A Hybrid Collaborative Filtering Approach for Recommendations

Sweta Sarkar¹, Pritha Ghosh²

¹²Assistant Professor, CSE, Dr. Sudhir Chandra Sur Degree Engineering College, DumDum, Kolkata-700074, India

Abstract—Collaborative recommender system has been an important and popular approach in making recommendations. However, it suffers with the cold start and sparsity problems. Therefore, to alleviate the problems, we have combined a set similarity and user evaluation method in collaborative filtering by introducing some additional weight parameters. Further, we optimize these parameters by Particle Swarm Optimization (PSO). We named it as hybrid approach. The hybrid approach is tested on movie recommendation to the user. The experimental result confirms that the mean absolute error with the other approaches is encouraging.

Keywords—Set Similarity, Recommendation, Cosine Similarity, Prediction, Collaborative Filtering, Ontology.

I. INTRODUCTION

With the rapid growth of the activities via Internet, the amount of information as well as new applications increases eventually every day. People are overwhelmed with the information as they enjoy the benefits bought by the Internet. It becomes more difficult to you to find the information that you really need from the Internet. Recommender systems help in personalizing the user’s need so that they can get the best information according to their choices.

Collaborative filtering is gaining its popularity as an important approach in recommendation system. It based on the similarity between the users. If user A is similar with user B, then according to the system user A and B have same interests or similar choices, so the recommender make recommendation to the user A based on user B’s visit history. Whereas in content based recommendation user’s past history is used in making new recommendation to them. Unlike content based recommendation, collaborative filtering does not require data from the item. But the collaborative recommendation has the cold start problem, that is, if a new item is introduced in the recommendation system and no users has used or visited it before, then the new item will be ‘ignored’ though it may be useful to some users. According to the traditional collaborative filtering, user-items sparsity problem is another issue in recommendation system.

In this paper, we have introduced a novel hybrid method based on user ontology to provide more efficient and effective recommendation to the user. Our research is focused on movie recommendation domains. The experiments show how the hybrid filtering is good to make recommendation.

II. BASIC MATERIALS AND RELATED WORK

In the traditional approach researchers had used many of the data mining techniques to categorize the users according to their visit history. Many researcher try to reduce the sparsity problem by using (Vozallis & Margaritis 2007) [7] SVD (singular value decomposition). SVD reduces the dimensions of the user-item matrix. But SVD will lose much useful information in dimensionality reduction. Many of the approaches have used the different features like (actors, directors, kind) in representing a movie domain. Some work has been done on predicting the user’s needs based on (Bentley & Ujjin, 2003) [3] PSO (particle swarm optimization) and the ant colony optimization (Bedi, Makkar, Sharma & Singh, 2007) [10]. The researcher focuses on the particle swarm optimization algorithm to fine-tune a profile matching algorithm within a recommender system, tailoring it to the preferences of the individual users. It helps in assigning weights to the different demographic features like user occupation, age, gender and different features in the movie domains like actor, kind of movies it belongs, directors etc and hence in making more accurate recommendation (Ujjin& Bentley, 2003) [3] have described Particle Swarm Optimization (PSO) recommender system in which PSO algorithm has been employed to learn personal preferences of users and provide tailored suggestions.

(Bedi & Sharma, 2011) [2] have proposed a Trust Based Ant Recommender System (TARS), using ant colony optimization. They have incorporated a dynamic trust between the users and have selected a small and best neighborhood based on biological metaphor of ant colonies.
Along with the predicted ratings, displaying additional information for explanation of recommendations regarding the strength and level of connectedness in trust graph from where recommendations are generated, items and number of neighbors involved in predicting ratings can help active user make better decisions. Also, new users can highly benefit from pheromone updating strategy known from ant algorithms as positive feedback in the form of aggregated dynamic trust pheromone defines “popularity” of a user as recommender over a period of time. The performance of TARS is evaluated using two datasets of different sparsity levels viz. Jester dataset and Movie Lens dataset (available online) and compared with

Traditional Collaborative Filtering based approach for generating recommendations.

The volume of the data will continue to increase over time. Discrete wavelet transformation is an approach to enhance the scalability of memory based collaborative filtering. In particular wavelet transformation approach is proposed and applied to both synthetic and real-world recommending system.

