Credit Based Collaborative Filtering Approach: An Improvement in Recommender Systems

Isha Mangal¹, Deepali Bajaj²
¹(Vivekananda College, Delhi University, Delhi, India)
²(Shaheed Rajguru College of Applied Sciences For Women, University of Delhi, Delhi, India)

ABSTRACT: In recent years Internet has emerged as an important tool for gaining information. But people confronts with abundance of data. High performance information searching is required to cope with this situation. Recommender systems search appropriate documents or filter out inappropriate documents from several information streams in order to match with a user’s general interests. Recent developments in recommender systems implements user classes as per the demographic data. Collaborative Filtering is the most vital component of recommender system as it recommends items by considering the ratings of similar users. We argued that additional factors have an important role to play in guiding recommendation. In this paper, we propose the design of Credit Based Collaborative Filtering (CBCF) approach for recommender systems which employs prioritized features of items to improve the efficiency of recommendations. In our suggested plan, each recommender is assigned with a credit value which signifies the goodness of a recommender. The profile similarity in combination with credit value influences the decision making ability of recommender system. Further this credit value gets updated after every recommendation as user gives feedback about likeness of the item. Eventually these modified credit values make the recommender system learn and thereby to improve the prediction accuracy in comparison to the classical recommender systems. We have supplemented our approach with a case study of a movie recommender system. A comparison of the generated recommendations using CBCF approach to the classical approach establishes the validity of the proposed system.

Keywords: Recommender System, Collaborative Filtering, Profile Similarity, CBCF approach, Credit Values

I. INTRODUCTION

In today world, e-commerce is an excellent platform for buying and selling products and services. Products and services can be simple or complex like web hosting services, movies, music, electronic products, financial assistance, computers etc. Online Shopping is still a point of confront for the users as well as for web site owners. Recommender system provides automated assistance for such complex judgment task.

Recommender system has emerged as an important tool to tackle the problem of information overload. Before the emergence of recommender system, users were facing too much data to be able to effectively and efficiently filter out the pieces of relevance [1]. Recommender system produces individualized recommendations as output and thus guides the users in a personalized way to find items of his or her interest and significance from a much larger domain of information [2]. Thus recommender system is an individualized proactive information retrieval engine that makes correct predictions [3] which are captured by combining ideas like user profile, collaborative filtering and machine learning.

Collaborative filtering is the most familiar and extensively implemented technique for recommender system [4]. It uses aggregate ratings of objects, identifies commonalities between users on the basis of their ratings and generates new recommendations afterwards. Thus collaborative filtering is an inter-user comparison method [5].

In this paper, we have improved the recommendation efficiency of traditional recommender system by considering the feature of items. Each feature is assigned some numeric weighted values. Each recommender has a credit value that tells the goodness of his/her prediction. Recommender System will
choose highly rated items of a good recommender having highest credit value. Thus this method of demographic information about items finds the most relevant recommendations.

This paper is organized as follows: Section II pertains to related works and research in this area. Section III discusses about the background concepts for Credit Based Collaborative Filtering (CBCF). The proposed CBCF approach is discussed in section IV. Section V elaborates the suggested approach using a case study on movie recommender system. Results and data analysis is being done in section VI. Section VII summarizes conclusion and outlines future work in this area.

II. RELATED WORK

Recommender systems for automatically suggested items of interest to users have become increasingly essential in fields where mass personalization is highly valued [6]. A. Felfernig and R. Burke suggested a constraint based recommender system which has the capability of recommending complex products and services [7]. Philip Bonhard with Clark Herris has also contributed to improve recommender system using profile similarity. They emphasized on profile similarity between decision maker and recommender, familiarity of the recommender and rating overlap with a particular recommender that influence the choice of decision maker[8]. Mustnasar Ali Ghazanfar and Adam Prugel –Bannett have discussed a unique scalable cascaded hybrid recommendation approach by combining the rating, feature and demographic information about items [5]. Tariq Mahmood with Francesco Ricci has stated the reinforcement learning techniques to learn an optimal, user adapted interaction strategy, by encoding relevant information as features describing the state of information [9].

Bonhard and Sasse [10] suggest that matching people according to their profiles in terms of their hobbies and interest rather than item ratings or user neighbourhood statistical data alone - and explain recommendations in terms of similarity, would make it easier for user to judge the appropriateness of a recommendation [8].

Presently, most of the recommender systems are non trust based systems but recent studies emphasized that trust must be inherent component of recommender systems.

Zeigler et al. [11] have proposed a trust based recommender system in which every agent maintains degree of trust for a small subset of the overall community of agents. The users can set degree of trust on the peers explicitly and where ever degree of trust is not given explicitly, trust is predicted using propagation and trust decay in the node chains.

In the model presented by Massa et al. [12], users explicitly provide trust values on any number of peers. The peers on which values are not given; trust values are predicted using propagation of the explicit trust values provided for the intermediate nodes for the network.

