

An Advance Recommendation System Formulated on End-User Interest and Rating Difference

Priyanshi Singh

Department of Computer Engineering,
Smt. Kashibai Navale College of Engineering,
Savitribai Phule Pune University,
Pune, India.

Prof. Dr. V. V. Kimbahune

Department of Computer Engineering,
Smt. Kashibai Navale College of Engineering,
Savitribai Phule Pune University,
Pune, India.

Abstract— In recent years, recommendation systems have grown in popularity and are now used in a variety of fields, including books, news, movies, research papers, search requests, social tags and business items. When it comes to creating intelligent recommendation systems that can be trained to provide better recommendations, collaborative filtering is most widely used algorithm. Currently, collaborative filtering has been profitably utilized in personalized recommendation systems. It is used by most websites, including Amazon Prime, YouTube, Netflix and Hotstar, as part of their advanced recommendation section. This method is used to provide users with recommendations based on the preferences and dislikes of other users. However, under the stipulation of extremely sparse rating data, the traditional scheme of similarity among users is relatively straightforward. Moreover, it does not consider the end-user's interest which results in pitiable performance. Due to invariability in score differences, the accuracy of the recommendation systems is hampered. So, here a system is proposed to improve the performance of the traditional collaborative filtering algorithm by considering both end-user interest and score difference.

Keywords—Collaborative filtering; End-user's interest; Score differences; Rating Prediction; Recommendations; Score forecast.

I. INTRODUCTION

The advancement of the internet and e-commerce has made our lives easier by allowing us to browse for billions of needed items digitally. Such highly regarded Internet sites as Amazon.com, YouTube, Netflix, Yahoo, TripAdvisor, Last.fm, and IMDb use recommender systems extensively. Furthermore, several media organizations are now designing and launching RSs as part of their subscriber services. For example, Netflix, the online movie rental service, awarded a million-dollar reward to the team that first improved the efficiency of its recommender system by a significant margin. As e-commerce Web sites grew in popularity, a pressing need arose for providing recommendations based on the filtering of all available options. Users were having a difficult time deciding which items (products and services) to purchase from the vast array of options available on these websites. It frequently causes users to become distracted, causing them to make poor decisions. Instead of providing a profit, the proliferation of options began to reduce users' happiness. It was understood that, having a variety of options is beneficial, having more options is not always better. Many scholars have conducted extensive studies on the recommendation methods, making significant progress in this field. However, data scarcity has

always been a major factor in the recommendation's poor accuracy. Researchers have suggested a growing number of excellent algorithms, such as neighborhood-based CF (Collaborative Filtering) and model-based CF, to make full use of existing data.

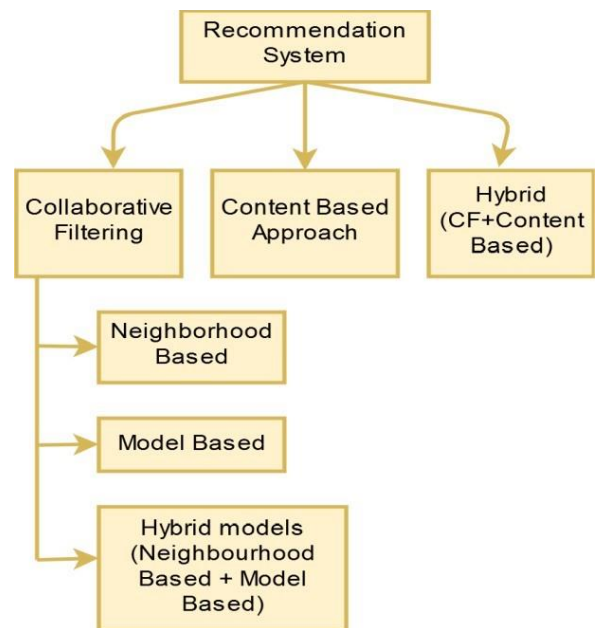


Fig. 1. Types of Recommendation System

User-based CF and item-based CF algorithms are the two types of neighborhood-based CF algorithms and their fundamental principles are intertwined. For instance, when two users' neighbors are identical, user-based CF assumes them to be similar. Obviously, the choice of the nearest neighbor is essential. On the other hand, choosing the right similarity would aid in improving the recommended consistency and applicability to the recommended algorithm. The researchers consider the score data that most accurately represents the user's interest. They devised tests of resemblance including cosine similarity, Pearson correlation coefficient, and modified cosine. Others also proposed similarity measures such as Salton similarity and Jaccard similarity, which take into account the number of objects among one's own and neighbors' possessions. are provided.

The main idea behind user-based collaborative filtering algorithms is to find out the users with similar ratings in order to target users according to historical data and its rating data, and to take them as neighbors, then organize them into assorted directory according to their favorite objects. In reality, the user rating of the items is not only correlated to user similarity, but also user interest, because the user interest also occupies a certain proportion. Hence, one of the crucial aspects that have to be taken into account is user interest similarity. The conventional collaborative filtering algorithms disregard such a scenario, in which two users rated on multiple items differently, but there is a high similarity among the users. One of the understanding is some users incline to rank high when they rate an item, while others tend to rate it inferior. Thus the proposed algorithmic approach attains an edge over the existing collaborative filtering by incorporating these prominent features in the improved recommendation systems.

