JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-1 E-ISSN: 2347-2693

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

Manu M N[1*], Ramesh B[2]
[1]Department of Computer Science & Engineering, SJBIT, Bengaluru
[2]Department of Computer Science & Engineering, MCE, Hassan
Visvesvaraya Technological University, Belagavi
Karnataka, India

Available online at: www.ijcseonline.org

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

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

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

I. INTRODUCTION

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

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

A. Types of recommendation systems

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

Content-based filtering: also called as cognitive filtering. It recommends items based on a userpast history and the detail of the items [4][32]. In a content-based recommendation system, the user history (profile) refers to builda user profile to describe what type of item user likes. Text documents are used as source in content-based filtering systems.

Collaborative filtering: Popular and widely used technique [5]. It works based on analyzing a huge history data on user activities, behavior, preferences and predicting the users like or dislikes based on similarity of other users. Neighborhood methods focus on relationships between items and users alternatively.

Demographic recommendation system: It provides item based recommendations on user'sdemographic profile [5]. For an example, recommend the items or productsbased on the age of the user, users habits etc.

Knowledge-based recommendation system: It recommends based on how item useful for user, how certain features meets user's preferences and needs based on specific domain knowledge [5]. Here the utility of the recommendation for the user directly interpreted to similarity score.

Community-based recommendation system: In this, system recommends an item based on preferences of users community or friends. Example Social networks,

Hybrid recommendation system: Combining content-based filtering and collaborative filtering could be more effective in some scenario [6]. For that Hybrid recommendation system demonstrated as most useful and powerful approach can be implemented.

B. Key challenges

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

- Cold start: If the user has no previous history or behavior. In this, user has to put more efforts to construction of their user profile and preferences.

- Scalability: Large amount of datasets with users could need large performance power. And also how it deal with big data and stream data produced by active users with items [7][8][9].
- Sparsity: Many a time's ratings on items by users are very low (less), Because of this problem; popular items will have less rating [10].

C. Applications recommendation system

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

- LinkedIn: Most popular business-oriented social networking site. It is used to forms recommendations for users, job, organization or company and group etc [11]. LinkedIn uses open source Apache Hadoop technologyto recommendation. LinkedIn uses item-based collaborative filtering, which is used to find relationshipsbetweenpairsof items.
- Amazon: Most popular e-commerce site uses Amazon item-to-item collaborative filtering technique. Whenever customers buy an item in this site to purchase, it recommends itemsof similar purchased by others users [12].

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

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

II. BIG DATA RECOMMENDATION SYSTEM

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

A. Recommendation algorithms

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

Collaborative filtering consists following techniques:

- Item based recommendation
- User based recommendation

Item-basedrecommendation: Similarities are calculated between items and make recommendations [14]. This can be computed off line because items usually don't change.
User-basedrecommendation: Recommend items by finding similar users [15]. Because of the dynamic nature of users it is facing scaling problem.

III. RELATED WORKS

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

The nature of big data makes it necessary for us to store the data in distributed file system architecture and Hadoop (HDFS) is a widely used tool for doing so. Map Reduce concept is widely used for analysis of Big data. The author [16] gives a brief knowledge on Both Hadoop (HDFS) architecture and Map Reduce Technique. The working of Map Reduce architecture is clearly explained using the four components namely Name node, Data node, Job Tracker, Task Tracker.
In real time,Analysis of streaming data for low cost is becoming very important. The data streams arrive at a high speed; therefore algorithms must work under very strict space and time constraints. Sampling and using distributed systems are two techniques are used to deal with big data. The author [17] also mentions few tools which are handy for these tasks namely Scribe, Cassandra, S4, Storm, Mahout, Moa, R, Pegasus, Graphlab. The author also states that sentimental analysis is a classification problem, where the classification is between positive or negative groups.
In E-Commerce era RS has played very important role in practice and research [18]. Collaborative filtering has proved that one of utmost important for its ease in both theoretical and practical. Collaborative filtering [19] predicts the active customers interest based on accumulated rating information of the on the same wave length customers in a chronological database.
The data scalability is considered as one of the major problem in collaborative filtering, which leaders to the bad performance. In this papers [20][21],Scalability is one of most difficult challenges in recommendation system. To solve this, author introduced k-means algorithm for item clustering prediction in item based collaborative filtering (CF). K-means algorithm was used in item clustering prediction and it was more scalable and accurate technique than any other approaches used earlier.
In this paper [22], multi-criteria collaborative filtering was considered to measure similarity between two users.

