KEYWORD BASED RECOMMENDER SYSTEM FOR ELECTRONIC PRODUCTS USING WEIGHT BASED RECOMMENDATION ALGORITHM IMPLEMENTED ON HADOOP

[1]NAYANA VAIDYA, [2]KHACHANE A.R

[2]HOD, Information Technology Department
[1,2] Vidyalankar Institute of Technology Wadala Mumbai
Email: [1] nayana_vaidya@hotmail.com, [2]ajitkumar.khachane@vit.edu.in

Abstract: Recommender system is a topic which falls under the domain of information retrieval, data mining and machine learning. Recommender systems are widely used by famous websites like Amazon, Flipcart, Netflix, Facebook, twitter and many others. There are various types of recommender systems like collaborative filtering, content based filtering and hybrid recommender system. Recommender systems can be used for recommending various products like books, movies, music and any products in general. Various researchers uptil now have developed various algorithms to improve the accuracy of recommender systems and provide good quality recommendations. Algorithm and approach used determines the quality of recommendations. In this paper we are proposing a keyword based recommender system for recommending electronic products. We recommend the products based on keywords. We are using weight based recommendation approach. Since we are taking into account the previous user preferences it also falls under the category of user based collaborative filtering. Recommender system also needs to handle big data. So in order to provide scalability we are implementing it in Hadoop using mapreduce . We have implemented the product using java and the database used is mysql. The integrated development environment used is Netbeans IDE 8.0.2

Index terms: information retrieval, data mining, machine learning, collaborative filtering, content based filtering, hybrid recommender system, keyword based recommendation, weight based recommendation, big data, hadoop, mapreduce, java, MYSQL

I. INTRODUCTION

We will first see what is a recommender system.
The first major recommender systems emerged in the mid-1990s (Resnick & Varian, 1997)
Ecommerce business is growing. There are a number of ecommerce websites ,to name some of them we have flipcart, amazon. If you see flipcart there is a menu which shows recommendations. If you see Amazon it shows “your browsing history”, ”Recommendations for you”, improve your recommendations”,”your recently viewed items and featured recommendatios”.Facebook provides a feature of recommendations in terms of “People you may know”. If you visit jabong.com and you add a product to bag,it will display “Recommended for you”. These recommendations are similar to the item you have purchased before. So several ecommerce websites have recommendation engines. So recommendation systems are gaining importance these days. There is too much of information overlaod for the users and people spend too much time in selecting the product. Recommendation systems make their jobs easier. Most of the algorithms are based on users searching patterns, browsing history, previously purchased items. One researcher talked about how to reduce the traditional limitation of cold start and how it can be overcome using fuzzy collaborative Filtering. There are mainly 2 types of recommender systems-collaborative filtering and content based filtering and demographic Based. The main problem in collaborative filtering is, it is not able to provide recommendations if no ratings exist. This is known as cold start problem. In content based filtering there is no variety in recommendations. User will not get to see different items which he has not seen or purchased before. Another problem that is common in recommender systems is scalability.

Recommender systems work well with small amounts of data ,but as the data set increases the algorithms cannot cope up. Before internet if people wanted to see a movie or read a book or visit a place they asked their friends. People prefer recommendations from friends rather than strangers. So recommendation engines are developed which combine the data from the social networking sites like facebook and twitter or Qoura or Linkedin and give recommendations of items to users of their interest. An item is a product the user is interested in. Recommender systems are used to recommend a number of products such as movies, books, CDs, news, restaurants and many other products in general. They give advice to the user in selecting an item from a vast choice which in turn helps in increasing sales. Recommender systems have a large effect on the performance of the ecommerce websites.

In content based filtering users past purchases are seen and a product is recommended similar to his past purchases.

In collaborative filtering ratings given to products are compared and users which have similar ratings show similar interest in the product. So products are recommended based on the similarity of ratings between the users.

Recent research has demonstrated that a hybrid approach, combining collaborative Filtering, and content-based filtering could be more effective in some cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model.