A. What is Collaborative Filtering?

The growth of the Internet has made it much trickier to efficiently access data from all the available online information. The overpowering amount of data necessitates mechanisms for proficient data filtering. Collaborative filtering is one of the methods used for dealing with this difficulty.

Collaborative filtering is a procedure that can strain out items that a user might like based on reactions by similar users. The algorithm works by finding a similar set of users based on their liking. It combines items they like to create a ranked list of suggestions. A lot of investigation has been done on collaborative filtering (CF), and the most accepted formulations are based on low-dimensional factor models (model-based matrix factorization). The incentive for collaborative filtering comes from the thought that people often get the best recommendations from someone with tastes similar to themselves. Collaborative filtering encompasses techniques for matching the public with similar interests and making recommendations on this basis.

Collaborative filtering algorithms often need (1) users' lively participation, (2) an effortless way to signify users' interests, and (3) algorithms that are capable to go with the public's similar interests.

Usually, the working of a collaborative filtering system is:

1. A user shows his / her preferences by rating items (e.g. books, movies, or CDs) of the system. These ratings can be viewed as a predictable representation of the user's significance in the alike domain.
2. The system matches this user's ratings in opposition to other users' and finds the people with the most "similar" tastes.
3. With similar users, the scheme recommends items that the alike users have rated greatly but not yet been rated by this user (the most probably the absence of rating is frequently considered as the unfamiliarity of an entry).

The CF techniques are generally separated into 2-types:

1. Memory-Based Approach
2. Model-Based Approach

The memory-based approach uses cosine similarity whereas the model-based approach uses machine learning.

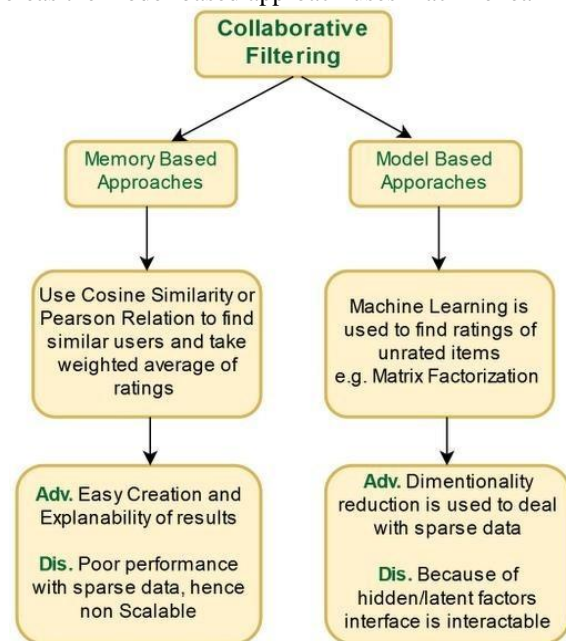


Fig. 2. Types of collaborative filtering approaches.

II. LITERATURE SURVEY

In this section, we are going to discussed some past research that have been done in the field of collaborative filtering algorithm, their benefits, limitations and technologies used.

Sr. No.	REFERENCE PAPER NAME	NAME OF AUTHORS	DESCRIPTION	BENEFITS	LIMITATIONS
1.	An improved collaborative recommendation algorithm based on optimized user similarity [1]	Hao Chen, Zhongkun Li, Wei Hu (Springer Science+ Business Media New York 2015)	A balancing factor is added to the traditional cosine similarity algorithm, which is used to calculate the project rating scale differences between different users.	Experimental results show that the method can effectively solve the problem of the high similarity of neighbor set caused by user rating scale of individual project differences, which enhanced the similarity of user accuracy and perfected the quality of the recommended	When the selected neighbor set is small, the results obtained by the traditional collaborative filtering is not ideal; obviously there is a data sparse problem. However, the improved recommendation algorithm has better recommendation results even if the system has the problem of data sparsely.

2.	Collaborative Filtering Service Recommendation based on a Novel Similarity Computation Method [2]	Xiaokun Wu, Bo Cheng, and Junliang Chen (IEEE Transactions on Services Computing)	In this paper, ratio-based method is proposed to calculate the similarity. The similarity between users or between items can be computed by comparing the attribute values directly.	Based on the new similarity measurement method, they proposed a prediction method and the experimental results show that new method is superior.	It is seen that the complexities of all the methods used here are $O(mn)$. Although these methods have the same complexity, they involve diverse operations and additional calculations.
3.	A multi-level collaborative filtering method that improves recommendations [3]	Nikolaos Polatidis, Christos K. Georgiadis (EL SEVIER publication)	This paper proposes the use of an e-recommendation collaborative filtering method by dividing user similarity, as offered by Pearson Correlation Coefficient (PCC) or Cosine Similarity.	The proposed method can be applied in different online domains that use collaborative recommendation systems, thus improving the overall user experience.	In collaborative recommendations, particularly in e-commerce, various users tend to give only negative or only positive ratings in order to affect the revenue of particular items or services. These types of users should be detected and removed.
4.	Collaborative filtering recommendation based on dynamic changes of user interest [4]	Ibtissem Gasmia, Hassina Seridi-Bouchelaghemb, Labar Hocinea and Baareh Abdelkarimc Mokhtar, Annaba University, Annaba, Algeria (INTELLIGENT Decision Technologies, vol. 9, no. 3.)	The proposed algorithm endows each score with a weight function which keeps user's recent, long and periodic interest, and attenuate user's old short interest.	A new item-based collaborative filtering algorithm is proposed to exploit genre information in each item and make dynamic changes over time of user's ratings.	Under the framework of item-based collaborative filtering, this paper focuses on modelling time effects in collaborative filtering by introducing a new simple weight function that not depends on any threshold and takes into account the changing of user interests over time.
5.	An effective collaborative filtering algorithm based on user preference clustering [5]	Jia Zhang, Yaojin Lin, Menglei Lin, Jinghua Liu (Springer Science +Business Media New York 2016)	In this paper, a cost-effective collaborative filtering algorithm based on user preference clustering is put forward to reduce the consequence of the data scarcity.	For identifying different typical users, the essential work in this paper is to plan a framework to assign users into user groups with contrasting preferences. Therefore, the neighbor users of the progressive user can be found with conformable preference.	This paper does not take into account the concept of user interest and rating differences which in turn hampers the performance of the recommendation systems.

