EPiC Series in Computing

Volume 58, 2019, Pages 345–355

Proceedings of 34th International Conference on Computers and Their Applications

EPiC
Computing

A Novel Neighborhood Calculation Method by Assessing Users' Varying Preferences in Collaborative Filtering

Pradeep Kumar Singh[1], Pijush Kanti Dutta Pramanik[2], Narayan C. Debnath[3], and Prasenjit Choudhury[4]

[1] Dept. of Computer Science and Engineering, National Institute of Technology Durgapur , India
pksingh3009se@gmail.com

[2] Dept. of Computer Science and Engineering, National Institute of Technology Durgapur , India
pijushjld@yahoo.co.in

[3] School of Computing and Information Technology, Eastern International University, Vietnam
narayan.debnath@eiu.edu.vn

[4] Dept. of Computer Science and Engineering, National Institute of Technology Durgapur , India
prasenjit0007@yahoo.co.in

Abstract

To recommend an item to a target user, Collaborative Filtering (CF) considers the preferences of other similar users or neighbors. The accuracy of the recommendation depends on the effectiveness of assessing the neighbors. But over the time, the mutual likings of two individuals change; hence, the neighbors of the target user also should change. However, this shifting of preferences is not considered by traditional methods of calculating neighborhood in CF. As a result, the calculated set of neighbors does not always reflect the optimal neighborhood at any given point of time. In this paper, we argue for considering the continuous change in likings of the previous similar users and calculating the neighborhood of a target user based on different time periods. We propose a method that assesses the similarity between users in the different time period by using K-means clustering. This approach significantly improves the accuracy in the personalized recommendation. The performance of the proposed algorithm is tested on the MovieLens datasets (ml-100k and ml-1m) using different performance metrics viz. MAE, RMSE, Precision, Recall, F-score, and accuracy.

keywords: Recommendation systems, Collaborative Filtering, Top-N neighbor, K-means clustering, User similarity, Time variance, Personalized recommendation

1 Introduction

Collaborative Filtering (CF) is the most common filtering approach used by today's recommendation engines [12]. It tries to find similar users in terms of preference by assessing the closeness of ratings given by them to similar items. It is assumed that if the ratings given by two users to the similar items are similar, then the users might have similar likings [1, 3]. By so, the CF identifies a set of similar users called neighbors of a target user to whom an item will be recommended [4]. Definitely, by this approach, the accuracy of the recommendation depends on the accuracy of calculating the neighbors. There are several similarity measures which do this job successfully, but most of them often fail to consider the changing preferences of the users while finding similar users [12].

Because the taste and preferences of an individual change over time, the list of neighbors of a particular user also change. For example, Table 1 shows a list of five users and eight movies with the rating information from year 2001 to 2003.

Table 1: **Users' rating in different years**

Year	2001			2002			2003	
Movie User	A	B	C	D	E	F	G	H
User 1	3.5	4.5	1	2	4	3	4.5	4
User 2	3	4	1.5	1.5	2	1	2	1.5
User 3	0.5	1.5	0.5	1.5	4	3.5	2.5	3
User 4	1	1	0.5	2	0	3	0	0
User 5	0	0.5	0.5	0	0	1	4.5	4

Table 1 represents the changing rating behavior of User 1. The most similar user of User 1 is User 2, User 3, and User 5 in year 2001, 2002, and 2003 respectively.

The traditional similarity metrics consider all available data, i.e., the complete table for calculating neighbors. But, from the table, it is evident that the preferences of users have been changed over the years, which has been reflected at dissimilar ratings for the same item. Therefore, if the similarity is calculated based on the old ratings, the estimated neighborhood will not be optimal which will result in an inaccurate recommendation. We have proposed a novel neighborhood calculation method that considers the temporal shifting preferences of the users.

