An Approach for Recommender System by Combining Collaborative Filtering with User Demographics and Items Genres

Saurabh Kumar Tiwari
Department of Information Technology
Samrat Ashok Technological Institute
Vidisha, M.P., India

Shailendra Kumar Shrivastava, PhD
Department of Information Technology
Samrat Ashok Technological Institute
Vidisha, M.P., India

ABSTRACT
With the explosion of service based web application like online news, shopping, bidding, libraries great amount of information is available. Due to this information overload problem, to find right thing is a tedious task for the user. A recommender system can be used to suggest customized information according to user preferences

Collaborative filtering techniques play a vital role in designing the recommendation systems. The collaborative filtering technique based recommender system may suffer with cold start problem i.e. new user problem and new item problem and scalability issues. Traditional K-Nearest Neighbor Technique also suffers with user and item cold start problem.In this paper recommender system generates suggestions for user by combining collaborating filtering on transaction data with rating predicted with user demographics and item similarity. The final rating is weighted sum of ratings computed from transaction data, user data and item data. The advantage of proposed system that recommender system can deal with cold start in case of "new user" or “new item” .and Also system has low MAE and RMSE in comparison of traditional collaborative filtering based on K-Nearest Neighbor approach.

Keywords
Recommendation System, Collaborative Filtering, Cold start, demographic filtering, K-Nearest Neighbor Method.

1. INTRODUCTION
With the dramatically fast and explosive growth of knowledge on the market over the Internet, World Wide Web has become a robust platform to store, spread and retrieve data likewise as mine helpful data. As a result of the properties of the large, diverse, dynamic and unstructured nature of web data, web data analysis has encountered lots of challenges, like scalability, multimedia system and temporal problems etc. Due to this large amount of information finding interesting information is a tedious and time spending task for the user. The scope of ‘Recommendation system’ here comes into light. Recommendation Systems are software tools and techniques that deal with information overload by providing interesting suggestions and recommendations to users [9]. These recommendations help users to make decision as which item to buy or which music to listen or which online news to read. Recommendation Systems are primarily focused on type of items like ‘book recommendation system’ or ‘music recommendation system’ etc.

In its commonest formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that haven't been seen by a user. Intuitively, this estimation is typically based on the ratings given by this user to other items. The recommendations provided by RS may be personalized or non-personalized. The first, personalized RS may provide different recommendation for a diverse set of users according to their interests respectively. The other variant, non-personalized RS will give a similar set of recommendations to different users like ‘Top 10 books’ or ‘Top 10 songs’ etc. Second, non-personalized RS are simple and recommendations are easy to generate. They may observe in online magazines or in online newsreaders. These styles of non-personalized recommendations don't seem to be usually addressed by RS analysis [9]. Recommender systems work as information processing systems which gather mostly of three categories user data, items data and transaction involving users and items with preferences.

User may have different characteristics and preferences. In order to generate personalized recommendations RSs uses a diverse range of user information. User information can be modeled in many ways and the selection depends on technique of recommendation. User can also be narrated with their online behavior or navigational patterns. Item refers to the object which will be recommended to user. News, music, books, movies etc all are items in context of recommendation system. Item can be described with attributes associated with it. For example a movie can be described with movie name, director, genre, cast etc. Transactions are the recorded interaction between user and item. Transaction mostly is tabular data that record important information during human computer interaction. Transaction may contain user feedback explicitly. Also user tastes can be understood by looking into the transactions. User preferences are measured in terms of online navigational patterns or ratings provided by the user.

Data for recommendation system may be implicit or explicit. Implicit data are recorded from user click streams, hyperlink navigation while explicit data is found in form of ratings or feedback provided by user for an item. Recommendation systems embrace processes that are conducted for the most part by hand, like manually making cross-sell lists and actions that are performed for the most part by PC, like collaborative filtering. The latter referred as automatic recommendation systems. Automatic recommendation systems are specialized data processing systems that are optimized for interaction with customers instead of marketers. They need been explicitly designed to require advantage of the real-time personalization opportunities web based services, accordingly, the algorithms focus additional on real-time and just-in-time learning than on model-building and execution [6].

In Collaborative filtering based RS the user will be suggested

items that people with similar interested and preferences liked in the past. In a CF recommendation application, in order to suggest items to user, the collaborative filtering recommendation system looks for the "peers" of user, i.e., set of users that have similar interest in item. Then, only the items that are most liked by the "peers" of user would be suggested [1].

In demographic filtering RS, it is assumed that the users with common demographics will also have same tastes and preferences [2]. Many websites adopt simple and effective personalization solutions supported demographics. For example, users are dispatched to specific websites supported their language or country. Or suggestions may be customized according to the profession or age of the user [9].The other type is Hybrid filtering, in which RS generates recommendations combining features of different filtering techniques. Most common combinations are Collaborative filtering with Content based or Collaborative filtering with demographic filtering [1, 2].