III. GAP ANALYSIS

Numerous recommendation systems have been designed and deployed and nearly all of them have used content-based filtering (CBF) and/or collaborative filtering (CF) as their fundamental techniques.

In [1], A balancing element is applied to the conventional cosine similarity algorithm, which is used to quantify the project rating scale disparities between different users, to deal with problems that arise in traditional collaborative filtering recommendations, such as data scarcities, cold start, recommendations, consistency and timeliness. But, however, when calculating the user similarity, the above algorithms don't take user interest into account and neglect the influence of different rating standards between users on the similarity calculation.

In [3], To improve the precision of recommendations user similarity can be divided, as offered by Pearson Correlation Coefficient (PCC), to different levels and add constraints to each level. It is shown that by modifying the user similarity, which is a value from -1 to 1 , according to the constraints that each user belongs, the accuracy of their commendations is improved.

In [4], To address the issue of the influence of time on user's interests, a new item-based collaborative filtering algorithm is proposed to exploit genre information in each item and imitate dynamic changes over a period of time of user's preferences. The proposed algorithm assigns each score with a

weight function that keeps the user's recent, long and periodic interest, and faded the user's old short differences and hence lags in providing valuable recommendation systems.

In [5], A new impelling collaborative filtering algorithm based on user preference clustering is designed to decrease the consequence of the data sparsely as collected data is extremely sparse in the user-item rating matrix, meantime many present similarity measure methods using in collaborative filtering are ineffective, which result in the poor operation. This method considers users who have different eating habits, and different typical users are defined to generate user groups with different preferences. The above algorithms don't take user interest into account and neglect the consequence of different rating standards between users on the similarity computation.

Hence, from the above analysis, it is evident that most of the traditional recommendation algorithms focuses on the user rating about that product or services, and during this recommendation, the system doesn't consider the user rating differences and also the interest of users. This drawback probably misleads the correct recommendation to the user. Hence we are proposing a collaborative filtering algorithm that is based not only on user rating difference but also considering user interest in the product and services what the user likes.

IV. ALGORITHM

A. Traditional Collaborative Filtering Algorithm

A traditional collaborative filtering algorithm is used to analyze the user’s historical behavior information, to find the preference relationship between users or items, then to give recommendation.

The data with a collection of items and a set of users that have responded to any of the items, in order to play with recommendation algorithms, is required.

The reaction can be rating on a scale of 1 to 5 or viewing an item, adding it to a wish list. Usually working with such data is represented using matrix format. Each row consists of ratings given by user, and each column consist of ratings received by an item.

For instance, a matrix with 5 users and 5 items is shown below.

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

Fig. 3. Rating Matrix

Here, users have ranked some of the products on a scale of 1 to 5 in the matrix. The third object, for example, received a 4 from the first customer. The cells in the matrix are usually empty as users only score a few objects. It’s unlikely that every user will rate or react to each and every item present. Hence, the matrix with empty cells is called sparse, while the filled matrix is called dense.

To generate such a matrix, CFA retrieves the user and the movie ratings given by the user from a database, and for each record it creates a vector called user-item vector, resulting in an n-user-item vector, and thus a user-item matrix with ratings associated with each cell is produced. Traditional CFA performs statistical operations such as calculating user similarities and locating a user’s k-nearest-neighbors. To recommend movies to a specific person, Traditional CFA uses previously determined neighbors (users) to estimate the ranking for each un-rated movie. It then sorts the movies based on scores, resulting in high-rated movies being prioritized when recommending the list of movies to end users or targeted user.

Matrix construction, finding similarity, finding k-nearest neighbor, and predict rating are the four main steps in the Traditional CFA procedure. The computation of similarities is the most critical step in the whole algorithm. Figure 4 shows a visual representation of Traditional CFA.