To include the temporal factor while determining the neighborhood, we have considered users' ratings for each year. We aim to find the cluster of most similar users for an optimized cluster of years. For this, the optimized K-means clustering algorithm (Elbow method) is applied over the years. Our proposed approach is based on the hypothesis that two users will be more similar if their yearly rating pattern on co-rated items and number of ratings given per year are similar.

Our approach will ensure that the set of neighbors of a user gets refreshed at certain time intervals, thus remain updated. This will improve the accuracy in the recommendation compared to using traditional similarity metrics.

The rest of the paper is organized as follows. Section 2 mentions some related works in this direction. Section 3 presents the proposed method and the solution approach. Section 4 does a comparative analysis of the proposed approach on the MovieLens datasets using the performance metrices such as MAE, RMSE, Precision, Recall, F-score, and accuracy. Section 5 concludes the paper.

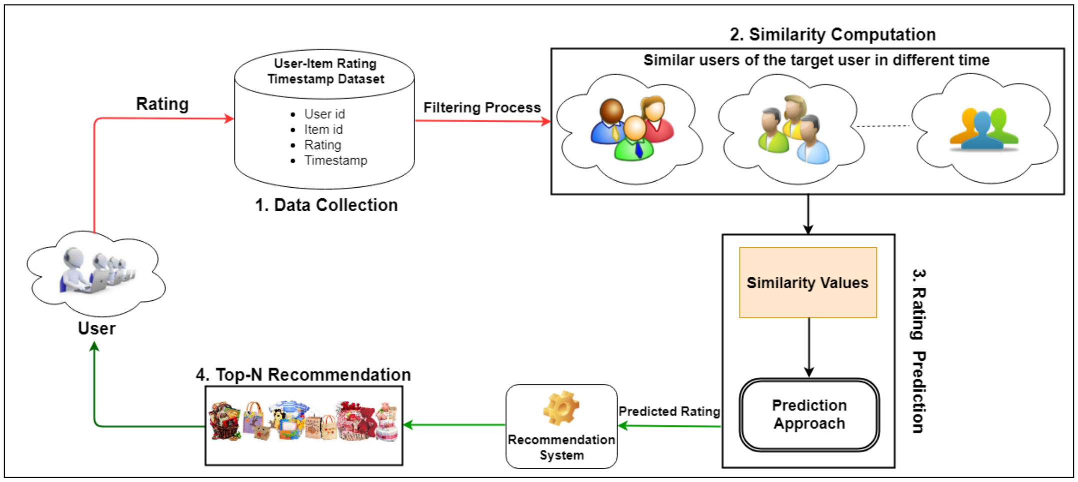

Figure 1: Framework of the proposed collaborative filtering approach.

2 Related Work

Several approaches have been proposed to improve the user-user and item-item similarity calculation. In this direction, Kant et al. have introduced a modified similarity measure that combines both similarities of users and items in rating prediction [5]. Lee et al. [8] have included a new attribute, i.e. temporal information, to improve the performance of CF. Their temporal information includes user purchase time and item launch time with the rating information to find the more personalized neighbors in different time interval. Clustering is a collection of items or users that have similar beaviour. The performance of the CF can be enhanced using the clustering concept. Therefore, Koohi et al. [7] have provided a new method in CF based on subspace clustering to find the best neighbors. Their approches have been tested on Movielens 100K, Movielens 1M and Jester datasets. Najafabadi et al. have utilized k-means clustering and association rule mining in their proposed approach to provide more personalized recommendations [6].

3 Proposed Recommendation Approach

The proposed neighborhood calculation approach in this paper is based on the hypothesis that the two users will be similar if their rating patterns are similar in the different time period. The proposed idea consists of four major steps, i.e., data collection, similarity computation, rating prediction, and top-n recommendation as shown in figure 1. The descriptions of these steps are given below.

3.1 Data Collection

The used dataset in this paper mainly considers user id for identifying the user, item id for identifying the item, a rating of the user on the item, and timestamp, i.e., the time when a user gives a rating. In this experiment, a matrix Y_{uc} (users' yearly contribution) is included that represents the total number of rating provided by a user in a particular year.

