International Journal of Science Engineering and Advance Technology

Individualised Recommender System Using Keywords With Efficient Similarity Computation Employing Mapreduce

1K. Sindhuja1,2D. Srinivas
1M.Tech (Student), 2Asst. Professor in Dept. of Computer Science and Engineering
Kakinada Institute of Engineering & Technology ,Korangi, E.G.dt, A.P.

ABSTRACT

The advancement in E-commerce enabled the websites to overwhelm users with numerous services. Often times, users find it a challenging task to choose the appropriate and best service from the availabe services. To provide the user with the most appropriate options, Recommender system, which is an information filtering technique, can be employed. Traditional recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences. A individualized recommender system addresses the versatile requirements of different users. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. However, finding similarity, which is heart of any recommender system, plays a vital role. The best similarity computation method contributes massively for an efficient recommender system.

The proposed work implements three similarity computation methods. These methods are compared with the existing methods and the results proved that the proposed techniques outperformed the existing techniques. To improve its scalability and efficiency in big data environment, proposed approach is implemented on Hadoop, a widely-adopted distributed computing platform using the MapReduce parallel processing paradigm.

Keywords

Big Data, Hadoop, MapReduce, Recommender systems, Collacorative filtering.

1. INTRODUCTION

The progress in technology in the area of Electronic commerce has enabled businesses to open up their products and services to a monolithic client base. As the rival among businesses becomes very boisterous, users are presented with a multitude of choices and hence information overburden. In the recommender system, the ill of information overload will be revealed when the user is provided with too many choices to opt from, most of which may not be wanted by the user. Hence an efficient scheme would capture the user preferences from his former purchase history and use the same for future individualized recommendation.

The preferences of user are saved every time he purchases something. We can collect the preferences of the user from his explicit feedback. But, this feedback often times may not be plausible, as it may result in frustrating the user. One way of acting upon around this problem would be to accumulate implicit feedback about the user preferences by using the user profile and by mining it.

The aim of this project is to use the Hadoop effectively creating recommender system that provides the user fast and best system. This project mainly focuses on the similarity computations in collaborative filtering. This project will be using an efficient similarity computation method known as Dicemin. Along with Dicemin, other efficient methods such as Jaccardmin and Lim are also implemented. The main concern of the scheme is the ability to recommend the user, the itinerary that fits best in his selection, as per the reviews of existing users. The system takes in the user keyword as input and generates a list of itineraries that the user is free to select from depending on his personal preference. The input essentially contains the user choice which is here called as current user keyword. A sorted list of recommended results is then displayed by performing suitable operations in map-reduce in Hadoop.

2. RELATED WORK

The ability for e-commerce to open up their products and services is enhanced by progress in technology. The myriad of choices available to the user may lead to confusion. Although this is a boon to any customer, this boon should be harvested further so as to use it in the best way.

This problem can be circumvented by using recommender systems. As suggested by the name, recommender systems recommend the most appropriate and apt services

or items to the user irrespective of user preferences. Websites use these systems as marketing tools so as to increase their revenue. This will be done by presenting to the user such products which are most likely to be bought by the customer. An internet site using a recommender system can exploit knowledge of customers' likes and dislikes to build an understanding of their individual needs and thereby increase customer loyalty.

Collaborative filtering is used in the most former recommender systems. It is also named as social filtering. These algorithms don't focus on internal qualities of the items to be recommended but will focus on the behavior of the users. This approach is similar to the nature of "real-life recommendations". We can assume that these algorithms have a semantic affinity to both the concept of collaborating individuals and the process of finding persons with similar interest. On the other hand, Content-based systems focus on the internal nature of items, or on the content of items. These systems utilize two main classes of algorithms, either from the field of information retrieval or attribute-based filtering systems. A content-based approach favors the semantics of the content over social interactions or user behavior. In some application domains, the content of an item may be crucial to every application. This means that systems with a severe focus on item content should use a content-based approach rather than a social approach (i.e. on the actual content, not on user interaction).

Knowledge-based recommender systems will be used in scenarios where content based filtering and collaborative filtering won't work out. For this method of recommendations we need explicit knowledge on item assortment. A major advantage in using Knowledge-based Filtering is that it is free from cold start problem. So knowledge based filtering will be best suited for applications where we have some explicit knowledge on the item. Hybrid systems take advantage of above methods and metrics by merging them as per the requirement. Generally, hybrid recommender systems would calculate ratings using a number of "internal algorithms", and combine these in a unique metric to allow ordered ranking. In some cases, the first results of the internal algorithms are saved component-wise in a vector and crafted into a single-dimensional rating for ranking.