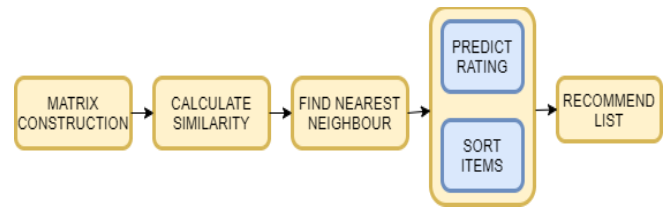


Fig. 4. Work-flow model of Traditional Recommendation system

B. Similarity Metrics

There are several methods for computing similarities, such as Euclidean Distance, Jaccard Similarity, Cosine Similarity (COS), Adjusted Cosine(ACOS), and Person Similarity (Pearson Correlation Coefficient, PCC). They are calculated as follows:

1) Euclidean Distance:

The interval between two points in space is called Euclidean distance. The difference between two users can also be expressed as a ranking, with the closer the distance, the higher the similarity between users. The Euclidean distance measures the similarity between user a and user b.

$$SimE(a, b) = \frac{1}{1 + \sqrt{(\sum_{i \in I_{a,b}} (R_{a,i} - R_{b,i})^2)}} \quad (1)$$

where, $R_{a,i}$ and $R_{b,i}$ are user a and b's ratings on item i respectively, and $I_{a,b}$ is the set that user a and b assessed together.

2) Jaccard Similarity:

There are several possible comparisons between users that do not have a shared ranking object in the user rating matrix. The standard approach involves calculating the degree of co-rating similarity between users, but there is no way to discover possible comparisons between users. As a result, despite their high similarity, some users could not have reliable forecasts.

$$SimJ(a, b) = \frac{|I_a \cap I_b|}{|I_a \cup I_b|} \quad (2)$$

where, a and b are the count of rating item sets of user a and b, respectively.

3) Cosine-based similarity:

Cosine Similarity evaluates the similarity of two vectors in space by computing their cosine. The smaller the angle and the closer the two vectors are, the closer the cosine value is to 1. The user rating can be represented as a space vector, with the cosine value increasing as the user similarity increases.

$$SimC(a, b) = \frac{\sum_{i \in I_{a,b}} R_{a,i} R_{b,i}}{\sqrt{\sum_{i \in I_a} R_{a,i}^2} \sqrt{\sum_{i \in I_b} R_{b,i}^2}} \quad (3)$$

where, $R_{a,i}$ and $R_{b,i}$ are rating of user a and b on items i respectively, and $I_{a,b}$ denotes the set that user a and b co-evaluated.

4) *Adjusted cosine similarity:*

In the user rating method, various parameters are used; some users choose high ratings, while others choose low scores. The user rating scale factor is not taken into account by the cosine similarity, so the Adjusted Cosine similarity (a, b) is changed by subtracting the user average value from the Cosine similarity.

$$SimC'(a, b) = \frac{\sum_{i \in I_{a,b}} (R_{a,i} - \bar{R}_a)(R_{b,i} - \bar{R}_b)}{\sqrt{\sum_{i \in I_a} (R_{a,i} - \bar{R}_a)^2} \sqrt{\sum_{i \in I_b} (R_{b,i} - \bar{R}_b)^2}} \quad (4)$$

where, \bar{R}_a represents the average of the user ratings on item I_a , \bar{R}_b represents the average of the user ratings on item I_b .

5) *Find Neighbour and Predict rating*

After calculating the user similarity, K nearest Neighbours of target user are selected. Now, the target user may be somewhat close to some users while being very different from others. As a result, the ratings given to a specific object by users who are more similar should be given more weight than those given by users who are less similar, and so on. A weighted average approach can be used to solve this problem. This method involves multiplying each user's rating by a similarity factor determined using the predictive rating formula. [7].

$$R_{ap} = \bar{R}_a + \frac{\sum_{i \in N_a} Sim(a,i) * R_{ip}}{\sum_{i \in N_a} |Sim(a,i)|} \quad (5)$$

- where a is the target user
- p is set of items
- R_{ap} : rating of user a against item/movie p
- N_a : set of all neighbors of a
- \bar{R}_a : average rating of user a

C. *Disadvantages of Traditional Similarity Metrics*

With the growth of e-commerce, the number of users and products has grown exponentially, resulting in an increasingly sparse consumer rating matrix, with even less popular rating items between users. Using traditional similarity calculations are difficult to get real recently Neighbor Set. For example, user a gets a set of scores as (1, 2, 1), user b's (4, 5, 4) and user c's (5, 4, 5). It can be seen from these scores that the user b has a high similarity with the user c, but the result of using the Pearson correlation coefficient is that the user a has a high similarity with the user b, so that we cannot get true Nearest neighbor.

Such 3 kinds of similarity methods are all based on calculation of similarity measurement of vector, and match strictly between object attribute. Among them, method of cosine similarity measurement regards users' score as a vector, it measures similarity among users though cosine angle of vector, however, it does not include of statistical characteristics of users' score; On the basis of cosine similarity, revised cosine similarity subtracts users' average score of items. Method of relevant similarity evaluates users' similarity according to the item scored commonly by both sides, the relevant similarity and the revised cosine similarity would be same if all of the score items among users are common score items. Although items score of users can reflect greatly degree of similarity of users, but different items score of users also changes as their interests changing, and vector-based calculation way of similarity exist defects in matching the nearest neighbor collections.