Table 2: **A matrix of users' yearly contribution**

Year User	Y_1	... ...	Y_j	... ...	Y_k
$User_1$	$Y_{uc}(1,1)$	... ...	$Y_{uc}(1,j)$	... ...	$Y_{uc}(1,k)$
...	...	... ...	...	... ...	...
$User_i$	$Y_{uc}(i,1)$	... ...	$Y_{uc}(i,j)$	... ...	$Y_{uc}(i,k)$
...	...	... ...	...	... ...	...
$User_m$	$Y_{uc}(n,1)$	... ...	$Y_{uc}(m,j)$	... ...	$Y_{uc}(n,k)$

Table 2 shows a matrix of size m×k, where m and k represent the number of users and the number of years respectively. If a user i has rated n items in j^{th} year then the value of $Y_{uc}(i,j)$ will be n.

3.2 Similarity Computation

Second steps of the proposed framework calculates the similarity value a target user in different time interval. For this, optimal k-means clustering algorithm is applied to find the top-n similar user of the target user in different time. The procedure of finding optimal number of cluster of similar users is shown in Algorithm 1.

Algorithm 1. : Finding optimal number of clusters of similar users

```
Input: $Y_{uc}$ dataset.
Output: Optimal number of cluster of similar users.
Procedure:
For cl = 1 to k, compute the k-means clustering algorithm on $Y_{uc}$, where k represents the total number of years
For each cl, calculate the SSE (sum of squared error) using $\sum_{cl=1}^{k} \sum_{u \in CL_{cl}} dist(u, CL_{cl})^2$. where CL represents a set of clusters CL=($CL_1$, $CL_2$,..., $CL_{cl}$...$CL_k$) and dist is a function that calculates the distance between user u and cluster centroid.
Plot the curve of SSE for each cluster cl=1 to k.
The location of a bend (knee) in the plot where cl and SSE value will be low, is considered as the optimal number of cluser $CL_o$.
```

3.3 Rating Prediction

The similarity values from the previous step are used in the prediction of rating of target item. In previous step, we find the optimal number of clusters of similar users. Hence, the equation for predicted rating becomes $\hat{r_{ui}} = \frac{\sum_{cl=1}^{k} CL(u,cl)}{\sum_{cl=1}^{k} |CL(u,cl)|}(\bar{r_u} + \frac{\sum_{v \in N_i(u)} sim(u,v)(r_{vi} - \bar{r_v})}{\sum_{v \in N_i(u)} |sim(u,v)|})$. Here, $\hat{r_{ui}}$ denotes the predicted rating of target user u on the item i and C(u,c) represents a binary matrix that shows the belonging nature of user u in cluster c. If user u belongs to the cluster c the value of C(u,c) will be 1 otherwise 0. $\bar{r_u}$ and $\bar{r_v}$ show the average rating of user u and v respectively. r_{vi} represents the rating of user v on item i, whereas sim(u,v) identifies the similarity between user u and v.

3.4 Top-N Recommendation

Based on the predicted rating of the items, a top-n list of items is generated in CF-based recommender system to recommend to the target user.

Algorithm 2 represents the complete steps in the proposed recommendation approach.

Algorithm 2. : Recommendation of top-n list to the target user

```
Input: User-Item rating dataset, A set of users (U), items (I) and Time (Y) when a user gives his rating.
Output: A list of top-n items to the target user u based on the predicted rating using proposed algorithm.
Procedure:
For $\forall i \in U$, $\forall j \in Y$, calculate $Y_{uc}(i,j)$
Apply the optimized k-means clustering algorithm (Elbow method) on the $Y_{uc}$ matrix and compute the optimal number of clusters CL=($CL_1$, $CL_2$,..., $CL_c$...$CL_k$).
For $\forall i \in U$, $\forall j \in I$, if $R_{ij}$ == 0 then,
$\hat{r_{ui}} = \frac{\sum_{cl=1}^{k} CL(u,cl)}{\sum_{cl=1}^{k} |CL(u,cl)|}(\bar{r_u} + \frac{\sum_{v \in N_i(u)} sim(u,v)(r_{vi} - \bar{r_v})}{\sum_{v \in N_i(u)} |sim(u,v)|})$ // $R_{ij}$ means rating of user u on item i.
Generate a list of top-n items based on the predicted rating.
```