CF algorithm is a classic individualized recommendation algorithm, which is widely used in many commercial recommender systems. In CF based systems, users receive recommendations based on people who have similar tastes and preferences, which can be further classified into item-based CF and user-based CF. In item-based systems, the predicted rating depends on the ratings of other similar items by the same user. While in user-based systems, the prediction of the rating of an item for a user depends upon the ratings of the same item rated by similar users and in this work, we take the advantage of a user-based CF algorithm to deal with our problem.

Similarity should be computed among the users using which the individualized ratings and individualized recommendations can be provided. The following are some of the existing similarity computation methods:

Base method is one of the computationally efficient method which may not produce desired outputs. In this method, the weight vector will be comprising only binary values which are 0 and 1. Here, in this vector, 0 represents absence of entity while 1 represents presence of the entity. This method simply finds out the magnitude of intersection of the binary values.

The formula for base method is as follows:

$$Base(w_1,w_2)=||BIN(w_1) \cap BIN(w_2)||$$

DiceBin also operates on the vectors which comprise of binary weights. That means any weight in a vector will be having either zero or one as their values. The DiceBin gives the similarity as the ratio of twice the intersection of binary vectors to the sum of all the weight values in both the weight vectors. The similarity of weight vectors can be calculated using the following formula.

$$DiceBin(w_1,w_2)= \frac{2||BIN(w1) \cap BIN(w2)||}{||BIN(w1)+bin(w2)||}$$

JaccardBin method computes the similarity between two vectors by calculating the ratio of intersection of the two vectors to the union of two vectors. The weight vectors used by this method are also binary vectors. Despite its similarity with DiceBin, the JaccardBin has its own pros. The similarity computation will be carried out as follows:

$$JaccardBin\ (w_1,w_2)= \frac{||BIN(w1) \cap BIN(w2)||}{||BIN(w1) \cup bin(w2)||}$$

In *service oriented computing*, Cloud computing is a successful paradigm which changed the way in which computing infrastructure is used and abstracted. Sharing resources is the major goal of this one. Cloud computing can provide effective platforms to process large volumes of data which facilitates parallel computing. This has gained significant amount of attention recently. There are numerous cloud computing tools available, such as Hadoop, MapReduce of Google, etc. Other examples include the Dynamo of Amazon.com and the Ask.com's Dryad of Microsoft and Neptune etc. Hadoop is the most popular among all open source cloud computing platforms which supports mass data storage with good fault tolerance and MapReduce programming framework. MapReduce. proposed by Google is a popular distributed implementation model which is inspired by Lisp programming language's map and reduce operations.

ISSN 2321-6905

Thus the cloud computing tools aforementioned can be used to improve the scalability and efficiency of service recommendation methods in the "Big Data" environment.

3. PROPOSED MODEL

This paper proposes computation of user preferences based on the preferences provided by the user. In the following discussion, we will be referring the user to whom we have to provide recommendations, as current user and the users who have already written their reviews, as existing users.

The existing approach takes into consideration, the preferences of the user. This recommends the services to the user by using the reviews written by the existing users. We extract the preferences of existing users by extracting keywords from their reviews.

Extraction of existing user preferences:

Data Structures:

For the purpose of identifying the existing user preferences, we use two data structures such as Domain thesaurus and Keyword candidate list.

First we use Keyword candidate list for extracting keywords from the existing user's reviews. The keyword candidate list consists of numerous keywords related to the jargon of the recommendation area. Whenever a existing user used a keyword which is in the keyword candidate list, it will be considered. The usage of a keyword by a user represents the users' interest in the domain.

