I.J. Information Technology and Computer Science, **2018, 4, 73-80**
Published Online April 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2018.04.08

Hike the Performance of Collaborative Filtering Algorithm with the Inclusion of Multiple Attributes

Barkha A. Wadhvani
Department of Computer Science, Government Engineering College Modasa, India
E-mail: barkha.wadhvani@gmail.com

Sameer A. Chauhan
Department of Computer Science, Government Engineering College Modasa, India
E-mail: sachauhan@gecmodasa.org

Received: 02 June 2017; Accepted: 09 March 2018; Published: 08 April 2018

***Abstract*—**At a recent time, digital data increases very speedily from small business to large business. In this span of internet explosion, choices are also increases and it makes the selection of products very difficult for users so it demands some recommendation system which provides good and meaningful suggestions to users to help them to purchase or select products of their own choice and get benefited. Collaborative filtering technique works very productive to provide personalized suggestions. It works based on the past given ratings, behavior and choices of users to provide recommendations. To boost its performance many other algorithms and techniques can be combined with it. This paper describes the method to boost the performance of collaborative filtering algorithm by taking multiple attributes in consideration where each attribute has some weight.

***Index Terms*—**Recommendation system, Collaborative filtering algorithm, Multiple attributes, Distributed computation, Compute intensive tasks, TOPSIS, Hadoop.

I. Introduction

In this digitization era, everything goes digital whether it is businesses, shopping, news or banking. People love to buy or search online. But at recent, the data are increases day by day which makes information and choices also growing. It confuses users to select the right product or it may take much time to choose a product. So they need some suggestions for products or information from somewhere. So here recommendation system comes into the vision. It helps users to find or suggest right product and saves users time and energy.

Recommendations can be provided by many ways – Content-based filtering, Collaborative Filtering based, Association rule-based etc. In content-based filtering, it uses the contents or keywords from users profile, product history, users feedbacks etc to provide suggestions for the products. In association rule-based, it recommends the products that are associated well with other products. In collaborative filtering, it recommends products by their past given ratings to the products.

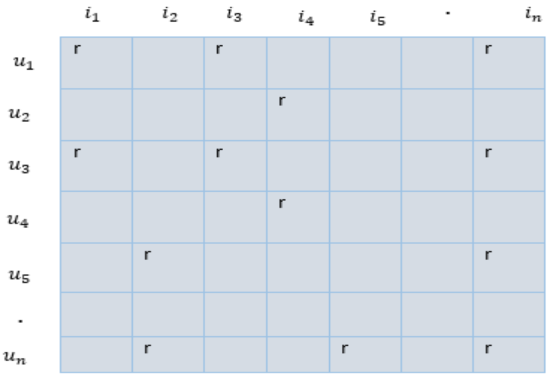
A. Collaborative filtering process

Collaborative filtering is the most commonly used and powerful approach. It helps to provide more individualized recommendations by collectively considering rating provided by users according to some similarity. There are two types of methods [1] for collaborative filtering – Model-based and Memory based approach. In a Model-based method, it builds a model from users past ratings or based on their profile or other attributes or characteristics like Location, Features, and Interests to recommend products to users. Clustering, Bayesian, neural network etc are used to develop the model. Once the Model is built, it will be very difficult to change. If any features changes then new model is to be built according to changes. In a memory based model, according to users rating given in the past to some products, it will predict the vacant rating for likely products. For Memory based Collaborative filtering, User-Item rating Matrix is required as shown in figure 1. u states user, i represents products and r specifies the rating given by a particular user to some product.