Authors estimated the users overall ratings by using nonadditive indifference indices.

To achieve better performance in recommender system, authors [23] designed new technique called item-based collaborative filtering framework. In this, users were rated items directly without any biasing by using Pearson's method.

With respect to data stream mining, J. Chandrika and K.R AnandaKumar [24] discussed the importance of resources adoption and quality awareness in streaming technology. Newly proposed novel framework was designed to generalizestreaming techniques. They proposed resource awareness and quality awareness system.

Most of the recommendation systems depend on similarity measure and this similarity measure depends on user ratings for neighbor's selections. Authors [34] analyzed that cosine and Pearson's similarity measures are not improving the recommender system performance.Different genres of movies and average ratings of users are considered to compare the performance of recommendation systems.

In this paper [26], authors joined the two approaches by combined the advantages of model based collaborative filtering and memory based collaborative filtering to gainmore accurate recommendation than traditional collaborative filtering. Initially it employed memory based collaborative filtering to solve the sparsity problem by filling the empty ratings of the matrix (user-item) and finally item based collaborative filtering technique was used to form the nearest neighbors of each items.

This paper [27] explains how to reduce the power consumption in event recognition system using big data analytics to handle huge data in real time. The proposed framework provides system scalability to perform under parallel, remote, and distributed in computing environment.

IV. APACHE MAHOUT AND HIVE

Apache Mahout is Machine Learning library of set of algorithms for open source Apache Hadoop [33]. It is Apache Software Foundation product with the goal of scalable performance and building scalable algorithms in the areas of machine learning. To produce distributed, scalable algorithms mahout's implemented MapReduce paradigm in Apache hadoop. In this work we used hadoop based MapReduce implementations to recommend items to users.

Hive is used for SQL queries over petabytes of data in hadoop [35]. Data summarization, querying and analysis of big data can do using hive tools and also using this developer's access hadoop data.

A. Recommendation System using APACHE Mahout

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

V. ALGORITHMS FOR SIMILARITY

List of similarity algorithms in Mahout [29]

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

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

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

$$sim(a,b) = \frac{\sum_i (r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_i (r_{a,i} - \bar{r}_a)^2 \sum_i (r_{b,i} - \bar{r}_b)^2}}$$

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

$$sim(a,b) = \frac{\sum_i r_{a,i}\, r_{y,i}}{\sqrt{\sum_i r^2{}_{a,i}}\sqrt{\sum_i r^2{}_{b,i}}}$$

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

VI. EXPERIMENTAL STUDIES

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

A. Data set description

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

Table 1: $user\ X\ item$ rating matrix mXn

User (u)	Items (i)			
	i1	i2	i3	i4
u1	2	3	2	4
u2	?	5	4	5
u3	4	1	1	2
u4	1	5	4	?
u1	2	3	2	4

B. Experimental Hadoop setup description

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

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

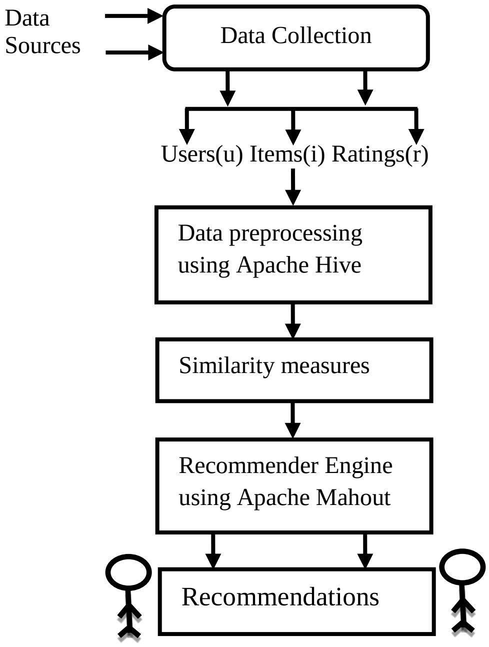

Fig.1. A novel architecture for mahout based recommendation

Step 1: Data Collection
MovieLens data set was available in .zip format in below mentioned link. Using Wget command, first we downloaded the data set into our experimental set up system. The downloaded file has to unzip and stored in fileml-latest.We can view this available data set using Apache Hive commands.

