An Adaptive Framework for Enhancing Recommendation Using Hybrid Techniques

Radhya Sahal[1], SaharSelim[1]and Abeer ElKorany[2]

[1]Phd student at Computer Science Department, Faculty of Computers and Information Cairo University, Egypt

[2]Associate professor at Computer Science Department, Faculty of Computers and Information, Cairo University, Egypt

ABSTRACT

Recommender systems provide useful recommendations to a collection of users for items or products that might be of concern or interest to them. Several techniques have been proposed for recommendation such as collaborative filtering, content-based, knowledge-based, and demographic filtering. Each of these techniques suffers from scalability, data sparsity, and cold-start problems when applied individually resulting in poor recommendations. This paper proposes an adaptive hybrid recommender system that combines multiple techniques together to achieve some synergy between them. Collaborative filtering and demographic techniques are combined in a weighted linear formula. Different experiments applied using movieLen dataset confirm that the proposed adaptable hybrid framework outperforms the weaknesses resulted when using traditional recommendation techniques.

KEYWORDS

Recommender System, Collaborative Filtering, Demographic Filtering, cold start, sparisty scalability

1. INTRODUCTION

Recommender Systems (RSs) were proposed as a computer-based intelligent techniques whose purpose was to assist with the problem of information overloading. By generating suggestions about new items for a particular user, she/he receive items recommended on basis of their previously purchase records. New users will need to rate appropriate number of items to allow the system to capture their preferences, and thus enable providing reliable recommendations. Thus, the most common form of input for a recommender system is that of ratings of past items. Another type of input is the demographic data regarding the user or item in mind which is usually hard to obtain and is normally collected explicitly from the user or manually from item catalogues [1]. Different techniques have been applied in RS such as Collaborative Filtering (CF)which is considered to be the most successful approach, as it makes its recommendations based on ratings provided by users who are similar to the active user [2]. Collaborative recommendation systems expect a user’s interest in new items based on the recommendations of other people with similar interests. CF are classified into two sub-categories: memory-based CF and model-based CF [3]. Memory-based approaches make a prediction by considering the entire collection of previous rated items by a user; examples include User-Based CF algorithms [4]. Model-based approaches learn a model from set of ratings and use this model for providing prediction; examples include item-based CF. Model-based approaches are more scalable than User-Based approaches [5]. Another type of recommender system depends on utilizing demographic attributes of users in order to produce their recommendations with the help of pre-generated demographic clusters.

DOI:10.5121/ijcsit.2014.6204

In present recommender system, users obtain recommended items based on ratings they have already provided. When a new user connects to the system, his/her list of ratings is empty, which makes it hard for the system to provide recommendations. This problem is well-known as cold start problems [6]. Large and sparse database also affect the efficiency and accuracy of recommendation process.

This paper proposes an adaptive framework that combines different recommendation techniques in order to overcome the problems of sparse and large data set as well as the cold start problem. Furthermore, the proposed model improves the recommendation performance for large scale dataset. The basic idea is to calculate similarity among users using both their demographic attributes as well as their pre-purchased items in order to identify the proper set of similar neighbors for each user. Then, by applying adaptable linear combination, those similarities are combined to generate new similarity value. Finally, the Top N neighbors based on new calculated similarities is used to recommend items. We compared the results of our hybrid approach to traditional recommender system that applies CF by usingMovieLens dataset. Experimental results showed that our approach outperformed others when ratings are very sparse. Experiments show that enhancing traditional CF techniques using demographic attributes, helps new users to overcome the cold start problem.

The reminder of this paper is organized as follows. Section 2 surveys related works in this area and provides an overview of several major approaches for recommender system. Our proposed framework is presented in section 3. Section 4 analyzes the experimental evaluation, followed by the conclusion and future work in section 5.

2. LITERATURE REVIEW

Recommender system provides a user with suggestions about an item. Typical examples include recommendation of books, CDs, and other products at Amazon.com [7], and movies, DVDs, and VHS videos at Movielens [8]. This section presents different RS approaches with their limitations

2.1 Recommender system techniques

A variety of techniques have been proposed as the basis for recommender systems such as: collaborative, content-based, knowledge-based, and demographic techniques.

- **Collaborative:** The type of system makes recommendations byutilizing rating information for different users. Collaborative systems identify candidate users with a rating history similar to the active user and produce recommendations using this neighborhood.
- **Content-based**: The system uses two sources to generated recommendations from two sources; the features of the products as well as the ratings that a user has given them. Content-based recommenders consider recommendation as a user-specific classification problem and use classifier to learn what the user's likes and dislikes based on product features [17]. It recommends items similar to those the user preferred in the past.
- **Demographic:**A demographic recommender provides recommendations based on a demographic attributes of the user. Demographic information is used to detect the types of users that like a certain object. Demographic data is extracted from user profiles and represented as vectors. Then, similarities among those user vectors were calculated for final prediction to be generated [17].

- **Knowledge-based**: A knowledge-based recommender predicts items using delivered knowledge about a user's needs and preferences. This knowledge will contain explicit functional knowledge about how specific product features satisfy user needs [17].
- **Hybrid System:** A hybrid recommender system is one that combines multiple techniques together to accomplish a better performance of filtering, and also take the synergy of each.