Widely accepted taxonomy classifies recommendation methods into 'Memory based approach' and 'Model based approach'. Memory-based methods usually use similarity metrics to obtain the distance between two users, or two items, based on each of their attributes. Model based use RS information to create a model that generates the recommendations [1]. Memory-based algorithms use the full table to calculate their prediction. They use similarity measures to choose users or items that are the same as the active user. Now, the prediction is calculated from the ratings of those similar users or items. Most of those algorithms will be classified as user-based algorithms or item-based algorithms depending on whether the process of getting neighbors is concentrated on finding similar users [5]. In Model based approach ,the design and development of models using machine learning, data mining algorithms can enable the system to learn to recognize complicated patterns based on the learning information, then generate intelligent predictions for test data or real-world information, based on the learned models[4]. Model based algorithms use the collections of ratings to learn a model and this model is employed for generating rating predictions [1].

The organization of the paper is as follows: In Section 2, related work is briefly discussed. In 3rd section, the proposed system is elaborated, which combines item-based collaborative filtering with user clusters based on demographics and genre based item similarity in a hybrid approach. In Section 4, the performance of the proposed system is discussed to show how it achieves a reduced MAE and successfully solving the cold start problem. In 5th section, the conclusion of paper is presented

2. RELATED WORK

In collaborative filtering based recommendation system, system generates ratings for the active user based on the rating given by the recommender system users who are much similar to active user. If two users rate an item similarly, users are considered similar in the recommendation system [2]. The easiest and original implementation of this approach [23] suggests to the active user the items that different users with same preferences within the past. The similarity in style of two users is calculated supported the similarity within the rating history of the users. this can be the rationale why [6] refers to collaborative filtering as "collaboration among users of recommender system." collaborative filtering is taken into account to be the foremost common and wide enforced technique in RS.

In Table 1, first we have to estimate the potential favorable opinion of Steve about Harry potter, one can use the similarity of her with those of John.

Table 1: Recommendation Process in Nutshell

Person / Movie	John	Boby	Steve
Titanic	5	1	5
The Reader	1	5	2
Harry Potter	4	2	?

Alternatively, one can note that ratings of Titanic and Harry potter follow a same pattern, which shows that people who liked the former might also like the later [17]. An example given in Table 1 will give brief knowledge about collaborative filtering.

Collaborative filtering techniques mostly relied upon K nearest neighbors methods to predict recommendations for the user. In Article recommender by GroupLens it was first introduced. There are two version used of k Nearest Neighbor approach in collaborative filtering

2.1 User-based collaborative filtering

In this collaborative filtering approach, recommendation items are predicted on the basis of finding recommendation system users with same item preferences to the active user.The methodology can be illustrated in following three steps [16]:

1. By using a particular similarity measure recommendation system produce a set of similar users to the active user 'u'. The selected K users are the K closest (similar) neighbor to active user 'u'.
2. Once k close neighbors are found to active user 'u', predictions are generated for item 'i' by using any one of following aggregation approach, the average, weighted sum, and the weighted adjusted aggregation.
3. To have top n recommendation, n items will be chosen from the similar items that are close neighbor of active user.

User based collaborative filtering suffers with scalability problem.

2.2 Item-based collaborative filtering

As the number of users increases User to user based kNN suffers from scalability problem. To overcome this drawback new method called item to item K-NN is introduced by Sarwar et al. [17] and Karypis. The item-based approach investigates the set of items rated by target user and calculates their similarity with the target item i and then chooses k most similar items$i1, i2,\dots ik$.
Their representing similarities $t\ i1,\ t\ i2\ \dots\ t\ ik$ are also computed at the same time. Formerly the most similar items are discovered, after that by taking a weighted mean of the target user's ratings on these similar items the prediction is calculated. Similarity computation and

the prediction generation are two important factors which make item-based recommendation more powerful. For similarity computation basically different types of similarity measures are used and weighted sum and regression used for prediction computation.
Collaborating filtering based recommendation system also faces some issues [1, 2].

a) **New User Problem**
It is identical drawback like content-based systems. in order to generate correct recommendations, the system should initially learn the user's preferences from the ratings that the user offers. Many techniques are projected to deal with this drawback. Most of them use the hybrid recommendation approach, which mixes content-based and collaborative techniques.

b) **New Item Problem**
New items are added frequently to recommender systems. Collaborative systems trust only on user's preferences to generate recommendations. Therefore, till the new item is rated by a considerable range of users, the recommender system would not be able to recommend it. This downside may also be addressed exploitation hybrid recommendation approaches, represented in the next section.

c) **Sparsity**
In any recommender system, the amount of ratings already obtained is typically very little compared to the amount of ratings that require to be expected. Effective prediction of ratings from a little range of examples is very important. Also, the success of the collaborative recommender system depends on the provision of an important mass of users. for instance, in the movie recommendation system, there could also be several movies that are rated by solely few individuals and these movies would be recommended terribly seldom, although those few users gave high ratings to them. Also, for the user whose tastes are uncommon compared to the remainder of the population, there will not be the other users who are significantly similar, resulting in poor recommendations [9].