To get MovieLens data:

1. Wgethttp://grouplens.org/datasets/movielens/latest/
2. unzip ml-latest.zip

Step 2: Data Preparation using Hive
After data collected, put it in a .CSV file using Apache Hive. In this stage, we are using Apache Hive for preparing the data as required. Here input file 000000_0 contains comma separated formatted data like <UserId, ItemId, Rating>
hadoop fs -tail /manu/userdemo/mctab1input/000000_0

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

Step 3: Similarity Measures
Choosing a similarity measure for produce best recommendation and requires careful testing, evaluation and research. In this paper, we are considered a Mahout

item based similarity called SIMILARITY_LOGLIKELIHOOD to produce recommendation.

Step 4: Recommender engine using Apache mahout

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

$ mahoutrecommenditembased

-s SIMILARITY_LOGLIKELIHOOD

-i/input path file (where input file stored in file)

-o /Output path file (Where we want to stored back) --numRecommendations10 (Number of recommendation is depends on Requirements)

-i /manu/mahout/input -- Input .CSV format for recommender job

-o/manu/mahout/output/part-r-00000 -- Output file contains recommendation

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

Step 5: Recommendations result

```
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6473
                Total time spent by all reduces in occupied slots (ms)=10074
                Total time spent by all map tasks (ms)=6473
                Total time spent by all reduce tasks (ms)=10074
                Total vcore-seconds taken by all map tasks=6473
                Total vcore-seconds taken by all reduce tasks=10074
                Total megabyte-seconds taken by all map tasks=1618250
                Total megabyte-seconds taken by all reduce tasks=2518500
        Map-Reduce Framework
                Map input records=1669
                Map output records=26941
                Map output bytes=16040195
                Map output materialized bytes=12103284
                Input split bytes=144
                Combine input records=0
                Combine output records=0
                Reduce input groups=943
                Reduce shuffle bytes=12103284
                Reduce input records=26941
                Reduce output records=943
                Spilled Records=53882
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=183
                CPU time spent (ms)=11510
                Physical memory (bytes) snapshot=412823552
                Virtual memory (bytes) snapshot=1720848384
                Total committed heap usage (bytes)=297795584
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1621642
        File Output Format Counters
                Bytes Written=90426
15/10/11 06:26:46 INFO driver.MahoutDriver: Program took 320349 ms (Minutes: 5.33915)
[root@sandbox ~]#
```

Fig.2. Screenshot of mahout recommendation process runs through.

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

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

Time taken::: 15/10/19 18:27:06 INFO driver.MahoutDriver: Program took 320349 ms (Minutes: 5.33915)

mid [mid:rating, ...]

1[1560:5.0,25:5.0,69:5.0,484:5.0,1194:5.0,481:5.0,474:5.0,68:5.0,449:5.0,549:5.0]
2[895:5.0,282:5.0,515:5.0,333:5.0,462:5.0,234:5.0,129:5.0,88:5.0,237:5.0,347:5.0]
3[285:5.0,248:5.0,693:5.0,124:5.0,137:5.0,654:5.0,150:5.0,129:5.0,865:5.0,47:5.0]
4[275:5.0,12:5.0,895:5.0,282:5.0,690:5.0,121:5.0,1238:5.0,237:5.0,234:5.0,79:5.0]
5[582:5.0,403:5.0,47:5.0,156:5.0,237:5.0,67:5.0,1016:5.0,608:5.0,128:5.0,276:5.0]
6[237:5.0,224:5.0,617:5.0,582:5.0,209:5.0,226:5.0,1204:5.0,608:5.0,156:5.0,403:5.0]
7[32:5.0,13:5.0,449:5.0,355:5.0,895:5.0,350:5.0,565:5.0,68:5.0,571:5.0,549:5.0]
8[234:5.0,121:5.0,237:5.0,47:5.0,282:5.0,275:5.0,88:5.0,515:5.0,514:5.0,89:5.0]
9[275:5.0,210:5.0,514:5.0,234:5.0,347:5.0,121:5.0,258:5.0,194:5.0,462:5.0,527:5.0]
10[226:5.0,137:5.0,582:5.0,965:5.0,403:5.0,276:5.0,608:5.0,156:5.0,237:5.0,745:5.0]

Above are the top ten recommendations put forth by the system based on single-criteria collaborative filter implementation using Apache Mahout in Big data for the dataset fed.