In our proposed recommender system, credit values have been exploited to recommend products. Feedbacks are taken from the target users and degree of likeliness systematically updates the credit value. So the system learns from the response of the user.

III. BACKGROUND

This section describes the three techniques that together form the basis for CBCF approach.

a) Collaborative Filtering

In the collaborative filtering approach, the recommender system would identify users who share the same preferences (e.g. rating patterns) with the active user, and propose items which the like-minded users favored.

For our suggested approach, we have used a system to generate the recommendations for the movies. It works by matching together users with similar opinions about movies. Each member of the system has a 'neighborhood' of other like-minded users. Ratings from these neighbors are used to create personalized recommendations for the target user. We have categorized movie domain in six pools of movies. All movie items fall into one or more pool(s).
b) Profile Similarity

Profile Similarity is one of the possible factors that might be used to influence recommendations and predictions. Users have more confidence and reliance to other users of the same pool.

In the suggested approach, we have grouped users into different pools depending on liking or disliking for the movies. As a new user wishes to take recommendations from recommender system, his predictions will come from the pool of similar profile.

c) Credit Value

The credit value $C$ indicates the goodness of the recommendations of a user. Initially the $C$ of user $u_i$ is the ratio of number of recommendation given by $u_i$ to the total number of recommendations in recommender system. This value indicates the degree of interaction of a recommender with the system. More the value better is the credit value of that recommender. As the recommendations goes correct as per the liking of the user, credit value of the recommender $u_i$ is increased and vice-versa.

We have developed an integrated approach of collaborative filtering, profile similarity with credit values called CBCF which assist users in getting good recommendations as shown in Fig.1.

![Collaborative Filtering](image)

Fig. 1. Credit Based Collaborative Filtering Approach

IV. PROPOSED APPROACH

In this section CBCF approach is discussed to generate recommendations for movies followed by a case study on movie recommender system.

The following steps represent the sequence of events that take place that enable users to extract information about movies of their interest as per CBCF algorithm.

1. To start with, genres of movies are listed to the users. He will opt for one of the category of his interest.
2. Recommender System will find relevant movies from plethora of alternatives in the same genre opted by user in the preceding step.
3. Recommender system will select highest credit value and find the best rated movies of that recommender. Recommender system will detail out movies in a descending order with highest rated movies at the top followed by movies of recommender of lesser credit values.
4. User will select movie $m_i$ from the aggregated list.
5. On the basis of his liking of $m_i$, user will rate the features $f_i$ of movie.
6. The degree of likeliness is utilized to update the credit value of recommender.
7. Credit value will be upgraded or degraded as per the user response to $m_i$.

As more and more recommendations are generated by recommender system, credit values are updated accordingly. High credit value will reflect more reliable recommenders. System prediction accuracy enhances with each prediction and a substantial and consistent improvement in system efficiency is attained by systematic learning process on the basis of actual experience of users.

V. CASE STUDY

In this section, we have taken into account a case study on ‘Movie Recommender System’. We have collected the real life data and created a dataset for learning the individual ratings and empirically validating our method.

We now describe the data used in our experiments. The data was collected through survey for one month. Users were selected at random for inclusion. All users selected had rated at least 20 movies. This data has been cleaned up – users who had not given ratings of all the features of a movie were removed from this data set. In data sets, no demographic information about users is included. Each
user is represented by a u-id. Each movie is represented by a m-id.
This data set consists of:

- 3000 ratings from 60 users on 100 movies.
- Each user has rated all the features of a movie on an integer scale of 1(bad) to 10(excellent).
- Each movie may fall into one or more category.

We have extracted the major features of a movie and numeric weights are allocated according to their preference as shown in Table I.

<table>
<thead>
<tr>
<th>FEATURES (f_i)</th>
<th>F1 DIRECTION</th>
<th>F2 ACTING</th>
<th>F3 MOVIE PLOT</th>
<th>F4 MUSIC</th>
<th>F5 SCREENER PLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIGHT S(w)</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table I . Weighted features of m-id

The training data (Movies) has been categorized according to the genre like: action, comedy, horror, patriotic, romance and family. Users have given their personalized ratings about their like and dislike for all the features in consideration. Thus pools are generated for collaborative filtering.

As a new user enters the recommender system who wants to seek recommendation, specify the movie category of his interest. Using profile similarity, system chooses the matched category and then uses the collaborative filtering to seek for the appropriate match in the categories of movies according to the genre. And eventually recommendation will emerge from the pool of users taste only.

For our example, a new user u_5 have chosen “Action pool”.

**Pool : Action**

<table>
<thead>
<tr>
<th>User Id u-id</th>
<th>Movie Id m-id</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>M1</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>U3</td>
<td>M2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>U9</td>
<td>M6</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Table II . An example of Action Pool

Let W = { w_1, w_2, w_3, w_4, w_5} where w_i is a weight associated with each feature f_i of a movie.