The memory-based approach is divided into two methods – user based and product based collaborative filtering. In user based, similar users are group together and based on their product ratings, the vacant ratings for products are to be predicted.

$$S_{u1,u2} = \frac{\sum_{i\in I}(r_{u1,i})(r_{u2,i})}{\sqrt{\sum_{i\in I}(r_{u1,i})^2}\sqrt{\sum_{i\in I}(r_{u2,i})^2}} \qquad (1)$$

$S_{u1,u2}$ is the similarity among user-1 and user-2. The similarity is calculated by cosine coefficient [1] formula by picking similarity among vectors of both users product ratings. I is the list of products rated by both of the users. If both users rate products with similar ratings then more similarity they get. Then user's vacant rating is to be found according to other users which are similar to that user.



	i_1	i_2	i_3	i_4	i_5	.	i_n
u_1	r		r				r
u_2				r			
u_3	r		r				r
u_4				r			
u_5		r					r
.							
u_n		r			r		r

Fig.1. Memory-based Collaborative Filtering User-Item rating Matrix

In Product based, product similarities are to be found using cosine coefficient. Now the vacant rating of the particular product is predicted using product which is similar to that product.

$$S_{i,j} = \frac{\sum_{u \in U} (r_{u,i})(r_{u,j})}{\sqrt{\sum_{u \in U} (r_{u,i})^2} \sqrt{\sum_{u \in U} (r_{u,j})^2}} \tag{2}$$

$S_{i,j}$ is the similarity between product i and product j and U is the list of users who rated both products i and j.

To predict the rating for a product, we can take a weighted average of all the ratings on that product with its similar products rating according to the following formula.

$$P(u,i) = \frac{\sum_{All\ products\ rated\ by\ user\ u} (r_{u,j} * S_{i,j})}{\sum_{All\ products\ rated\ by\ user\ u} (S_{i,j})} \tag{3}$$

P(u,i) is rating prediction for item i to user u.

B. Challenges of Collaborative filtering

Collaborative filtering suffers from many challenges [1]. Challenges are listed below:

- **Sparsity**: This problem happens when users rated very few products so it becomes very difficult to predict the ratings for maximum products because of lack of ratings as collaborative filtering predicts ratings based on users past ratings only.
- **Scalability**: As users and products increase nowadays, it also increases the computation time to provide the recommendation.
- **Shilling Attacks**: It occurs when people give favorable ratings to their product and negative ratings to their opposition so that their products can highly recommend to other users which make recommendations biased.
- **Gray Sheep**: This problem arises for the users whose likings does not match with other users so recommendation becomes difficult for them.
- **Cold Start**: Cold start problem occurs when new users and new products come in the market. New user or new product does not have any ratings so it makes recommendations difficult.

The rest of the paper is structured as follows. In Section II we discussed Hadoop and its tools and also presents how Collaborative Filtering Algorithm works as Big Data instance. Section III presents the analysis and related work. Section IV represents the multiple attribute decision-making method to select products according to their attributes suitability. Section V introduced the proposed method and its steps. Section VI represents the Experiments analysis including Dataset, Performance Parameters and Results. Section VII represents the discussion about how proposed method solves various challenges of collaborative filtering. Finally, Section VIII summarized the conclusion of this paper. Section IX specifies the future work.

II. Product Recommendation towards Big Data

The growing amount of data and need to survey that data in a timely manner for various reasons has created a barrier in big data analysis techniques [2]. Big data refers different types of data formats like structured, semi-structure and unstructured data and has various aspects – volume, velocity, value and veracity [3]. Big data application is divided into data-intensive application and compute intensive application [4]. Data-intensive application worked on massive data which is beyond the potential of storage and mostly in various formats. So if it is not possible for storage then it is not possible to process that data. In Compute intensive application data are not that much high but computation complexity is bit high so that it takes much time to process that data. Product recommendation is the remarkable illustration of Big data Computation. Memory-based Collaborative Filtering is the compute-intensive application as it requires more time for evaluating similarity among users or products to predicting ratings. Now as suppose there are millions of products or millions of users then it will take additional time for computations of similarities and predict the ratings. To get the computations time efficient and storage efficient various distributed computing platforms are required. Distribution computing platform is the integration of multiple nodes which are connected by some network to complete a common work [5]. [6] discussed different approaches to mounting the performance of collaborative filtering algorithm. They have reviewed that there are various distributed computing platforms Hadoop platform, Cloud platform, Cuda etc to solve the scalability problem and also try to resolve other challenges by Topsis, social trust inclusion, clustering etc approaches. We required to distribute our work to multiple nodes to process it parallelly or distributively. For that, any processing model is required. We mainly concentrate on Hadoop as a distributed environment and Map reduce as a processing model. Hadoop is a distributed platform to store, manage and process the data. Hadoop and its components for storage, processing, resource management are discussed in [7,8]. In [9] authors discussed Hadoop Distribution File System (HDFS) which is used to support Big data assuredly. Another file system supports very small block size in memory so access time will be increased. HDFS supports block size of 64MB or 128MB so that it will get higher bandwidth for large sequential read or write. Map Reduce is a programming model for data processing. For this model, our data should be in key and value form.