4 Comparative Analysis

The MovieLens datasets ml-100k and ml-1m have been collected to compare the performance of traditional CF alorithms and the proposed CF approach [9, 11]. The ratings of these datasets are within the year 1997 to 2003 and belong in the range of 1 to 5 with 1 increment, 1 denotes the lowest rating whereas 5 represents the highest rating. The dataset ml-100k has 100000 rating information of 943 users and 1682 movies whereas ml-1m consists 1000209 ratings of 6040 users and 3952 items. These datasets have 93.695% and 95.809% sparsity respectively. We use the equation sparsity=$\frac{Total\ no.\ of\ missing\ ratings\ in\ the\ dataset*100}{Total\ no.\ of\ users*Total\ no\ of\ items}$ to calculate the sparsity of dataset. The datasets ml-100k and ml-1m are modified into different datasets based on their percentage of trained dataset and test dataset. The details of these modified datasets are shown in table 3.

Table 3: **Details of datasets used in the comparative analysis**

Collected Dataset	Modified Dataset	Trained Dataset (%)	Test Dataset (%)
ml-100k	Dataset 1	45	55
	Dataset 2	40	60
	Dataset 3	35	65
ml-1m	Dataset 4	45	55
	Dataset 5	40	60
	Dataset 6	35	65

For comparative analysis, we use Pearson Correlation as a similarity metric and Mean Centering prediction approach for rating prediction [2, 10]. The equations of traditional similarity metric and prediction approach are shown in table 4.