2.2 Collaborative filtering recommender systems

As mentioned above, CF systems provide a user with recommendations of items by considering both user's previous likings as well as the preferences of other users who have similar tastes [9].Thus, in CF the user has to rate a number of items. When a new user, having no rating, it would not be able to calculate similarities. Besides, if a user's rating records sparsely, the recommendation performance is very poor. In the following, different systems that apply CF are presented.The work of [10], assign different weights to neighbouring users/items by applying significance weighting schemes. They proposed various heuristics for significance weighting scheme and empirically evaluated them on two different datasets, to show how these schemes affect the performance of a RS. Furthermore, they claimed that the conventional weighted sum prediction formula used in item-based CF is not correct for very sparse datasets. Thus, they provided another prediction formula and empirically evaluated it.Bellogín and Castells in [11], investigated the adaptation of clarity-based query performance predictors to predict neighbour performance in CF. They explored the use of performance prediction techniques to enhance the selection and weighting of neighbours in CF. A predictor is proposed and introduced in a KNN CF algorithm to produce a dynamic variant where neighbour ratings are weighted based on their predicted performance.In [12], Xu et al. formulated the Multiclass Co-Clustering (MCoC) problem and proposed a solution to it. They also proposed a unified framework to extend the traditional CF algorithms by utilizing the subgroups information for improving their top-N recommendation performance. Their approach is considered as an extension of traditional clustering CF models. They examined their approach on three real world data sets to demonstrate the effectiveness of their proposed approach. Experimental results showed that using subgroups is a promising way to further improve the top-N recommendation performance for many popular CF methods.Lu et al. [13] investigated online Collaborative Filtering techniques for building live recommender systems where the CF model can evolve on-the-fly over time. They proposed a new framework of Second Order Online Collaborative Filtering, for building realistic solutions for online recommender systems. They presented the Confidence Weighted Online Collaborative Filtering (CWOCF) method, which attempts to maintain the distributions of both the user and item vectors by updating both the first order and the second order information of the model in the online learning process. In contrast to the existing first order algorithms which only exploit the first order information, their algorithms converge significantly faster and thus achieve much lower values of RMSE and MAE. They conducted extensive experiments on four different data sets of different sizes and sparsity levels, in which the promising results validate the effectiveness of their proposed algorithms. Unlike the above mentioned methods which overcome one of the key problems of applying CF, our proposed is able to adapt itself and mitigate the weakness of CF.

2.3 Demographic Recommender Systems

Demographic Recommender Systems make recommendations based on demographic categorization [14]. They utilize user attributes, classified as demographic data, in order to produce their recommendation. Compared to the other approaches, the advantage of this type of system is that it only uses the demographic data of user such as gender, age, education, etc, and may not need the history of users' ratings, or the textual description or the knowledge about items. Therefore, new users can get recommendations before they rate any items. Many researchers tried to enhance the Collaborative Filtering and overcome cold start problem by using demographic attributes. Vozalis et al. [14] and Ghazanfar et al. [15], proposed an approach enhancing the Collaborative Filtering, which compute the user similarity based on demographic data by computing the cosine similarity. Also Chen and He [16], proposed a Collaborative Filtering Algorithm Based on demographic vector. They generated user demographic vector based on the user information, then calculated two users' similarity. They generated similarity using cosine or Pearson-Correlation. Then, predicted item rates by top N similar neighbours. They applied their experiments on Movielens dataset. The results showed that the quality of recommendations improved, although the new user has no initial ratings. Therefore, we utilize the demographic attributes of users in order to overcome the limitation of CF using weighted scheme.

2.4 Problems of current recommender techniques

Each of above mentioned techniques have known shortcomings, such as the well known cold-start problem for collaborative and content-based systems (what to do with new users with few ratings) and the knowledge engineering bottleneck in knowledge-based approaches [17]. In the following, we will illustrate different problems associated with collaborative filtering (CF) techniques. CF required a large amount of existing data about a user in order to make accurate recommendations. This approach often suffers from three problems: scalability, sparsity and cold start [15].

Scalability: In many of the environments that these systems make recommendations in, there are millions of users and products. Thus, a large amount of computation power is often necessary to calculate recommendations.

Sparsity: The sparsity problem refers to a situation in which transactional or feedback data is sparse and insufficient to identify similarities in consumer interests. Users rate only a small number of items and hence a very sparse user-item matrix is available.

Cold Start: The cold-start problem refers to the situation in which a new user or item has just entered the system. It illustrates the importance of addressing the sparsity problem.

3. AN ADAPTIVE HYBRID MODEL FOR ENHANCED RECOMMENDATION

Traditional approached for CF based recommendations are classified into: K-nearest neighbour, association rules, and Matrix factorization[16]. The most wide spread method of CF is the neighbourhood-based approach (also known as "k Nearest Neighbours" or KNN). KNN method is applied by searching for those users who are most similar to the active user, and analyze their ratings to determine which items may interest to active user. KNN performs computation on the entire database in order to identify the top K most similar users to the active user from the training database. Similarity of the rating patterns between active users and entire database is calculated using either Pearson-Correlation, cosine similarity, or others. However, this method suffers from two fundamental problems: data sparsity and cold start problem which affect the accuracy of recommendation process. As a result, the accuracy of the method is often relatively deficient. Furthermore, for large data set, computational time affect the performance of the recommendation process. Moreover, it is difficult to determine preferences of new user as her rating list is empty.

The proposed model aims to overcome those problems by applying an adaptable linear combination using different recommendation techniques to enhance the accuracy of recommendation and via Triangular Matrix in order to speed up the calculation of similarity.

