10.31341/jios.45.2.10
Open Access
UDC 004.85:004.89
Original Scientific Paper

Hybrid Recommendation Using Temporal Data for Accuracy Improvement in Item Recommendation

Desabandhu Parasuraman *desabandhu.p@vit.ac.in*
School of Information Technology and Engineering
Vellore Institute of Technology, Vellore, India.

Sathiyamoorthy Elumalai *esathiyamoorthy@vit.ac.in*
School of Information Technology and Engineering
Vellore Institute of Technology, Vellore, India.

Abstract

Recommender systems have become a vital entity to the business world in form of software tools to make decisions. It estimates the overloaded information and provides the suitable decisions in any kind of business work through online. Especially in the area of e-commerce, recommender systems provide suggestions to users on the items that are likely based upon user's true interest. Collaborative Filtering and Content Based Filtering are the main techniques of recommender systems. Collaborative Filtering is considered to be the best in all domains and always outperforms Content Based filtering. But, both the techniques have some limitations like data sparsity, cold start, gray sheep and scalability issues. To overcome these limitations, Hybrid Recommender Systems are used by combining Collaborative Filtering and Content Based Filtering. This paper proposes such kind of hybrid system by combining Collaborative Filtering and Content Based Filtering using time variance and machine learning algorithm.

Keywords: Recommendation Systems (RS), Collaborative Filtering (CF), Content Based Filtering (CBF), Hybrid Recommender System (HRS, Machine Learning (ML).

1. Introduction

The aim of a Recommender System (RS) is to suggest a meaningful recommendations to maximum number of users on the items they are interested upon [1].RS is used maximum in E-Commerce sites which offers many number of products to the customers and customers face the challenging of choosing the right product [2]. Existing RS demands the evolution of hybrid techniques which collaborates the best available techniques in order to get the best result [3].

Generally, CF reports the better results than CBF since it directly depends upon the user ratings on the items [4]. Concept of CF is very simple like a user who had similar interest in past will also have similar interest in future [5].

In contrast, content-based filtering techniques rely on item descriptions and generate recommendations from items that are similar to those the target user has liked in the past, without directly relying on the preferences of other users[6]. CBF could be much more easily understood as users who were interested on particular items with certain attributes will have interest on same kind of items in future [5]. Machine learning algorithm is used for text categorizing in producing the accurate RS in content based book recommendation system[7].

Generally, collaborative systems report a better performance than content-based approaches, but their success relies on the presence of a sufficient number of user ratings [4]. Such systems have the drawback that they suffer from the item cold-start problems which occur when recommendations must be made on the basis of few recorded ratings [5]. These problems arise because the similarity analysis is not accurate enough [8]. In these situations the use of a content-based approach appears as an alternative. Nevertheless, this approach has its own limitations. For example, the keywords used to represent the content of the items might not be very representative. Also, content-based approaches suffer the limitation of making accurate recommendations to users with very few ratings [5]. These limitations can be overcome by combining both the CF and CBF which leads to HRS [6].

In this paper, an algorithm is proposed to overcome the limitations of CF and CBF to provide better recommendations. This is possible by proposing a new framework by combining both the CF and CBF. Here, time variance and a machine learning algorithm are used in the implementation of HRS.

2. Literature Survey

New Bayesian model is proposed to overcome the problem in making the perfect hybrid RS by combining the CBF and CF [4]. Content based collaborative filtering (CCF) is an approach proposed by combing both the similarities by a neighborhood model [9]. Three way aspect model is proposed to consider the rating data, content data and user preferences of web services [10]. Recommender system based on collaborative **F**iltering (RECF) algorithm is built to explore the similarity between the semantic relations and short item descriptions [11]. Most of the hybrid techniques combine with collaborative filtering with other techniques in a weighted way [5]. Additional stacked de-noising auto encoder (aSDEA) is proposed as a novel deep presentation learning approach to overcome the problems of cold start and data sparsity problems [8]. Influence of visual information in restaurant recommendation system is advocated using CBF and CF by modelling the heterogeneous items in same space [12]. Hybrid CF method for multiple interests and multiple content proved the CF applied simultaneously on user and items performs better than the CF on user and CF on items [13]. A Hybrid content based and item based technique for recommending TV programs is proposed by to eliminate the issues in CF by using matrix factorization technique [14]. Cinema screen recommender agent provides hybrid RS engine to predict the interest of viewers on actors, directors and film genres. CBF lets the data space to include new films in recommendations [15]. iExpand is a novel collaborative-filtering-based recommender system by expanding user interest in the way of personalized ranking [16]. iExpand introduces three layer scheme which performs more accuracy in ranking recommendations. Fuzzy preference tree-structured is a comprehensive tree matching method, which matches two tree structured data by considering all its information of the properties of the tree [17]. Two categories of hybrid recommendation methods are proposed for restaurant recommendation system. One is by taking visual features and user preferences and another is taking visual features as intermediate to update pocket matrix [12].

3. Proposed Methodology

User profiles and items are the basic infrastructure of the methodology to be proposed. Proposed methodology consists of three phases, i) collaborative filtering with time variant, ii) Content based filtering with winnow algorithm, iii) Combing content based filtering, collaborative filtering and time variance.