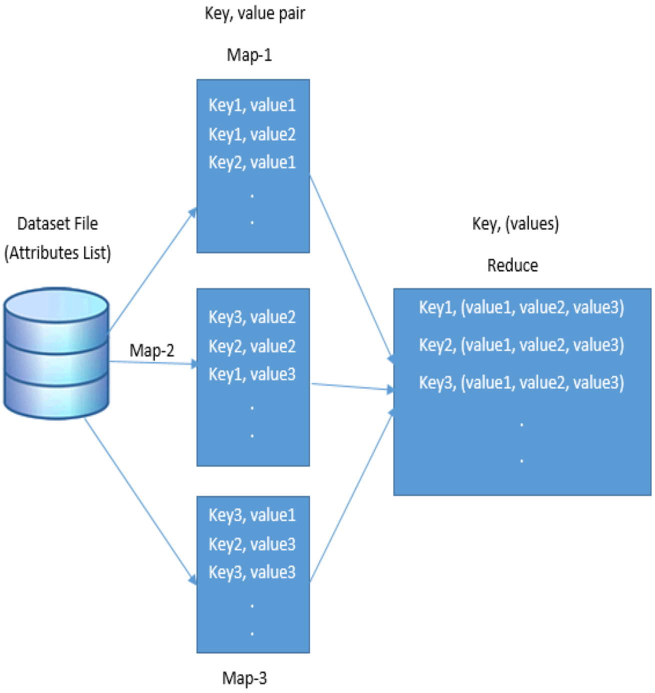


Fig.2. Map reduce Process

Figure 2 shows the map reduce process. Firstly our data are moved to the map function and mapper will divide it into multiple chunks of data and then distributes that chunks into different nodes for processing. Reduce phase splits into two processes - shuffle and reduce. Shuffle phase performs the sorting on map inputs on Key values and passed it to reduce phase. Reducer phase combines the output of every node and any task can be done [10].

III. Related Work

The algorithm is chosen based on the type of recommendations required. In [11], authors determined the self-organizing mapping (SOM) to optimize the improved k-means (IK) clustering in collaborative filtering. They concluded that this approach reduced the Mean Square Error and overcome scalability by using clustering and distributed platform. [new one] creates clusters according to users information according to their choices of genres or category and based on that they provide the suggestion. But a limitation of K-means is it generates only predefined or fixed number of clusters so it will provide recommendations only from its own cluster so may be essential information lost. [12] proposed ClubCF approach based cloud computing for big data applications. Primarily services are merged into the clusters to make it scalable as the cluster has the much lesser services than the entire system. It costs less online computation time and provides more accurate results as in each cluster, services are more related to each other. They concluded that ClubCF solved the sparsity challenge to some degree. But the limitation is the number of clusters is predefined.

With the inclusion of trust in collaborative filtering and users which are connected by socially or in other someway or considering users features, raises the prediction accuracy of product rating [13,14]. [15] presented a parallel approach that is based on social relationship to provide more accurate, scalable and trustworthy recommendations. All users have different choices and moreover, if users are socially connected then it doesn't necessary that their choices are same. So it is not possible to thoroughly trust social users and based on that provide recommendations. Cold start user problem can be solved using this approach to some extent but cold start product problem cannot be solved. In [16] authors compared both the approach product based and user-based collaborative filtering and result showed that the execution time improves by 30% with every add-on of a node into the Hadoop cluster. And they conclude that product based approach has more scalability than the user based approach. To unite the ability of some common node to process the big scale data in less time pipeline approach with product based approach can be used by creating Hadoop cluster [17]. It helps to utilize all the computing resources to make them all busy.