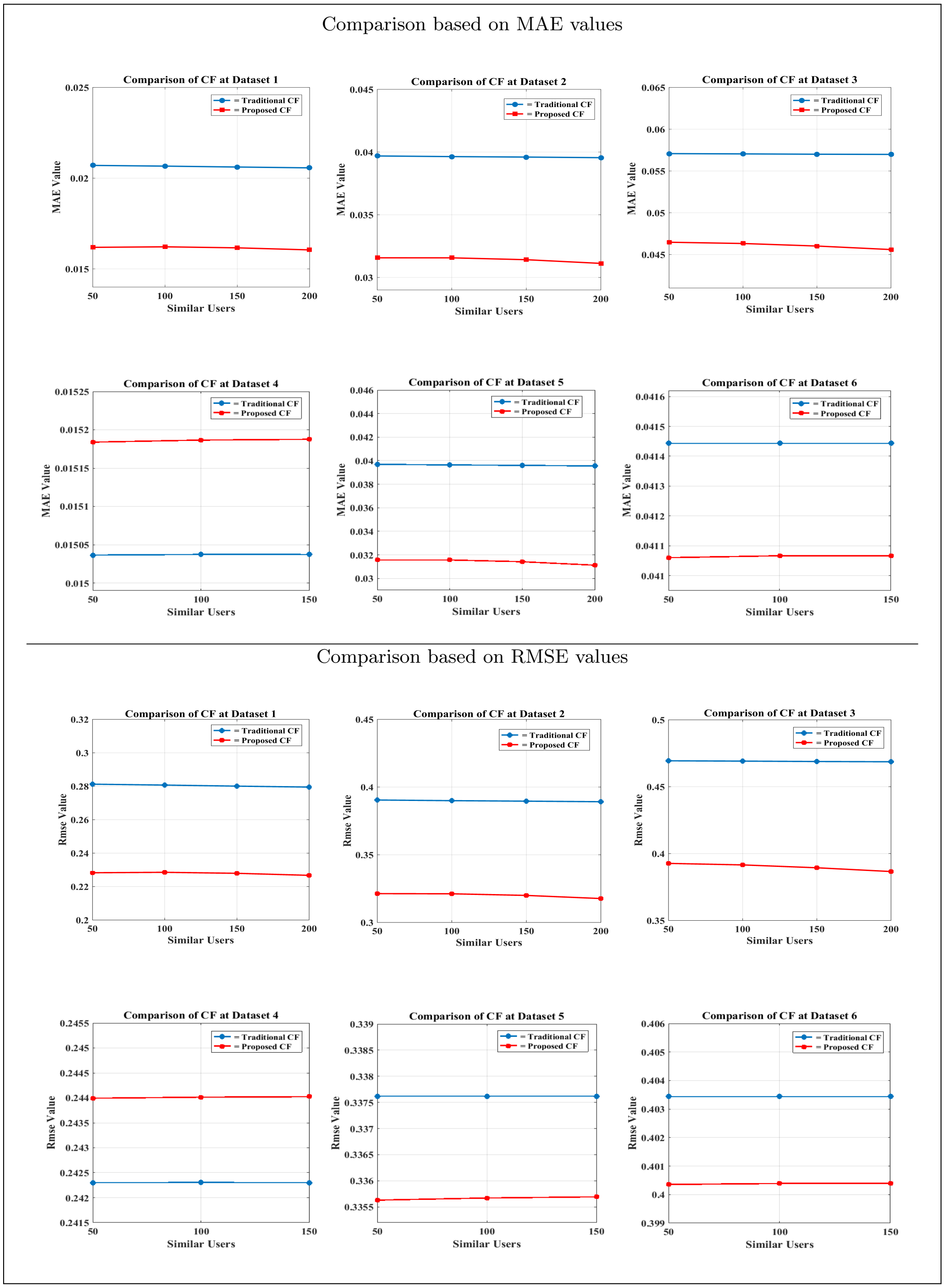

Figure 2: Comparison between traditional CF and proposed approach based on MAE and RMSE.