Several works focuses on the user-item sparsity problem. Examples of these efforts include: using agents to filter and augment missing ratings, default or average voting (Vozalis, 2003) [11], using trust models to infer missing values and clustering. In our model we have used the column average voting values to infer the missing ratings. And in (Ghani& Fano, 2002) [5] in order to better understand customers, an ontology based approach is used to reason the products' attributes in a semantic way. However, they both lack the efficiency when modeling, organizing and extracting the semantics of the products.

A. System Framework

In our experiment we have proposed a hybrid recommendation system that combines the set similarity and the evaluation similarity between the users. Our task is to provide a detailed structure of the proposed model by identifying its building blocks.
Collaborative Filtering: It provides recommendation to an active user in the system based on their similarity with the other users (Grant, Shawn, McCalla, Gordon) [6] in the movie recommendation domain.

User Ontology in Movie Recommendation Domain

![User Feature Diagram](image)

Fig.(ii) Shows user hierarchy in the movie recommendation domain. (Sarwar, B, Karypis, G.Riedl) [9]

We can abstract 5 features of a user in the movie recommendation domain. They are Age, Gender, Occupation, Zip code and how the user has evaluated a movie.

B. Data Pre-processing and Clustering

Input data is stored in a form user-item rating matrix, in which users represent the rows, and the items represent the columns and the values in the cell represent the ratings, expressed by a user about an item. In our paper we have replaced the missing values by taking the column average and replaced each zero entry with respective column average values. Movie lens dataset consisting of ratings ranges in the discrete scale of 1 to 5. Specifically we calculate the column average of each column in the user-item matrix and fill all the slots in the same column that have no values, and denoted by 0 by that average;

\[ r_{ij} = \begin{cases} r_{ij} & \text{if the user } u_i \text{ has not rated the item } i_j \\ r_i & \text{if the user } u_i \text{ has rated the item } i_j \end{cases} \]  \hspace{1cm} (1)

Where \( r_{ij} \) is the rating of item \( i_j \) by user \( u_i \)

C. Particle Swarm Optimization

Particle Swarm Optimization is an algorithm capable of optimizing a non-linear and multidimensional problem which usually reaches good solutions efficiently while requiring minimal parameterization. The algorithm and its concept of "Particle Swarm Optimization"(PSO) were introduced by James Kennedy and Russel Ebbart in 1995.

However, its origins go further backwards since the basic principle of optimization by swarm is inspired in previous attempts at reproducing observed behaviour of animals in their natural habitat, such as bird flocking or fish schooling, and thus ultimately its origins are nature itself. These roots in natural processes of swarms lead to the categorization of the algorithm as one of Swarm Intelligence and Artificial Life.

The basic concept of the algorithm is to create a swarm of particles which move in the space around them (the problem space) searching for their goal, the place which best suits their needs given by a fitness function. A nature analogy with birds is the following: a bird flock flies in its environment looking for the best place to rest (the best place can be a combination of characteristics like space for all the flock, food access, water access or any other relevant characteristic).

Based on this simple concept there are two main ideas behind its optimization properties:

- Single particle (which can be seen as a potential solution to the problem) can determine "how good" its current position is. It benefits not only from its problem space exploration knowledge but also from the knowledge obtained and shared by the other particles.
- Stochastic factor in each particle's velocity makes them move through unknown problem space regions. This property combined with a good initial distribution of the swarm enable an extensive exploration of the problem space and gives a very high chance of finding the best solutions efficiently.

Every PSO have their own defined fitness function. The specification of this function then depends on the problem being optimized (especially in its dimensions).This function represents how good the \( i \) particle's position in the multidimensional space is relatively to the desired goal.PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest.

Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called lbest.
After finding the two best values, the particle updates its velocity and positions with the following equation:

\[ v_i^{t+1} = v_i^t + c_1 r_1^t (pBest_i - x_i^t) + c_2 r_2^t (gBest_i - x_i^t) \]  

(2)

Present[] = present[] + \( v_i^t \)  

(3)

\( v_i^t \) is the particle velocity, present[ ] is the current particle (solution). \( pBest_i \) and \( gBest_i \) are defined as stated before. \( Rand() \) is a random number between (0, 1). \( c_1 \) and \( c_2 \) are learning parameters usually taken as 1.

D. The Flowchart of PSO Algorithm

III. THE HYBRID APPROACH

This section will give clear view about the collaborative recommendation approaches used in the proposed model.