[18] proposed the multi-attribute group decision-making method – Technique of preference of similarity to ideal solution (TOPSIS). Paper concludes that Topsis is a very useful method to deal with the multi-attribute decision process. It is used to select the best service or product among the similar kind of services [19]. [20] used TOPSIS to select the best Engineering college according to various criteria of colleges and according to facility needed or prioritized by the students. In [21] authors represents the topsis for manufacturing enterprises for selecting optimal services and collaborative filtering to predict missing QoS values.

IV. Multiple Attribute Decision-making Method

Multiple attribute decision-making method is used to select the best alternative according to several attributes. Here we have used Technique of preference of similarity to ideal solution (TOPSIS) for multiple attribute decision making. It calculates closeness rank for each product based on shortest geometric distance from the positive ideal solution (PIS) and the longest geometric distance from the negative ideal solution (NIS). Each product's attributes have some numerical value and then it assigned some weight and according to that weight and attribute value, it computes the closeness rank to take some decision [22]. Closeness rank is generated between 0 to 1 and shows that how the product is closed to the ideal solution. The highest the closeness rank the ideal product it represents.

Below shows steps to calculate the closeness rank for all products [23].

Step-1: Construct normalized decision matrix.

Normalize scores or data are as follows:

$$\left[y_{ij} = x_{ij} / [x_{ij}^2]^{\frac{1}{2}} \right] \tag{4}$$

for i = 1 , …, m, j = 1, …, n.
m is Number Of Products or alternatives, n is Number of attributes, x is attribute value, y is a Normalized matrix, i is a product, j is an attribute of a product.

Step-2: Construct the weighted normalized decision matrix.

Suppose we have a set of weights for each attribute wj for j = 1,…n.

Multiply each column of the normalized matrix by its correlated weight.

An element of the new matrix is:

$$z_{ij} = w_j y_{ij} \tag{5}$$

z is normalized weighted matrix,w is weight

Step-3: Determine the ideal and negative ideal solutions.

Ideal solution be,

$$A^* = \{z_1^*, K\,K\,, z_n^*\}, z_j^* = \{\max(z_{ij})\} \tag{6}$$

zj* matrix representes maximum attribute value for each attribute

Negative ideal solution be,

$$A^{'} = \{z_1^{'}, K\,K\,, z_n^{'}\}, z_j^{'} = \{\max(z_{ij})\} \tag{7}$$

zj' matrix representes minimum attribute value for each attribute**.**

Step-4: Calculate the separation measures for each alternative.

The separation from the ideal alternative is:

$$S_i^* = \sum [z_j^* - \max(z_{ij}))^2]^{\frac{1}{2}} \quad ,i = 1, \ldots m \tag{8}$$

Similarly, the separation from the negative ideal alternative is:

$$S_i^{'} = \sum [z_j^{'} - \max(z_{ij})^2]^{\frac{1}{2}} \quad , i = 1, \ldots, m \tag{9}$$

Step-5: Calculate the relative closeness value Ci*

$$[C_i^{'} = S_i^{'} / (S_i^* + S_i^{'})] \quad , 0 < C_i^* < 1 \tag{10}$$

Relative closeness value is between 0 to 1.

V. Proposed Method

The proposed work concentrate to solve challenges of collaborative filtering technique. The proposed method is traditional product-based collaborative filtering with the inclusion of multiple attributes. It takes two dataset files – first is user – product rating (review) dataset file from which product similarities are to be found and second is the products or users attributes file which helps to calculate the rank of product based on its attributes value.

Proposed work is mainly divided into four steps.

Step-1: Generate product – product similarity so that based on that product similarities, vacant product ratings can be predicted. To calculate the product – product similarity, cosine correlation coefficient that is shown in formula (2) is used.

Step-2: Calculates the closeness rank of all products based on multiple attributes. Some weight is attached on each attribute of product. Weight can be decided by some survey process or by the attributes priorities given by the users. Closeness rank of the product is calculated using

Topsis method or any multiple attribute decision-making method.