2.3 Similarity Measure:

Memory-based CF algorithms check for the complete or a sample of the user-item data to create a prediction. Every user is a part of a group of people with similar interests. By identifying the supposed neighbors of a current user (or active user) a prediction of tastes on new items for him or her are going to be generated. The neighborhood-based collaborative Filtering rule, a current memory-based CF rule, uses the following steps:

1. calculate the similarity or weight: calculate the similarity or weight, w_{ij}, that reflects distance, correlation, or weight, between two users or 2 items, i and j;
2. Generate a prediction for the active user by taking the weighted average of all the ratings of the user or item on a definite item or user, or employing an easy weighted average [17].

When the task is to build a top-N recommendation, we want to search out k most similar users or items (nearest neighbors) once computing the similarities, so aggregate the neighbors to urge the top-N most frequent items as the recommendation.

Similarity computation between items or users could be a essential step in memory-based collaborative filtering algorithms. For item-based CF algorithms, the essential plan of the similarity computation between item i and item j is initial to figure on the users who have rated each of those items so to apply a similarity computation to work out the similarity, w_{ij}, between the two co-rated items of the users [4]. For a user-based CF algorithmic rule, we tend to initial calculate the similarity, w_{uv}, between the users u and v who have each rated a similar items. There are many various ways to work out similarity or weight between users or items.

2.3.1 Correlation-Based Similarity

In this case, similarity w_{uv} between two users' u and v, or w_{ij} between two items i and j, is computed by computing the Pearson correlation or different correlation-based similarities. Pearson correlation measures the extent to that two variables linearly relate with one another [4]. For the user based algorithmic rule, the Pearson correlation between user u and v is

$$w_{uv} = \frac{\sum_{i \in I} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{vi} - \bar{r}_v)^2}} \quad (2)$$

Where i ∈I summations are over the items that both the users u and v have rated and $\bar{r}_u$ is the average rating of the co-rated items of the u[th] user.
For the item-based algorithm, denote the set of users' u∈U who rated both items i and j, then the Pearson Correlation will be

$$w_{ij} = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{uj} - \bar{r}_j)^2}} \quad (3)$$

Where r_{ui} is the rating of user u on item i, $\bar{r}_i$ is the average rating of the i[th] item by those users.

2.3.2 Vector Cosine-Based Similarity

The similarities between two documents are often measured by treating every document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors [39]. This formalism may be adopted in collaborative filtering, that uses users or items rather than documents and ratings rather than word frequencies. Formally, if R is that the m×n user-item matrix, then the similarity between 2 items, i and j, is outlined as the **cos** of the n-dimensional vectors cherish the ith and jth column of matrix R. Vector cosine similarity between items i and j is given by

$$\boldsymbol{w_{ij}} = \cos(\vec{\boldsymbol{i}}, \vec{\boldsymbol{j}}) = \frac{\vec{\boldsymbol{i}} \bullet \vec{\boldsymbol{j}}}{\| \vec{\boldsymbol{i}} \| * \| \vec{\boldsymbol{j}} \|} \quad (4)$$

Where "•" denotes the dot-product of the two vectors. To get the desired similarity computation, for n items, an n×n similarity matrix is computed [4]. For example, if the vector $\vec{A} = \{x_1, y_1\}$, vector $\vec{B} = \{x_2, y_2\}$, the vector cosine similarity between $\vec{A}$ and $\vec{B}$ is

$$w_{ij} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\| \vec{i} \| * \| \vec{j} \|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{(x_1^2 + y_1^2)} \sqrt{(x_2^2 + y_2^2)}} \quad (5)$$

2.3.3 Adjusted Cosine Similarity

Adjusted cosine similarity is also a similarity measure which

is used in collaborative filtering based recommender system. It is used in the case in which difference in every user's use of rating scale is considered [8]

$$s(i,j) = \frac{\sum_{u \in U'} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U'} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U'} (r_{u,j} - \bar{r}_u)^2}} \quad (6)$$

Where U' is referred to the set of users who had given ratings to both item i and j and the average rating of user u is $\bar{r}_u$.

In an approach, item based method is used to alleviate sparsity and user clusters are formed to achieve high scalability. It also combines item based and user based collaborative by providing a weighted average of predictions [38]. However, these algorithms do not give solution for the cold start problem.

