different than SVD, its uses L2 regularization where SVD uses L1 regularization.

2. It minimizes two loss functions alternatively it first holds user matrix fixed and runs gradient descent with item matrix, and then holds item matrix fixed and runs gradient descent or user matrix

3. ALS runs its gradient descent in parallel across multiple partitions of the underlying training data from cluster of machines

4.8 Flow Chart of the proposed method

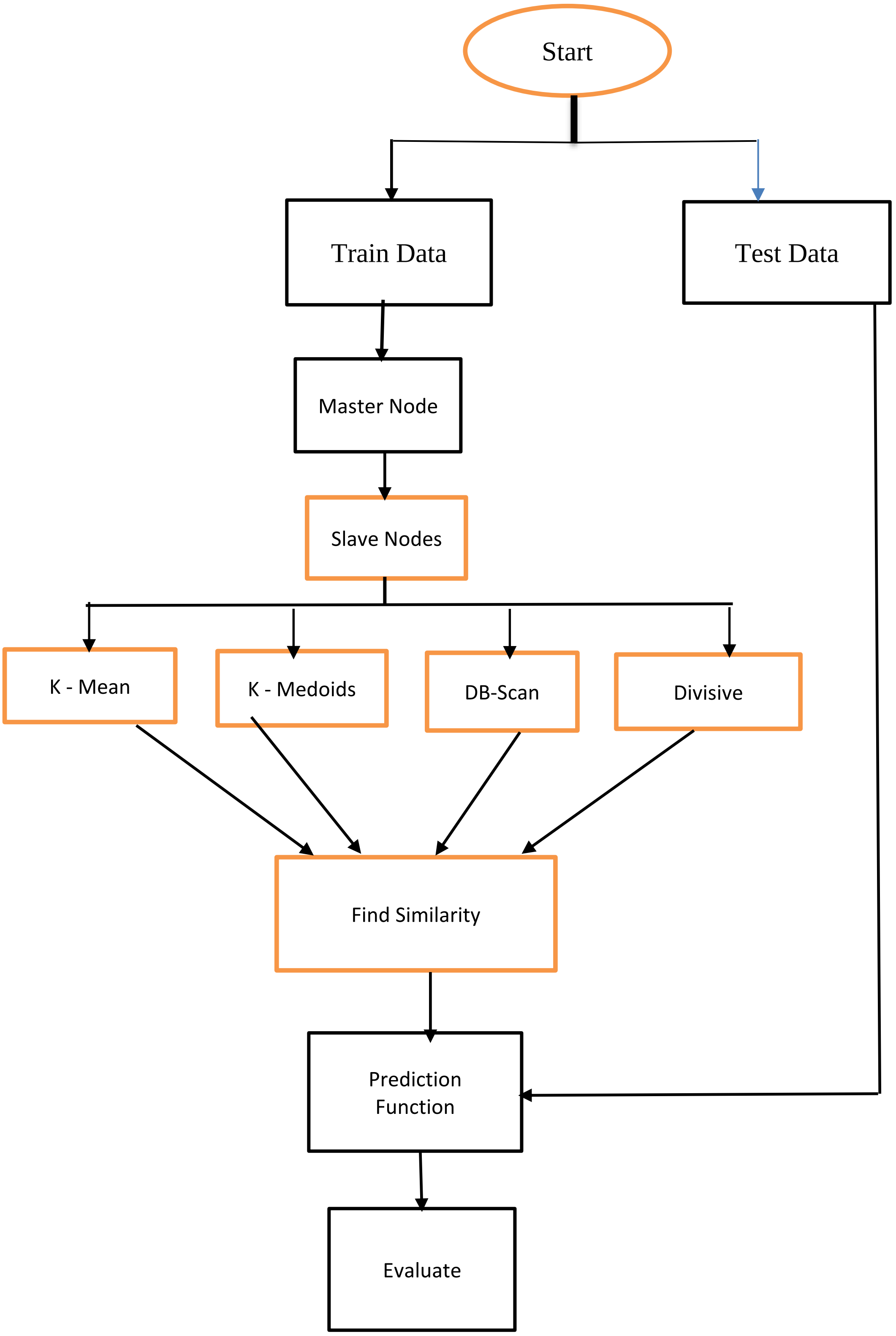


Figure 4.1: Flowchart of The Work Flow

4.9 Book-Crossing Dataset

The BookCrossing (BX) dataset changed into gathered with the aid of Cai-Nicolas Ziegler in a 4- week move slowly (August / September 2004) from the Book-Crossing network with kind permission from Ron Hornbaker, CTO of Humankind Systems. It carries 278,858 users (anonymized but with demographic records) offering 1,149,780 ratings (explicit / implicit) approximately 271,379 books.

http://www.informatik.uni-freiburg.de/~cziegler/BX/

The Book-Crossing dataset comprises 3 tables.

