Personalized Movie Recommender System using Rank Boosting Approach on Hadoop

Ms. Annies V Jose
Department of Computer Science and Engineering
University of Calicut NCERC, PAMPADY, Thrissur - 680597

Ms. Jini KM
Department of Computer Science and Engineering
University of Calicut NCERC, PAMPADY, Thrissur - 680597

Abstract

Today we are living in an era of Big Data. Large numbers of services are available to customers; from these services it is difficult for them to choose those that are most appropriate for them. In this scenario a wide variety of service recommender systems will guide the user in selecting the most appropriate one. But these traditional service recommender systems will not work well with Big Data environment; they will experience scalability and efficiency problems as it has to work on huge amount of data. Most of the existing recommender system will provide same rating and ranking of services to different customers. As a solution to this we propose service recommender systems that operate on MapReduce Framework on Hadoop platform. Our service recommender system uses keywords to indicate user preferences and a variation of collaborative filtering algorithm called user based collaborative filtering is used to provide recommendation to customers. It also uses rank boosting approach for combined preferences, which will help to rank the results according to the preference of the current user and produce the recommendation list accordingly.

Keywords: Efficiency, MapReduce, RankBoosting, Scalability, User-based Collaborative Filtering

I. INTRODUCTION

In the modern era the amount of data available over the internet has increased beyond expectations and it is commonly called with the term Big Data. Big data refers to collection of both structured and unstructured data that is very large and beyond the processing capability of traditional database management systems. Earlier we were dealing with only terabytes of data but now we are coming across petabytes, exabytes, zetabytes of data and will soon come across yottabytes of data. Big Data processing turned out to be an overhead for most of the companies and organizations. Big Data brings new opportunities and features in academic and industrial scenario.

Like most Big Data applications, Big Data tendency has also began to show its impact on service recommender systems. Now large variety of services is available to users and users will face the problem of selecting the appropriate service for them. Service recommender systems will act as an important tool in guiding the user to opt the appropriate services for him/her. The effect of service recommender system can be found on different scenarios like online shopping sites [1], movie recommender systems, Facebook etc.

A wide variety of algorithms are now available for service recommendation. The field of service recommendation began its development since 90's, commonly used recommendation methods are Collaborative Filtering, Content based and hybrid recommender systems.

A. Content-Based Recommendation System

Content based filtering, which is also known as cognitive filtering recommends items based on the content of items and a profile of user preferences. In these types of recommendation systems keywords are used to describe the content of the items and user preferences [2]. Various services that are referred by the current user in the past is considered and keywords of those services are retained, then the similarity between those past preferences and the current services will be considered and those services that seems to be most appropriate for the user will be recommended.

B. Collaborative Filtering-Based Recommendation System

Collaborative filtering recommends services to current user, based on the items that the people with similar taste have preferred in the past. These systems can be classified into two categories as item-based collaborative filtering and user-based collaborative filtering.

1) Item-based collaborative filtering

In this method similarity between different elements in the dataset are calculated, this is done by considering all the users who have rated both of the items [3]. A wide variety of methods can be adopted for calculating similarity.

2) User-based collaborative filtering

In this method users having same taste to that of the current user will be identified and item will be recommended to the user based on these previous user preferences [4].

C. Hybrid Recommendation system

Hybrid recommendation system can be considered as a combination of both content-based recommendation system and collaborative filtering based recommendation system [5]. Several approaches can be used in implementing hybrid recommendation systems like combining the results of both the algorithms, incorporating the feature of one to other algorithm etc. Hybrid recommendation systems will produce more accurate and efficient results while compared with others.

As we have seen various types of recommendation system, for implementing personalized service recommendation system user based collaborative filtering method is taken into account. The user based collaborative filtering will try to identify users who are having similar taste to that of the current user and will recommend services based on that those users have preferred in the past**.**

II. USER PREFERENCE BASED RECOMMENDATION SYSTEM

User preference based recommendation system is a keyword based technique for service recommendation. This method uses keywords to indicate user's preferences and quality of services [7]. User-based Collaborative Filtering algorithm is used to find and provide appropriate recommendations to users. It calculates personalized rating of services for each user according to his personal preferences and provides the recommendation list. The first element of the recommendation list will be the one that is most suitable for the user and it will be arranged in the decreasing order of appropriateness.