D. *Improved Collaborative Filtering Algorithm*

From the previous discussion, we concluded that Traditional CFA fails to capture the major aspects while recommending the movies to user. In our proposed Improve CFA, we have considered two major factors user rating difference and user interest similarity, which are explained in details in further section.

1) *Rating Difference:*

In our proposed improved CFA, we have highly emphasized on user rating difference, reason being that some users tend to rate high while some used to rate to low. In spite of such high difference, there is high possibility of those user ratings vectors having similar absolute mean difference vector derived from rating vector.

Consider, users A and B with ratings $R_A = \{4,3,4,5\}$ and $R_B = \{2,1,2,3\}$, \bar{R}_A and \bar{R}_B be the average or mean ratings of user A and B. From above rating vectors, we get $\bar{R}_A = 4$ and $\bar{R}_B = 2$. Now, finding the absolute mean vector of A and B using below formula,

$$R'_A = R_A - \bar{R}_A \quad (6)$$

Hence, we get new rating vector R'_A and R'_B as $R'_A = \{0,1,0,1\}$ and $R'_B = \{0,1,0,1\}$. Here we can observe that if we would have calculated the similarity between such vectors using traditional CFA, we would have get different results in spite of having similar difference rating vector. Hence we can conclude that R'_A and R'_B are same hence the user rating vectors are also same in some respect. Hence to improve the accuracy of similarity calculation we need to add rating difference factor.

When calculating the user rating difference, a user rating difference factor is added to the similarity calculation. Firstly, the rating difference is calculated between user A and B, as follows:

$$sumDiffer(u_a, u_b) = \sqrt{\sum_{i \in I_{a,b}} (R_{a,i} - R_{b,i})^2 / M} \quad (7)$$

where, $\text{sumDiffer}(u_a, u_b)$ denotes the rating difference between user u_a and u_b , $I_{a,b}$ is the common rating of user u_a and u_b , and M is the count of $I_{a,b}$ [1].

The user rating difference factor as follows:

$$\omega(u_a, u_b) = \lambda^{\text{sumDiffer}(u_a, u_b)}, \quad 0 < \lambda < 1 \quad (8)$$

where $\omega(u_a, u_b)$ is the rating difference factor between user u_a and u_b , $0 < \lambda < 1$ is the weight of the score difference factor. Next, the rating difference factor is added to the traditional algorithm to get the improved algorithm, so that to get the Collaborative filtering algorithm base on rating difference.

To get the final similarity metric, we need to include rating difference factor while predicting the rating, hence improve rating algorithm can be formulated as

$$\text{SimD}(u_a, u_b) = \text{Sim}(u_a, u_b) * \omega(u_a, u_b) \quad (9)$$

where, $\text{Sim}(u_a, u_b)$ is the traditional similarity algorithm, $\text{SimD}(u_a, u_b)$ is improved algorithm, which includes rating difference factor.

2) User Similarity based on Interest

In fact, it has been discovered that user ratings are not only related to user similarity, but also to user interest. It is possible that if two users have a high percentage of a specific category of movie, such as the adventure genre, whether or not they have identical ratings for those movies, we can still draw some conclusions that they are both highly affected by such category of movie, and therefore those users are highly similar to each other.

Hence we can conclude that user similarity of interest is high when they like same genre movie. The user interest similarity can be calculated as:

$$H_{u,g} = N_{u,g} / N_u \quad (10)$$

where g is genre of movie
 N_u is the user's rating for all movie and $N_{u,g}$ is user's rating for all movie with genre g .
 $H_{u,g}$ is user interest in movie's belonging to genre g .

Hence user interest Similarity SimI can be calculated as:

$$\text{SimI}(u_a, u_b) = \frac{\sum_{g \in n} H_{u_a, g} * H_{u_b, g}}{\sqrt{\sum_{g \in n} H_{u_a, g}} \sqrt{\sum_{g \in n} H_{u_b, g}}} \quad (11)$$

where $\text{SimI}(u_a, u_b)$ is user interest Similarity for user u_a and u_b and n is the set of all movie genres.

To formulate the final equation to find the user similarity based on rating difference and similarity interest, we would combine equation (9) and (11) as:

$$\text{Sim}_D(u_a, u_b) = \text{SimD}(u_a, u_b) * \alpha + \text{Sim}_I(u_a, u_b) * (1 - \alpha) \quad (12)$$

where $\alpha \in (0,1)$ which is the weight of similarity of users' interest in the whole users' similarity.

E. Improved Collaborative Filtering Algorithm Steps:

Based on all above formulas, the proposed algorithm for Improved Collaborative filtering algorithm can be concluded as follows:

- Step1:** Read MovieLens dataset into dataframes. Pivot the user_genre and rating dataframes as user_genre_matrix and user_item_matrix respectively.
- Step2:** Calculate the similarity between users by using Traditional CFA and obtain the user similarity matrix.
- Step3:** Calculate the rating difference factor using (9) and formulate it in a matrix.
- Step4:** Calculate the interest of each user using (11) and formulate it in a matrix.
- Step5:** Combine the user rating and user interest similarity matrices from step 3 and step 4 to obtain a comprehensive similarity matrix by considering their weighted calculation.
- Step6:** Use KNN to obtain Top-N neighbor set according to comprehensive similarity matrix.
- Step7:** Predict end-user's score for a particular movie according to Top-N neighbors.
- Step8:** Generate an output list of recommendations to the end-user.