A. Set Based Similarity Algorithm

Set based similarity algorithm is used to measure similarity between two users based on their demographic features. We compute the set based similarity algorithm on our dataset using the given formula:

\[ \text{Sim}_\text{Set}(U_1, U_2) = \frac{\sum_{i=1}^{n} 2|S_{U_1i} \cap S_{U_2i}|}{|S_{U_1i}| + |S_{U_2i}|} \]  

(4)

\( S_{U_1i} \) is the data set of the feature \( i \) for user \( U_1 \) and \( S_{U_2i} \) is the data set of the feature \( i \) for the user \( U_2 \). \(|S_{U_1i}|\) and \(|S_{U_2i}|\) stands for the element number of feature \( i \) to user \( U_1 \) and \( U_2 \) respectively. Where \( n \) is the total feature sets.

For our experiment we have four demographic features so here \( n=4 \).

B. User Evaluation Based Similarity Algorithm:

Evaluation based similarity algorithm computes the similarity between two users based on their ratings given to an item. For our experiment we have used cosine similarity algorithm to compute similarity between two users. The two most common similarity algorithms used in movie recommendation system are the cosine similarity and Pearson’s correlation coefficient. Cosine similarity algorithm is defined as:

\[ \text{Sim}_\text{Eva}(U_1, U_2) = \frac{\sum_p R_{U_1,j} R_{U_2,j}}{\sqrt{\sum_p (R_{U_1,j})^2} \sqrt{\sum_p (R_{U_2,j})^2}} \]  

(5)

(2) where \( j \) is the total item set in the movie recommendation domain. \( R_{U_1,j} \) is the rating given by a user \( U_1 \) to a movie \( j \) and \( R_{U_2,j} \) is the rating given by a user \( U_2 \) to a movie \( j \).

C. A Hybrid Similarity Algorithm:

Combining the similarity in the above two sections, we made a hybrid recommendation system. Assuming that \( \text{sim}(U_1, U_2) \) is the hybrid similarity between the users, the hybrid similarity algorithm is as follows:

1. Set a weight \( W_i \) to every edges emerges out from a user node in Fig: (ii) and such that \( \sum W_i = 1 \)
2. \( \text{sim}(U_1, U_2) = 0 \)
3. while(feature \( f_i != NULL \))
   
   \{ \n   if(feature \( f_i != "evaluation" \))
   
   \( \text{sim}(U_1, U_2) = \text{sim}(U_1, U_2) + w_i * \text{Sim}_\text{Set}(U_1, U_2) \)
   
   Else
   
   \( \text{sim}(U_1, U_2) = \text{sim}(U_1, U_2) + w_i * \text{Sim}_\text{Eva}(U_1, U_2) \)
   
   \( f_i = f_i \text{ next feature} () \)
   \}

4. return \( \text{sim}(U_1, U_2) \);
International Journal of Emerging Technology and Advanced Engineering

D. Predicting A Movie By Using Resnick’s Prediction Formula:

We use formula 3, to predict item i’s ranking given by user u.

\[ R_{u,i} = \frac{\sum_{v \in \varphi(u)} \text{sim}(u,v) \cdot (R_{v,i} - \bar{R}_u)}{\sum_{v \in \varphi(u)} \text{sim}(u,v)} \]

(6)

Where \( R_{u,i} \) is the predicted rating on item i by user u generated by our proposed hybrid recommendation model. \( \varphi(u) \) is the set of nearest neighbors of user u. For our model we have considered the top 20 nearest neighbor. \( \bar{R}_u \) and \( \bar{R}_v \) stand for the average rating scores to the relative items evaluated by user u and v respectively. \( R_{v,i} \) is the rating score to item i given by user v.

E. Various Utility of the Recommendation Approaches:

1. If the user is a new comer then the recommender can better use the set similarity based algorithm to make recommendations to a user. This approach is suit to tackle the cold start problem with the new user.

2. If the user has used the recommendation system for a large period of time and he has given evaluation score to a large amount of movies then recommender can better use collaborative filtering using evaluation similarity approach.

3. If the user has the situation between (1) and (2), then recommender can better use collaborative filtering using hybrid approach to make efficient prediction. And later in the experimental evaluation part we have seen that hybrid approach has maximum accuracy measures so it is more preferable to use collaborative hybrid filtering to make prediction.