Majority of existing Recommender Systems obtains an overall numerical rating, as input information for the recommendation algorithm. This overall rating depends only on one single criterion that usually represents the overall preference of user *u* on item *i*. To overcome the problems in the existing recommendation system, combined preferences based rank boosting algorithm is used. . In the rank boosting algorithm, it gets the input as combined preferences [6], based on the preferences it process the similarities with the reviews of the existing users then it provides the ranking to the services. Based on the ranking provided to the services we generate the output recommendations. Finally it generates high similarity matching results as the recommendation list to the end users for their combined preferences.

Traditional service recommendation methods will not work well in Big Data environment; they will experience scalability and efficiency problems as they have to deal with huge amount of data, which is beyond their control. In order to overcome these issues "Personalized service recommendation method" is implemented in a MapReduce framework on Hadoop [4]. The proposed algorithm for service recommendation will be split to multiple MapReduce phases so that scalability and efficiency related problems can be resolved easily. It is a keyword based approach where it uses keywords to indicate the quality of candidate services.

A. Working of Recommender System

The system architecture is depicted in figure 1 and the execution of the personalized service recommendation system can be explained with the help of following steps:

1) Loading Datasets

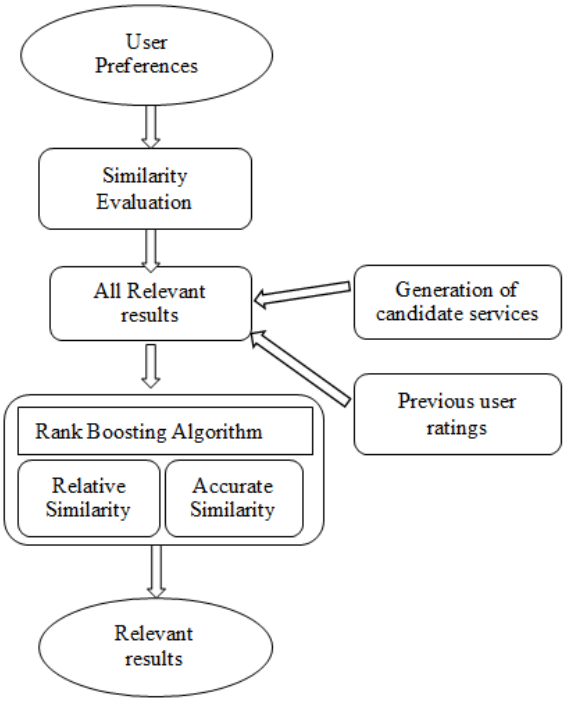

Fig. 1: System Architecture

For implementing user preference based service recommendation system we use MovieLens dataset. The dataset consists of movie information, user information and rating information. As an initial step datasets are loaded as it is present without applying any processing.

2) Preprocessing

The loaded MovieLens dataset may contain some unwanted contents like ambiguous information, extra symbols and special characters etc. In the preprocessing step all the three datasets are preprocessed for removing any extraneous information that is present in it. In the preprocessing step all the unwanted portions of the datasets are removed and it is stored in a structures manner, which reduces the complications of further steps.

3) HDFS upload and categorization

All the preprocessed and modified datasets will be uploaded to Hadoop Distributed File System, which is highly fault tolerant distributed file system designed to run commodity hardware. HDFS will provide high throughput access to application data and provides efficient processing of large datasets. In this stage of service recommendation all the preprocessed form of movie information, user information and rating information will be uploaded to the HDFS.

Categorization process will classify the movies according to its categories like Horror, Animation, Children, and Comedy etc. Movies belonging to each category will be identified and will be arranged according to the category to which they belong. This will help the processing of further steps in recommendation system.