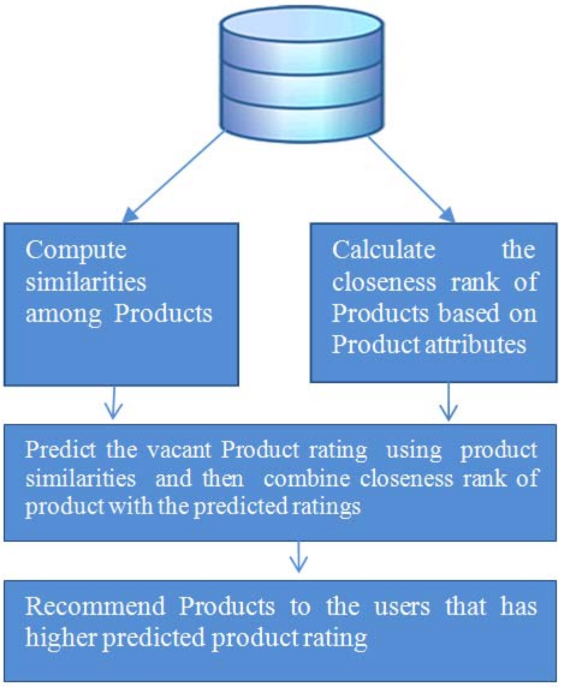

Compute similarities among Products
Calculate the closeness rank of Products based on Product attributes
Predict the vacant Product rating using product similarities and then combine closeness rank of product with the predicted ratings
Recommend Products to the users that has higher predicted product rating

Fig.3. Collaborative filtering with inclusion of multiple attributes

Step-3: Predict the vacant product ratings using product similarity (formula (3)) and after that combine (multiply) that predicted rating with ranks of products. For the product whose value is not predicted its by default value is 1.

$$FP(u,i)= P(u,i) * \text{rank value of product } i \tag{11}$$

P(u,i) is rating prediction for item i to user u, FP(u,i) is the final predicted rating of product.

Step-4: Above step generates the final predicted value for all products. Final predicted value depends on both past ratings given by users and closeness rank of the product. So products which are having good attributes values and past ratings values higher can only get the higher final predicted rating. Finally, recommend or lists the top-N products to the users which has highest predicted value.

VI. Experimental analysis

Experiments are conducted on distributed environment Hadoop which executes the job parallelly to complete it in tolerable time.

A. Dataset

To facilitate proposed work, we have used Yelp dataset [24]. It includes local businesses like dentists, hair stylists, restaurants, mechanics etc. The dataset contains 4.1M reviews and 947K tips by 1M users for 144K businesses and 1.1M business attributes. From that, we have taken Restaurant business as our product and two dataset files for our implementation. First dataset file that we have used is user-product rating file. Which represents the ratings given by users to products. The format for the file is as follows: userId, ProductId, Rating. Second dataset file that we have used is product Information file. That file contains ProductId, Attributes of that product.There are thousands of attributes are available to that particular business but we have only considered few attributes for our experiment. Product attribute dataset needs to be preprocessed before it is actually used in our experiment. For example, a Noise level attribute has given three values either Silent, Average or Noisy and it converted into 3,2,1 respectively because to compute closeness rank of each product we need numerical values only.

Restaurants Attributes that we have included for our implementation are the Noise level, Stars, and its value is between 0 to 5, Credit Card Availability attribute has its values are 0 for Not available and 1 for available, Price range attributes is given in $ and has values 1, 2 or 3, Wheelchair accessibility attributes has two values 1 for available and 0 for not available, Good for Kids, Good for Groups attribute again has two attributes 0 and 1, Wifi has three values 1 for No wifi, 2 for free wifi and 3 for paid wifi. These attributes respective weights are 8,7,6,5,4,3,2,1.