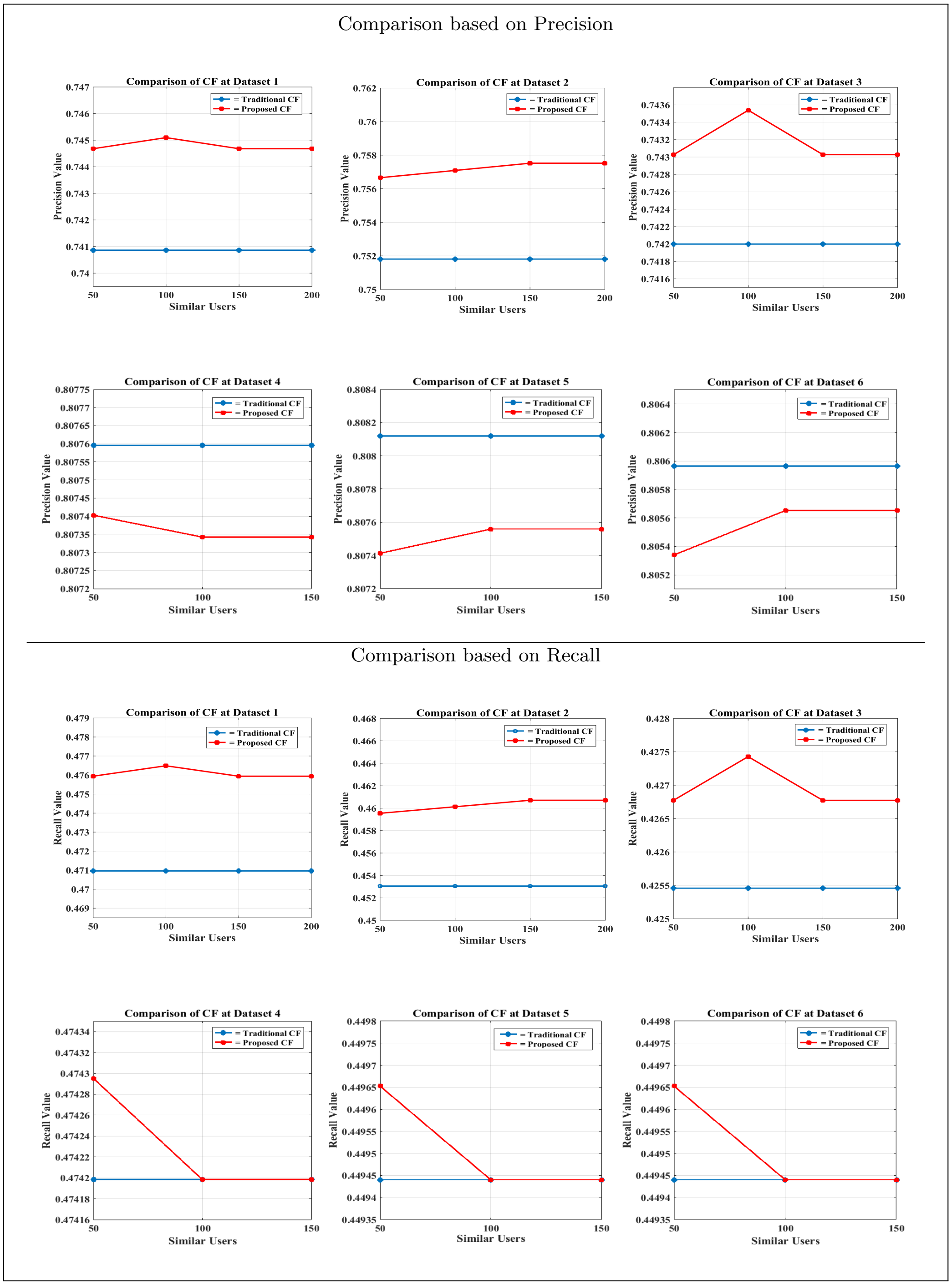

Figure 3: Comparison between traditional CF and proposed approach based on Precision and Recall.