4) Capture current user's preferences

The preference of the current user who is in search of appropriate movie will be collected. Since we are using the approach of combined preference, user will have an opportunity to select multiple categories according to his preference. Here the user will be given a set of keywords and from that, the user can select the keywords that are related to the category of movies and those words will be stored in the keyword list according to its order of preference.

Based on the preference set supplied by the user, the relevant results will be identified and all those movies that are irrelevant will be filtered out. Intersection operation will help to identify the relevant results. All the relevant results will be uploaded to the HDFS.

5) Similarity measurement

Next step is to measure the similarity between the preference of the current user and that of previous users who have already given their reviews and ratings for movies. Two methods are adopted for calculating the similarity Relative similarity computation method and Accurate similarity computation method. Before the similarity measurement, reviews of users that are not related to what the current user prefers are filtered out and it is done by employing intersection concept. The intersection of the keyword list of current users and previous users are considered and those reviews that return a null value will not be considered for similarity measurement.

a) Relative similarity Measurement

Jaccard coefficient is used to measure the relative similarity. Jaccard coefficient is used to compute the similarity and diversity of sample sets. Similarities between the preferences of the current and previous users are computed with the help of following equation.

$$Sim(CK, PK) = Jaccard(CK, PK) = \frac{|CK \cap PK|}{|CK \cup PK|} \qquad (1)$$

b) Accurate Similarity Measurement

Accurate similarity measurement uses a cosine based approach. In this approach the preference of the current and previous users will be transformed into *n*-dimensional weight vector, which can be named as preference weight vector $\overrightarrow{W_p} = [w_1, w_2, \dots\dots, w_n]$, n is the number of words in keyword list, w_i is the weight of the keyword k_i in the keyword list. If k_i is not present in the keyword list then its weight w_i will be taken as 0. Preference weight vector of the current and previous users can be noted as $\overrightarrow{W_{CP}}$ and $\overrightarrow{W_{PP}}$ respectively.

Analytic Hierarchy Process model is used to measure the weight of the preference keyword set of the current user. Firstly a pairwise comparison matrix is constructed in terms of relative importance between each pair of keywords. The pairwise comparison matrix $A_m = (a_{ij})_m$ must satisfy the following properties, a_{ij} represents the relative importance between two keywords and m is the number of keywords in the preference keyword set of current user:

1) $a_{ij} = 1 \qquad i = j = 1,2,3, \dots\dots, m$
2) $a_{ij} = \frac{1}{a_{ij}} \qquad i, j = 1,2,3, \dots\dots, m \; and \; i \neq j$
3) $a_{ij} = \frac{a_{ik}}{a_{jk}} \qquad i, j, k = 1,2,3, \dots\dots, m \; and \; i \neq j$

After this, the weight can be calculated using the following function:

$$w_i = \frac{1}{m} \sum_{j=1}^{m} \frac{a_{ij}}{\sum_{k=1}^{m} a_{kj}} \qquad (2)$$

Similarity computed using cosine-based approach can be defined as follows:

$$sim(AK, PK) = \cos\left(\overrightarrow{W_{CP}}.\overrightarrow{W_{PP}}\right) = \frac{\overrightarrow{W_{CP}}.\overrightarrow{W_{PP}}}{\|\overrightarrow{W_{CP}}\|_2 \times \|\overrightarrow{W_{PP}}\|_2} \qquad (3)$$

$$= \frac{\sum_{i=1}^{n} \overrightarrow{W_{CP,i}} \times \overrightarrow{W_{PP,i}}}{\sqrt{\sum_{i=1}^{n} W^2\ CP,i}\sqrt{\sum_{i=1}^{n} W^2\ ,PP,i}}$$

Where $\overrightarrow{W_{CP}}$ and $\overrightarrow{W_{PP}}$ are the preference weight vectors of current and previous users respectively.

6) Calculation of personalized ratings and ranking process

On the basis of the similarity computed in the previous steps further filtering will be carried out, for that we consider a threshold value δ, if $sim(AK, PK_j) < \delta$ the preference keyword set of the previous users PK_j will not be considered further. Once the most related users are figured out personalized ratings for each current user will be computed and recommendation list will be provided to them.

Personalized ratings *pr* is calculated based on weighted average approach.