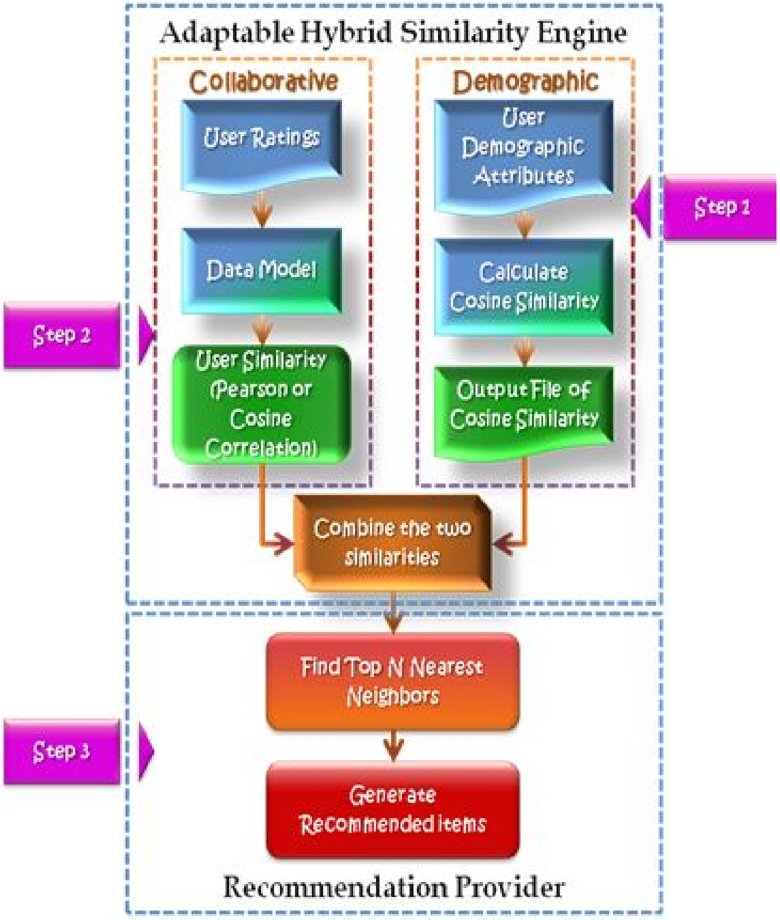
Adaptable Hybrid Similarity Engine
Collaborative
User Ratings
Data Model
User Similarity (Pearson or Cosine Correlation)
Demographic
User Demographic Attributes
Calculate Cosine Similarity
Output File of Cosine Similarity
Step 1
Step 2
Combine the two similarities
Find Top N Nearest Neighbors
Generate Recommended items
Step 3
Recommendation Provider

Figure1: Framework for adaptive Hybrid Recommender System

The main phases of the proposed hybrid recommendation model are:

- Generate new similarity by combining demographic with User-Based CF similarity.
- Get top N Nearest Neighbours using the new combined similarity.
- Generate top M recommendations based on top N similar neighbours to the active user.

The first two phases are applied using adaptable RS engine while the third step is accomplished through recommendation provider module.

3.1 Adaptable Hybrid Similarity Engine

The most crucial step in applying collaborative filtering technique is to calculate similarity between users in order to generate a list of Nearest Neighbours as recommendation partners for an active user. As illustrated in figure 1, the proposed hybrid recommender engine combines similarity calculated by two different recommendation techniques using a linear formula. There are several examples of weighted hybrid recommenders [17] which combines evidence from both recommenders in a static manner. Unlike such approaches, the proposed hybrid similarity engines dynamically applies adaptable weighted equation to linearly combine similarity computed using both demographic similarity and Pearson or Cosine similarity. The general equation form (Eq.1.) is used to calculate the overall similarity between an active user and her/his neighbor:

$$CombinedSim = \propto \cdot\, DFSim + (1 - \propto) \cdot CFSim \qquad \text{Equ(1)}$$

where:

α: Real value varies from 0 to 1
CF: Collaborative filtering similarity
DF: Demographic similarity

The value of α is chosen based on the current input status. For example, for new user, since collaborative similarity will not generate the set of Nearest Neighbours. Accordingly, when the engine detects a new user α is set to 1 and thus demographic similarity is used to calculate the Nearest Neighbours. On the other hand, since quality of the recommendations of a CF improves by increasing the number neighbours for sparsity dataset;different values for α with appropriate CF technique (either Pearson or cosine) have been tried in order to get the best result. Mean Absolute Error (MAE) and precision has been used for measuring the improvement of the quality of recommendation and thus select the value of α. Thus, the engine works as follow: it start by identifying the type of target user (either new, low rated data, or high rated), calculate both demographic and CF (if needed) similarity, and hence apply the appropriate value for α.

3.1.1 Calculation of Demographic Similarity

Demographic information about users or items could be used in prediction. This approach is based on an assumption: If two people are similar in the age, occupation, gender and other physical attributes, then they may both have a common interest in the same item. In this case, user's similarity is calculated without considering their rating. In MovieLens data set, demographic attributes are: gender, age, and occupation. A user demographic vector *usdemog*, is generated for each user which decomposes 3 fields each representing user demographic features. Nominal values are replaced by numeric values as indicated by Table1. Demographic Similarity phase consists of the following steps:

1) Generates user demographic vector based on the user’s information (gender, age, occupation).
2) Calculate the demographic similarity between target users and others using cosine similarity.
3) Identify the top similar users

Table 1. Structure of the User Demographic Vector encoding, $usdemog$

Feature	Encode	Feature Contents	Comments
Gender	0	Male	describing the user gender
	1	Female	
Age	0	1-9	each user belongs to a single age grouping
	1	10-19	
	2	20-29	
	3	30-39	
	…	…	
Occupation	1	Teacher	each user has a specific occupation
	2	Doctor	
	…	…	

3.1.2 Calculation of Collaborative Filtering Similarity

CF method identifies set of like-minded users who tend to rate similarly to an active user as her/his neighbours. Then, calculate the similarity between the active user to each of the users in the candidate neighbours set and get top K users. In this paper, we will use some of the traditional similarity techniques such as Pearson Correlation and Cosine. Pearson Correlation (PC) is a well-known metric that compares ratings where the effects of mean and variance have been eliminated is the Pearson Correlation (PC) similarity [18]:

$$PC(u,v) = \frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{v,i}-r_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)^2}\sum_{i\in I_{uv}}(r_{v,i}-\bar{r}_v)^2}(2)$$