To solve the problems of scalability and sparsity in the collaborative filtering, an approach is given in [39] in which personalized recommendation methods joins the user cluster and item cluster. Ratings given by users on items are used for user cluster, and each users cluster is given by a cluster centre. User's neighbor is calculable by computing similarity between active user and cluster centres. Then, the given approach employs the item based collaborative filtering based on clustering to generate the recommendations. This offers scalable and correct recommendation then traditional approach by providing recommendation combining user cluster and item cluster based collaborative filtering.

In recent times numerous enhancements to traditional approach of collaborative filtering are proposed including change in user's preferences with reference to time [40] tackling the sparsity and scalability problem, trust on users, and evolution of hybrid recommender system.

To improve the prediction quality of item-based collaborative filtering, some algorithms take the attributes of items into consideration while predicting the preference of a user [41].

There is an attempt to cope with Item cold start using a hybrid method which first clusters items using the rating matrix and then uses the clustering results to build a decision tree to combine novel items with existing ones [42].

Collaborative filtering, content based filtering and demographic filtering have been combined to solve the cold start problem [43]. However, they do not address the scalability problems of user based collaborative filtering.

Clustering of users has been used to solve the scalability problem of user based algorithms. In one approach, a cascaded hybrid model first clusters users based on demographic data and then applies user based collaborative filtering to each cluster [44].

In [45] a metric is given to estimate similarity between users is given, which can be applied in collaborative filtering technique in recommender systems. The metric is formulated by the use of a linear combination of values and weights. Values are computed for every pair of users for which the similarity is calculated, at the same time as weights are computed just once ,using a preceding step in which a genetic algorithm extracts weighs from the recommender system which depends on the precise nature of the data from every recommender system. This results in significant improvements in quality of prediction and recommendation and performance.

In [46], a hybrid algorithm is proposed by combining the ratings and content data to overcome item cold-start problem. In this approach initially items are clustered based on the rating matrix and clustering results and item content data are utilized to create a decision tree to associate the prominent items with the existing items. Considering the constantly increasing ratings on novel item, there is a tendency to present predictions of this method can be associated with the traditional collaborative-filtering strategies to meet with higher performance with a coefficient. Tests performed on data set show the development of recommender approach in handling the item side cold-start problem. In many real recommender systems, great portion of items are new items and recommending new items to consumers is a key success for online enterprisers. A hybrid approach is developed which exploits not only ratings space but also attributes of items for item cold-start recommendation.

In [47] an enhanced collaborative filtering recommendation algorithm is proposed based on dynamic item clustering method. Item space is divided into clusters dynamically by introducing a similitude threshold model. They stated with experiments that by employing dynamic item clustering method recommender system can convince the requirement of increasing amount of users and consumers in huge e-business systems. The stated collaborative Filtering recommendation algorithm works comparatively good in providing recommendation with minimize resource consumption.

Hybrid approaches have also been proposed to improve the accuracy of predictions. Adaptive weighted prediction has been used to calculate final ratings from user-based and item-based approaches [48].The method in [30] uses Pareto dominance to carry out a pre-filtering process to eliminate less delegate users from the k-nearest neighbor selection procedure while retain the most promising ones. The computations from the MovieLens and Netflix websites show vital improvement in quality measures.

In [3] K-Nearest-Neighbor (KNN) classification is employed to be used on-line to spot clients/visitors click stream knowledge, matching it to a specific user group and advocate a tailored browsing choice that meet the necessity of the precise user at a selected time. They stated that the K-NN classifier is clear, consistent, simple, easy to know, high affinity to have desirable qualities and straightforward to implement than most alternative machine learning algorithms specifically once there is very little or no previous information regarding data distribution.In [8], item ratings from item based collaborative filtering recommender techniques are associated with ratings computed from user clusters based on demographics in a weighted manner. The stated solution is scalable and successfully overcome user based cold start. At performance front proposed recommender system generate recommendation with comparatively reduced MAE and better coverage to nearest neighbor based collaborative filtering recommenders. However, item clod start is not addressed.

3. PROPOSED SYSTEM

Collaborative filtering based recommender system suffers from scalability, sparsity and cold start situations like new user occurs or new item occurs. These problems have been discussed above.

The proposed system is shown in figure 1.In recommender system there are three sources of data exist-

Transaction data- contains ratings for items provided by users.

User Demographics-like age, gender, occupation or location

Item genres-item may belong to one or more genres e.g. a movie may belong to action, comedy genres.

In proposed system all three types of data are used. For a user-item pair ratings are estimated from all three types of data and final rating is computed as weighted sum of three ratings.

Working of system is discussed in following sections-

3.1 Rating Prediction from Transaction Data

Rating from transaction data are predicted using K-Nearest neighbor classification technique. K-nearest neighbor [11] is mostly used algorithm for collaborative filtering based techniques.