VII. CONCLUSION AND FUTURE ENHANCEMENT

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

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

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

REFERENCES

[1] Dr. Sarika Jain, Anjali Grover, Praveen Singh Thakur, Sourabh Kumar Choudhary, "Trends, problems and solutions of recommender system," International Conference on Computing, Communication and Automation (ICCCA2015), IEEE 2015, ISBN:978-1-4799-8890-7/15.

[2] BakirKarahodza, DzenanaDonko, HarisSupic, "Temporal dynamics of changes in group user's preferences in recommender systems," MIPRO 2015, , Opatija, Croatia, 25-29 May 2015

[3] Burke R, "Hybrid web recommender systems,"in: The Adaptive Web, pp. 377–408. Springer Berlin / Heidelberg (2007)

[4] Peter Brusilovsky, Alfred Kobsa, Wolfgang Nejdl, "The adaptive web: methods and strategies of web personalization," page no. 325, Volume: 4321, 2007, ISBN: 978-3-540-72078-2.

[5] ParitoshNagarnaik, A.Thomas, "Survey on recommendation system methods," IEEE Sponsored 2nd International Conference on Electronicsand Communication System (ICECS 2015), 2015

[6] D.Deepika, K.Pugazhmathi, "Efficient Indexing and Searching of Big Data in HDFs", International Journal of Computer Sciences and Engineering, Volume-04, Issue-04, Page No (237-243), Apr -2016

[7] Poornima Sharma, Varun Garg , Prof. Randeep Kaur , Prof. Satendra Sonare , "Big Data in Cloud Environment", International Journal of Computer Sciences and Engineering, Volume-01, Issue-03, Page No (15-17), Nov -2013.

[8] Bo Xie, Peng Han, Fan Yang, Shen, "An efficient neighbor searching scheme of distributed collaborative filtering on p2p overlay network," database and expert systems applications pp. 141–150, Springer 2004.

[9] Mantripatjit Kaur and Gurleen Kaur Dhaliwal, "Performance Comparison of Map Reduce and Apache Spark on Hadoop for Big Data Analysis", International Journal of Computer Sciences and Engineering, Volume-03, Issue-11, Page No (66-69), Nov -2015

[10] YiBo Chen, "Solving the sparsity problem in recommender systems using association retrieval," Academy Publisher, Journal of Computers, Vol. 6, No. 9, September 2011.

[11] Lili Wu, "Browsemap: Collaborative Filtering At LinkedIn," October 23, 2014

[12] Greg Linden, Brent Smith, Jeremy York, "Amzon.com: recommendations- item-to-item collaborative filtering," Industry report, Published by the IEEE Computer Society 1089-7801/03/$17.00©2003 IEEE Internet computing

[13] Jai Prakash Verma, Bankim Patel, Atul Patel, "Big Data Analysis: Recommendation System with Hadoop Framework," Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on 13-14 Feb. 2015, pp.92–97, DOI: 10.1109/CICT.2015.86.

[14] Seikyung Jung, Juntae Kim, Herlocker, "Applying collaborative filtering for efficient document search," Web Intelligence, Proceedings 2004, IEEE/WIC/ACM International Conference,pp. 640-643, DOI: 10.1109/WI.2004.10126.

[15] ShivlalMewada, Umesh Kumar Singh, "Measurement Based Performance of Reactive and Proactive Routing Protocols in WMN ", International journal of Advance Research in Computer science and software Engineering,Volume-1,Issue-1, December 2011.

[16] Sarita Sharma, Priyanka Agiwal, Rakesh Gaherwal, Shivlal Mewada and Pradeep Sharma, "Analysis of Recovery Techniques in Data Base Management System", Research Journal of Computer and Information Technology Sciences, Vol-4, Issue-3, pp.(4-8), March 2016 . -ISSN 2320 – 6527, DOI: dx.doi.org/10.13140/RG.2.2.23964.49289

[17] Albert Bifet, "Mining Big Data in Real Time," Informatica 37 (2013) 15–20.

[18] Chong-Ben Huang, Song-Jie Gong, "Employing rough set theory to alleviate the sparsity issue in recommender system", Proceeding of the Seventh International Conference on Machine Learning and Cybernetics (ICMLC2008), IEEE Press, 2008, pp.1610-1614.