- BX-Users

  Contains the customers. Note that person IDs (`User-ID`) have been anonymized and map to integers. Demographic statistics is furnished (`Location`, `Age`) if available. Otherwise, these fields comprise NULL-values.BX-Books.

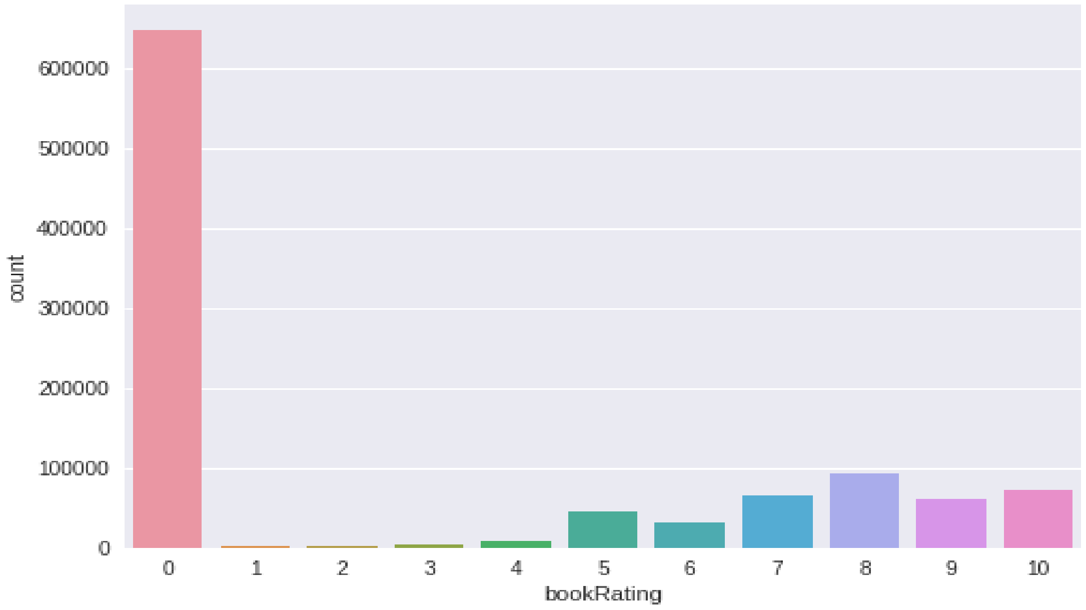
- Bx-Books

  Books are recognized by using their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based records is given (`Book-Title`, `Book-Author`, `Year-Of-Publication`, `Publisher`), acquired from Amazon Web Services. Note that during case of numerous authors, only the primary is furnished. URLs linking to cover snap shots also are given, acting in 3 one-of-a-kind flavours (`Image-URL-S`, `Image-URL-M`, `Image-URL-L`), i.E., small, medium, massive. These URLs point to the Amazon internet website.

- BX-Book-Ratings

  Contains the book rating statistics. Ratings (`Book-Rating`) are either express, expressed on a scale from 1-10 (better values denoting higher appreciation), or implicit, expressed by 0.