F. Rationale of Using PSO

The PSO algorithm has been used to attain feature weights for the active user, and hence help tailor the matching function to the user’s specific personality and tastes. In our experiment we have incorporated PSO to fine out the weights of \( w_1, w_2, w_3, w_4, w_5 \), and \( w_6 \) such that it should satisfies the equality constraint as mentioned above. We have repeated PSO and have obtained 20 such sets of data of five weights. Fitness function used for our data sets

\[ \text{MAE} = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N} \]

(7)

Other variants we have considered for the PSO Initial Population size=20

Initial weight=0.9
Final weight=0.4
\( c_1 \) and \( c_2 = 2 \)

IV. Evaluation and Experimentation Results

A. Experimental Data set

In this paper we have worked on Movie Lens dataset. Grouplens Research has collected and made available rating data sets from Movielens website (http://www.movielens.umn.edu). This dataset consists of 100000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. Simple demographic information for the user are age, gender, occupation and zip code. Different features have used in identifying a movie in a recommendation system.

Movie is classified by 19 different genres. 1 indicates that movie is of that genre and 0 indicates that movie does not belong to that genre, and movie can be in several genres at once.

- movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children’s | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western.

In the user-item matrix each entry is a tab separated list of user-id| item-id| rating| timestamp.

For ease of understanding we have ignored the timestamp attribute and a fragment of dataset comprising 100 users and 1682 items is used for our experiment. The range of our score is 1 to 5. 1: very bad; 2: bad; 3: average; 2: good; 4: very good; 5: excellent.

B. Experiment metrics:

MAE (MeanAbsoluteError) (Amira, Hiroo, Ikuo, Yasuo&Shingo, 2012; Sarwar, Karypis&Riedel, 2001) [1] can be used to measure statistical accuracy. In our experiment we compute MAE on the test set of each user, and then averaged over the set of test sets.

\[ \text{MAE} = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N} \]

(4) Where \( p_i \) is the predicted rating of user i and \( q_i \) is the actual rating and N is total users. Decision support accuracy metrics evaluate how effective a predictor help users in selecting high quality items. ROC (Receiver Operating Characteristic) can be used to measure decision support accuracy. In our experiment we considered that if a movie score is equal or above 4 (4 or 5) then we think that a movie is worth recommending to a user else if a score is less than 4 movie is rejected. The ROC-4 is as following:

\[ \text{ROC} - 4 = \frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} y_i} \]

(8)
\( x_i = \begin{cases} 1, & \text{if } p_i \geq 4 \text{ and } q_i \geq 4 \\ 0, & \text{otherwise} \end{cases} \quad (9) \)

\( y_i = \begin{cases} 1, & \text{if } p_i \geq 4 \\ 0, & \text{otherwise} \end{cases} \quad (10) \)

Where \( p_i \) is the predicted score for item \( i \) and \( q_i \) is the actual score for item \( i \) given by a user. If ROC-4 is larger, more accurate we think our recommender would be.

C. Experimental Results:

Our experiment is set as the following:

First we performed traditional k means algorithm (Ray & Turi) [8] on the entire data set (943 users and 1682 items).

Since the k-means method aims to minimize the sum of squared distances from all points to their cluster centre, this should result in compact clusters. We can therefore use the distances of the points from their cluster centre to determine whether the clusters are compact.

For this purpose, we use the intra-cluster distance measure, which is simply the distance between a point and its cluster centre and we take the average of all of these distances, defined as:

\[
\text{intra} = \frac{1}{N} \sum_{i=1}^{N} \sum_{x \in c_i} |x - z_i|^2 \quad (11)
\]

Where \( N \) is the total data points, \( k \) is the total number of cluster, \( z_i \) is the centre of the cluster \( c_i \). We obviously want to minimize this distance.

We can also measure the inter cluster distance that is the distance between clusters centre, which we want to be as big as possible. The inter cluster distance is defined as:

\[
\text{inter} = \min \left( \| z_i - z_j \|^2 \right) \quad (12)
\]

We calculate the distance between clusters and take the minimum of this value, as we want the smallest of the distance to be maximized, and the other larger values will be automatically be bigger than this value.

\[
\text{validity} = \frac{\text{intra}}{\text{inter}} \quad (13)
\]

Since we want to minimize the intra cluster distance and this value is in the numerator, so we want to minimize the validity measure. And according to the k means algorithm we want to maximize the inter cluster distance and this value is in the denominator, again we want to minimize the validity measures. Therefore, the clustering which gives a minimum value for the validity measure will tell us what the ideal value of \( K \) is in the k-means procedure. The experimental results are shown in Table.