B. Evaluation parameter

The evaluation parameters that we have used here are Speedup and F1 score. Speedup parameter is used to measure the scalability metric of the approach. It analyses the execution performance change of a fixed size problem as the number of processors expands.

$$SpeedUp = \frac{T_a}{T_b} \tag{12}$$

T_a is single node running time, T_s is amount of parallel execution time with number of processors

The other parameter that we have taken here is the F1 score. The F1 score is the harmony mean of Precision and Recall parameter. There are many ways to measure precision and recall. Basically Recall parameter shows that how complete our results are. In our approach recall is how completely or maximum rating predictions are possible using this approach. Precision [1] measures how much your results are correct or reliable. The precision parameter we have measured for different threshold values. It shows how much correct results we get considering different threshold values. We have split our dataset into training dataset (80%) and test dataset (20%) and analyze the results for precision and recall.

$$\Pr ecision = \frac{True\ Positive}{True\ Positive + Flase\ Positive} \tag{13}$$

True Positive=PR>=Theashold & AR>=Threshold
False Positive= PR>=Theashold & AR<Threshold
Where,

PR is Predicted Rating and AR is Actual Rating

$$\text{Re}\,call = \frac{Hit\ test}{Test\ set} \qquad (14)$$

Hit test =products which appear in both sets are put in a set called hit set

$$F1 = \frac{2 * precision * recall}{precision + recall} \qquad (15)$$

The more F1 score shows more stability between Precision and Recall.

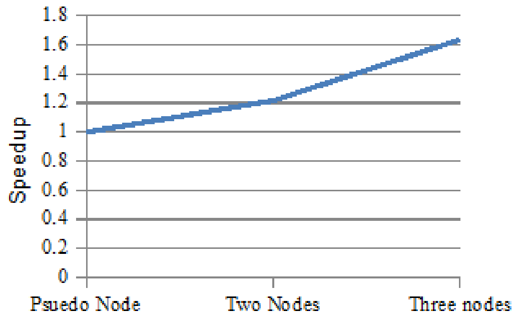


Fig.4. Speedup measure with Three Nodes

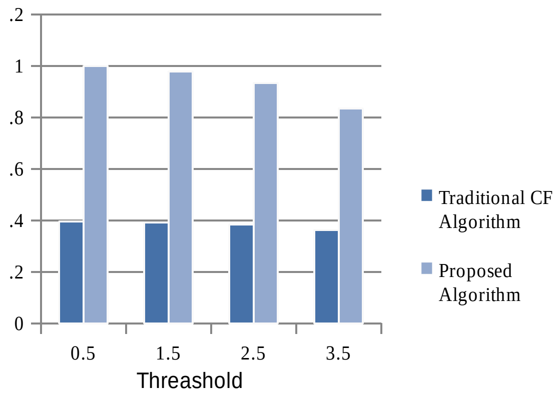


Fig.5. Comparision of Traditional and Proposed approach with F1 parameter

C. Results

We have executed proposed approach on Hadoop cluster with three nodes. The processing model that we have used is Map Reduce. The input data are divided into three nodes for execution so the time required for execution can be eliminated. Figure 4 shows the performance of approach with speedup metric. Here, with the growth of the number of nodes speedup is fairly increased. Table 1 represents the precision and recall values of Traditional CF and Proposed approach. Table 2 presents the comparison of both the approaches with the F1 score. Higher the precision, Recall and F1 represent better results. Figure 5 shows the graphical representation for comparison of both the conventional and proposed approach based on the F1 score. Precision, Recall, and F1 score is measured on some threshold values or on certain condition. The threshold is decided based on application's working or how efficient results you want to provide to the users. If you take very high threshold, the user will get only the most highest and desirable products only. In case of Lower the threshold, efficiency will be decreased as it includes all the products whose ratings are not that much higher. It shows that our proposed approach is outperformed than the traditional Collaborative Filtering approach.

Table 1. Comparision of Traditional and Proposed approach with Precision and Recall parameter

Threshold	Precision		Recall	
	Traditional	Proposed	Traditional	Proposed
0.5	1.0	1.0	0.246384	1.0
1.5	0.956672	0.957239	0.253824	1.0
2.5	0.868581	0.875680	0.238780	1.0
3.5	0.686564	0.716299	0.174110	1.0

Table 2. Comparision of Traditional and Proposed approach with F1 score

Threshold	F1 score	
	Traditional CF Algorithm	Proposed Algorithm
0.5	0.395359	1.0
1.5	0.391850	0.9781526
2.5	0.383877	0.9337203
3.5	0.362633	0.834701