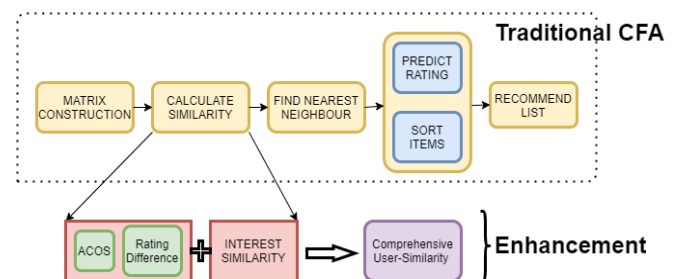


Fig. 5. Work-flow model of Improved Recommendation system

V. PROPOSED SYSTEM

The aim of our proposed system is to develop a Recommendation System using a modified collaborative filtering algorithm. This enhanced algorithm incorporates user interest and rating difference in the similarity calculation. The system first takes the user-item score matrix as an input that contains the ratings of the users against the items it has consumed. The user-item matrix also contains some unknown ratings that have to be predicted on demand. The enhanced similarity is calculated by incorporating the user rating

difference and user interest in two stages. In the initial stage, the user rating difference is added along with traditional rating similarity by weighting it by a rating difference factor. Adjusted cosine is used in traditional algorithm for user similarity calculation, as it has the lowest mean absolute error. The system is developed to reduce the adverse impact on recommendations when the two users of high similarity rate items differently. In the subsequent stage of similarity calculation, the user interest is taken into account as user ratings are highly influenced by user interest. The user interest similarity is calculated by incorporating the genres of the items. This enhanced similarity is used for calculating the comprehensive user similarity matrix and then the Top-N neighbor set is generated for predicting the users' unknown ratings which are used for recommending items.

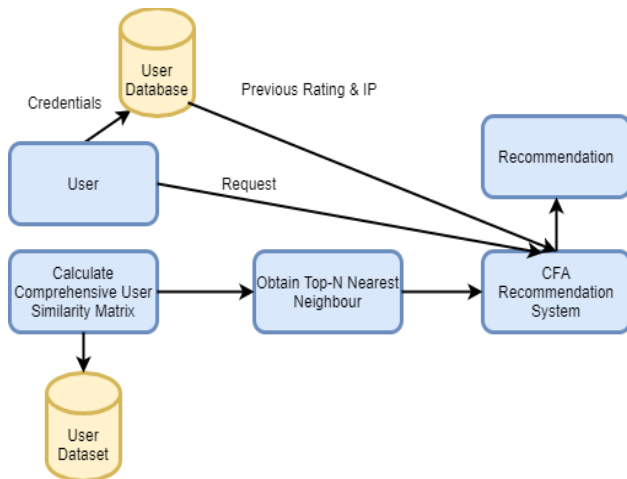


Fig. 6. Proposed Work-flow

VI. SYSTEM ARCHITECTURE

The system architecture for the recommendation system consists of modules namely the WebApp module, dataset, enhanced CFA engine, N-neighborhood selection module and database. The system is implemented using Python 3.6, Django framework, MySQL database and Pycharm Integrated development environment as development tool.

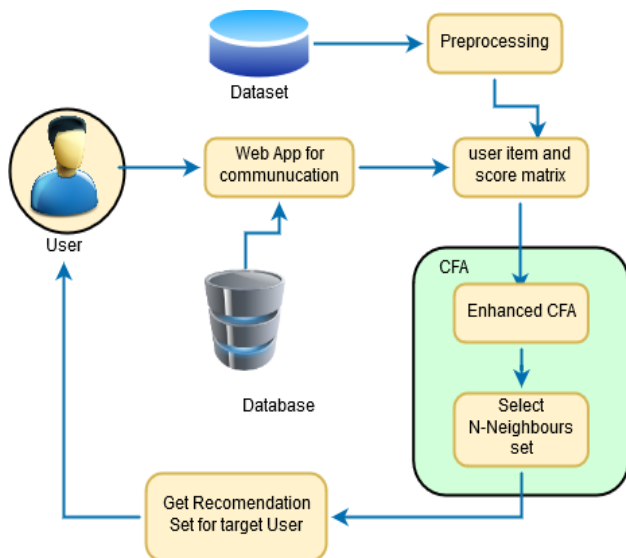


Fig. 7. System Architecture

The modules are as follows:

1) *WebApp Module*: The web application renders via interface for the user to interact and use the system by providing functionalities such as registration, login, enabling item-rating and request for recommendations.

2) *Dataset*: The dataset used is the MovieLens dataset provided by GroupLens which has a collection of 943 users with more than 100,000 ratings for 1682 movies [1]. The users have rated movies from range of 1-5, data sparsity being 93.7%. The dataset is in the form of user-item score matrix.

3) *Enhanced CFA module*: The enhanced CFA module consists of the recommendation model that is prepared by training the pre-processed dataset. The module comprises of the comprehensive user similarity matrix which is calculated by incorporating rating difference and user interest. This comprehensive user similarity is used for generating the Top-N neighborhood set which in turn used for recommendation.