Here K denotes the number of neighbors. Its primary virtues are simplicity and reasonably accurate results.

In the item to item version [2] of the kNN algorithm, the following two tasks are executed:

1. Determine k items neighbors for each item in the database;
2. for each item i not rated by the active user a, calculate its prediction based on the ratings of a from the k neighbors of i

K-nearest neighbor algorithm now provides rating for active user a based on transaction data. 'r_t' shows the rating estimated from K-nearest neighbor classification performed on transaction data

3.2 Rating Prediction from User Demographics

Users are partitioned in different cluster using K-means clustering algorithm [11] by using user's demographics.

Ratings are computed in following steps:

1. First similarity of active user's demographics is calculated from all clusters. Here Pearson correlation is used as similarity measure. Maximum similarity value decides the cluster for active user.
2. Rating for active user is given by multiplying similarity measure with average rating of cluster in which active user lies.

'r_u' is rating estimated by user clustering based on user demographics.

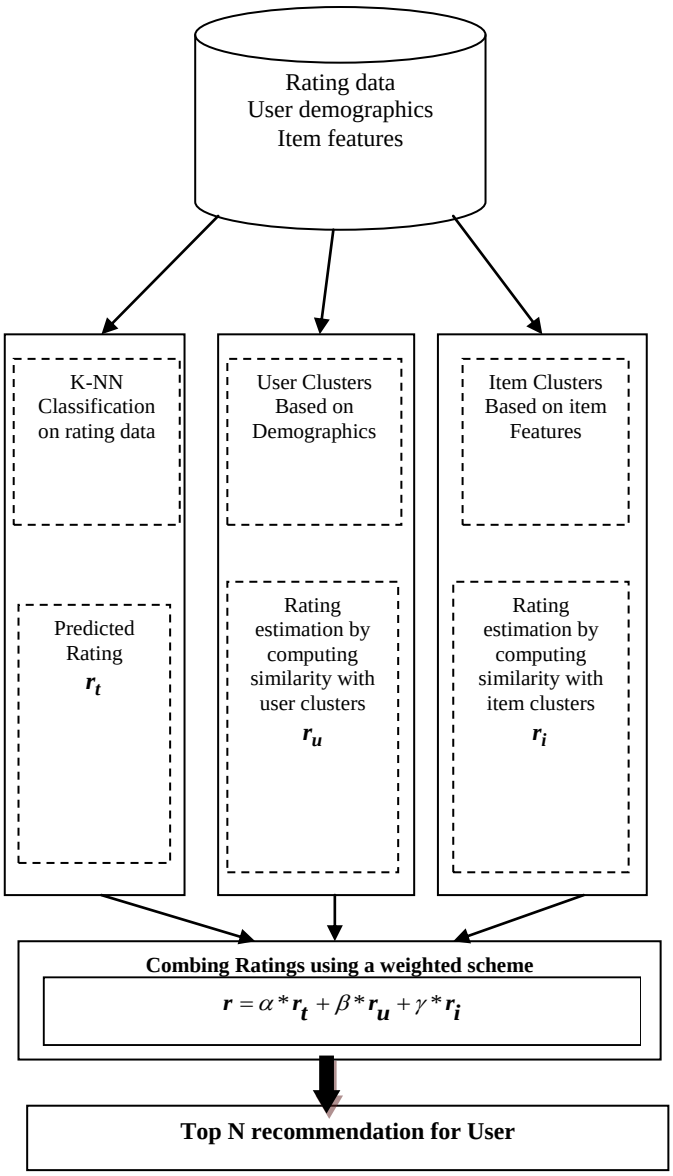

Figure 1: Proposed System Architecture

.

3.3 Rating Prediction from item genres

Also for item K-means partitioned based clustering algorithm is performed. Items are clustered by using their genres (for movie, music, books). Other features of items can also be used for item clustering.

Ratings are computed in following steps:

1. First similarity of the item, on which rating is to be predicted, is calculated from all clusters. Here again Pearson correlation is used as similarity measure. Maximum similarity value decides the cluster for current item.
2. Rating for item is given by multiplying similarity measure with average rating of items in the cluster to which current item is most similar.

'r_i' is rating estimated by item clustering based on item genres.

3.4 Combining Ratings

The resultant rating is estimated as weighted sum of three ratings ' r_t ', ' r_u ' and ' r_i ' in following manner

$$r = \alpha * r_t + \beta * r_u + \gamma * r_i \qquad (7)$$

Here

'r' is predicted rating.

' r_t ' is rating predicted using classification of transactions.

' r_u ' is calculated rating based on user similarity.

' r_i ' is calculated rating based on item similarity.

' α ', ' β ', ' γ ' are weights for different calculated ratings determined by experiment.