VII. Discussion

Our proposed method can solve all the challenges of collaborative filtering algorithm by combining it with Topsis which is multiple decision-making methods. It can solve the scalability by dividing computation work to numerous nodes and cut down the time required to perform the task. Traditional product based is not always able to predict the rating for all products because of Sparsity, Cold start, Grey sheep problems. To solve these issues, proposed method combined predicted rating with closeness rank of that product. As in case if the prediction is not possible for some products or users then closeness ranks of that product can still help to predict the rating. The proposed method can also solve the shilling attack at some extent as our method also considers the multiple attributes value of the products for the prediction that makes it less biased.

VIII. Conclusion

Collaborative Filtering technique is the most intensive technique to provide recommendations. Many other techniques can be integrated with collaborative filtering to solve obstacles of that technique. Moreover, it is not always reliable to rely just on past ratings to predict the ratings for users. By taking multiple attributes into the consideration with conventional collaborative filtering can refine the performance and can also provide more individualized and reliable outcome as it not only considers past users ratings but also includes the product or users attributes. Proposed work is implemented on distributed platform Hadoop to minimize the computation time.

IX. Future work

In our work, weights for different attributes of the product are statically given. In future, we will use some productive and objective method to find weight or weight can be taken from users according to their need or choice to suggest them products. To get more effective results, we can add Text reviews and votes given by users to different products into our experiment. Also, we can include tweets of users about various products and can give some weight to that tweets in our decision of recommendation of tweets. So with the help of combining multiple techniques and taking the decision from various sources can give more effective results.

Acknowledgment

The authors wish to thank those who provided necessary information regarding this research and supported to complete this research and help direct or indirect way. Authors also take this opportunity to show gratitude to Government Engineering College, Modasa for providing Environment to complete this research.

References

[1] Xiaoyuan Su and Taghi M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009.

[2] Rasim M. Alguliyev, Rena T. Gasimova, Rahim N. Abbaslı,"The Obstacles in Big Data Process", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.4, pp.31-38, 2017. DOI: 10.5815/ijitcs.2017.04.05.

[3] Pooja Chaudhary and Virendra Kumar Yadav. A Survey on Security Issues and the Existing Solutions in Big Data. International Journal of Computer Applications 162(1):33-37, March 2017.

[4] I. Gorton, P. Greenfield, A. Szalay and R. Williams, "Data-Intensive Computing in the 21st Century," in Computer, vol. 41, no. 4, pp. 30-32, April 2008. doi: 10.1109/MC.2008.122.

[5] Weiyan X., Wenqing H., Dong L., Youyi D. (2013) A Distributed Computing Platform for Task Stream Processing. In: Yang Y., Ma M., Liu B. (eds) Information Computing and Applications. Communications in Computer and Information Science, vol 391. Springer, Berlin, Heidelberg.

[6] Barkha A. Wadhvani, Sameer A. Chauhan, "A Review on Scale up the Performance of Collaborative Filtering Algorithm", International Journal of Science and Research (IJSR), https://www.ijsr.net/archive/v6i4/v6i4.php, Volume 6 Issue 4, April 2017, 2528 - 2533, DOI: 10.21275/ART20172947

[7] Bibhudutta Jena, Mahendra Kumar Gourisaria, Siddharth Swarup Rautaray, Manjusha Pandey,"A Survey Work on Optimization Techniques Utilizing Map-Reduce Framework in Hadoop Cluster", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.4, pp.61-68, 2017. DOI: 10.5815/ijisa.2017.04.07.

[8] Luis Emilio Alvarez-Dionisi,"Envisioning Skills for Adopting, Managing, and Implementing Big Data Technology in the 21st Century", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.1, pp.18-25, 2017. DOI: 10.5815/ijitcs.2017.01.03.

[9] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The Hadoop Distributed File System. In Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) (MSST '10). IEEE Computer Society, Washington, DC, USA, 1-10.

[10] Seema Maitrey, C.K. Jha, MapReduce: Simplified Data Analysis of Big Data, In Procedia Computer Science, Volume 57, 2015, Pages 563-571, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.07.392.