4) *Database*: The database consists of user and movie details. The user information such as authentication details movie ratings for specific user are stored in MySQL database. Movie details such as movie name, movie item are stored for providing information for rating.

VII. EXPERIMENTAL RESULTS

A. Experimental Environment

The laptop configuration used in the experiment has a Core i5-2450M, dual-core, 3.2GHz, 8GB DDR4 memory, Window10 operating system, the programming language is Python, the version is Python3.6, and the development tool is Anaconda3.

B. The Dataset

The MovieLens dataset, compiled by the GroupLens analysis project, contains 943 consumers that have collectively rated 1682 films with over 100,000 scores. The consumer ratings in this data collection vary from 1 to 5, and the data is 93.7 % sparse. In the experiment, 80 percent of the data was chosen at random as part of the training sample, while the remaining 20% was used to create the test set [1].

The dataset consists of many files which contains different information about users and movies. Out of all most important files are:

- u.item: movies list
- u.data: ratings list (given by users)

The ratings are selected in the file u.data, which is a tab separated list of user ID, item ID, ratings and timestamp. The initial 5 rows of the file appear like below:

user_id	item_id	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

Fig. 8. First 5 rows of MovieLens 100k Data

C. Metrics

The key goal of the proposed algorithm is to increase the recommendation algorithm's accuracy. We test the proposed algorithm using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) in order to calculate its accuracy. By comparing the scores between the user experimental prediction rating and the user actual rating, the MAE and RMSE can be calculated. The suggested algorithm performs best when the MAE and RSME values are poor. The suggested algorithm's user predictive rating set is $\{r_{p1}, r_{p2}, \dots, r_{pn}\}$, while the user actual rating set is $\{r_{a1}, r_{a2}, \dots, r_{an}\}$, with n being the number of predictive objects.

$$MAE = \frac{1}{n} \left(\sum_{i \in n} |r_{pi} - r_{ai}| \right)$$

$$RMSE = \frac{\sqrt{\sum_{i \in n} (r_{pi} - r_{ai})^2}}{n}$$

D. Results

To begin, the MAE values were determined using Euclidean distance, Jaccard similarity, Cosine Similarity, Adjust Cosine Similarity, and Person Similarity.

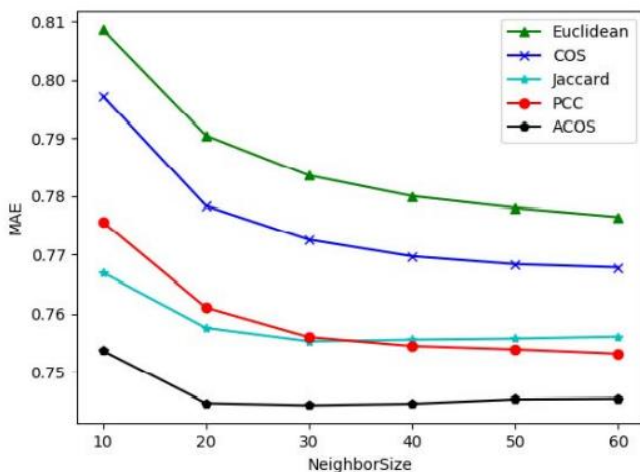


Fig. 9. MAE comparison of traditional algorithms.

Considering the factor of Difference in User Ratings Weight $\lambda = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and the number of neighbors $N = \{10, 20, 30, 40, 50, 60\}$, the result

obtained under various conditions is compared to establish the most suitable value of λ .

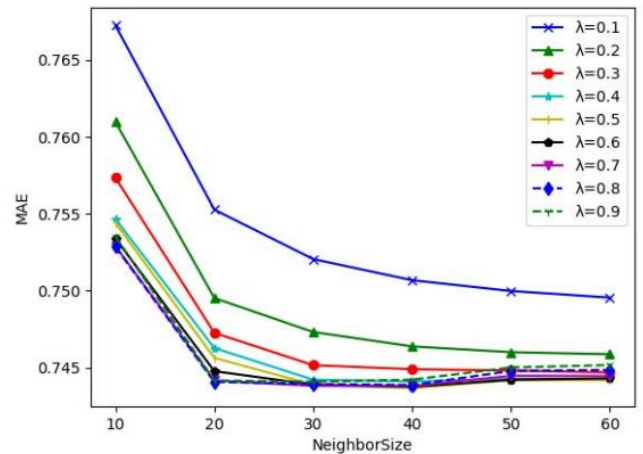


Fig. 10. MAE under different λ

The weight $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ denotes certain proportion of user interest in the entire user similarity. As a result, the most precise α must be determined before the user similarity can be calculated.