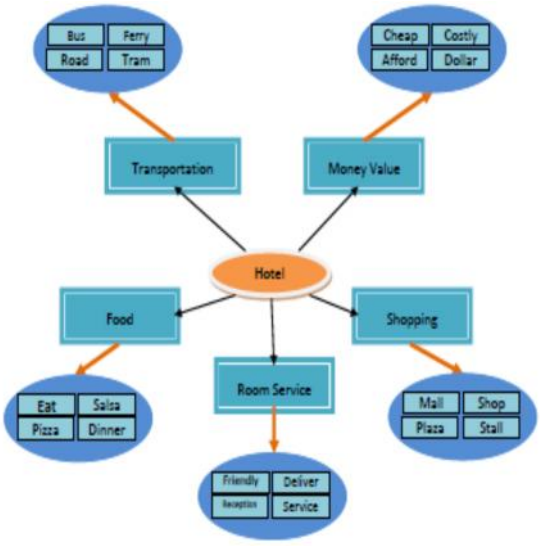
After the extraction of keywords is done, the next step is to map them into a domain thesaurus. It is a reference work which groups the keyword candidate list. All the keywords which are in the same domain thesaurus will have similar meaning or will be related to each other. Both these data structures work in such a way that a keyword used by a existing user reveals the interest of the user in a particular area or aspect.

The keyword extraction process is carried out as follows:

Keyword extraction:

For the purpose of knowing the preferences of the existing user we use the reviews written by the existing user. It is not the reviews that reveals directly about preferences of existing user. We should extract the key words from the existing user's reviews. For this we use domain thesaurus and key word candidate list.

We have two steps to find out the user preferences. First one is, to extract the keyword. For example, "spa" can be used in the reviews by the user and this could be a keyword. Using domain thesaurus we map the word "spa" to the domain "Fitness". If a keyword, is repeated number of times we will also consider the number of times the keyword is repeated.



Preference weight vector :

In cosine based approach we transform the keyword sets of the current user and existing user into n-dimensional weight vector. If the key word is not present in the preference key word set, then it's weight will be zero. The preference weight vector of existing user and current user are denoted as W_{AP} and w_{PP} respectively. The weight vector of the preference keyword set of the existing user is calculated by the Term Frequency/Inverse Document Frequency (TF-IDF).

The Term Frequency of the keyword is calculated by using the following formula:

$$TF = \frac{N_{pk_i}}{\sum_g N_{PK_i}}$$

Where, Npki is the number of occurrences of the keyword P_{ki} in all the keyword sets of the reviews commented by the same user u, g is the number of keywords in the preference keyword set of the user u.

The Inverse Document Frequency is the ratio of all reviews by the number of reviews containing the particular keyword. The formula is as shown below:

$$IDF=\log \frac{|R'|}{|r':P_{ki} \epsilon r'|}$$

Finally the weight of the keyword will be obtained by multiplying the term frequency with the inverse document frequency. This is as shown in the following formula.

$$W_{PK_i}=TF*IDF=\frac{Npki}{\sum_g N_{PK_i}} * \log \frac{|R'|}{|r':pki \epsilon r'|}$$

Extraction of current user preferences:

The importance degree of keywords of the current user should also be known, so as to provide him with best individualized ratings. For this purpose, current user has to select the importance degree of keywords by selecting a number in the scale of 1 to 5. Where 1 represents barely important, 3 represents moderately important and 5 represents as utterly important. The current user preferences have to be normalized using Analytic Hierarchy Process(AHP).

 ISSN 2321-6905

We use AHP to decide the weight of the keywords in the current user's preference keyword set. This is done as follows:

A pair wise comparison matrix which will be in terms of the relative importance between each two key words is constructed. Once the consistency of the matrix is checked, the weight is calculated using the following formula:

$$W_i = 1/m \sum_{j=1}^{m} \frac{a_{ij}}{\sum_{k=1}^{m} a_{kj}}$$

Here a_{ij} shows the relative importance between the two keywords

Similarity computation:

This step involves identification of the existing users who have the similar tastes to an current user. This is done by using the existing user's reviews. The preferences of the current user are also taken and are compared to the existing user's preferences. There are three methods of computing similarity.

Lin Similarity:

This method is proposed for calculating the semantic similarity of terms in a specific taxonomy in 1988. This measure depends on information content(IC) values allotted to the concepts in the taxonomy. In Lin's evaluation, the similarity between concept ci and cj not only depends on their shared information content, but also their own information content correspondingly. It assumes that the similarity among ci and cj can be calculated by the ratio between the amount of information needed to state the commonality.

The formula for calculation of similarity using Lim's method is as follows:

$$\text{Lin}(W_1,W_2)=\frac{\sum_{c_j \epsilon c_{1,2}} \left(A(W_1,c_j)+A(W_2,c_j) \right)}{\sum_j A(W_1,c_j)+\sum_k A(W_2,c_j)}$$

JaccardMin Similarity:

In this method, for calculation of similarity, only the smallest association weight is considered. It is a ratio of two components where the numerator is the smallest association which represents the sum of corresponding weights where both weights are non zeros or rated by both the current user and existing user. So consider the weight if and only if there is a non-zero rating given by both the users. On the other hand, the denominator doesn't have such restrictions. The denominator is simply the sum of all weights. The similarity can be calculated by taking the ratio of numerator and denominator.