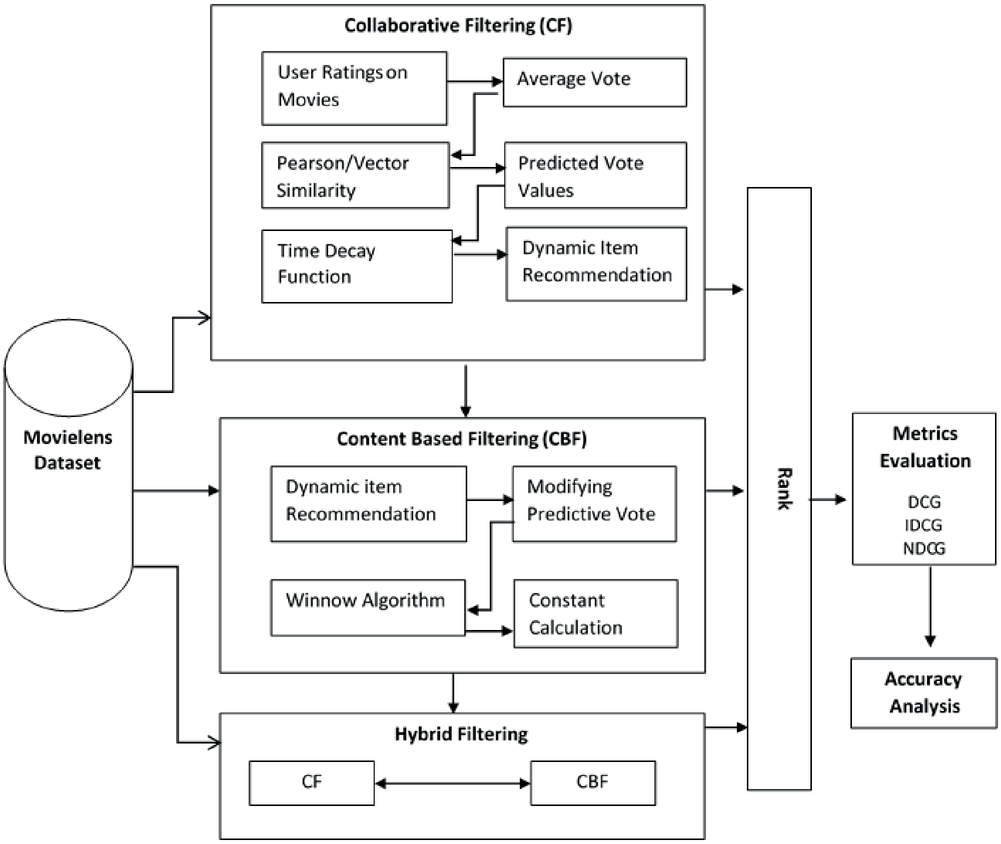

Figure 1. Proposed Approach

i) Collaborative Filtering with Time Variance

Normally, traditional item-based collaborative filtering is performed for item recommendations which never considers the dynamic interest of the user [3]. So, a dynamic item-based collaborative filtering is needed to outperform the traditional item-based collaborative filtering [6]. User profile and item profile are the basic information required to perform this methodology [3]. However, user's interest on different items will be available in user profile [18]. Finding the similarity between the users is the initial job.

RS will prioritize the users which has most similarity with active users. Most similar users will prefer most of the same items [1]. The question is how to find the priority of each user towards active user. Priority weightage is normally calculated using traditional Pearson correlation coefficient technique in collaborative filtering [19].

Though Pearson correlation coefficient is the traditional way to calculate weight of an entity technically, there are many other equations which can be used for the calculation of the similarity between users, one of them being Vector Similarity [20]. Vector similarity is used to measure how similar a content is suitable to the contents selected by a user [21]. Based on the research done, vector similarity outperforms the Pearson Correlation when the number of users is very large to predict the similarity [22].

In the dataset, set of items with their corresponding features information is available [23]. Here, Pearson correlation coefficient is applied to the dataset to find the priority weightage of each items on others and we get $P_{x,y}$. Users predicted vote on all items is derived using vector similarity. Standard method to calculate the weight W and average vote $\overline{\vartheta_i}$ is expanded using vector similarity $w_{a,j}$ and the predicated vote $P_{a,j}$ is derived finally. Once the users vote predicted by applying vector similarity, time delay is set for each and every day by fixing a decay rate λ. Average of the votes are calculated for traditional method without considering the time. So, the time decay λ (item) is incorporated on the average of the votes. By subtracting the normal vote average λ (item) from time decay vote average vote (item), the time variance outperform with the lesser values. So, the latest interest of users on items is recognized in collaborative filtering using time variance [6].

Step 1: The average vote $\overline{\vartheta_i}$ can be calculated for any user *i* is as

$$\overline{\vartheta_i} = \frac{1}{|C_i|} \sum_{j \in C_i} \vartheta_{i,j}$$

Where C_i is the set of items user *i* has voted for and $\vartheta_{i,j}$ is the vote of user *i* over the item j.