Values of ' α ', ' β ', ' γ ' are empirically decided in such a manner that

$$\alpha + \beta + \gamma = 1 \qquad (8)$$

Once combined rating is calculated, Top N items not yet seen are recommended to active user. K-Nearest Neighbor classification, user clustering and item clustering are performed offline and recomputed at a certain period of time. Hence it ensures scalability of system.

Now if there is a user who has not rated any item in the past the transaction rating is nothing. So the rating will be predicted on the basis of user demographic and the item genres. This solves user-side cold start problem.

Similarly if there is a new item which is not rated yet, again rating in transaction will be zero. So rating for that item to the active user will be computed by user demographics and item genres. This solves item-side cold start problem.

The system is evaluated using mean absolute error (MAE) and root mean square error (RMSE), given as following

$$MAE = \frac{1}{n} \sum_{u,i} \left| p_{u,i} - r_{u.i} \right| \qquad (9)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{u,i} (p_{u,i} - r_{u,i})^2} \qquad (10)$$

4. RESULTS

Data from the MovieLens [10] data set was used to test the system. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota and are a popular choice for research on recommendation systems. It consists of 100,000 ratings from 943 users on 1682 movies. The ratings are in the scale of 1-5 where 1 means 'Awful' and 5 means 'Must see'.

In the dataset, each user has rated at least 20 movies. Simple demographic info for the users such as age, gender, occupation and zip code is included. The ratings dataset U was divided into two training sets (UABASE and UBBASE) and corresponding test sets with exactly 10 ratings per user being withheld in the test set. The test sets UATEST and UBTEST were disjoint. The rating prediction was taken as the average of experiment results over the two datasets UA and UB.

Then, the MAE and coverage of UBCF, IBCF, DBCF and IDBCF were compared.

4.1 Experimental Setup

The proposed system is tested with MovieLens dataset [14]. First experiments were conducted to determine the number of neighbors for K-nearest neighbor classification. The values of k are tested in step of K=1, 3, 5……29, total 15 sets of values were tested. Also values of weights α, β, γ were tested with 11 sets of different α, β, γ values. Sets used in experiment is given in table 2

Table 2: Different sets of Weights

Set	' α '	' β '	' γ '
1	.04	0.3	0.3
2	0.4	0.4	0.2
3	0.4	0.2	0.4
4	0.5	0.3	0.2
5	0.5	0.2	0.3
6	0.6	0.1	0.3
7	0.6	0.2	0.2
8	0.6	0.3	0.1
9	0.7	0.2	0.1
10	0.7	0.1	0.2
11	0.8	0.1	0.1

The lowest MAE was obtained when α =0.6, β =0.1, γ =0.3 and K=5.Since dataset have rating scale 1-5 so in K-means algorithm number of cluster is chosen is 5.So user and item data is divided in 5 clusters respectively. The lowest mean absolute error and root mean square error is computed when number of neighbors is k=9 for classification algorithm. Figure 2 and 3 shows the sensitivity of MAE and RMSE with changing values of set described for α , β , and γ respectively. The system generates recommendation with lower MAE compare to traditional K-nearest neighbor based collaborative filtering techniques. Also it resolves the user based and item based cold start issues with a single mechanism.

Also figure 5 shows the improvement of proposed system over traditional k nearest neighbor based collaborative filtering. So the proposed system has almost equal MAE to IBCF and IDBCF but it solves cold start issues in recommender system, However IDBCF handles user cold start, proposed system is able to tackle item and user cold start with reduced MAE

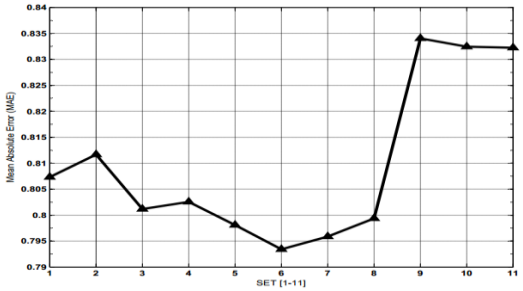

Figure 2: MAE vs Set of weights

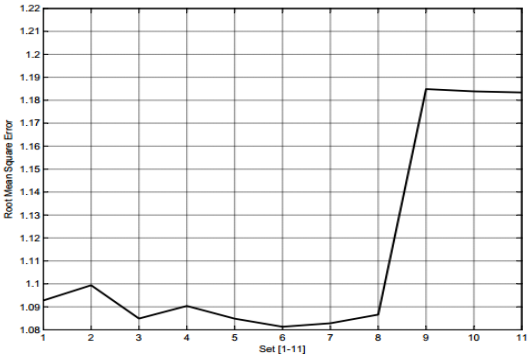

Figure 3: RMSE vs Set of weights

4.2 Comparison with Other Techniques

Here a comparison is given with some prominent collaborative filtering techniques based on MAE's.