Calculation of similarity is done using the following formula:

$$\text{JaccardMin}(w_1,w_2)=\frac{\sum_j min(A(W_1,c_j),A(W_2,c_j))}{\sum_j max(A(W_1,c_j),A(W_2,c_j))}$$

DiceMin similarity:

This is the most efficient similarity measure which will be producing minimum amount of error. DiceMin computes the similarity by taking the ratio of two components. The numerator is summation of all the corresponding weights of both the vectors. This indicates the amount of preference matching among the users. The multiplication of the weight implies the scaling of one user weight by other user's weight. The numerator comprises of scaled weights of the vectors.

The denominator comprises of sum of all the weights. This divides the numerator so that the result of the ratio can be considered as the similarity between the two users. The computation of similarity between the two users with their weight vectors can be deduced using DiceMin similarity using the following formula:

$$\text{DiceMin}(W1,W2)= \frac{2\sum_j min(A(W_1,c_j),A(W_2,c_j))}{\sum_j(A(W_1,c_j)+\sum_k A(W_2,c_j))}$$

Calculation of individualized ratings and generating recommendations:

Using similarity of the current user and existing user, we will filter them further by using a threshold for similarity computation. So, for calculation of individualized rating of an current user, we consider only the similarities which are greater than the thresholds. This process facilitates each user to get his own individualized ratings. We use a weighted average approach for calculation of individualized rating, P_r , regarding a service to the current user.

$$P_r = \bar{r} + k \sum_{ppk_j \epsilon \hat{R}} sim\left(APK.PPK_j\right) * (r_j - \bar{r})$$

Where $K=\frac{1}{\sum_{PPK_j \epsilon \hat{R}} Sim(APK,PPK_j)}$

After calculation of individualized ratings, top-k ratings will be presented to the user.

4. EVALUATION

For testing the efficiency of various methods used, the total dataset is divided into two parts of which one is 80 percent of the total dataset whereas the other will be of 20 percent of the total dataset. The large dataset is used as training dataset and small one is used as test dataset.

MAE and NMAE:

MAE is a statistical accuracy metric often used in Collaborative Filtering methods to measure the prediction quality. It is defined as the average absolute deviation between a predicted rating and the real rating.

$$MAE = \frac{\sum_{i=1}^{M} |q_i - p_i|}{M}$$

where qi and pi respectively denote the real rating and the corresponding predicted rating, and M is the number of the pairs of real ratings and predicted ratings < qi, pi >. The normalized mean absolute error (NMAE) metric is also used to measure the prediction accuracy, which is defined as:

$$NMAE = \frac{MAE}{\sum_{i=1}^{M} q_i / M}$$

The lower the MAE or NMAE presents the more accurate predictions.

The MAE and NMAE of the five similarity computational approaches are as follows:

Error	Cosine	PCC	Jaccard Min	Lin	Dicemin
NMAE	0.060434	0.073601	0.0514008	0.053369	0.051284
MAE	0.228899	0.279523	0.193497	0.200961	0.19305

Table 4.1: MAE and NMAE Values

The values from the table can be inferred from their pictorial representation given below.

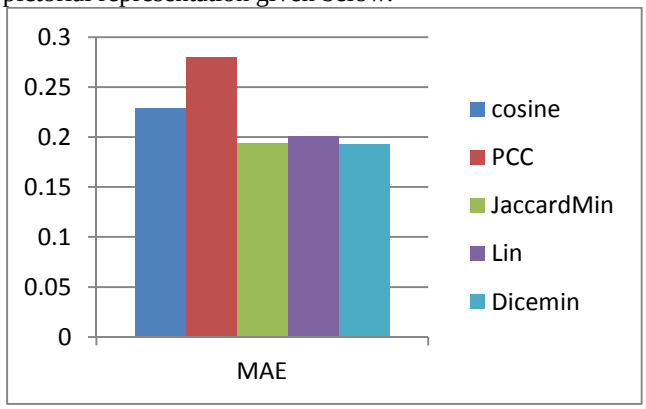


Fig. 4.1 :MAE Results

In the above bar chart, it can be observed that in the two existing methods which are Cosine based approach and Pearson Correlation Coefficient, Cosine based approach is better. On the other hand, among the three proposed methods, Lin is having the greater error. Finally DiceMin approach stood as the best approach for similarity computation with an error slightly less than that of JaccardMin.