Step 2: Weight using vector similarity can be calculated as

$$w_{a,i} = \frac{\sum_{j=1}^{m} \vartheta_{(a,j)} \left(2^{-\lambda\left(t-t_{a,j}\right)}\right) \vartheta_{(i,j)} \left(2^{-\lambda\left(t-t_{i,j}\right)}\right)}{\sqrt{\sum_{j=1}^{m}} (\vartheta_{(a,j)} \left(2^{-\lambda\left(t-t_{a,j}\right)}\right)^{2} \sum_{j=1}^{m} (\vartheta_{(i,j)} \left(2^{-\lambda\left(t-t_{i,j}\right)}\right)^{2}}$$

Step 3: The normalizing factor N as

$$N = \frac{1}{\sum_{i} |sim(a,i)|}$$

Step 4: Augment the vote value with the time variant factor

$$\vartheta_{i,j} = \vartheta_{i,j} * 2^{-\lambda\left(t-t_{i,j}\right)}$$

Step 5: Time dependent factor will arrive when the average vote appended in equation (1)

$$\overline{\vartheta}_{i} = \frac{1}{|C_{i}|} \sum_{j \in C_{i}} (\vartheta_{i,j} * 2^{-\lambda\left(t-t_{i,j}\right)})$$

where,

t - Current time stamp

$t_{i,j}$ - Time stamp when the user *i* voted for the item *j*.

λ - Decay rate.

Higher the value of λ faster will be the degradation of the vote.

Step 6: The weight calculation by the Pearson correlation co-efficient will be modified as

$$w_{a,i} = \frac{\sum_{j} \left(\vartheta_{a,j} \left(2^{-\lambda\left(t-t_{a,j}\right)}\right) - \overline{\vartheta}_{a}\right) (\vartheta_{i,j} \left(2^{-\lambda\left(t-t_{i,j}\right)}\right) - \overline{\vartheta}_{i})}{\sqrt{\sum_{j}} (\vartheta_{a,j} \left(2^{-\lambda\left(t-t_{a,j}\right)}\right) - \overline{\vartheta}_{a})^{2} \sum_{j} (\vartheta_{i,j} \left(2^{-\lambda\left(t-t_{i,j}\right)}\right) - \overline{\vartheta}_{i})^{2}}$$

Step 7: Finally, the predicted vote will also be modified as

$$P_{a,j=}\overline{\vartheta}_{a} + N \sum_{i=1}^{n} w_{(a,i)} (\left\{\vartheta_{i,j} * 2^{-\lambda\left(t-t_{i,j}\right)}\right\} - \overline{\vartheta}_{i})$$

ii) Content based filtering with winnow algorithm

Based on the past research in content based fileting, the popular algorithms used are Rocchio's Algorithm and Bayesian Classifier [9]. Both the algorithms require pre-specification of number of items used in dataset. Hence, a better option is needed to calculate content based filtering. Here, latest machine learning algorithm called winnow algorithm is able to serve the situation since it is a perceptron algorithm [24] and [1]. It converges quickly as it uses multiplicative scheme instead of additive scheme in weight update. Content based filtering vote is appended with collaborative filtering by multiplying with a constant μ.

$$\sum_{i=1}^{n} w_i x_i > \theta$$

where θ is the threshold

i) If the sample is correctly classified and do nothing.

ii) If the prediction is 1 but result was 0 perform a demotion step (like divide by α)

ii) If the prediction is 0 but the result was 1 perform a promotion step (like multiply by α)

$$\sum_{i=1}^{n} w_i x_i = \mu$$

iii) Hybrid Recommender System – combining content based filtering, collaborative filter, and Time Variance

Result of dynamic item-based collaborative filtering with time variance is considered as the initial step at this point. Winnow algorithm is applied on user ratings i.e. votes of users on items derived in collaborative filtering. Based on the observation, the input for winnow algorithm should be either 0s or 1s [25]. So the user votes ϑ are converted to 0s and 1s by elevating 1 to 3.9 as 0s and 4 to 5 as 1s.

The result of winnow CBF $\mu_{a,j}$ is calculated for content based filtering using online conversion tools. Later, the average of winnow is termed as a constant μ.Finally, by multiplying the constant with the winnow result $\mu_{a,j}$ and summing it with collaborative filtering gives the end result. The results of collaborative filtering, content based filtering and combining content based and collaborative filtering are ranked.

$$P_{a,j} = \overline{\vartheta}_a + N \sum_{i=1}^{n} w_{(a,i)}(\left\{\left(v_{i,j} * 2^{-\lambda(t-t_{i,j})}\right) - \overline{\vartheta}_i\right\} + \mu * w_{a,j})$$

where, μ = constant.

3.1. Algorithm

Input: $ItemSetVector\ IV = < i_i, i_2, \dots i_n >, UserVector = \{A, B, \dots P\}$