UBCF (user based collaborative filtering) and IBCF (Item based collaborative filtering) were executed using cosine similarity given in (4) and (5). IBCF is more accurate than UBCF.DBCF (Demographic based collaborative filtering) has higher MAE compare to IDBCF (Item and demographic based collaborative filtering). Table 3 gives comparison MAE of various collaborative filtering algorithms Figure 4 shows the comparison of MAE among the above discussed techniques with proposed system.

Table 3: Comparison of algorithms

Algorithms	Mean absolute error
UBCF	0.8485
IBCF	0.7865
DBCF	0.8373
IDBCF	0.7737
Default KNN	0.9042
Proposed	0.7963

.

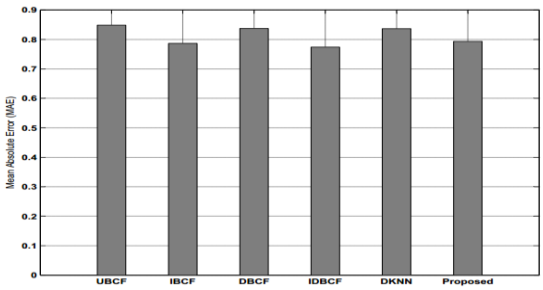

Figure 4: Comparison of algorithms

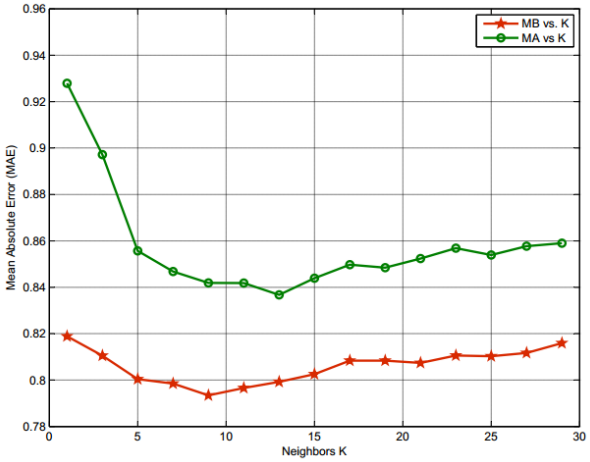

Figure 5: MB(proposed MAE) vs MK(Traditional MAE)

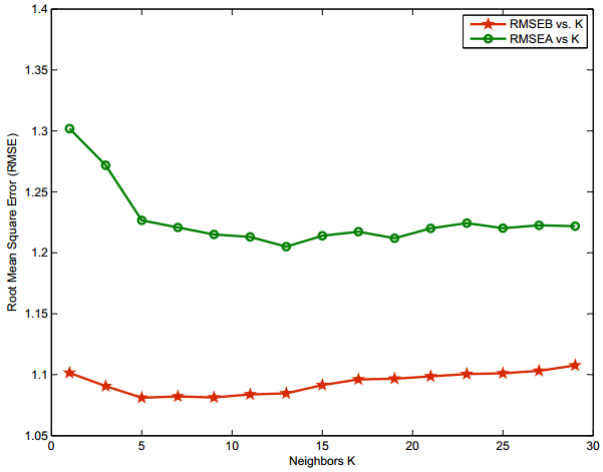

Figure 6: RMSEB (Proposed) vs RMSEK (Traditional)

5. CONCLUSION & FUTURE WORK

In this paper, a hybrid recommendation approach is proposed by combining nearest neighbor ratings prediction with rating computed from user demographics and item genres. In proposed system classification and user and item similarity computation is performed offline and recomputed after certain amount of time. New user cold start is resolved by generating immediate ratings based on user demographics while item cold start is addressed by using item cluster.

The system also achieves lower MAE than traditional K nearest neighbor algorithm used for collaborative filtering based recommendations. In this work recommendation is generated using correlation based similarity measure. In future other newly developed similarity measure can be used which may provide better performance.

6. REFERENCES

[1] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.

[2] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132.

[3] Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2014).

Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Applied Computing and Informatics.

[4] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

[5] Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Transactions on the Web (TWEB), 5(1), 2.

[6] Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. In Applications of Data Mining to Electronic Commerce (pp. 115-153). Springer US.

[7] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

[8] Gupta, J., & Gadge, J. (2015, January). Performance analysis of recommendation system based on collaborative filtering and demographics. In Communication, Information & Computing Technology (ICCICT), 2015 International Conference on (pp. 1-6). IEEE.

[9] Kantor, P. B., Rokach, L., Ricci, F., & Shapira, B. (2011). Recommender systems handbook. Springer.

[10] MovieLens dataset, http://www.grouplens.org/data/ (as of 2003)

[11] Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques: concepts and techniques. Elsevier.