In the above equations $sim(CK, PK_j)$ is the similarity of the preference list of current and previous users; *k* is the multiplier used as normalizing factor; $\hat{R}$ denotes the set of previous users keyword list remaining after filtering out; r_j is the rating of the review *j* and $\bar{r}$ is defined as the average rating of services.

Based on the above steps proper recommendation will be provided to the current user and the algorithm for its working is given below.

B. Algorithm 1:

Input: The preference keywords of the current user *CK*
The candidate services
The threshold δ in the filtering phase
The number *K*
Output: The services with Top-K highest ratings

1) for each service $ws_i \epsilon WS$
2) $\hat{R} = \emptyset, sum = 0, r = 0$
3) for each review R_j of service ws_i
4) process the review into preference keyword set PK_j
5) if $PK_j \cap CK = \emptyset$ then
6) insert PK_j into $\hat{R}$
7) end if
8) end for
9) for each keyword set $PK_j \epsilon \hat{R}$
10) $sim(CK, PK_j) = SIM(CK, PK_j)$
11) if $SIM(CK, PK_j) < \delta$ then
12) remove PK_j from $\hat{R}$
13) else $sum = sum + 1, r = r + r_j$
14) end if
15) end for
16) $\bar{r} = {}^{r}/_{sum}$
17) get pr_i by formula
18) end for
19) sort the services according to the personalized ratings pr_i
20) return the Top-K services with highest ratings

III. IMPLEMENTATION ON MAPREDUCE

To improve the scalability and efficiency of recommendation system in "Big Data" environment, we implement it in a MapReduce framework on Hadoop platform.

A. Personalized-RSM on MapReduce

Personalized-RSM on MapReduce, operates in three steps. Step1 is offline executed and step 2 and step 3 are online executed.

1) Step 1: The first step is to process the reviews for candidate services by previous users into their preference keyword sets and compute the average ratings for each candidate service. Map-I: Map $\langle i, j, r_{ij}, R_{ij} \rangle$ on *i* such that the tuples with same *i* are shuffled to the same node in the form of $\langle j, r_{ij}, R_{ij} \rangle$. Reduce-I: Take $\langle j, r_{ij}, R_{ij} \rangle$ as input and emit $\langle i, j, r_{ij}, PK_{ij}, \bar{r_i} \rangle$ for each input of Map-I. The output of Reduce-I
$O_1 = \{\langle i, j, r_{ij}, PK_{ij}, \bar{r_i} \rangle\}, 1\epsilon[1, N]$ will be used as the input of Map-II to calculate the similarity.

2) Step 2: The second step is to compute similarity between the current user and previous users. Map-II: Map on *i* and tuples with the same *i* are shuffled to the same node in the form $\langle j, r_{ij}, PK_{ij}, \overline{r_j} \rangle$. Reduce-II Take <CK> and $< j, r_{ij}, PK_{ij}, \overline{r_j} >$as input then emit $sim = \langle i, j, r_{ij}, S_{RSM}^{ij}, \overline{r_i} \rangle, i \in [1, N]$.
3) Step 3: The third step aims to calculate the personalized rating of each candidate service and present a personalized recommendation list to the current user. Based on the output of this step, the recommendation can be obtained. Map-III: Map $\langle i, j, r_{ij}, S_{RSM}^{ij}, \overline{r_i} \rangle$ on *i* so that the tuples with same *i* are shuffled to the same node in form of $\langle j, r_{ij}, S_{RSM}^{ij}, \overline{r_i} \rangle$. Reduce-III: Take $\langle j, r_{ij}, S_{RSM}^{ij}, \overline{r_i} \rangle$ as input and emit ranking list, where pr_i is the personalized rating of the current user to service *i.* The tuples of the output are ordered by the services id *i*, which is just the personalized service recommendation list to the current user.

B. Personalized-ASM on MapReduce

Personalized-ASM on MapReduce, which consists of four steps. Step 1 and step 2 are offline executed, and step3 and step 4 are online executed.