Output: $ItemRecommendationRanks\ \{R_1, R_2, \dots R_n\}$

```
01. For each element j in IV
    a. For each active user a in U
       i. Augment the rating of an active user "a" over a particular item j as
          $\vartheta_{i,j} = \vartheta_{i,j} * 2^{-\lambda(t-t_{i,j})}$          (3)
          ii.  Let the time Augmented Predicted rating of an active user “a” over a
               particular item j be
               $P_{a,j=}\overline{\vartheta}_a + N\sum_{i=1}^{n} w_{(a,i)}(\{\vartheta_{i,j}\} - \overline{\vartheta}_i)$   (1)
               where (i varies from 1 to n users, j varies from 1 to m items)
               $\overline{\vartheta}_i = \frac{1}{|C_i|}\sum_{j\in C_i}(\vartheta_{i,j} * 2^{-\lambda(t-t_{i,j})})$   (2)
               store it in array v []
          iii. Calculate the similarity weight of an active user over a particular item
               j using Pearson Similarity method/Vector Similarity as
               $w_{a,i} = \frac{\sum_{j=1}^{m}\vartheta_{(a,j)}\left(2^{-\lambda\left(t-t_{a,j}\right)}\right)\vartheta_{(i,j)}\left(2^{-\lambda\left(t-t_{i,j}\right)}\right)}{\sqrt{\sum_{j=1}^{m}(\vartheta_{(a,j)}\left(2^{-\lambda\left(t-t_{a,j}\right)}\right)^2 \sum_{j=1}^{m}(\vartheta_{(i,j)}\left(2^{-\lambda\left(t-t_{i,j}\right)}\right)^2}}$   (4)
                                        (OR)
               $w_{a,i} = \frac{\sum_j\left(\vartheta_{a,j}\left(2^{-\lambda\left(t-t_{a,j}\right)}\right)-\overline{\vartheta}_a\right)(\vartheta_{i,j}\left(2^{-\lambda\left(t-t_{i,j}\right)}\right)-\overline{\vartheta}_i)}{\sqrt{\sum_j(\vartheta_{a,j}\left(2^{-\lambda\left(t-t_{a,j}\right)}\right)-\overline{\vartheta}_a)^2\sum_j(\vartheta_{i,j}\left(2^{-\lambda\left(t-t_{i,j}\right)}\right)-\overline{\vartheta}_i)^2}}$   (5)
               store it in array w []
          iv.  Calculate the normalizing factor N as
               $N = \frac{1}{\sum_i|sim(a,i)|}$   (6)
     b.   End For
02.  End For
03.  Call WinnowDriver(w[].v[], n,m,w_output[],v_output[])
04.  For each element j in IV
     a.   For each active user a in U
          i.   Recalculate predicted vote as P(a,j) = P(a,j) + (w_output[j]*$\mu$)
     $P_{a,j} = \overline{\vartheta}_a + N\sum_{i=1}^{n} w_{(a,i)}(\left\{\left(v_{i,j} * 2^{-\lambda(t-t_{i,j})}\right) - \overline{\vartheta}_i\right\} + \mu * w_{a,j})$   (7)
                    $\sum_{i=1}^{n} w_i x_i > \theta$   (8)
                    $\sum_{i=1}^{n} w_i x_i = \mu$   (9)
               where $\mu$ is the average of w_output[j],
     b.   End for
05.  End For
```

4. Experiments and results

Movielens dataset ml-20m is used to implement the proposed approach [23]. Dataset consists around 20 million records of ratings, 465564 tag applications across 27278 movies and holds the data of 138493 users, links, tags, relevance, genres and genome scores. Dataset was created by users from January 09, 1955 and March 31, 2015. Timestamp is recorded at the time of user rating an item. That timestamp is one of the promoting element of our approach. Time stamp given in the dataset is in form of seconds whereas, we have converted into years for time decay since the dataset generated across 60 years. Since it is a 20 millions of records, we loaded the entire data in Oracle database in order to fetch the relevant information by writing SQL and PLSQL queries. The main requirement of our approach is sufficient data of the users, movies and ratings. We fixed a condition that, maximum number of items with the ratings of maximum and unique users. In this case, PLSQL queries are called to find out the required data and it's found that, 16 unique users have given rating for as many of 106 items.

Average of all users ratings are considered to calculate traditional CF by finding the predicted votes of Pearson correlation similarity and vector similarity. Both the similarities are calculated using python code and the values are stored in database. By analyzing the similarities, it is found that the Pearson similarity outperforms the vector similarity. So, we consider only the values of person correlation in the remaining part of the implementation of our approach. Now, the results of traditional CF is obtained and we consider only Pearson results and neglect vector similarity results from this step. Same method is followed for the proposed approach by implementing time decay. The current timestamp and rating time stamp is converted into years and the values obtained is again computed using Pearson similarity and stored in the database. Now, the predicated weights are derived by the average weight obtained from traditional CF and dynamic item based collaborative filtering by using time variance by using the equation (2). The end result of predicted weights by the traditional CF and proposed CF are converted to 0s and 1s as the inputs for winnow algorithm to obtain the results of CBF in equations (8, 9). Winnow algorithm is implemented by using c# code and the predicted weights for CBF is derived. Combining CBF with CF to obtain HF is attained as per the equation (7).HF is also computed for traditional HF and HF with time decay. The results of the proposed approach i.e. ranking of all items are shown in table 1 and figure 2.