<table>
<thead>
<tr>
<th>No of Clusters</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.9616</td>
</tr>
<tr>
<td>4</td>
<td>2.4349</td>
</tr>
<tr>
<td>5</td>
<td>2.4320</td>
</tr>
<tr>
<td>6</td>
<td>2.4101</td>
</tr>
<tr>
<td>7</td>
<td>2.2684</td>
</tr>
<tr>
<td>8</td>
<td>2.5344</td>
</tr>
<tr>
<td>9</td>
<td>1.6141</td>
</tr>
<tr>
<td>10</td>
<td>1.7293</td>
</tr>
<tr>
<td>11</td>
<td>1.7274</td>
</tr>
<tr>
<td>12</td>
<td>2.6920</td>
</tr>
<tr>
<td>13</td>
<td>2.7040</td>
</tr>
<tr>
<td>14</td>
<td>2.6776</td>
</tr>
<tr>
<td>15</td>
<td>2.6722</td>
</tr>
<tr>
<td>16</td>
<td>2.7119</td>
</tr>
<tr>
<td>17</td>
<td>2.6218</td>
</tr>
<tr>
<td>18</td>
<td>2.5962</td>
</tr>
<tr>
<td>19</td>
<td>2.0991</td>
</tr>
<tr>
<td>20</td>
<td>2.0980</td>
</tr>
<tr>
<td>21</td>
<td>2.0970</td>
</tr>
<tr>
<td>22</td>
<td>2.1493</td>
</tr>
<tr>
<td>23</td>
<td>2.0879</td>
</tr>
<tr>
<td>24</td>
<td>2.3642</td>
</tr>
<tr>
<td>25</td>
<td>2.4065</td>
</tr>
<tr>
<td>26</td>
<td>2.3983</td>
</tr>
<tr>
<td>27</td>
<td>2.2987</td>
</tr>
<tr>
<td>28</td>
<td>2.2970</td>
</tr>
<tr>
<td>29</td>
<td>3.3180</td>
</tr>
<tr>
<td>30</td>
<td>3.0639</td>
</tr>
<tr>
<td>31</td>
<td>3.2573</td>
</tr>
<tr>
<td>32</td>
<td>2.7202</td>
</tr>
<tr>
<td>33</td>
<td>3.1430</td>
</tr>
<tr>
<td>34</td>
<td>2.6356</td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>
Since cluster 3 would be very less for our data set we consider cluster 9 would be better. We then select first 100 users from the clustered data set and with 1682 item we calculate the MAE (mean absolute error) (Bresse, Hecherman & Kadie, 1998) [4]. We then can calculate similarity for each user with all other user using the three collaborative filtering approaches. We then select top N users (N=7) according to their value.

### Table II

Results of optimizing weights using PSO are shown

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3758</td>
<td>0.1332</td>
<td>0.1638</td>
<td>0.0821</td>
<td>0.2451</td>
</tr>
<tr>
<td>0.1975</td>
<td>0.2078</td>
<td>0.1383</td>
<td>0.1212</td>
<td>0.3352</td>
</tr>
<tr>
<td>0.4038</td>
<td>0.0354</td>
<td>0.1829</td>
<td>0.0738</td>
<td>0.3040</td>
</tr>
<tr>
<td>0.1305</td>
<td>0.3737</td>
<td>0.1326</td>
<td>0.0244</td>
<td>0.3398</td>
</tr>
<tr>
<td>0.2773</td>
<td>0.1590</td>
<td>0.1877</td>
<td>0.1583</td>
<td>0.2178</td>
</tr>
</tbody>
</table>

### Table III

Experimental Results of Various Collaborative Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE</th>
<th>ROC 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid Collaborative Filtering</td>
<td>0.8845</td>
<td>0.8320</td>
</tr>
<tr>
<td>Set Based Similarity Filtering</td>
<td>0.9156</td>
<td>0.8533</td>
</tr>
<tr>
<td>Evaluation Based</td>
<td>0.9238</td>
<td>0.8343</td>
</tr>
<tr>
<td>K Means</td>
<td>0.8996</td>
<td>0.8343</td>
</tr>
</tbody>
</table>

### V. CONCLUSION AND FUTURE WORK

This paper gives a novel collaborative filtering approach using user demographic feature in the movie recommendation domain. The user ontology based collaborative filtering can overcome the cold start problem. Through our experimentation, it proved that, hybrid approach can give better accurate result in recommendation. In future, we will enrich our experiment including more number of users.

---

**REFERENCES**