1) Step 1: The first step of the flowchart of personalized-ASM on MapReduce is same as step 1 of the flowchart of personalized-RSM on MapReduce.
2) Step 2: The second step is to process all reviews of each previous user into corresponding keyword sets respectively and takes advantage of TF-IDF measurement to calculate the preference weight vectors of the previous users. Map-II: Map $< j, R_{jv}^{`} >$ on j such that reviews of the same user j are shuffled to the same node in the form of$< R_{jy}^{'} >$. Reduce-II: Take $< R_{jy}^{'} >$ as input and emit$< j, \overrightarrow{W_{PP_j}} >$. $\overrightarrow{W_{PP_j}}$ is the preference weight vector of a previous user j. And the tuples $\left\{ \langle j, \overrightarrow{W_{PP_j}} \rangle \right\}$ will be used to calculate the similarity in Reduce-III.
3) Step 3: The third step is to compute the similarity between the current user and previous users. Map III: Map $\langle i, j, r_{ij}, PK_{ij}, \overline{r_i} \rangle$ on I, and tuples with the same I are shuffled to the same node in the form of $\langle j, r_{ij}, PK_{ij}, \overline{r_i} \rangle$. Reduce-III: Take $\langle \overrightarrow{W_{AP}} \rangle$,$\langle j, \overrightarrow{W_{PP_j}} \rangle$ and $\langle j, r_{ij}, PK_{ij,} \overline{r_i} \rangle$ as input, emits $sim = \langle i, j. r_{ij}, S_{ASM}^{ij}, \overline{r_i} \rangle, i \in [1, N]$ as output.
4) Step 4: The last step is same as the step 3 of the flowchart of Personalized-RSM on MapReduce. Based on the output of this step, we can present a personalized recommendation list to the current user and recommend the most appropriate services for him/her

IV. CONCLUSIONS AND FUTURE SCOPE

In this paper we have proposed a personalized service recommendation for Big Data applications. In this keywords are used to indicate user preferences, based on these keywords similarity between the current and previous users are identified using user based Collaborative Filtering algorithm. Most suitable services will be recommended to the current user. By deploying our recommendation system on MapReduce framework of Hadoop scalability and inefficiency problems in Big Data environment are solved to an extent. Current user will give his/her preference by selecting the keywords from the keyword candidate list and the preference of the previous users are extracted from their reviews. Using well formulated collaborative filtering algorithm appropriate services can be recommended to the user according to users' individual taste and preferences. The use of rank boosting approach provides an opportunity for the current user to select more than one keyword from the keyword candidate list. The user will get recommendation list as a combination of his preferences.

This service recommendation system can be further improved by considering user reviews for individual movies. As a future work we can incorporate the dictionary feature, which will help to identify words with same meanings. Thus improve the result produced by the recommendation system.

ACKNOWLEDGMENT

The authors of this paper would like to express sincere gratitude to the CSE department of Nehru College of Engineering and research Centre for offering their full support and guidance in proceeding with this project.

REFERENCES

[1] Greg Linden, Brent Smith and Jeremy York "Amazon.com Recommendations Item to Item Collaborative filtering," IEEE Internet computing, February 2003.
[2] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro "Content-based Recommender Sytems: State of the Arts and trends," Springer Science + Business Media, LLC 2011.
[3] Badrul Sarwar, George Karypis, Joseph Constan and John Riedl,"Item-based Collaborative Filtering Recommendation Algorithm," ACM 1-58113,348-0/01/0005, 2001.
[4] Zhi-Dan Zhao, Ming-Sheng Shang, "User-based Collaborative filtering Recommendation Algorithms on Hadoop," Third international conference on Knowledge Discovert and Data Mining, 2010.
[5] Robin Burke, "Hybrid Recommender Systems: Survey and Experiments".
[6] Yoav Freund, Raj Iyer, Robert E. Schapire, Yoram Singer, "An efficient Boosting Algorithm for Combining Preferences," Journal of Machine Learning Research, 2003.
[7] Shunmei Meng, Wanchun Dou, Xuyun Zhang, Jinjun "KASR: Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications," IEEE Trans. On parallel and distributed system, March 2014.