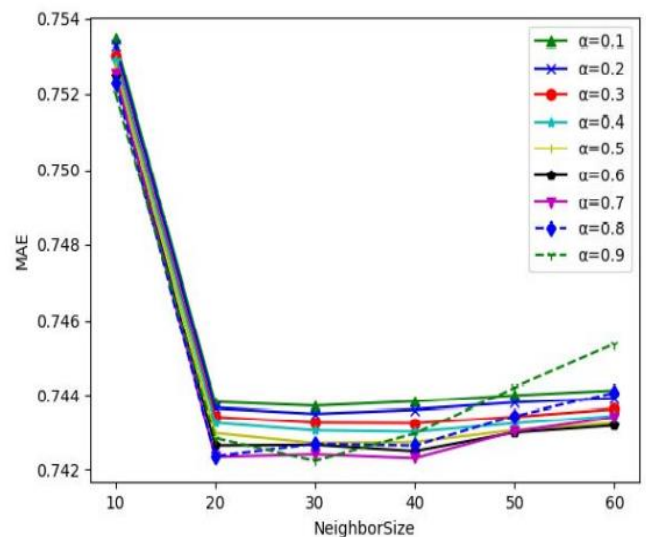


Fig. 11. MAE under different α

Here we can see from above graph that for α in range 0.6 to 0.8 we are getting a linear graph between MAE and neighbor-size. For neighbor-size = 40 $\alpha = 0.7$ achieves his low MAE after gradually decreasing when neighbor-size is below 40 and then gradually increasing by some margin when neighbor-size is greater than 40. For MovieLens dataset we got best result when neighbor-size is 40 and the weight of interest α is 0.7.

The values of MAE extracted in the figure above have a smaller contrast than the standard ACOS algorithm, indicating that the proposed algorithm is more reliable than ACOS. The explanation for this is that we added the rating difference factor and user interest similarities to the standard algorithm to make the collaborative filtering algorithm more efficient and

effective. Meanwhile, when the number of neighbors is 10, the MAE values are high due to the data-sparse problem caused by the small neighbor size. Even when faced with a data sparse challenge, the proposed algorithm outperforms the conventional algorithm.

VIII. EXPERIMENTAL RESULTS

Collaborative Filtering is the majority familiar technique used when it comes to the construction of intelligent recommendation systems that can learn to give better recommendations as more information about users is collected. We used this practice to build recommendation systems that give suggestions to a user resting based on the likes and dislikes of similar users. It works by searching a large group of people and finding a minor set of users with tastes similar to an exacting user. It collects a list of items they like and combines them to create a ranked list of suggestions.

This paper examines the efficiency of a shared filtering recommendation algorithm based on user rating difference and user interest using several experimental schemes. Firstly, the design ideas and algorithm steps are introduced, and then rating difference factor and user preference are considered in the standard similarity algorithm. Second, multiple algorithm parameters are calculated by a series of experiments. Finally, the paper's optimized algorithm is contrasted to the standard one. The proposed improved algorithm not only increases precision in general, but also produces a better result when dealing with sparse results. The experiment's drawback is that the experimental conditions cannot provide the best outcomes across different datasets. The next step is to figure out how to make the suggested algorithm automatically change the parameters based on various data sets and also make it scalable with large and real-world datasets. It should also take into account the user's preferences in terms of changing time and mood. One of the most critical aspects of our future research would be the protection of privacy.

ACKNOWLEDGMENT

I want to express my deepest gratitude to my respected guide Dr. V. V. Kimbahune for his valuable guidance that gave me proper direction and inspiration to complete this research work satisfactorily. He always stood by me and made it easy for me to accomplish the work in given time-frame. Also, I want to thank Prof. V.S. Deshmukh, Prof. P. A. Sonewar and Prof. R.A. Kudale as their suggestions were also very helpful and provided me a meaningful insight and motivation for the successful accomplishment of the given task. Last but not the least, I want to thank Dr. P.N.Mahalle Sir, for giving me this wonderful opportunity.

REFERENCES

- [1] Chen H,LiZ,HuW. An improved collaborative recommendation algorithm based on optimized user similarity [J]. Journal of Supercomputing, 2016, 72(7):2565-2578.
- [2] Wu X, Cheng B, Chen J. Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method [J]. IEEE Transactions on Services Computing, 2017, 10(3):352-365.
- [3] Polatidis N, Georgiadis C K. A multi-level collaborative filtering method that improves recommendations [J]. Expert Systems with Applications, 2016, 48:100-110.
- [4] Gasmis I, Seridi-Bouchelaghem H, Hocine L, et al. Collaborative filtering recommendation based on dynamic changes of user interest[J]. Intelligent Decision Technologies, 2015, 9(3):271-281.
- [5] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering [J]. Applied Intelligence, 2016, 45(2):230-240.
- [6] Park Y, Park S, Jung W, et al. Reversed CF: A fast collaborative filtering algorithm using a k -nearest neighbour graph[J]. Expert Systems with Applications, 2015, 42(8):4022-4028.
- [7] Kaleli C. An entropy-based neighbour selection approach for collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(C):273-280.
- [8] Park Y, Park S, Jung W, et al. Reversed CF: A fast collaborative filtering algorithm using a k -nearest neighbour graph[J]. Expert Systems with Applications, 2015, 42(8):4022-4028.
- [9] Pitsilis G, Knapkog S J. Social Trust as a solution to address sparsity-inherent problems of Recommender systems[J]. Computer Science, 2012.
- [10] Zhang X, Li Y. Use of collaborative recommendations for web search: an exploratory user study[M]. Sage Publications, Inc. 2008.
- [11] Pirasteh P, Jung J J, Hwang D. Item-Based Collaborative Filtering with Attribute Correlation: A Case Study on Movie Recommendation[M]// Intelligent Information and Database Systems. Springer International Publishing, 2014:245-252.