Also, for acquiring the similarity between two items i and j the ratings given by users that have rated both of these items is compared:

$$PC(i,j) = \frac{\sum_{u\in U_{ij}}(r_{u,i}-\bar{r}_i)(r_{u,i}-r_j)}{\sqrt{\sum_{u\in U_{ij}}(r_{u,i}-\bar{r}_i)^2}\sum_{u\in U_{ij}}(r_{uj}-\bar{r}_j)^2}(3)$$

While cosine similarity metric is used for estimating the similarity between two objects a and b in information retrieval that the objects are in the shape of two vectors X_a and X_b and calculating the Cosine Vector (CV) (or Vector Space) similarity between these vectors indicate the distance between them[19]:

$$\cos(X_a, X_b) = \frac{X_a \cdot X_b}{\|X_a\|^2 * \|X_b\|^2}(4)$$

In the context of item recommendation, for computing user similarities, this measure can be employed in which a user u indicates vector xu ∈ R|I| where xui = rui if user u has rated item i and for unrated item considers 0. The similarity between two users u and v would then be calculated as:

$$CV(u,v) = \cos(X_u, X_v) = \frac{\sum_{i\in I_{uv}} r_{ui}r_{vi}}{\sqrt{\sum_{i\in I_u} r_{ui}^2}\sqrt{\sum_{j\in I_v} r_{vi}^2}}(5)$$

where, I_{uv}once more indicates the items rated by both u and v. A shortcoming of this measure is that it does not examine the differences in the mean and variance of the ratings made by user's u and v.

3.2 Recommendation provider

The main performance measure of recommendation process is the cost of finding the k-nearest which is calculated by the number of probes needed to find the top ranked neighbours [20]. The main purpose of recommender provider is to apply those steps which illustrated in his section.

3.2.1 Enhancing selection of K-Nearest Neighbor Set (KNN)

In order to select top ranked neighbours, several authors reduce the high dimensionality of the user ratings matrix to a lower-dimensional setting which affects the accuracy of recommendation. Unlike those authors, we applied triangle matrix to speed up the calculation process, and hence to enhance the computational cost used for calculating similarity. The basic idea is simple, since the Sim_{ij} is the same as Sim_{ji}, we calculate only one of them. This method improves complexity and reduces time required to calculate similarities to less than half as we just calculate similarity between target user and $\mathrm{N} * (\mathrm{N} - 1)/2$ instead of calculating $\mathrm{N} * \mathrm{N}$ similarities as shown in Figure2.

user	1	2	3	4
1				
2	🟥			
3	🟥	🟥		
4	🟥	🟥	🟥	

Figure 2. Triangular Matrix

3.2.2 Generate recommendations using enhanced correlation

Having active user's Nearest Neighbour set, we can predict items for active user. Rating of an item k for $\mathrm{user}_{\mathrm{a}}$is calculated by computing the weighed sum of the ratings given similar users to $\mathrm{user}_{\mathrm{a}}$. Formally, using the notion,$\mathrm{pred}(\mathrm{a}, \mathrm{p})$denotes the prediction as:

$$pred(a,p) = \bar{r}_a + \frac{enh\_cor(a,b) \cdot (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} enh\_cor(a,b)} \qquad (6)$$

a, b: users

$r_{b,p}$: rating of user b for item p

N: set of Nearest Neighbors to active user

$\overline{r_a}$, $\overline{r_b}$: user's average ratings

4. EXPERIMENTAL EVALUATION

This section describes the dataset used in experiments and different quality metrics used to measure the performance of experiments, and then discuss the results.

4.1 Dataset

In this paper, MovieLens data set, 100K dataset is used for experimental. This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (gender, age, occupation, zip)

The data set provided by the GroupLens consists of three subordinate data sets: ratings, users, and movies. The rating data set contains users, movies, ratings and timestamps and the other two data sets contain the demographical explanation about users and the explanation of movie genres [21]. In this research, we used both ratings and users' file. The dataset is randomly divided into 80% of the training set and 20% of the test set respectively.

4.2 Performance Measurement

Several metrics have been proposed for measuring the accuracy of CF. In this paper, in order to validate the effectiveness of the proposed model, we use several prediction and recommendation quality measures. Mean Absolute Error (MAE) is used to measure accuracy of recommendations [22], [23]. Precision, recall and F1-measure are also used to evaluate the quality of recommendations of our experiments. Furthermore, performance is measured by comparing time used to calculate the overall experiment with and without using triangle matrix.

4.2.1 Mean Absolute Error (MAE)

MAE measures the average absolute deviation between a recommender system's predicted rating and a real rating assigned by the user. Formally, if n is the number of actual ratings in an item set, then MAE is defined as the average absolute difference between the n pairs. Assume that $p_1, p_2, p_3, \ldots, p_n$ is the prediction of users' ratings, and the corresponding real ratings data set of users is $q_1, q_2, q_3, \ldots, q_n$. See the MAE definition as following:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \quad (7)$$

The lower the MAE, the more precise the predictions would be, resulting for better recommendations to be formulated. MAE has been computed for different prediction algorithms and for different levels of sparsity [24].

4.2.2 Precision, Recall and F1-Measure

Another way to assess the quality of the recommendations made by the system, is to use the metrics of precision, recall and F1-measure [25, 26]. The table below shows the confusion matrix of number of recommendations used to calculate precision and recall (true positive, true negative, false positive and false negative).

