PARTITIONED RCF: AN IMPROVED REVERSED COLLABORATIVE FILTERING ALGORITHM FOR MAXIMIZING RECOMMENDATIONS

Himalay H. Pathak[1], Vimalkumar B. Vaghela[2]

[1]M.E. Scholar, Computer Engineering Department,
L. D. College of Engineering, Ahmedabad, Gujarat, (India)
[2]Ph.D. Scholar, Computer Science & Engineering Department,
Karpagam University, Coimbatore, Tamilnadu, (India)

ABSTRACT

Collaborative Filtering (CF) is the prime technology used in the field of the Recommendation System. A method named Reversed CF is been developed for purpose of reducing the pre-processing and query-processing time without any compromise in the accuracy. The drawback of this method is that it provides less number of recommendations. Thus, Partitioned RCF is been designed to solve the problem of the less number of the recommendations. In this method, dataset is partitioned and then the Reverse CF is been applied on each partition and the recommendations from each partition is been merged together. As a result, the recommendations are been maximized. Also, the execution time of Partitioned RCF is comparatively less than Reversed CF.

Keywords – k-Nearest Neighbor, Maximizing Recommendations, Partitioned RCF, Reversed CF.

1. INTRODUCTION

Collaborative Filtering is a form of memory based reasoning[2], specifically applied for the recommendation. It starts with the history of user's interest and preferences. According to this, the distance function is used to find the other users with similar interest and preferences. On the basis of this similarity the targeted user can be recommended for a particular item/object.

If the method used for selecting the group of users, with similar interest and preferences, is k-Nearest Neighbor[3]. Then, the Collaborative Filtering is said to be based on k-Nearest Neighbor[4].

In the Reversed CF[1], the k-nearest neighbors are been selected on the basis of the unrated items, unlike the traditional CF, where the nearest neighbors are selected on the basis of rated items. It is been observed that there is a limitation of the RCF, that it isn't able of recommending many items, in the case when the parameter k' is small enough.

To overcome this, we have proposed a method of partitioning[5] the dataset and then applying the RCF to each partition. The results obtained are in favour of proving that the partitioning helps to obtain more recommendations and thus fixing the drawback of the RCF.

One additional benefit of partitioning is that the execution time decreases notably. This adds to the efficiency of the Partitioned RCF methodology of predicting the recommendations.

II RELATED WORK

Reversed CF, is a technique of recommending the predictions on the basis of the unrated items using the k-Nearest Neighbor graph. The neighbourhood graph is been formed by greedy filtering[6].

The Reversed CF comprises of 2 steps:

i) The approximate construction of the k'-Nearest Neighbor graph; &
ii) Finding the k nearest rated neighbor of the unrated items based on k'NN graph, which is then used for the recommendation.

The condition for the selection of k is $k' > k$.

All the unrated and the rated neighbors are selected of a particular user, say, an active user.

For every k' unrated items of the k'NN graph, then rated items are been selected that are been rated by a particular active user on the basis kNN algorithm. All other than the most k nearest neighbor are been dropped off the selection.

Then the prediction of the ratings for the k' unrated items, is been done by the following equation:

$$p_{a,i} = \bar{r_i} + \sum_{n \in S[i]}' (r_{a,n} - \bar{r_n}) * sim(i,n) \quad (1)$$

$$where, \quad sim(i,n) = \frac{\sum_{c \in C_u} r_{c,i} * r_{c,n}}{\sqrt{\sum_{c \in C_u} (r_{c,i})^2} \sqrt{\sum_{c \in C_u} (r_{c,n})^2}} \quad (2)$$

Here, $p_{a,i}$ is the prediction of rating for active user a , for item i, $\bar{r_i}$ and $\bar{r_n}$ are the average ratings of user i and neighbor n respectively, $sim(i,n)$ is the similarity between i and n.

III PARTITIONED RCF

For a Recommender System based on the Collaborative Filtering, the recommendation is the prime part of it. It is been observed that there is a limitation of the RCF, that it isn't able of recommending many items, in the case when the parameter k' is small enough.

For fixing this issue of RCF, we have proposed a method, of partitioning the dataset into equal parts, and then applying RCF on each partition and finally combining each partition's recommendation.

The steps of the Partitioned RCF are as shown in the flow of the architecture below:

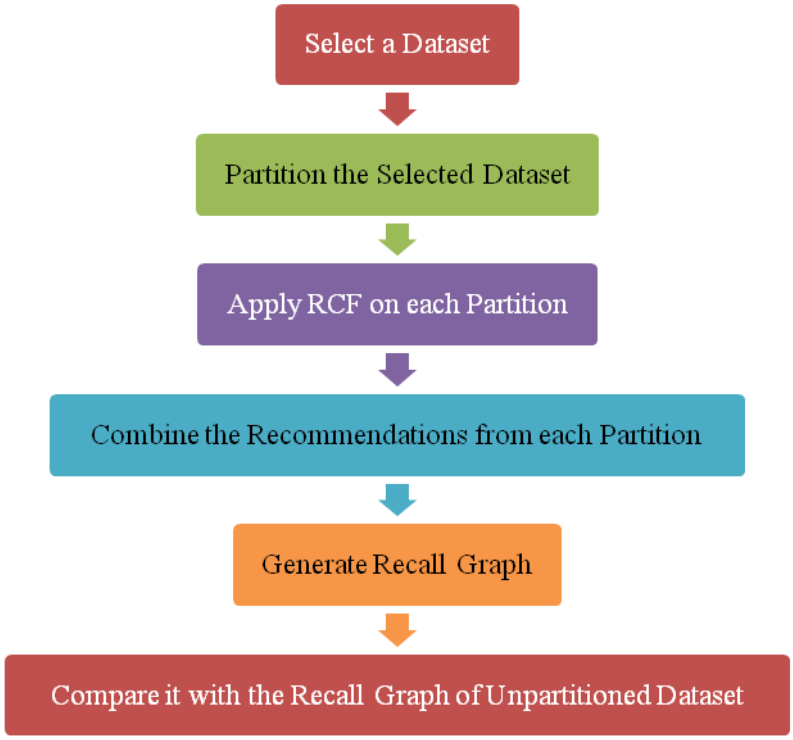


Figure 1: Flow of Partitioned RCF Architecture

The Time Complexity of the Partitioned RCF is $O[n*(n+u+r+m)]$ and the Space complexity of the Partitioned RCF is $O(n*m)$, where n is the number of partition, u is the unrated items of an active user, r is the rated items of an active user and m is the recommendation obtained from a single partition.

IV EXPERIMENTS

4.1 Experimental Setup

We will be using the MovieLens dataset is with 1,000,209 ratings of 3,952 movies rated by 6,040 users on a scale of 5-star (only whole-star rating allowed) and with at least 20 movies been rated by each user, as our testing database.

We will be using kNN as our algorithm for finding the neighbors from the neighbourhood graph, and here *k* refers to the number of the nearest neighbor of the active user. The value of k will not be chosen large as we are aimed to maximize the recommendation by keeping the value of the k small enough.

The value of *k'* will be taken 3[7] for finding the neighbors, of the unrated items from the graph. The value of *k* will be taken 1, when we will be predicting the ratings of the unrated item on the basis of the rated items by the active user.

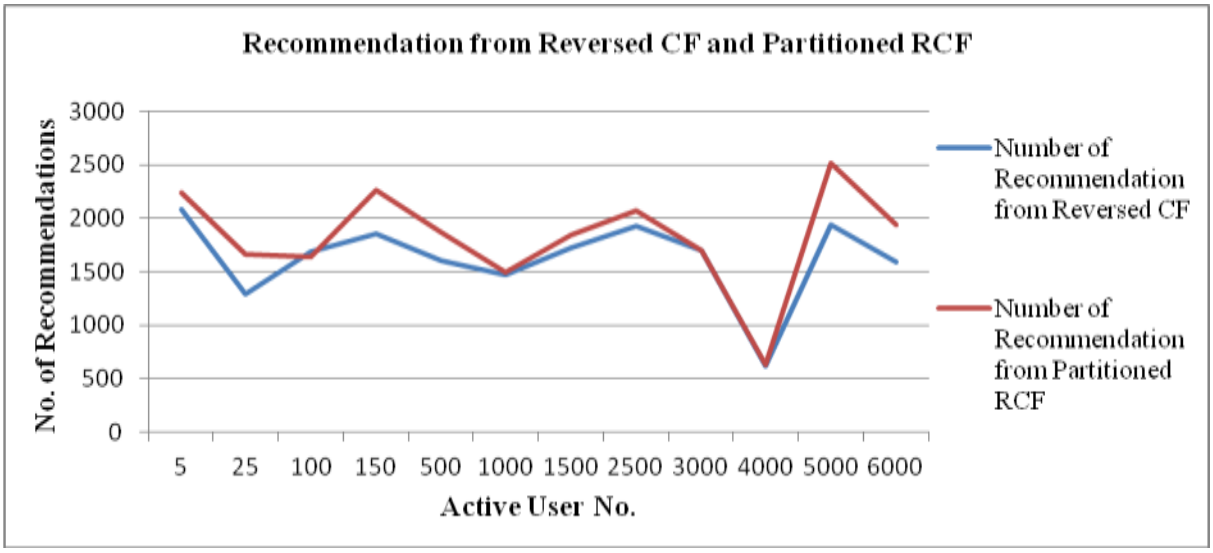
Here, the number of partitions created is 4. The partitions are been created in such a way that the rated items of an active user is been available in each partition. The reason of such partitioning is that the predictions are been made on the basis of the rated items of an active user.