Demographic Based Recommendation Systems are based on **demographic** characteristics of consumers and recommend a list of items that have good feedback from the consumers that are demographically similar to the target consumer

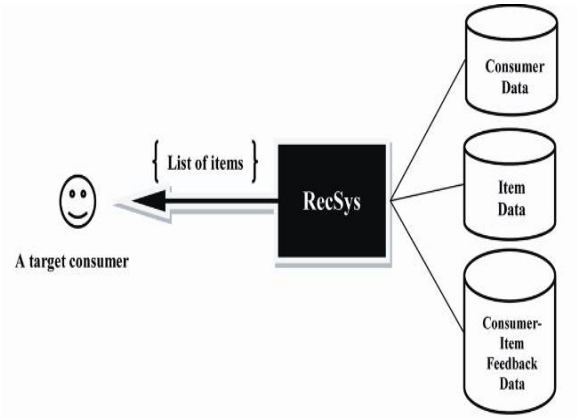

Figure 1,0 Recommendation System(RecSys) Model

Information overload problem is partly solved by Search Engines, but they do not give personalized recommendations. Recommendations are based on various properties such as popularity of items, ratings given by users, geographical location, users demographic information such as age, gender, education of users, users searching behaviour. Machine learning algorithms are used by various recommendation engines to find the similarity between items or users. Data is most important in a recommendation system.

II. SOME CHALLENGES IN RECOMMENDER SYSTEM

1) Cold Start problem: This problem occurs when a new user or a new item enters a website. No ratings are available to the item and the user until now has not purchased any item. So recommendations cannot be given to such a user and such an item. In content based filtering user profile is built based on users searching patterns. This user profile is then compared with the items characteristics. So if the item or user are new we have a cold start problem.

2) Sparsity of Data: This means less ratings are available to a particular item. Users generally rate a limited number of items.

3) Scalability: When the dataset is small recommender engine can give quality recommendations. When there is huge dataset or big data it is difficult to provide recommendations.

4) Changing user preferences:
Today I have a particular intention while browsing tomorrow, I have a particular intention while browsing. So it is difficulty to the recommendation engine.

5) Synonymy: There is certain degree of commanility between words which is not recognized by content based collaborative filtering algorithm.

6) Recommendation quality: User is sensible about incorrect recommendations. In such case user will lose trust in recommendation system and stop using it. So the challenge is to maintain the quality of recommendations.

III. MY PROPOSED SYSTEM:

I have developed a recommender system for recommending electronic products.

IV. CONFIGURATION

The product is developed using java and JSP. System is implemented on Hadoop. Netbeans 8 is used as the development environment.

There are 2 types of users in my recommender system-previous user and active user. The data set file used is in JSON(Java Script Object Notation) format. When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects.

Following are the features used in my recommender system-
1) Login
2) Registration
3) Add Product
4) Upload Data
5) Training
6) Logout.

The login feature can be used for login for both the previous user and the active user.

The registration feature is to register new users in the product.

Add product feature is to add the electronic products in the system.

When the user logs in as an active user he has the following features-
1) Upload reviews

2) View reviews
3) Recommend
4) Logout

V. ALGORITHM

Following is the algorithm used in my system. It is called as weight based recommendation.
There are 2 roles in the system-previous user and active user.
Previous user is the user which has given the reviews for the product. In our system we use text reviews. The text reviews are in JSON format.The textual user reviews are downloaded from the Amazon website.
The text reviews has an "overall rating" column which is numeric. If the overall rating is above 3 it is considered as a positive review. Like sentiment analysis the entire sentence is considered as positive or negative. If the overall rating is 3 or less than 3 then the rating is considered as negative.

Following is the algorithm:

1) User logs in as admin.
2) Admin uploads the dataset which is in the JSON format.
3) JSON data is stored in the database .
4) 2 files are created "positive_review.txt" and "negative_review.txt",
4)Then the data set has to be cleaned so "Training" is performed on the data.
5) The 2 text files are given as input to the Training process.
6) In the Training function we calculate the symbols count from "symbols.txt" file and calculate the stopwords from the "stopwords.txt" file.
7) We read the positive_review.txt file line by line.
8) We split the positive review text into keywords and store the keywords into an array .
9) We ignore the stopwords if they occur in the review.
10) We replace the symbols in the review with space
11) We replace the keyword which has "ing" in the end with a space eg. talking becomes talk.
12) We replace all numbers if any in the keyword with space.
13) List of positive words is stored in an array.
14) Positive words are also stored in the file "positive.txt".
15) Map Reduce calculates the frequency of keywords in the positive file
16) Above same process is repeated for the "negative _review.txt" file.
17) Negative words are also stored in the "negative.txt" file.
18) Map Reduce calculates the frequency of keywords in the negative file.
19) A database table contains the keyword, class of keyword whether positive or negative, frequency of the keyword and the weight of the keyword.
20) Weight of keyword is calculated using the formula:
double wt = Integer.parseInt(keyvalue[1]) * 1.0 / poscount;
double wt = Integer.parseInt(keyvalue1[1]) * 1.0 / negcount;
21) Once the training process is over admin user logs out.
22) An active user can login who wants recommendation.
23) Active user will select the product and provides some reviews for the product.
24) The above step is necessary as the user need to show interest in a particular product.
25) Once the user provides the reviews the reviews are uploaded in the database in a table.
26) When the user clicks on view reviews he can see the reviews provided by himself for the particular product.
27) Select keyword from the table which is similar to the keyword in the review and whose class='positive'.
28) Select keyword from the table which is similar to the keyword in the review and whose class='negative'.
29) keyword from review which is provided by the active user is searched into the database table.
30) Weight of all positive keywords is added.
31) Weight of all negative keywords is added.
32) if positive weight> negativeweight then recommend=true else false.
33) Suppose active user had given reviews for samsung mobile and LG TV.
34) So the keywords are Samsung, mobile, LG and TV
35) Selected related products would be [samsung mobile, samsung galaxy s8 mobile, samsung j7 mobile, Samsung refrigerator, Nokia n8 mobile, apple mobile, apple iphone mobile, Htc 10 mobile, LG smart TV, LG refrigerator, apple TV].
36) Finally the products are recommended whose class is positive.

VI. KEYWORD BASED RECOMMENDATION SYSTEM DESIGN

Following is the design of the system

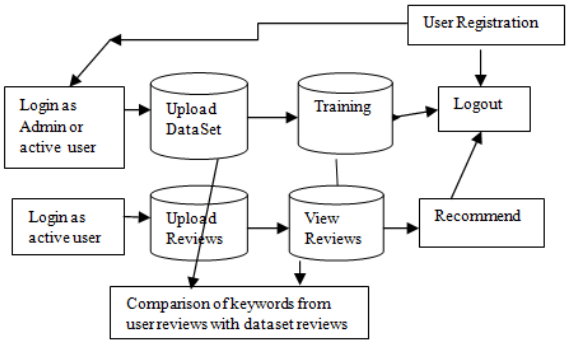

Fig.1.0 System Design Diagram

VII. INTRODUCTION TO HADOOP

We have implemented the system using Hadoop and used Java, JSP and MapReduce. We have used MapReduce to calculate the frequency of keywords. Hadoop is used for scalability and to handle big data ie a huge dataset file.
Now let us see what is Big data, Hadoop and MapReduce:

What is Big Data?
The data which is beyond storage and beyond the processing capacity is called as big data. Big data is generated from social networking sites like Facebook, Twitter, Linkedin and online shopping websites.
Big data is huge terabytes or petabytes of data which cannot be stored in your local system, so you have to approach a data center to store that huge data. We store the data in the datacenter which we want to process 70% of the data that is generated is unstructured data. The 3 terms volume, velocity and variety together is called as big data. This definition is given by IBM. If the data is big processing time must also be reduced. So in order to store huge data and have less processing speed we have Hadoop.Google introduced the idea of GFS(google file system) and a technique called as MapReduce in a white paper. From that Yahoo introduced the term HDFS (Hadoop distributed File System) and MapReduce. But both Google and Yahoo did not invent Hadoop. Hadoop was invented by a scientist called as “doug cutting” .Hadoop logo is a yellow elephant.2 terms used with Hadoop are HDFS and MapReduce. Hadoop is an opensource framework introuduced by Apache Foundation. Hadoop is given by cloudera.

VIII. HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

HDFS is a cluster of commodity hardware. Cluster is a set of machines in single Lan. Commodity hardware are cheap hardware like laptops. HDFS is a specially designed file system to store big data and it stores the data in parallel. In normal file system the data is stored in blocks in the harddisk. So suppose there is a block of 4kb and we are storing a 2kb file the remaining harddisk size is wasted. For HDFS the default block size is 64MB.So if we store a file of 30MB in the block the remaining space is released for storing other files. HDFS has services such as NameNode, SecondaryNameNode,JobTracker,data node, TaskTracker. First 3 services are master services. Last 3 services are slaveservices. Every master service can talk to each other, every slave services can talk to each other. In HDFS the input file of data is split into various files and stored in datanodes. HDFS has by default 3 replications. So the same file is copied in 3 datanodes whichever are free. Replication is for backup so that the data is not lost. All data nodes give a heartbeat signal to the name node and a block report after every short period of time. Name node stores metadata. But what if your name node metadata is lost there is back up of name node which is called as secondary name node which is to protect the single point of failure of name node. When you write a program to process the data job tracker comes into picture. Jobtracker sends a request to the name node. Jobtracker assigns the task to tasktracker. In HDFS there are input splits ie.the input file is split into more number of files and runs on different machines. The number of maps is equal to the number of input splits. TaskTracker sends a heartbeat signal to the job tracker.