[19] GediminasAdomavicius, Alexander Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". IEEE Transaction on Knowledge and Data Engineering, 2005.17(6): pp. 734-749.

[20] YiBo Huang, "An item based collaborative filtering using item clustering prediction," 2009 ISECS International Colloquium on Computing, Communication, Control, and Management (Volume: 4), Aug 2009, pp.54 – 56, 2009 IEEE, ISBN: 978-1-4244-4247-8,

[21] Gui-RongXue, Chenxi Lin, Qiang Yang, Wensi Xi, Hua-Jun Zeng, Yong Yu, Zheng Chen, "Scalable collaborative filtering using cluster-based smoothing," Proceedings of the ACM SIGIR Conference 2005, pp.114–121

[22] Yi-Chung Hu, "Non-additive similarity-based single-layer perceptron for multi-criteria collaborative filtering," Journal Neurocomputing, Volume 129, April, 2014, pp. 306-314.

[23] Z. R. Deng, X. Zhang, X. Deng, L. Xu, W. M. Huang, "An improvement of video recommender similarity measurement model," International Conference on Automation, Mechanical Control and Computational Engineering(AMCCE 2015), pp.675-680, 2015.

[24] J. Chandrika, K.R Ananda Kumar, "Data stream querying: challenges and issues," Int.Conf. on Computer Applications, ISBN: 978-981-08-7300-4, 2011.

[25] Y. K. Park, S. C. Park, W. S. Jung, S. G. Lee, "Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph," Expert Systems with Applications, vol. 42, no.8 pp.4022-4028, 2015.

[26] SongJie Gong, HongWu Ye, HengSong Tan, "Combining memory-based and model-based collaborative filtering in recommender system", 2009 Pacific-Asia Conference on Circuits, Communications and System, 978-0-7695-3614-9/09 ©2009 IEEE, DOI 10.1109/PACCS.2009.66

[27] Changbingchen, Xia Yang, Bong Zoebir, SivadonChaisiri, "A workflow framework for big data analytics: event recognition in a building," 2013 IEEE Ninth World Congress on Services, pp.21–28, 2013 IEEE, DOI: 10.1109/SERVICES.2013.29.

[28] Skategui, "Recommendation algorithms with Apache Mahout," [Online] Apr 11.

[29] R. Murugesh and I. Meenatchi, "A Study Using PI on: Sorting Structured Big Data In Distributed Environment Using Apache Hadoop MapReduce", International Journal of Computer Sciences and Engineering, Volume-02, Issue-08, Page No (35-38), Aug -2014,

[30] Dunning, "Accurate methods for the statistics of surprise and coincidence," Computer Linguist, vol. 19, no. 1, pp. 61-74, Mar, 2003.

[31] N. Rastin and M. ZolghadriJahromi, "Using content features to enhance performance of user-based collaborative filtering performance of user-based collaborative filtering," Int. journal of artificial intelligence and applications, vol. 5, no. 1, pp. 53-62, Jan 2014.

[32] F. Ricci, L. Rokach, B. Shapira, "Introduction to recommender systems handbook," New York: Springer, 2011, pp. 1-35.

[33] Manu M N, Anandakumar K R, "A current trends in big data landscape," 2015 IEEE International Conference on Computational Intelligence and Computing Research, IEEE 2015, 978-1-4799-7849-6/15.

[34] Qiao Cheng, Xiangke Wang, Dong Yin, YifengNiu, "The new similarity measure based on user preference models for collaborative filtering," Information and Automation, 2015 IEEE International Conference, Aug 2015, pp.577-582, 2015 IEEE, DOI: 10.1109/ICInfA.2015.7279353.

[35] Mazin S. Al-Hakeem, "A Proposed Big Data as a Service (BDaaS) Model", International Journal of Computer Sciences and Engineering, Volume-04, Issue-11, Page No (1-6), Nov -2016, E-ISSN: 2347-2693

[36] V.Vijayadeepa, Archana.G "Semantic Based Service Recommendation using Collaborative Filtering", SSRG International Journal of Computer Science and Engineering (SSRG -IJCSE), V3 (10), 13-19 October 2016. ISSN:2348 – 8387.

Authors Profile

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

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