4.9.1 Visual Understanding of the Data

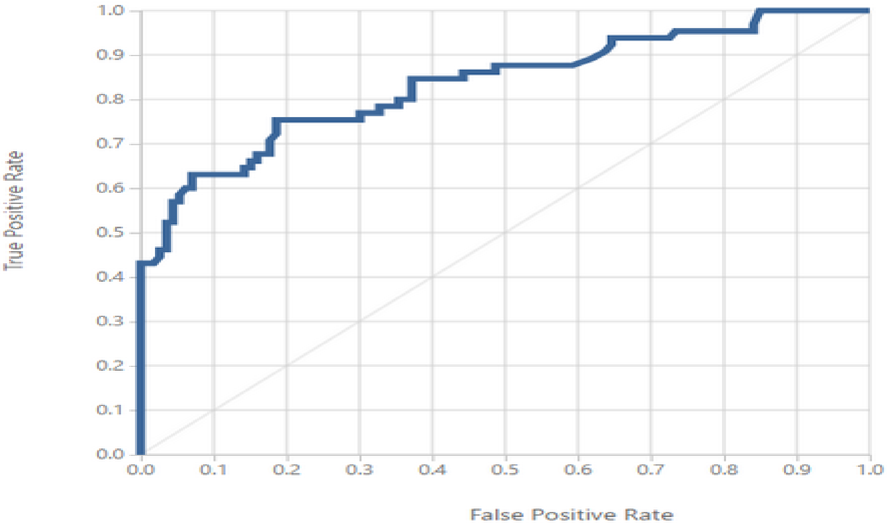


We can see that the Dataset has a excessive degree of Sparsity. Lot's of Cero's As taken from the Book-Crossing Dataset facts.

4.9.2 Experiment and Measurements

In this subsection we simply speak approximately our result. Below figures display the effects won by way of the usage of Book-Crossing dataset. According to figures, the result we acquired for all situation (MAE, precision, recall & f-measures) with our proposed technique the outcomes won't change. But the execution time might be much later in phrases of multi-node clustering.