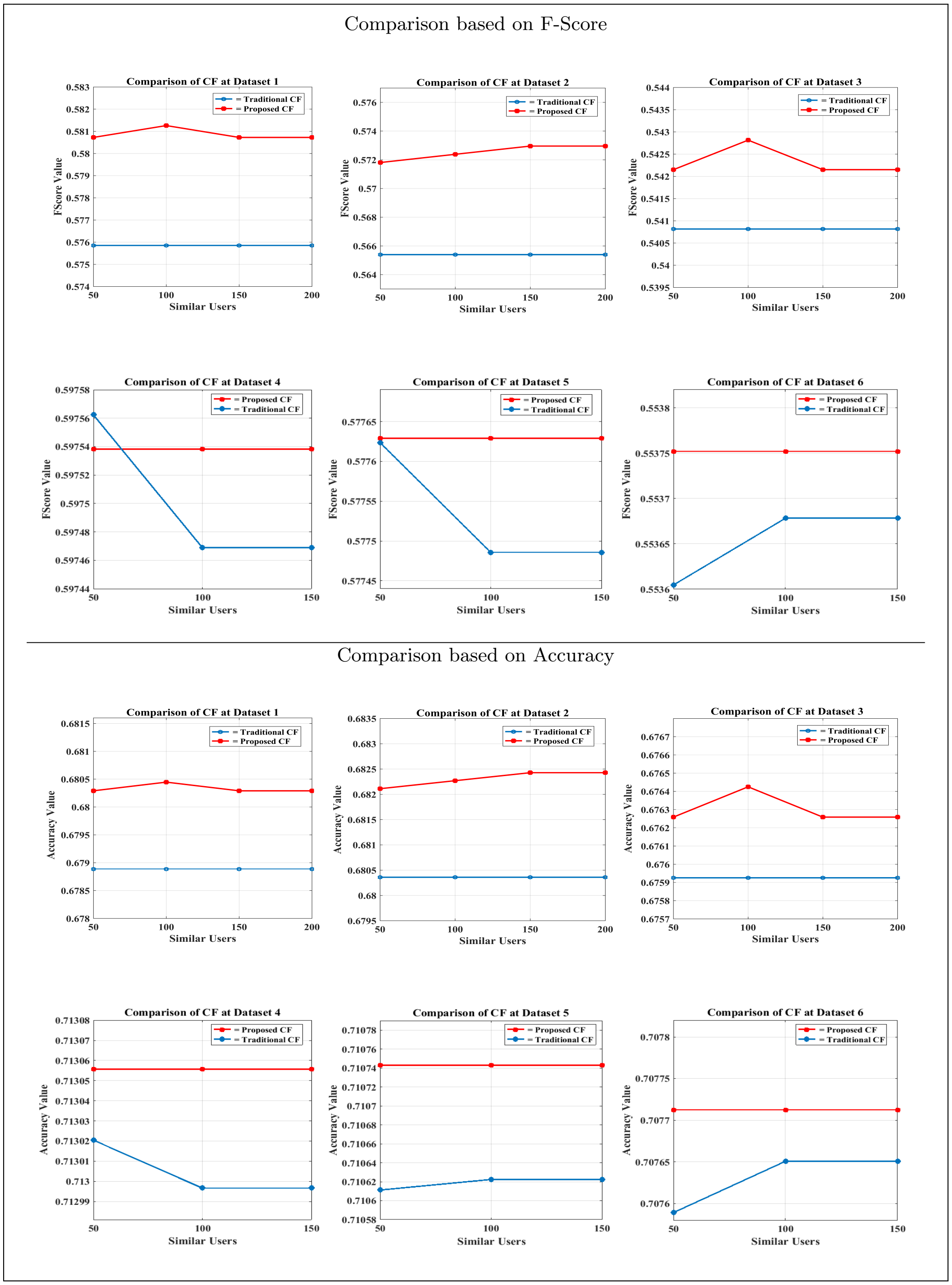

Figure 4: Comparison between traditional CF and proposed approach.

Table 4: **Used similarity metric and prediction approach**

Pearson Correlation	$sim(u,v) = \frac{\sum_{i\in I}(r_{i,u}-\bar{r_u})(r_{i,v}-\bar{r_v})}{\sqrt[2]{\sum_{i\in I}(r_{i,u}-\bar{r_u})^2}\sqrt[2]{\sum_{i\in I}(r_{i,v}-\bar{r_v})^2}}$
Mean Centering	$\hat{r_{ui}} = \bar{r_u} + \frac{\sum_{v\in N_i(u)} sim(u,v)(r_{vi}-\bar{r_v})}{\sum_{v\in N_i(u)} \lvert sim(u,v)\rvert}$

In table 4, $r_{i,u}$ and $r_{i,v}$ denote the rating of user u and v on item i. Six different metrics i.e. MAE, RMSE, Precision, Recall, F-Score, and Accuracy have been used for evaluation of the proposed approach [12]. The equations of computing MAE and RMSE values are as follows:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - \hat{q}_i|}{N} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - \hat{q}_i)^2}{N}} \tag{2}$$

Here, p_i and $\hat{q}_i$ show the predicted and actual rating of item i respectively. N represents the total number of predicted item. We consider the ratings above 3 as a high rating (recommended items), and less than 3 as a low rating (not recommended items). The classification of the possible results are shown in Table 5.