[11] Z. Ma, Y. Yang, F. Wang, C. Li and L. Li, "The SOM Based Improved K-Means Clustering Collaborative Filtering Algorithm in TV Recommendation System," 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, 2014, pp.

[12] R. Hu, W. Dou and J. Liu, "ClubCF: A Clustering-Based Collaborative Filtering Approach for Big Data Application," in IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 3, pp. 302-313, Sept. 2014.

[13] P. Yu, "Recommendation method for mobile network based on user characteristics and user trust relationship," 2016 IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou, 2016, pp. 1-6.

[14] Z. Gao, Z. Lu, N. Deng and K. Niu, "A novel collaborative filtering recommendation algorithm based on user location," 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, 2016, pp. 1-2.

[15] D. Lalwani, D. V. L. N. Somayajulu and P. R. Krishna, "A community driven social recommendation system," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 821-826.

[16] P. Ghuli, A. Ghosh and R. Shettar, "A collaborative filtering recommendation engine in a distributed environment," 2014 International Conference on Contemporary Computing and Informatics (IC3I), Mysore, 2014, pp. 568-574.

[17] Z. L. Zhao, C. D. Wang, Y. Y. Wan, Z. W. Huang and J. H. Lai, "Pipeline Item-Based Collaborative Filtering Based on MapReduce," 2015 IEEE Fifth International Conference on Big Data and Cloud Computing, Dalian, 2015, pp. 9-14.

[18] W. Zhoul and W. Jiang, "Two-phase TOPSIS of uncertain multi-attribute group decision-making," in Journal of Systems Engineering and Electronics, vol. 21, no. 3, pp. 423-430, June 2010.

[19] Manel Mejri and Nadia Ben Azzouna. Scalable and Self-Adaptive Service Selection Method for the Internet of Things. International Journal of Computer Applications 167(10):43-49, June 2017.

[20] T. Miranda Lakshmi, V. Prasanna Venkatesan, A. Martin,"An Identification of Better Engineering College

with Conflicting Criteria using Adaptive TOPSIS", International Journal of Modern Education and Computer Science(IJMECS), Vol.8, No.5, pp.19-31, 2016.DOI: 10.5815/ijmecs.2016.05.03

[21] Wenyu Zhang, Shixiong Zhang, Shuai Zhang, Dejian Yu," A novel method for MCDM and evaluation of manufacturing services using collaborative filtering and IVIF theory",Journal of Algorithms & Computational TechnologyVol 10, Issue 1, pp. 40 – 51

[22] Hamdani Hamdani, Retantyo Wardoyo, Khabib Mustofa, "A Method of Weight Update in Group Decision-Making to Accommodate the Interests of All the Decision Makers", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.8, pp.1-10, 2017. DOI: 10.5815/ijisa.2017.08.01

[23] Topsis, sept, 2017. Retrived from, https://en.wikipedia.org/wiki/TOPSIS

[24] Dataset, Jan, 2017. Retrived from, https://www.yelp.com/dataset_challenge/dataset

Authors' Profiles

Barkha A. Wadhvani is Assistant Professor in ITM Vocational University, Vadodara, Gujarat. She has 1 year of Teaching Experience in Computer Engineering Department. She obtained her bachelor degree of Computer Engineering in GCET, Vallabh, Vidhyanagar, India (2014). And obtained her Master of Engineering in Computer science and engineering from Government Engineering College, Modasa, India (2017). Her Area of interest is Big Data, Data Mining and Machine learning. She can be reached at barkha.wadhvani@gmail.com

Sameer A. Chauhan is a Assistant Professor in Government Engineering College, Modasa. He has a more than a decade of Teaching experince. He has completed his Master of Engineering from IIT, Rorkee. His area of interest is Cloud Computing, Grid Computing and Data Mining. He can be reached at sachauhan@gecmodasa.org.

How to cite this paper: Barkha A. Wadhvani, Sameer A. Chauhan, "Hike the Performance of Collaborative Filtering Algorithm with the Inclusion of Multiple Attributes", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.4, pp.73-80, 2018. DOI: 10.5815/ijitcs.2018.04.08