Precision-Recall



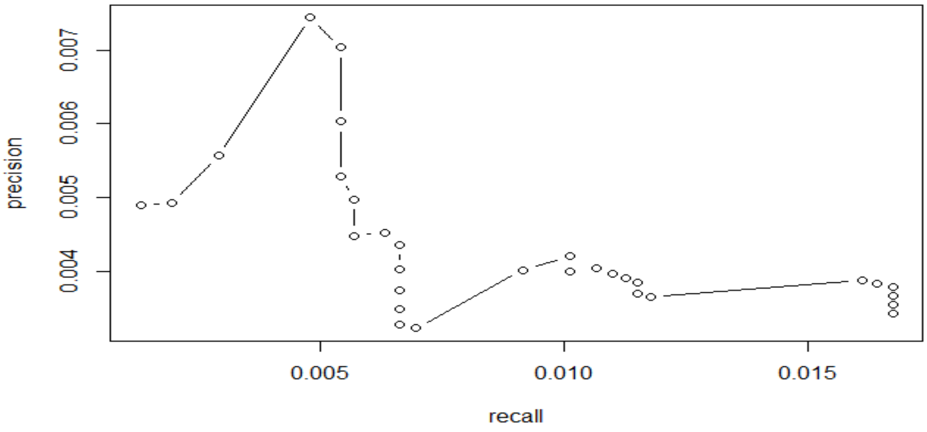


Figure 4.2 : Precision-Recall

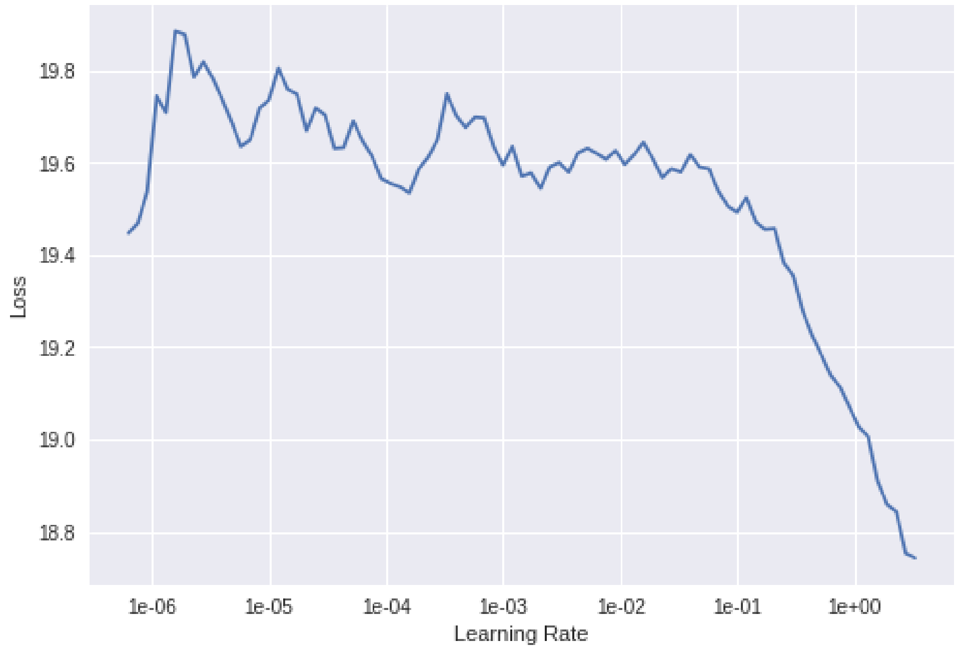


Figure 4.3 : Learning Rate

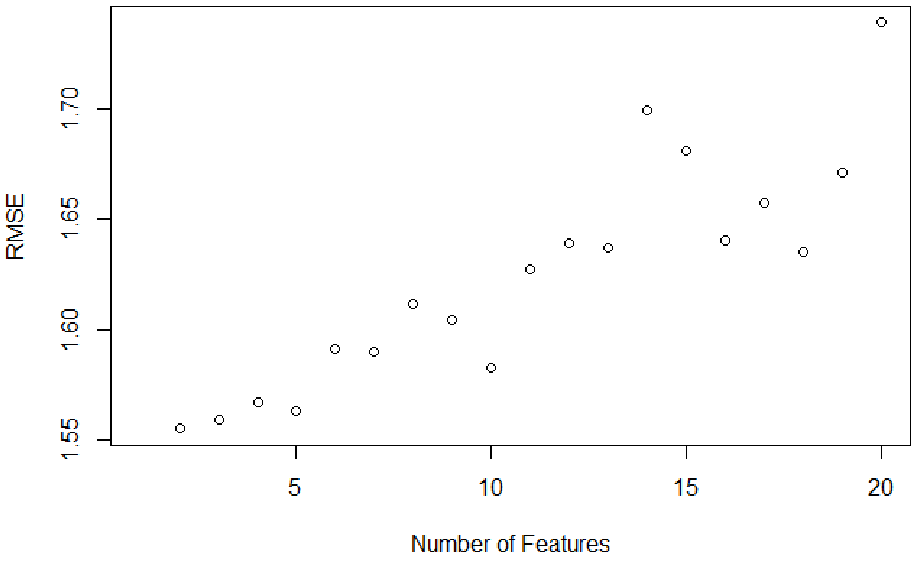


Figure 4.4: RMSE for Different Numbers of Feuatures

4.10 Diversity

In order to construct variety into the recommendation machine, we can examine the similarity between books. We will propose a e book this is maximum dissimilar to some other eBook. To start, permit's take a look at the first 10 books and primary 10 users.