Movie Id	Rating Rank	CF Rank	CBF Rank	HF Rank	Movie Id	Rating Rank	CF Rank	CBF Rank	HF Rank
1	31	32	1	14	**1197**	15	17	1	9
10	80	76	3	59	**1198**	2	7	1	6
19	106	106	4	106	**1200**	20	15	3	25
32	51	51	1	18	**1208**	34	54	4	72

34	59	60	3	51	**1210**	24	52	3	46
39	99	98	4	98	**1214**	6	8	3	21
47	20	19	1	10	**1221**	18	31	3	34
50	3	14	1	8	**1222**	32	37	1	15
110	12	39	1	16	**1240**	43	24	4	66
150	69	55	3	47	**1258**	47	59	3	50
153	103	103	4	103	**1265**	59	63	3	53
165	80	75	4	80	**1270**	27	6	1	5
208	105	104	4	104	**1291**	32	40	1	17
231	103	102	4	102	**1387**	20	16	3	26
253	65	58	3	49	**1517**	77	64	4	75
260	10	5	1	4	**1527**	47	56	3	48
296	1	2	1	1	**1580**	51	25	3	30
316	95	95	4	95	**1682**	69	50	3	45
318	3	11	3	24	**1704**	43	65	3	54
344	87	89	3	62	**1721**	77	79	4	82
356	43	33	3	35	**1732**	12	4	1	3
364	75	81	4	84	**1923**	87	80	4	83
367	94	93	4	93	**1961**	34	57	4	73
377	86	73	3	57	**1968**	47	23	4	65
380	90	88	4	90	**2011**	56	41	3	39
457	77	90	3	63	**2028**	43	72	3	56
480	51	35	3	36	**2115**	40	42	3	40
500	96	94	4	94	**2174**	59	25	3	30
527	20	43	3	41	**2571**	12	12	4	64
541	27	18	3	27	**2617**	87	86	4	89
586	101	99	4	99	**2628**	102	105	4	105
588	34	46	3	43	**2683**	84	71	4	79
589	40	20	1	11	**2716**	27	29	3	33
590	73	87	3	61	**2762**	51	78	4	81
592	69	53	4	71	**2858**	69	83	4	86
593	15	38	4	67	**2916**	76	49	4	70
595	59	70	4	78	**2918**	51	60	3	51
597	100	101	4	101	**2959**	6	1	3	20
608	27	27	1	12	**2987**	58	34	3	36
648	96	97	4	97	**3578**	34	48	3	44

733	80	82	4	85	**3793**	65	77	3	60
736	98	100	4	100	**3996**	59	47	4	69
780	90	84	4	87	**4022**	90	96	4	96
858	5	13	1	7	**4027**	47	66	3	55
924	84	91	4	91	**4226**	34	28	3	32
1036	24	10	3	23	**4306**	56	45	4	68
1073	80	85	4	88	**4886**	65	74	3	58
1089	24	22	3	29	**4993**	9	30	1	13
1097	34	36	3	38	**4995**	59	67	4	76
1101	90	92	4	92	**5349**	73	62	4	74
1136	18	9	3	22	**5445**	65	69	4	77
1193	10	21	3	28	**5952**	15	44	3	42
1196	6	3	1	2	**7153**	40	68	1	19

Table 1. Ranking of movies - Result of proposed approach

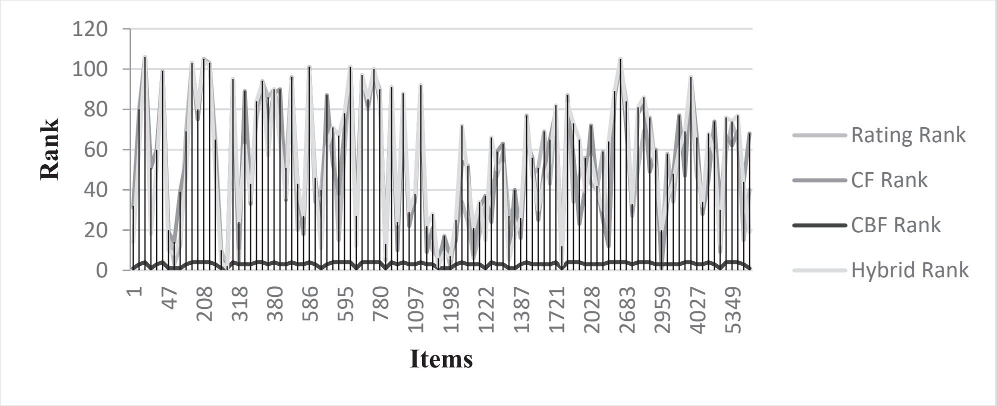

Figure 2. Ranking of movies - Result of proposed approach.

5. Discussion

Main objective of the proposed approach is to rank the items in each filtering techniques. Compare the results of traditional CF, CBF and HF with CF, CBF and HF with time decay and to prove the hybrid approach can recommend the items to the user more efficiently. Table 1 and figure 2 illustrate the ranking of movies with traditional method and proposed approach. Here, for Movie Id 1, the rating rank i.e. the rank of traditional method is 31. Whereas in proposed approach, CF rank is 32, CBF rank is 1 and HF rank is 14. This result explains that the Movie Id 1 has obtained 31st rank in traditional method and 14th rank in proposed approach for the purpose of recommending to the users. Therefore, the Movie Id 1 is elevated due to time decay implementation and highlighted to the user with priority while recommending. Moreover, in order to prove the accuracy in ranking items, we used the ranking metrics like DCG (Discounted cumulative gain), IDCG (Ideal discounted cumulative gain) and NDCG (Normalized discounted cumulative gain) which are the most efficient metrics used in the area of item ranking. DCG is a weighed sum of the degree of relevancy of the ranked items [26]. Weight is a decreasing function of a position or rank of the item and it is called as discount [26]. IDCG represents the list of relevant items (|REL|) in the corpus up to position p [26]. DCG and IDCG rewards good rank that return good answers or the items which come up with maximum gain comparing with other positions [26]. the ranks which NDCG normalizes DCG by IDCG and NDCG always measure in between 0 to 1 and the values which is near to 1 is termed as good in quality of rank [26].