Weighted sum(s_i) of all the features of a tuple is calculated as:

\[ s_i = \sum_{j=1}^{n} W_i f_i \]

where w_i is the weight of the feature of an item, r_i is the ratings of feature of an item given by the user u_i, n is the number of features of an item (here n=5).

This is a sample data which is taken from the data set of 3000 ratings. Initially the credit value \( \zeta \) of each recommender is represented using 1-D array CV [ ]. Credit values are computed as follows:

\[
\zeta_r = \frac{\text{No. of recommendations given by recommender } u}{\text{Total no. of recommendations}}
\]

(2)

Using the above equation, calculated credit value \( \zeta \) of four users is shown below:

CV [ ] = \{0.25, 0.33, 0.25, 0.17\}

This array indicates the degree of interaction of a recommender with the system. More the value better is the credit value \( \zeta \) of that recommender. According to CBCF approach recommender system will choose highly rated movies of a good recommender having the highest \( \zeta \).
The tuples are presented to the user in decreasing order of their weighted sum $s_i$. Once the user gets the aggregated list of recommendations, user selects the item from the list and uses it. The degree of likeliness is utilized to update the credit value of the recommender. The credit value is updated in such a manner that it increases for those recommenders who have given good recommendations and decreases otherwise.

Based on the response of the user, updation factor $\alpha$ can be calculated as:

$$\alpha = \sum_{i=1}^{n} \left( ru_{fi} - rr_{fi} \right) * w_{fi} \quad (3)$$

where $n$ is the number of features of an item, $ru_{fi}$ is the rating of feature $i$ of a movie given by the user, $rr_{fi}$ is the rating of feature $f_i$ of a movie recommended by the recommender, $w_{fi}$ is the weight associated with feature $f_i$ of a movie.

The value of $\alpha$ may be negative or positive. The positive value indicates that the user likes the movie suggested by the recommender and negative value indicates the dislike for the same. Thus the credit value updates accordingly as shown below:

$$CV \left[ u_i \right] = CV \left[ u_i \right] \pm \alpha \quad (4)$$

Once the user receives the recommendation he can now selects the movie from the recommended list and assigns a degree of likeliness to that movie.

Suppose the movie chosen by $u_5$ is $m_4$ which was recommended by $u_2$ and he has given the rating as per his liking:

<table>
<thead>
<tr>
<th>$u$</th>
<th>$m$</th>
<th>$F1$</th>
<th>$F2$</th>
<th>$F3$</th>
<th>$F4$</th>
<th>$F5$</th>
<th>$S_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_5$</td>
<td>$m_4$</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8.8</td>
</tr>
</tbody>
</table>

The updation factor $\alpha$ for user $u_2$ is shown below using equation (3)

$$\alpha = \frac{(9 - 9) * 3 + (9 - 8) * 3 + (9 - 8) * 2 + (8 - 7) * 1 + (8 - 7) * 1}{5} = 0.14$$

Therefore, $\alpha$ of $u_2$ is increased by 0.14 using equation (4). If the value of $\alpha$ comes out to be negative, then the value $\alpha$ of recommender is decreased by the same calculated value. Therefore, the positive value of $\alpha$ signifies that the user likes and appreciates the movie. The updated CV $[\ ]$ array is as follows:

$$CV \left[ \right] = \{0.25, .50, 0.25, 0.17\}$$

If the movie seen by the user is suggested by two or more recommenders, then $\alpha$ is updated for all those recommenders.

VI. EXPERIMENTAL RESULTS

To give strength to our case study, we have observed experimental results in contrast to conventional recommender system. In our suggested CBCF approach, weights are given to the feature of movie items. During the survey, we also gathered an overall rating of movie items from users.

Above graph shows the comparative analysis of proposed CBCF approach with traditional approach prevalent in earlier recommender systems. The ratings given by new user and recommender as per CBCF approach are closely correlated. Thus proposed approach is more reliable than conventional recommender system approaches.

![Fig.2. CBCF approach v/s Traditional approach](image)

![Fig.3. Time variant credit values of recommenders](image)
As system make more and more predictions, credit values are updated accordingly and machine learns with each new prediction. Thus over the time domain system’s accuracy, reliability and truthfulness will be enhanced.

VII. CONCLUSION AND FUTURE WORK

Recommender System have been developed to address the abundance of choices we face in taste domain (movies, restaurants, web pages, TV Programs/ Show/Episode, Video on-demand, Music, Books, News, Images etc.) when shopping or going out.

In this paper we propose a new breed of recommender system that follows CBCF approach which provides personalized recommendations based on credit values. CBCF approach further updates these values depending upon the likeliness of item being recommended. Results revealed that segmenting the movie into weighted features provides potentially better recommendations. We have supplemented our work with a case study on movie domain to generate recommendations for movies.

Our work can further be strengthened by including fractional ratings to movie items rather than just integer values. This approach can be supplemented by demanding the users to rate more movies and even less popular movies.

REFERENCES