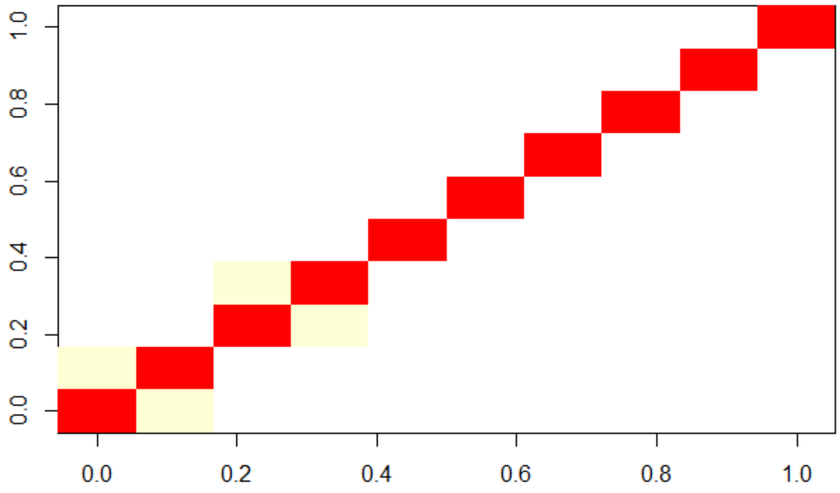


Figure 4.5: Book Similarity for the First 10 Books

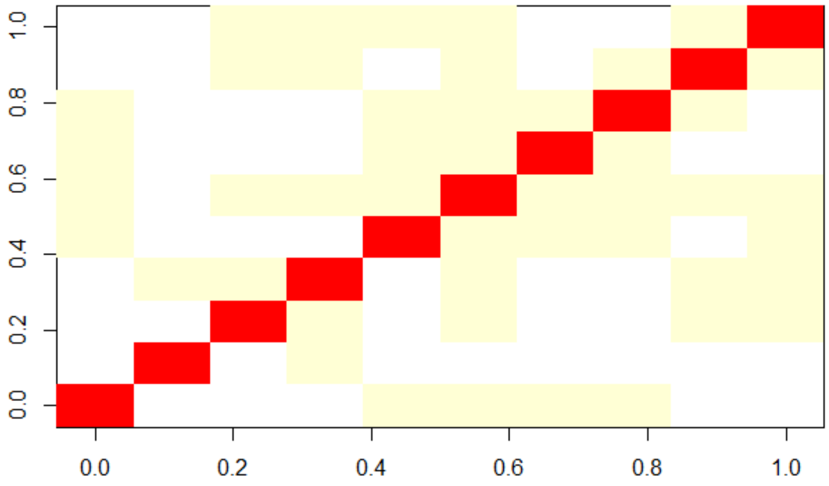


Figure 4.6: Book Similarity Among the first 10 Users

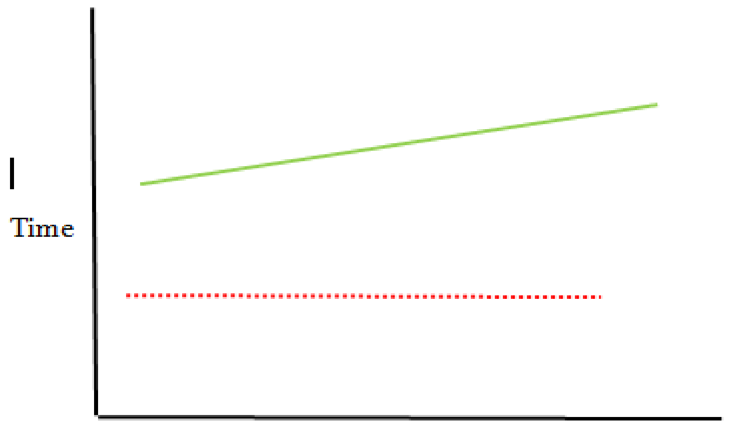


Figure 4.7 : Run Time

Here, Dotted line curve denoting the time for using single machine. And the solid line denoting the time for using multi-node clustering. From the above, graph we can clearly see that if we increase the number of nodes/machines in our Recommender system the total execution time will gradually decrease. Which is a great relief in Big data computing

Chapter 5

Performance Evaluation

5.1 Recommender System Evaluation

We can evaluate recommender systems by five following steps:

1. Establish high – level target of the system
2. Establish particular processes which are selected for the system
3. Importing a dataset to evaluate the system
4. Creating system level matrices/formula
5. Experiment and assess the results

5.2 Establish Goals

We have to first define the goals or objectives of the system before going into evaluation. Recommender system helps users to perform better tasks. Before assessing a recommender system it is important to detect the objectives of the system. Our main purpose is to establish an efficient recommender system.

5.3 Detect Tasks

For our recommender framework we have to ensure some tasks that the recommender will do for the users.

- Users have freedom of selection. Users should have full access to the terms and policy of the system.

- User have fixed amount of time and resources, and he will have full access to the resources within possible restrictions.
- Users can access all relevant events occurring for a specific content.
- User should know the features of an item.

5.4 Importing Dataset

To implement recommender system we need to import datasets which contains information about users and items. Datasets are usually tabular. Dataset can be small or large based on the objectives for the system. Large datasets can occur larger computation time which can lead to scalability problem for a recommender system. We use book rating dataset for our work.

5.5 Establish Metrics

We need to implement several matrices for our computation like mean absolute error, precision, recall and F-measure.

5.5.1 Mean Absolute Error

Mean absolute error (MAE) assesses the average deviation between a user’s rating and predicted rating.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

©easycalculation.com

Important thing to notice that other matrices show improvements when the MAE value decreases. So this measurement should be counted as a potential method.

5.5.2 Precision

Precision is when recommender predicts most accurate rating for the user. The ability of a measurement to be reproduced consistently.

$$precision = \frac{tp}{tp + fp}$$

Where tp= true positive and fp = false positive.