$$DCG_P = \sum_{i=1}^{P} \frac{rel_i}{log_2\,(i+1)}$$

$$IDCG_P = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{log_2\,(i+1)}$$

$$NDCG_P = \frac{DCG_P}{IDCG_P}$$

where,

P - Position of rank

rel_i – Relevancy of ranked items

|REL| - list of relevant items

Therefore, the results are tabulated in order with the ranking metrics and results ensure that the proposed system is accurately recommending the items by ranking the items in all three recommendation techniques. The results of the ranking metrics are showed in table 2, 3, 4 and figures 3, 4, 5 respectively. Here, the accuracy of ranking metrics shown in table 2, 3, 4 is listed with top N movies i.e. Top 10, 20, 50, 70 and 100. The table 2 contains the result of DCG metric of proposed and traditional approaches comparing the techniques. For example in top 100, the DCG-CF, DCG-CBF and DCG-HF are 80.0, 43.15 and 173.21 in proposed approach and the DCG-CF, DCG-CBF and DCG-HF are 78.96, 42.01 and 170.21 in traditional method. The table 3 contains the result of IDCG metric of proposed and traditional approaches comparing the techniques. For example in top 100, the IDCG-CF, IDCG-CBF and IDCG-HF are 84.54, 49.68 and 191.55 in proposed approach and the IDCG-CF, IDCG-CBF and IDCG-HF are 83.98, 47.25 and 189.22 in traditional method. The table 4 contains the result of NDCG metric of proposed and traditional approaches comparing the techniques. For example in top 100, the NDCG-CF, NDCG-CBF and NDCG-HF are 0.94, 0.86 and 0.90 in proposed approach and the NDCG-CF, NDCG-

CBF and NDCG-HF are 0.94, 0.83 and 0.88 in traditional method. Since the NDCG measure from 0 to 1 [26], the result which is near to 1 is observed as the highest cumulative gain. The results of all the metrics used infers that in top N movies (with top 10 to 100 movies), the ranking accuracy in terms of cumulative gain of proposed approach is high comparing to the traditional method.

Based on the results of ranking metrics used (DCG, IDCG and NDCG) in the proposed approach, it is proved that the CF, CBF and HF with time decay outperforms traditional recommending approaches. Moreover, the ranking metrics of proposed and traditional approaches results ensure that the cumulative gain of the ranks is higher in HF comparing to CF and CBF. Also, the cumulative gain of CF is higher than CBF. Hence, the proposed approach also proves the universal myth that the HF outperforms CF, CBF and CF outperforms the CBF.

	Proposed Approach			Traditional Method		
TOP N Movies	DCG - CF	DCG - CBF	DCG - HF	DCG - CF	DCG - CBF	DCG - HF
TOP 10	17.06	12.99	45.13	17.11	12.12	43.54
TOP 20	26.11	17.65	64.23	25.78	16.23	63.21
TOP 50	47.89	27.40	107.07	47.17	25.01	105.21
TOP 70	61.89	35.67	138.95	61.12	33.26	135.21
TOP 100	80.0	43.15	173.21	78.96	42.01	170.21

Table 2. Results of DGC ranking metric

	Proposed Approach			Traditional Method		
TOP N Movies	IDCG - CF	IDCG - CBF	IDCG - HF	IDCG - CF	IDCG - CBF	IDCG - HF
TOP 10	20.59	18.17	59.58	20.53	18.03	58.24
TOP 20	31.38	27.70	90.50	31.47	26.32	87.25478
TOP 50	55.45	39.42	140.01	55.40	37.90	137.21
TOP 70	68.33	45.01	165.00	68.17	44.015	160.05
TOP 100	84.54	49.68	191.55	83.98	47.25	189.22

Table 3. Results of IDGC ranking metric

	Proposed Approach			Traditional Method		
TOP N Movies	NDCG - CF	NDCG - CBF	NDCG - HF	NDCG - CF	NDCG - CBF	NDCG - HF
TOP 10	0.82	0.71	0.75	0.79	0.69	0.71
TOP 20	0.83	0.63	0.70	0.81	0.60	0.68
TOP 50	0.86	0.69	0.76	0.85	0.66	0.73