Table 2. confusion matrix that accumulates the numbers of true/false positive/negativerecommendations

	Relevant	Irrelevant	Total
Recommended	*tp*	*Fp*	*tp + fp*
Not Recommended	*fn*	*tn*	*fn + tn*
Total	*tp + fn*	*fp + tn*	*N*

Precision: Precision or true positive accuracy is calculated as the ratio of recommended items that are relevant to the total number of recommended items (i.e. measures the ability of the system to suggest content that is truly relevant for users):

$$Precision = \frac{tp}{tp+fp} = \frac{Correctly\ Recommended\ Content}{Total\ Recommended\ Content} \tag{8}$$

where Correctly Recommended Content is the set of relevant recommendations made to the user and Total Recommended Content is the total number of recommendations made to the user.

Recall: Recall or true positive rate is calculated as the ratio of recommended items that are relevant to the total number of relevant items. It measures the system's ability to gather the relevant content to the specific user. (i.e. it measures the amount of contents that fit the user's preferences, recovered by the system).

$$Recall = \frac{tp}{tp+fn} = \frac{Correctly\ Recommended\ Content}{Relevant\ Content} \tag{9}$$

where Correctly Recommended Content is the set of relevant recommendations made to the user and Relevant Content is the set that are interesting according to the user's likings.

F1-Measure: It tries to combine precision and recall into a single score by calculating different types of means of both metrics. The F1-measure or F1-score is calculated as the standard harmonic mean of precision and recall. It corresponds to the Formula:

$$F1 - Measure = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2*Precision*Recall}{Precision+Recall} \tag{10}$$

4.3 Experimental Implementation

In order to show the effectiveness of the proposed model, we compared it with traditional CF and demographic filtering. The main purpose of this experiment is to find out the effect of changing the number of users on the quality of the recommendation as well as the evaluate search cost of neighbours. Thus, we select sample of users according to their ratings (high ratings, low ratings & no ratings). Experiments are applied for each one of those categories (each category contains more than 20 users) and the average is presented. Users in each categoriesare compared with the whole dataset to get the top KNN.

1) ***Exp 1:*** Users with Low Rating (LR).
2) ***Exp 2:*** Users with No Rating (Cold Start Problem)
3) ***Exp 3:*** Users with High Rating (HR).

For each experiment, we applied different techniques for calculating similarities between users:

1- Collaborative Filtering (CF)

a. We applied both Pearson & Cosine to calculate similarity.

2- **Demographic Filtering (DF)**

3- **Weighted Sum which combines DF & CF similarities**

   a. We examined both Pearson (PC) & Cosine (COS)to calculate similarity for CF.

   b. We varied Alpha between (0.1, . . . , 0.9)

For all the above experiments we varied the Nearest Neighbor (NN 10, 100). Although using 10 Nearest Neighbors gives higher accuracy for all experiments, it was significant to repeat all those techniques using NN equal to 100 in order to compare the usage of the proposed model for big and sparse dataset which currently affecting the performance of any recommendation systems. We computed the MAE to measure the accuracy for each method. We also calculated the precision, recall & F-Measure for top 20 users to measure the quality of ranked items. Finally, performance of the proposed model is evaluated by measuring consumed time to accomplish the recommendation process.

4.3.1 Experiment 1: Low Ratings Users

Users with few ratings might face the sparsity problem and accordingly CF would lead to poor results. Therefore, we slightly change the value of α and compute accuracy and quality of recommended items as listed in Table 3.

Table 3. Results of applying different similarity approaches for users having low ratings

Low Rating	**MAE**		**Precision**		**Recall**		**F-measure**	
Exp Name	**NN=10**	**NN=100**	**NN=10**	**NN=100**	**NN=10**	**NN=100**	**NN=10**	**NN=100**
CF_Cos	0.62	0.5	9%	2%	6%	1%	7%	1%
CF_PC	0.65	0.64	13%	2%	8%	1%	10%	2%
DF	0.8	0.75	13%	1%	8%	1%	10%	1%
W(0.1)_Cos	0.74	0.88	9%	1%	6%	1%	7%	1%
W(0.1)_PC	0.64	0.64	12%	2%	7%	1%	8%	2%
W(0.3)_PC	0.65	0.72	11%	3%	6%	2%	8%	2%
W(0.5)_PC	0.8	0.5	11%	2%	6%	1%	7%	1%
W(0.7)_PC	0.85	0.5	9%	2%	5%	1%	6%	1%
W(0.9)_Cos	0.74	0.33	11%	1%	7%	1%	8%	1%
W(0.9)_PC	0.61	0.2	9%	2%	5%	1%	7%	2%

1) Accuracy Evaluation using MAE for high rating users:

According to Table3 and Figure 3, adjusting the value of Alpha to 0.9 with using Pearson Correlation gave the best results for both NN =10 or NN = 100 which guarantee that increasing the effect of demographic attribute in calculating similarity for sparse dataset enhance the accuracy of recommendation.