5.5.3 Recall

Recall is the proportion of correct positive classifications from cases that are actually positive.

$$\text{Recall} = \frac{TP}{TP + FN}$$

5.5.4 F-Measure

If classification is binary , the F1 score also called as F-measure is a measurement of test’s accuracy. It relates both precision and recall . Its actually the harmonic mean of precision and recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

Summary

In this chapter, MAE, precision, Recall, and F-Measure are used to evaluated the performance of the similarity metrics. In our experiment section, we have shown that our proposed method a time efficient multi node clustering system quite better for book rating data.

Chapter 6

Conclusion

In this chapter, firstly, first of all, the summary of the research outcomes supplied in this book are discussed. Secondly, few guidelines for destiny research are discussed

6.1 Summary of Thesis

In this paper, we first of all added multi-node clustering methods to implement recommendation device. Then we pointed out the general overall performance of our device. We also mentioned approximately exceptional styles of multi-node clustering techniques in Apache Spark. We tried to offer an interview of the bloodless-start problem in the recommender system and how the problem may be solved in one-of-a-kind ways. Then we display our proposed approach combining one of a kind sorts of clustering algorithm. After producing prediction feature we evaluated our venture's performance. The general execution time decreases through margin in phrases of single machine.

6.2 Future Works

1. In future we will introduce more approaches related to clustering. We are hoping for a better time efficient and accurate approach.

2. We will focus on user's other information not only rating.

3. We will try to implement our algorithm in social media sites or e-commerce sites to detect multiple users and recommend in a efficient way.

4. Since clustering needs more servers and hardware to establish one, monitoring and maintenance is hard. Thus we will try to increase the infrastructure.

Bibliography

1. Sasmita Panigrahi, Rakesh Ku. Lenka, Ananya Stitipragyan 2016,Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark, IIT Bhubaneswar, Bhubaneswar, Odisha.
2. Ariyam Das, Xiangrui Meng, Ishan Upadhyaya, Ameet Talwalkar, Collaborative Filtering as a Case-Study for Model Parallelism on Bulk Synchronous Systems.
3. Rakesh K. Lenka, Rabindra K. Barik, Sasmita Panigrahi and Sai S. Panda, 2018.An Improved Hybrid Distributed Collaborative Filtering Model for Recommender Engine using Apache Spark.
4. Collaborative filtering recommendation algorithm based on Hadoop and Spark, 2015 IEEE International Conference.
5. Dean, Je_rey, and Sanjay Ghemawat. "MapReduce: simpli_ed data Processing on large clusters." Communications of the ACM 51.1 (2008): 107-113
6. Schelter, Sebastian, Christoph Boden, and Volker Markl. "Scalable similarity-based neighborhood methods with MapReduce Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012.
7. Apache Hive. http://hadoop.apache.org/hive
8. Jeffrey Dean and Sanjay Ghemawat,“Map-reduce: Simplied Data Processing on Large Clusters”,in Proc.of OSDl, 2004, pp.137-150
9. Ungar LH, Foster DP. “Clustering methods for collaborative filtering”. In AAAI workshop on recommendation systems (Vol. 1, pp.114-129), Jul 26 1998

10. Zaharia, Matei, et al."Spark: cluster computing with working sets."Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. 2010.

11. Zhao, Zhi-Dan, and Ming-sheng Shang. "User-based. collaborative-_Filtering recommendation algorithms on hadoop." Knowledge Discovery and Data Mining,2010. WKDD'10. Third International Conference on. IEEE, 2010.

12. Dean, Je_rey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.

13. Chowdhury, Mosharaf, Matei Zaharia, and Ion Stoica. "Performance and Scalability of Broadcast in Spark."

14. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (WWW '01). ACM, New York, NY, USA, 285-295.

15. . Pallavi Singh, Saurabh Anand, Sagar B.M ‘Big Data Analytics with apache spark’ International journal of computer applications. Volume 175-No.5, October 2017

16. Sreekanth Rallapalli, Gondkar R R, Apache Spark and Hadoop Based Big Data Processing System For Clinical Research. International Journal of Applied Engineering Research. Volume 13, Number 10 (2018) pp. 7488-7492

17. Malika Bendechache, A-Kamel Tari , M-Tahar Kechadi, ‘Parallel and distributed Clustering Framework for Big Data Spatial Mining’ The International Journal of Parallel, Emergent and Distributed System Vol. 00, No. 00, Month 2011, 1–21

18. Billy Peralta, Pablo Espinance and Alvaro Soto. ‘Enhancing K-means Using Class Labels. May 9,2013