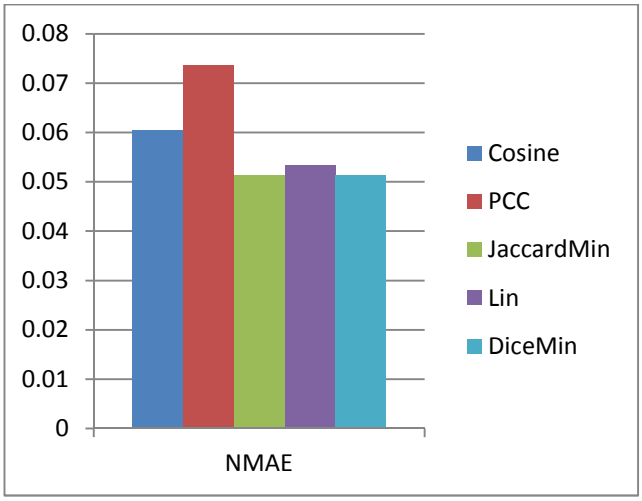


Fig. 4.2:NMAE Results

In the above graph, it can be deduced that minimum error is obtained when the similarity computing technique is DiceMin. The second best similarity computation technique is JaccardMin. It is clear from the above graph that all the three proposed techniques for computing similarity outperformed the two existing techniques. Of the two existing techniques, the cosine based approach is efficient when compared to the Pearson Correlation Coefficient.

5. CONCLUSIONS

The similarity computation is addressed by an efficient similarity computational approach, DiceMin. Other similarity computation approaches such as JaccardMin and Lim are also implemented and compared. The improved efficiency of the recommender system is proved by decreased Mean Absolute Error and Normalized Mean Absolute Error. Moreover, to improve the scalability and efficiency of the system in "Big Data" environment, we have implemented it on a MapReduce framework in Hadoop platform. Finally, the experimental results demonstrate that this significantly improves the accuracy of service recommender systems over existing approaches.

In the future work, distinguishing the positive and negative preferences of a user can be done so as to make the predictions more accurate.

REFERENCES

[1] Shunmei Meng, Wanchun Dou, Xuyun Zhang and Jinjun Chen, **"**KASR: A Keyword-Aware Service Recommendation Method on MapReduce for BigData Applications" IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 12, pp. 3221-3231, 2014.

[2] Pablo Gamallo Otero and Stefan Bordag, "Is singular value decomposition useful for word similarity

extraction?" Language resources and evaluation, Vol. 45, No. 2, pp. 95-119, 2011, Springer.
[3] Akihiro Yamashita, Hidenori Kawamura, and Keiji Suzuki. "Similarity Computation Method for Collaborative Filtering Based on Optimization" Journal of Advanced Computational and Intelligent Informatics, Vol.14 No.6, 2010.
[4] Y. Jing and W. Croft, "An association thesaurus for information retrieval," Proceedings of RIAO, Vol. 94, No. 1994, pp.146-160, 1994.
[5] Tom White, "Hadoop:The Definitive Guide, Second Edition", O'REILLY, 2010
[6] Jimmy Lin and Chris Dyer, "Data-Intensive Text Processing with MapReduce", Morgan and Claypool, 2010
[7] https://www.coursera.org/learn/recommender-systems
[8]https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
[9] https://Sifaka.cs.uiuc.edu/~wang296/Data/index.html
[10] https://bigdatauniversity.com/resources
[11] http://stackoverflow.com/

Ms. **K. Sindhuja** is a student of Kakinada Institute of Engineering &Technology (KIET), Korangi. Currently She is pursuing her M.Tech (CS) (13B21D0536) from this college. She received her graduation from Vignan Institute of engineering for women, Duvvada in The year 2012. Her Area of interest is Big data

Mr. **D.Srinivas** is working as Assistant Professor in KIET. He has 6 years of teaching experience.
He completed his B.tech. from KIET in 2007.He completed his M.tech from GIET Rajahmundry in 2010. His Areas of interests are DBMS & Networks He had published his paper in International Journal of computer science & Technology.