TOP 70	0.90	0.79	0.84	0.89	0.75	0.82
TOP 100	0.94	0.86	0.90	0.94	0.83	0.88

Table 4. Results of NDGC ranking metric

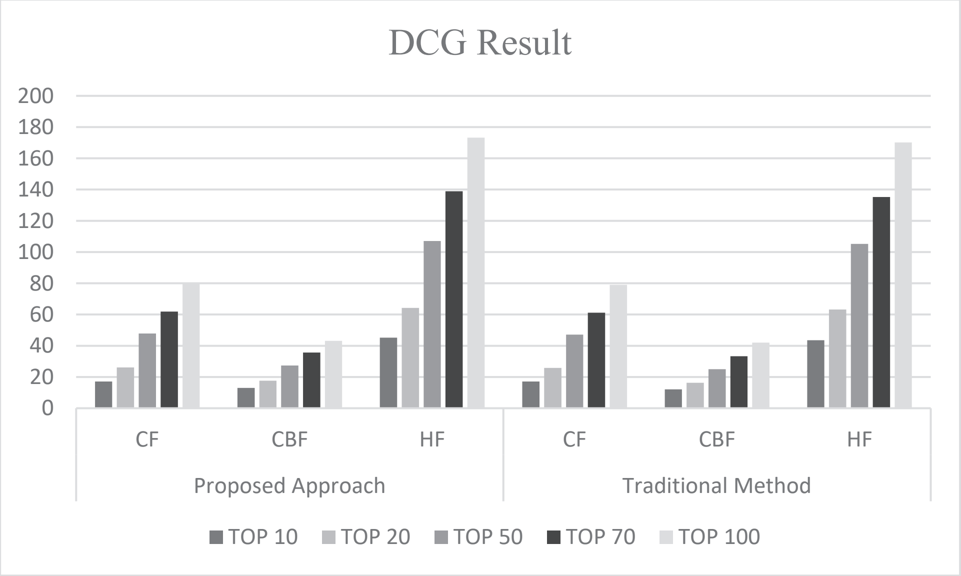

Figure 3. Results of DGC ranking metric

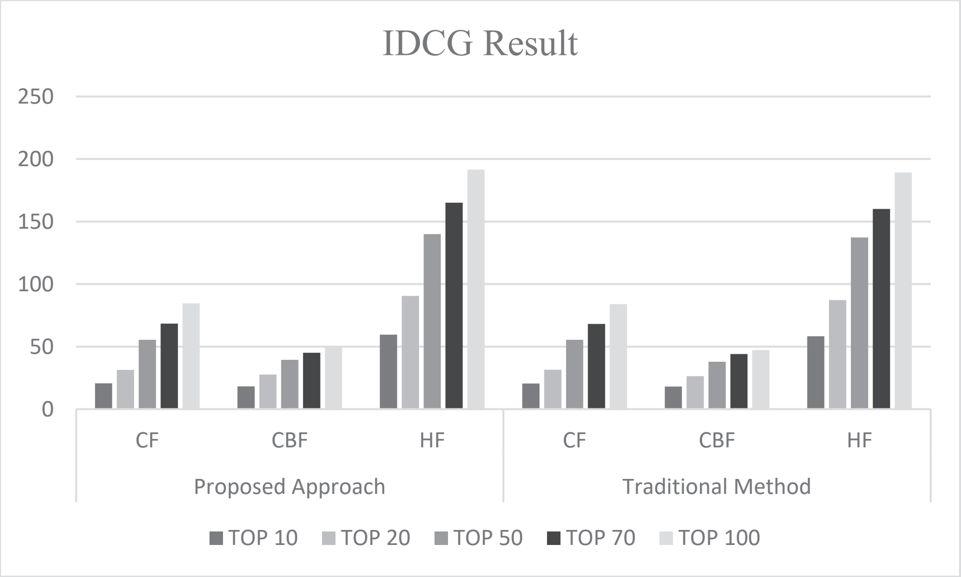

Figure 4. Results of IDGC ranking metric

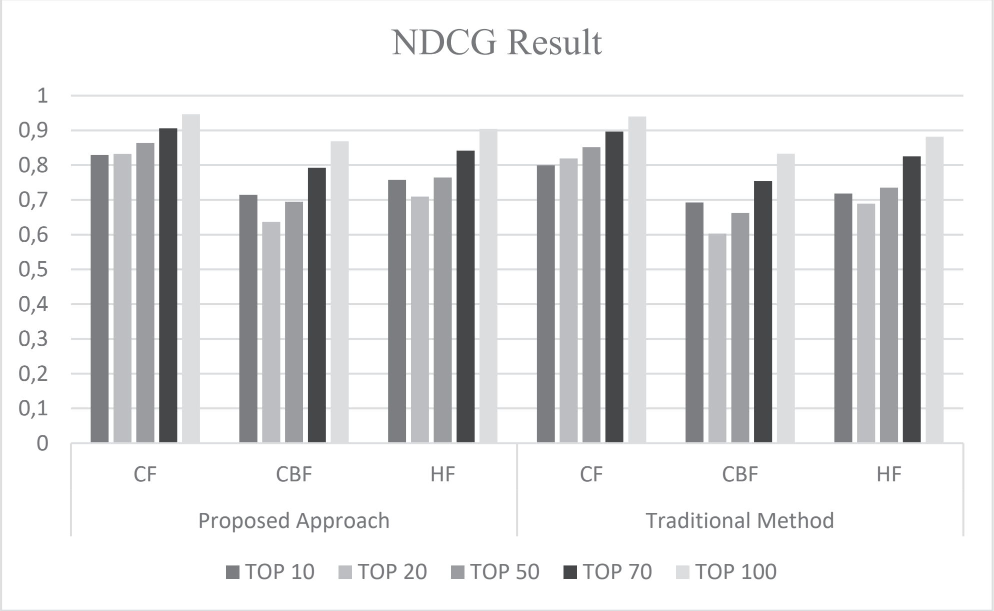

Figure 5. Results of NDGC ranking metric

The results infer that the most of the values satisfies the theory of metrics used. Proposed approach outperforms the traditional approach and HF outperforms CF, CBF and CF outperforms the CBF.