4.2 Overall Comparison

The recommendations obtained from both the methods are been as shown in the following table:

Table 1: Recommendations from Reversed CF and Partitioned RCF

Active User No.	Number of Recommendation from Reversed CF	Number of Recommendation from Partitioned RCF
5	2086	2242
25	1294	1665
100	1690	1638
150	1858	2259
500	1602	1865
1000	1468	1490
1500	1726	1836
2500	1924	2070
3000	1693	1694
4000	614	633
5000	1935	2515
6000	1592	1940
Average	**1623.5**	**1820.6**



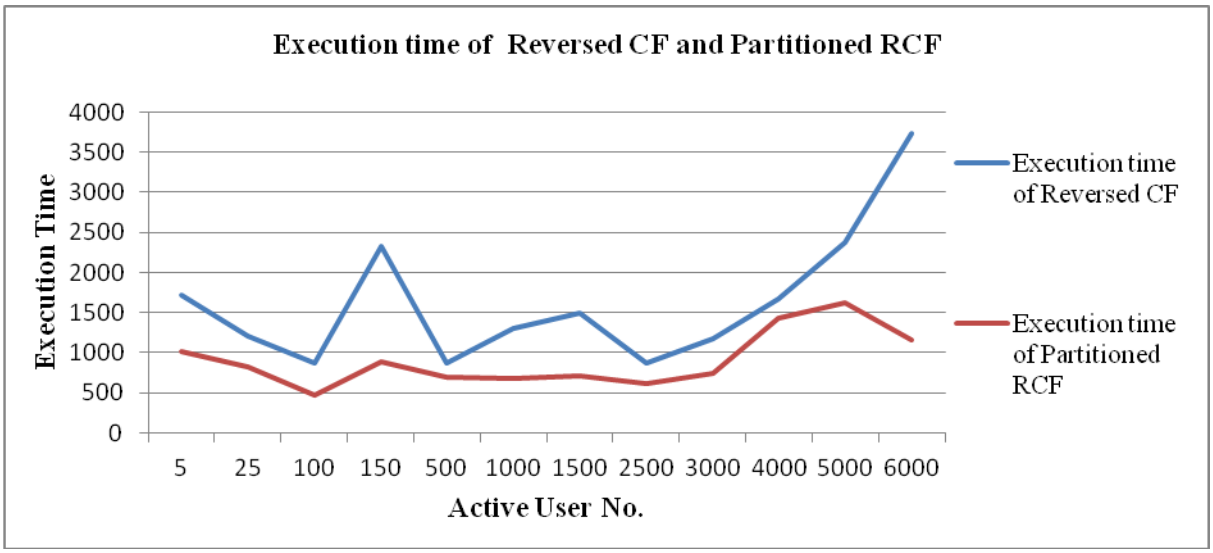
Graph 1: Recommendation from Reversed CF and Partitioned RCF

From the experimental results and the graph, we can state that the number of recommendations obtained from the Partitioned RCF approach is more as compared to that of the Reversed CF approach. Thus, the proposed method is able to fix the drawback of the Reversed CF method.

The execution time of both the methods is been as shown in the following table:

Table 2: Execution Time of Reversed CF and Partitioned RCF

Active User No.	Execution time of Reversed CF	Execution time of Partitioned RCF
5	1716	1023
25	1203	819
100	871	468
150	2320	885
500	878	701
1000	1311	678
1500	1501	718
2500	864	617
3000	1173	751
4000	1665	1427
5000	2376	1618
6000	3737	1165
Average	**1634.6**	**905.8**



Graph 2: Execution time of Reversed CF and Partitioned RCF

From the experimental results and the graph, we can state that the execution time of the Partitioned RCF approach is less as compared to that of the Reversed CF approach. Thus, the proposed method is efficient enough in terms of time as compared to Reversed CF method.

V CONCLUSION

The recommendation obtained on applying RCF using kNN on the dataset with a small value of k', is less. To maximize the recommendation by the Recommender System, we have proposed the method of partitioning the dataset, as Partitioned RCF.

The dataset is first partitioned and then RCF is been applied on each partition. And then the recommendations from each partitioned dataset are merged. On analysis we found that Partitioned RCF provides more recommendations with respect to the recommendations obtained from the RCF. Also, the execution time of the Partitioned RCF is less as compared to the execution time of RCF.

REFERENCES

1. Park, Youngki, et al. "Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph." *Expert Systems with Applications* 42.8 (2015): 4022-4028.
2. Berry, Michael JA, and Gordon S. Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.
3. Han, Jiawei, and Micheline Kamber. "Data mining: concepts and techniques." (2001).

4. Zhi-Hang, Tang, Shu Yuan-Jie, and Ouyang Wenmin. "A SURVEY OF COLLABORATIVE FILTERING BASED ON NEAREST-NEIGHBORS." *IJITR* 3.4 (2015): 2207-2212.
5. Lu, Zhigang, and Hong Shen. "A Security-assured Accuracy-maximised Privacy Preserving Collaborative Filtering Recommendation Algorithm."*Proceedings of the 19th International Database Engineering & Applications Symposium*. ACM, 2015.
6. Park, Youngki, et al. "Fast collaborative filtering with a k-nearest neighbor graph." *Big Data and Smart Computing (BIGCOMP), 2014 International Conference on*. IEEE, 2014.
7. Vaghela, Vimalkumar and Himalay Pathak. "Optimization of the Neighbor Parameter of k-Nearest Neighbor algorithm for Collaborative Filtering". Springer, 2016