Table 5: **Classification of the possible results of a recommendation of an item to a user**

Type of Ratings	**Prediction**	
	Recommended (Predicted High Rating)	Not Recommended (Predicted Low Rating)
Actual High Rating	True-Positive (t_p)	False-Negative (f_n)
Actual Low Rating	False-Positive (f_p)	True-Negative (t_n)

Hence, using table 5, the equations of Precision, Recall, F-Score, and Accuracy become:

$$Precision = \frac{\#t_p}{\#t_p + \#f_p} \tag{3}$$

$$Recall = \frac{\#t_p}{\#t_p + \#f_n} \tag{4}$$

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

$$Accuracy = \frac{\#t_p + \#t_n}{\#t_p + \#t_n + \#f_p + \#f_n} \tag{6}$$

Here, # denotes the 'number of'.

Figure. 2 shows the MAE and RMSE values at different datasets. Based on MAE and RMSE values, the proposed CF algorithm provides less prediction error than the traditional CF algorithm. In figure 3, the proposed CF gives more precision value at different similar number of users. For less number of similar users it provides less recall values than the traditional CF whereas, for high number of similar users it provides approx same recall values. The proposed CF attains high f-score and accuracy than the traditional CF as shown in figure 4. Therefore, from figure 2 to 4, for different similar users the proposed CF outperforms the traditional CF algorithm.

5 Conclusion

Since the Collaborative Filtering based recommender systems recommend an item to a target user on the basis of the items preferred by the similar users or neighbors of the that user the effectiveness and accuracy of the recommendation depends on the correctness of the neighbors of the target user. Considering the similarity between users over a long period often leads to incorrect neighborhood calculation because the taste and preferences of an individual change along with time. This paper proposes a novel method to calculate the neighborhood of a user by considering the changed preferences of the users which were considered as neighbors in the past. The top-n neighbors of a user are calculated per year basis. The proposed method calculates the total number of ratings provided by a user at different time years and apply the optimized K-means clustering to find the optimal number of clusters of similar users for an optimized cluster of years. In other words, it calculates which users are more similar in which years, then it computes the set of most similar users for the longest period of times. The proposed solution is compared with the traditional CF algorithm on the basis of performance metrics such as MAE, RMSE, Precision, Recall, and Accuracy applied on the different highly sparse datasets, i.e. ml-100 k and ml-1m. The comparison results establish the advantage of our proposed method.

References

[1] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40:66–72, 1997.

[2] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[3] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning*, 51:785–799, 2010.

[4] Alípio M. Jorge, João Vinagre, Marcos Domingues, João Gama, Carlos Soares, Pawel Matuszyk, and Myra Spiliopoulou. *Scalable Online Top-N Recommender Systems*, pages 3–20. Springer International Publishing, 2017.

[5] Surya Kant and Tripti Mahara. Merging user and item based collaborative filtering to alleviate data sparsity. *International Journal of System Assurance Engineering and Management*, 9:173–179, 2018.

[6] Maryam Khanian Najafabadi, Mohd Mahrin, Suriayati Chuprat, and Haslina Sarkan. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, 67, 2017.

[7] Hamidreza Koohi and Kourosh Kiani. A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Systems with Applications*, 83:30–39, 2017.

[8] Tong Queue Lee, Young Park, and Yong-Tae Park. A time-based approach to effective recommender systems using implicit feedback. *Expert Syst. Appl.*, 34:3055–3062, 2008.

[9] Chenyang Li and Kejing He. Cbmr: An optimized mapreduce for item-based collaborative filtering recommendation algorithm with empirical analysis. *Concurrency and Computation: Practice and Experience*, 29, 2017.

[10] Bamshad Mobasher. Data mining for web personalization. In *The adaptive web*, pages 90–135. Springer, 2007.

[11] MovieLens. Grouplens. https://grouplens.org/datasets/movielens/, 2018. Online; Last accessed 16 August 2018.

[12] Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, and Prasenjit Choudhury. A comparative study of different similarity metrics in highly sparse rating dataset. In *Data Management, Analytics and Innovation*, pages 45–60. Springer, 2019.