6. Conclusion

Initially, time variant is implemented in traditional collaborative filtering to obtain dynamic item based collaborative filtering. Secondly, machine learning algorithm is implemented in content based filtering. Later, the content based filtering is combined with collaborative filtering to obtain hybrid filtering. When combining the content based and collaborative filtering, the ranks clearly infers that the items rank derived in hybrid filtering by user's ratings computed with person similarity and time variance is either equivalent or higher than the ranks of content based filtering and collaborative filtering. Hence, the proposed algorithm improves the accuracy of item recommendation by combining the collaborative filtering with time variance and content based filtering with winnow algorithm. The accuracy is proved by evaluating the results with the DCG, IDCG and NDCG ranking metrics.

References

[1] P. Melville and V. Sindhwani, "Recommender systems," in *Encyclopedia of machine learning*, Springer, 2011, pp. 829–838.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of recommendation algorithms for e-commerce,” in *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000, pp. 158–167.

[3] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.

[4] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks,” *Int. J. Approx. Reason.*, vol. 51, no. 7, pp. 785–799, 2010.

[5] E. Çano and M. Morisio, “Hybrid recommender systems: A systematic literature review,” *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, 2017.

[6] D. Parasuraman and S. Elumalai, “Improving the accuracy of item recommendations in collaborative filtering using time-variant system,” *Electron. Gov. an Int. J.*, vol. 13, no. 4, pp. 324–338, 2017.

[7] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” in *Proceedings of the fifth ACM conference on Digital libraries*, 2000, pp. 195–204.

[8] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, “A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems.,” in *AAAI*, 2017, pp. 1309–1315.

[9] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang, “Content-Based Collaborative Filtering for News Topic Recommendation.,” in *Association for the Advancement of Artificial Intelligence*, 2015, pp. 217–223.

[10] L. Yao, Q. Z. Sheng, A. Segev, and J. Yu, “Recommending web services via combining collaborative filtering with content-based features,” in *Web Services (ICWS), 2013 IEEE 20th International Conference on*, 2013, pp. 42–49.

[11] J. He, H. H. Zhuo, and J. Law, “Distributed-Representation Based Hybrid Recommender System with Short Item Descriptions,” *arXiv Prepr. arXiv1703.04854*, 2017.

[12] W.-T. Chu and Y.-L. Tsai, “A hybrid recommendation system considering visual information for predicting favorite restaurants,” *World Wide Web*, vol. 20, no. 6, pp. 1313–1331, 2017.

[13] Y. Li, L. Lu, and L. Xuefeng, “A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce,” *Expert Syst. Appl.*, vol. 28, no. 1, pp. 67–77, 2005.

[14] A. B. Barragáns-Martinez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Inf. Sci. (Ny).*, vol. 180, no. 22, pp. 4290–4311, 2010.

[15] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: combining collaborative and content-based filtering," *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 35–41, 2006.

[16] Q. Liu, E. Chen, H. Xiong, C. H. Q. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 1, pp. 218–233, 2012.

[17] D. Wu, G. Zhang, and J. Lu, "A Fuzzy Preference Tree-Based Recommender System for Personalized Business-to-Business E-Services," *IEEE Trans. Fuzzy Syst.*, vol. PP, no. 99, pp. 1–1, 2014.

[18] J. A. Iglesias, P. Angelov, A. Ledezma, and A. Sanchis, "Creating evolving user behavior profiles automatically," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 854–867, 2012.

[19] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Leveraging prior ratings for recommender systems in e-commerce," *Electron. Commer. Res. Appl.*, vol. 13, no. 6, pp. 440–455, 2014.

[20] F. S. Gohari and M. J. Tarokh, "Classification and Comparison of the Hybrid Collaborative Filtering Systems," *Int. J. Res. Ind. Eng.*, vol. 6, no. 2, pp. 129–148, 2017.

[21] F. Liu, H. Nguyen-Le, and C. C. Ko, "Vector similarity-based detection scheme for multi-antenna FH/MFSK systems in the presence of follower jamming," *IET Signal Process.*, vol. 2, no. 4, pp. 346–353, 2008.

[22] X. Yang, Y. Guo, and Y. Liu, "Bayesian-inference-based recommendation in online social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 642–651, 2013.

[23] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, 2016.

[24] V. Dani, O. Madani, D. M. Pennock, S. Sanghai, and B. Galebach, "An empirical comparison of algorithms for aggregating expert predictions," *arXiv Prepr. arXiv1206.6814*, 2012.

[25] N. Aston, T. Munson, J. Liddle, G. Hartshaw, D. Livingston, and W. Hu, "Sentiment analysis on the social networks using stream algorithms," *J. Data Anal. Inf. Process.*, vol. 2014, 2014.

[26] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T.-Y. Liu, “A theoretical analysis of NDCG ranking measures,” in *Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013)*, 2013, vol. 8.