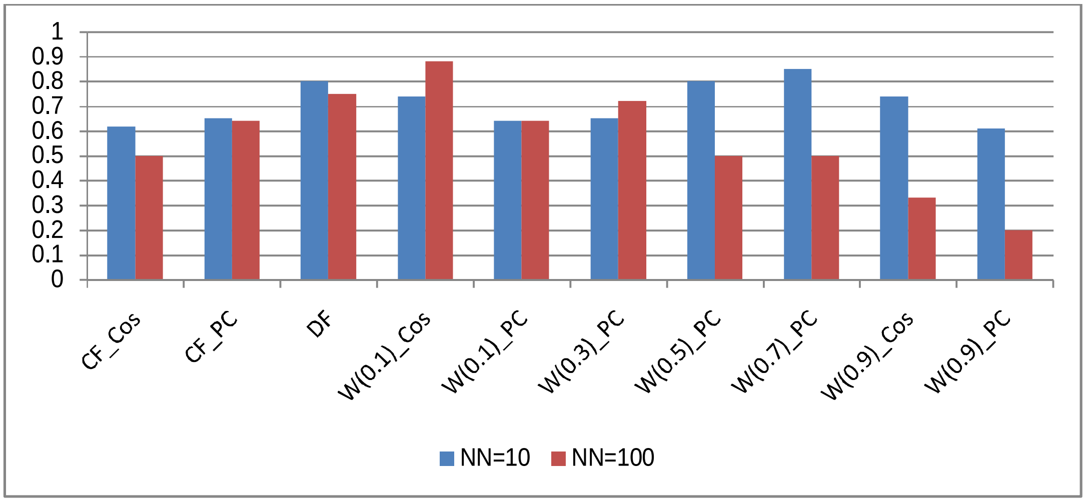

Figure 3. The comparison of MAE for Low Ratings Users

2) Evaluation using Precision, Recall and F1-Measure

Results in table3 and Figure 4 show that best precision was obtained for NN=100 when applying weighted formula with different value of α (0.3, 0.7and 0.9 respectively) which match with our assumption of using dynamic RS model to obtain the best recommendation accuracy especially with large dataset . On the other hand, for NN=10 (small neighbour), traditional techniques (CF and DF)still provide good results.

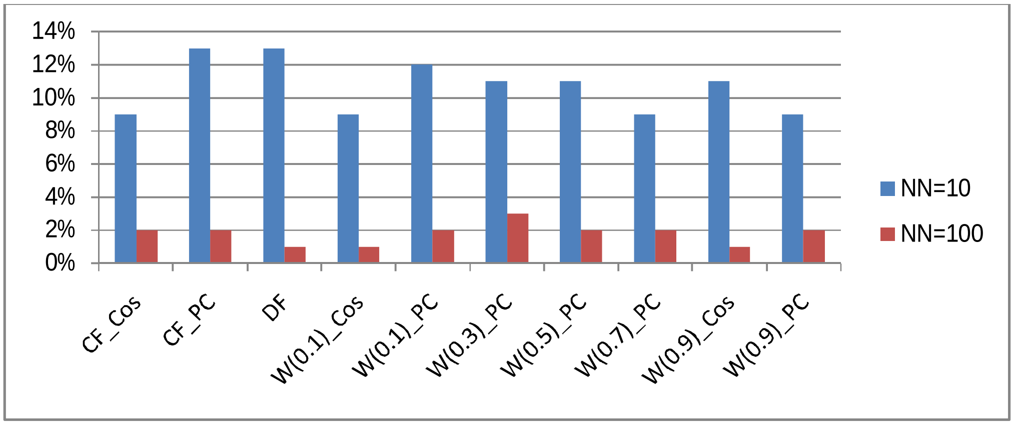

Figure 4. The comparison of the predictive Precision for Low Ratings Experiments

4.3.2 Experiment 2: No Ratings Users

New users face the cold start problem. These users having no ratings make it impossible for Collaborative Filtering to find similar users. In this case when applying equation1, α is set to 1 and only DF similarity is applied as shown in Table 4. Experiment shows that, in case the value of MAE improve with NN = 100 (large scale dataset).

Table 4. Results of applying Demographic Similarity for users with no ratings

NN	MAE	Precision	Recall	F-measure
10	0.73	17%	6%	8%
100	0.55	7%	2%	4%

4.3.3 Experiment 3: High Ratings Users

Table 5. Results of applying different similarity approaches for users having high ratings

High Rating	MAE		Precision		Recall		F-measure	
Exp Name	NN=10	NN=100	NN=10	NN=100	NN=10	NN=100	NN=10	NN=100
CF_Cos	0.90	0.95	14%	25%	2%	3%	3%	5%
CF_PC	0.89	0.77	45%	24%	5%	3%	9%	5%
DF	0.93	0.82	53%	19%	6%	2%	11%	4%
W(0.1)_Cos	0.79	0.86	21%	21%	2%	2%	4%	4%
W(0.1)_PC	0.86	0.91	44%	23%	5%	3%	9%	5%
W(0.3)_PC	0.77	0.77	47%	21%	5%	2%	10%	4%
W(0.5)_Cos	0.76	0.89	32%	22%	4%	2%	7%	4%
W(0.5)_PC	0.79	0.69	48%	21%	5%	2%	10%	4%
W(0.7)_PC	0.81	0.66	51%	19%	6%	2%	10%	4%
W(0.9)_Cos	0.92	0.81	54%	20%	6%	2%	11%	4%
W(0.9)_PC	0.86	0.69	49%	18%	5%	2%	10%	4%

Results for applying different approaches for users with high ratings are given in Table 5 which shows that best accuracy result (MEA) for large dataset (NN=100) is obtained when setting α to 0.7 and 0.9. While best precision is obtained when either applying CF alone or weighted formule that increase the weight of CF (i.e for α = 0.1, .3,0.5 respectively)

1) Accuracy Evaluation using MAE for high rating users:

According to Table5, the value of MAE intensely affected by changing the value of α (it drops to 0,66 for α= 0.7). Thus, in the following experiments, we study the effect of changing set of neighbours on both MEA and precision usingα = 0.9.