IX. MAPREDUCE AND ITS WORKING

MapReduce splits the input file into blocks. We are using MapReduce to count the frequency of keywords from the “positive.txt” and count the frequency of keywords from the “negative.txt” file. Files have unstructured text format. MapReduce could be used in HealthCare domain, weather forecasting etc. MapReduce runs in the form of key,value pairs.For every input split and mapper there is a RecordReader. Mapper and Reducer understands only key, value pairs. There are 4 file formats recognized by the mapper.

1. Textinput formats
2. Keyvalue text input format
3. Sequence file input format.
4. Sequence file as text input format

Record reader reads one record at a time from the input split.
MapReduce does parallel processing. In Hadoop there are box classes like IntWritable, LongWritable etc. Output of mapper is key, value pair. There is shuffling phase after mapper phase. To remove the duplicate keys there is a shuffling phase. Reducer gives the final output in the part-00000 file. Another file is the _success file and _logs file.

CONCLUSION AND FUTURE WORK

In this paper, we have proposed a recommendation system which uses JSON data set. Here, the user's choices are indicated through keywords and recommendations are generated using a weight based algorithm. The active user can select a product and give their reviews. Based on their reviews whether positive or negative the product is recommended to the user. keywords from the reviews provided by active user is compared with similar keywords in the training set. If positive weight exceeds negative weight then the product and similar other products are recommended to the user. Reviews of existing users are extracted from their ratings. The system provides personalized and accurate recommendations. KBRS is implemented in Hadoop to increase the efficiency and improve scalability if the data set is big data. At present only text reviews are extracted. Currently we have considered the entire sentence of the review as positive or negative depending on the overall rating whether 3 or above. So even though a word is positive in a review text and the review is the negative review file the keyword will be considered as negative. This flaw we will try to work in future.

REFERENCES

[1] Sanjay k.Dwivedi,Chandrakala Arya, "A Survey of news recommendation approaches",2016 International Conference on ICT in Business Industry and Government.

[2] Yingtong Dou,Hao Yang,Xiaolong Deng, "A Survey of Collaborative Filtering Algorithms for Social Recommender Systems",2016 12th International Conference on Semantics,Knowledge and Grids(SKG).

[3] Sunil Kumar Khatri,Sonam Gupta,"A novel approach for improving recommender systems",2016 5th International Conference on Reliabilty,Infocom Technologies and Optimization(Trends and Future Directions)(ICRITO).

[4] FeiYue Ye, Haolin Zhang ,"A collaborative filtering Recommendation based on users interest and correlation of items"," School of computer engineering and Science ,Shanghai university

[5] Ferdaous Hdioud ,Bouchra Frikh,Brahim Ouhbi,"A Comparison Study of some algorithms in recommender Systems ," Colloquium in Information Science and technology Year 2012,IEEE conference publication"

[6] Quing Wang,",Design and Implementation of recommender Systems Based on Hadoop",2016 7th IEEE International Conference on software Engineering and Service Science(ICSESS),IEEE Conference Publication.

[7] www.igi-global.comwebsite
Wikipedia.

[8] Sanjay k.Dwivedi,Chandrakala Arya, "A Survey of news recommendation approaches",2016 International Conference on ICT in Business Industry and Government.

[9] Bushra Alhijawi,YouSef Kilani,"Using gentic algorithms for measuring the similarity values between users in collaborative filtering recommender systems",2016 IEEE /ACIS 15th International Conference on Computer and Information Science(ICIS).

[10] Anshul Gupta, Hirdesh Shivhare, Shalki Sharma, "Recommender Systems using fuzzy c-means clustering and genetic algorithm based weighted similarity measure",2015 International Conference on Computer ,Communication and Control(IC4).

★★★