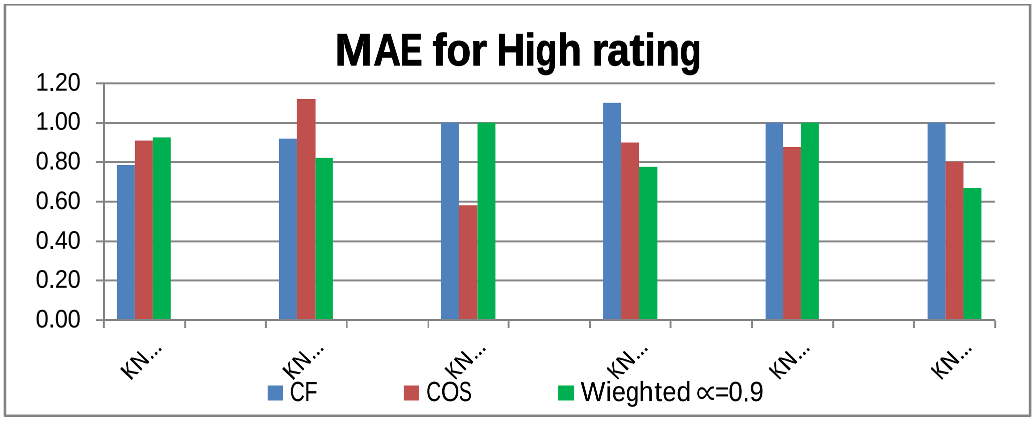

Figure 5. The comparison of MAE for High Ratings Experiments

As shown in Figure5, the value of MEA slightly change with increasing the neighbours size which confirm that we can set value of α to 0.9 and obtain the optimum accuracy with increasing the set of neighbours.

2) Evaluation using Precision

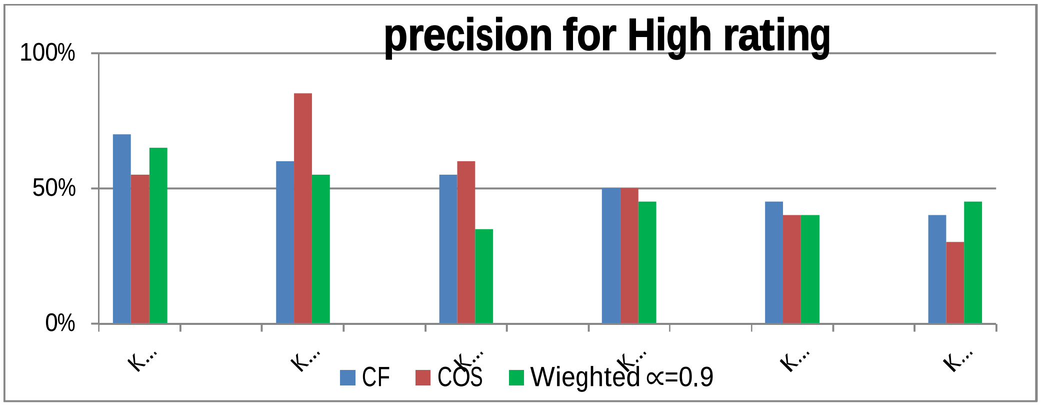

Figure 6. The comparison of the predictive Precision for High Ratings

ExperimentsResults in Figure 6 show that best precision was obtained when using α 0.9 when NN =100 large data size

4.4 Performance Evaluation (Time complexity)

According to the fact that the Sim_{ij} is the same as Sim_{ji}, similarities among users is calculated using triangular matrix. This resulted in decreasing runtime and enhanced space complexity. Furthermore, we tried our model using hash tables and all the multiples for different number of users between 10 and 50 with 100 Nearest Neighbours. Then, we calculate the total runtime used to generate recommendations which enhanced the time complexity to a remarkable improvement as shown in Figure 7.

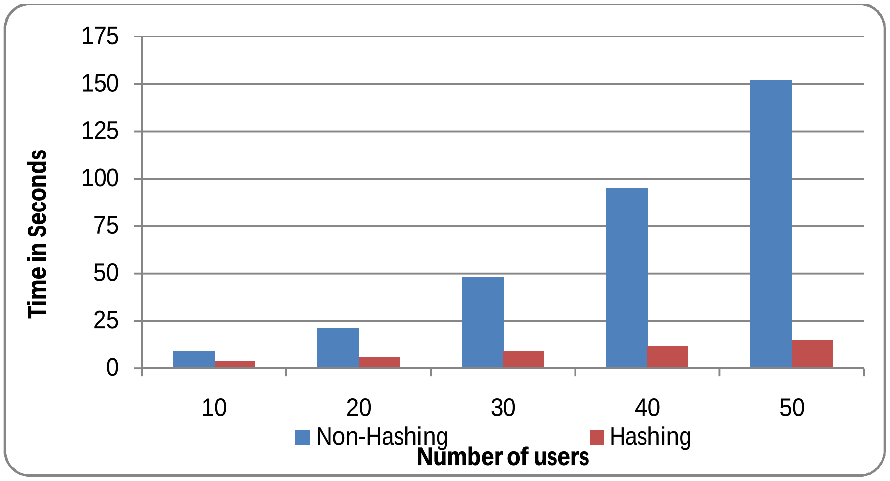

Figure 7. The comparison of using Hashing and Non-Hashing lookup into triangular array of demographic similarity when KNN=100 and different number of users in MovieLens dataset

According to Figure 8 using hashing has improved the time complexity up to 95.7% for the first 100 users and KNN=100.

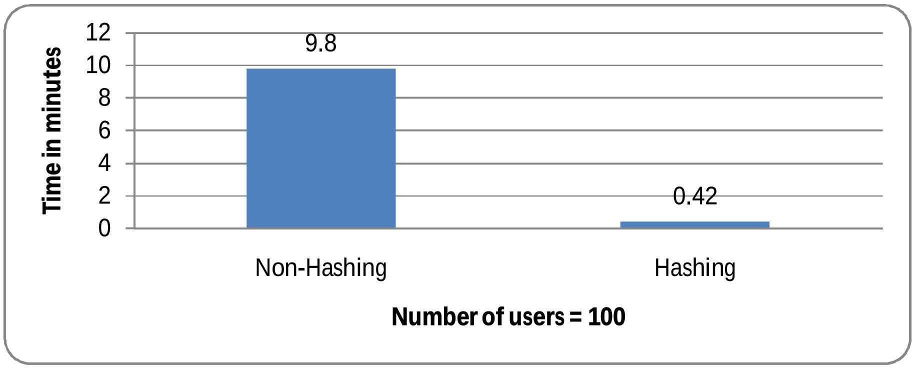

Figure 8. The comparison of using Hashing and Non-Hashing lookup into triangular array of demographic similarity for first 100 users when KNN=100 in MovieLens dataset

5. CONCLUSION & FUTURE WORK

Recommender systems have attracted a lot of attention in the past decade due to its vital role as an information filtering technique. One of the most crucial techniques in RS is Collaborative Filtering which depends on the fullness of the users' rating. Such that the more information users provide, the more accurate the recommendation will be. On the other hand, demographic attributes which is obtained from user profiles during registration were heavily used to overcome the cold start problem. Current work in applying RS in different filed showed that there's no approach suitable for all users. Furthermore, for web 3.0 where floods of information are there due to big data, RS will suffer from sparsity and low performance. This paper presented an adaptive hybrid recommendation framework that applies a weighted equation to linearly combine similarity computed using both demographic similarity and Pearson or Cosine similarity. The proposed adaptive model is generally applicable to any big dataset as it adjusts the linear formula a according to the status of target user. Experimental result applied on MovieLens shows that , the proposed model improves the recommendation accuracy and was able to overcome the sparsity and cold start problems that face traditional CF. Furthermore, it enhances performance of recommendation process(time and space) by using triangular matrix to calculate similarity between users. In the future, the proposed model could be improved by calculating similarity among users by constructing a user profile which is extracted from their behaviour and context information. Furthermore, we ca consider other implicit knowledge which is obtained from different social networks. Moreover, the proposed model could be adopted and applied to helps a user to select the best services from different Cloud.

References

[1] M. Vozalis and K. G. Margaritis, "Collaborative Filtering Enhanced by Demographic Correlation," in Proceeding of the AIAI Symposium on Professional Practice in AI, part of the 18th World Computer Congress, Toulouse, France, 2004.

[2] F. Ortega, J.-L. Sánchez, J. Bobadilla, and A. Gutiérrez, "Improving collaborative filtering-based recommender systems results using Pareto dominance," vol. 239, 2013.

[3] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, pp. 734–749, 2005.

[4] D. H. John S. Breese, and Carl Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," presented at the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item based collaborative filtering recommendation algorithms," in Proceeding of the 10th international conference on World Wide Web, ACM New York, NY, USA, 2001, pp. 285–295.

[6] D. Maltz and E. Ehrlich, "Pointing the way: Active collaborative filtering," in Proceeding of SIGCHI conference on Human factors in computing systems (CHI'95), Denver, Colorado, USA, 1995, pp. 202-209.

[7] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, pp. 76-80, 2003.

[8] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender Systems," in Proceedings of Intelligent User Interfaces, 2003, pp. 263-266.

[9] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," presented at the The Adaptive Web of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 2007.

[10] M. A. Ghazanfar and A. Prugel-Bennett, "Novel Significance Weighting Schemes for Collaborative Filtering: Generating Improved Recommendations in Sparse Environments," presented at the DMIN'10, the 2010 International Conference on Data Mining, 2010.

[11] A. Bellogín and P. Castells, "A Performance Prediction Approach to Enhance Collaborative Filtering Performance " presented at the ECIR, 2010.

[12] B. Xu, J. Bu, C. Chen, and D. Cai, "An Exploration of Improving Collaborative Recommender Systems via User-Item Subgroups," presented at the International World Wide Web Conference Committee, Lyon, France, 2012.

[13] J. Lu, S. Hoi, J. Wang, and P. Zhao, "Second Order Online Collaborative Filtering," in JMLR: Workshop and Conference Proceedings, 2013, pp. 325-340.

[14] M.J. Pazzani, "A Framework for Collaborative, Content-based and Demographic Filtering," Artificial Intelligence Review, vol. 13, pp. 393-408, 1999.

[15] S. Lee, J. Yang, and S.-Y. Park, "Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem," Discovery Science, 2007.

[16] R. M. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," in ICDM '07 Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, IEEE Computer Society Washington, DC, USA, 2007, pp. 43-52.

[17] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in ACM SIGIR Workshop on Recommender Systems: Implementation and Evaluation, Berkeley, CA, 1999.

[18] M. Nilashi, K. Bagherifard, O. Ibrahim, H. Alizadeh, L. A. Nojeem, and N. Roozegar, "Collaborative Filtering Recommender Systems," Research Journal of Applied Sciences, Engineering and Technology, vol. 5, pp. 4168-4182, 2013.

[19] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," User Mod. User-adapted Interac., vol. 10, pp. 147-180, 2000.

[20] A. M. Roumani and D. B. Skillicorn, "Finding the Positive Nearest-Neighbor in Recommender Systems," in Proceedings of the 2007 International Conference on Data Mining, Las Vegas, Nevada, USA, 2007, pp. 190-196.

[21] "MovieLens Dataset," ed. Research lab in Department of Computer Science and Engineering at the University of Minnesota, Twin Cities.

[22] H. qin-hua and O. wei-min, "Fuzzy collaborative filtering with multiple agents," Journal of Shanghai University (English Edition), vol. 11, pp. 290-295, 2007.

[23] S. Gong and C. Huang, "Employing Fuzzy Clustering to Alleviate the Sparsity Issue in Collaborative Filtering Recommendation Algorithms," in Proceeding of 2008 International Pre-Olympic Congress on Computer Science, World AcademicPress, 2008, pp. 449-454.

[24] S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," Journal of Software, vol. 5, july 2010.

[25] Carrer-Neto, W. Hernández-Alcaraz, M. Luisa, Valencia-García, R. García-Sánchez, and Francisco, "Social knowledge-based recommender system. Application to the movies domain," Expert Systems with Applications, vol. 39, 2012.

[26] D. M. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2, pp. 37-63, 2011.