[12] Schafer, J. (2009). The Application of Data-Mining to Recommender Systems. Encyclopedia of data warehousing and mining, 1, 44-48.

[13] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4), 395-416.

[14] Geyer-Schulz, A., & Hahsler, M. (2002, May). Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory. In proceedings WEBKDD (pp. 100-114).

[15] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5-6), 393-408.

[16] Thorat, P. B., Goudar, R. M., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. International Journal of Computer Applications, 110(4).

[17] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

[18] Billsus, D., & Pazzani, M. (1997, June). Learning probabilistic user models. InUM97 Workshop on Machine Learning for User Modelling.

[19] Fischer, G. (2001). User modelling in human–computer interaction. User modelling and user-adapted interaction, 11(1-2), 65-86.

[20] Mahmood, T., & Ricci, F. (2007). Towards Learning User-Adaptive State Models in a Conversational Recommender System. In LWA (pp. 373-378).

[21] Berkovsky, S., Kuflik, T., & Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. User Modeling and User-Adapted Interaction, 18(3), 245-286.

[22] Berkovsky, S., Kuflik, T., & Ricci, F. (2009). Cross-representation mediation of user models. User Modeling and User-Adapted Interaction, 19(1-2), 35-63.

[23] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer Berlin Heidelberg.

[24] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000, October). Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd ACM conference on Electronic commerce (pp. 158-167). ACM.

[25] Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating "word of mouth". In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

[26] Sheth, B., & Maes, P. (1993, March). Evolving agents for personalized information filtering. In Artificial Intelligence for Applications, 1993. proceedings., Ninth Conference on (pp. 345-352). IEEE.

[27] Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. User modelling and user-adapted interaction, 10(2-3), 147-180.

[28] Zhang, Y., Callan, J., & Minka, T. (2002, August). Novelty and redundancy detection in adaptive filtering. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 81-88). ACM.

[29] Bobadilla, J. E. S. U. S., Serradilla, F., & Hernando, A. (2009). Collaborative filtering adapted to recommender systems of e-learning. Knowledge-Based Systems, 22(4), 261-265.

[30] Ortega, F., SáNchez, J. L., Bobadilla, J., & GutiéRrez, A. (2013). Improving collaborative filtering-based recommender systems results using Pareto dominance. Information Sciences, 239, 50-61.

[31] Bobadilla, J., Ortega, F., & Hernando, A. (2012). A collaborative filtering similarity measure based on singularities. Information Processing & Management, 48(2), 204-217.

[32] Boley, D., Gini, M., Gross, R., Han, E. H. S., Hastings, K., Karypis, G. & Moore, J. (1999). Document categorization and query generation on the World Wide Web using webace. Artificial Intelligence Review, 13(5-6), 365-391.

[33] Pirolli, P., Pitkow, J., & Rao, R. (1996, April). Silk from a sow's ear: extracting usable structures from the Web. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 118-125).

ACM.

[34] Etzioni, O. (1996). The World-Wide Web: quagmire or gold mine? Communications of the ACM, 39(11), 65-68.

[35] R.Malarvizhi, K.Saraswathi "Web Content Mining Techniques Tools & Algorithms – A Comprehensive Study" International Journal of Computer Trends and Technology (IJCTT) ,V4(8):2940-2945 August Issue 2013

[36] Sharma, K., Shrivastava, G., & Kumar, V. (2011, April). Web mining: Today and tomorrow. In Electronics Computer Technology (ICECT), 2011 3rd International Conference on (Vol. 1, pp. 399-403). IEEE.

[37] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.

[38] Hu, R., & Lu, Y. (2006, November). A hybrid user and item-based collaborative filtering with smoothing on sparse data. In Artificial Reality and Telexistence--Workshops, 2006. ICAT'06. 16th International Conference on (pp. 184-189). IEEE.

[39] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7), 745-752.

[40] Zhang, Y., & Liu, Y. (2010, April). A Collaborative filtering algorithm based on time period partition. In Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on (pp. 777-780). IEEE.

[41] Puntheeranurak, S., & Chaiwitooanukool, T. (2011, July). An Item-based collaborative filtering method using Item-based hybrid similarity. In Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on (pp. 469-472). IEEE.

[42] Sun, D., Luo, Z., & Zhang, F. (2011, October). A novel approach for collaborative filtering to alleviate the new item cold-start problem. In Communications and Information Technologies (ISCIT), 2011 11th International Symposium on (pp. 402-406). IEEE.

[43] Chikhaoui, B.; Chiazzaro, M.; Shengrui Wang, "An Improved Hybrid Recommender System by Combining Predictions," in Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on , vol., no., pp.644-649, 22-25 March 2011

[44] Moghaddam, S. G., & Selamat, A. (2011, October). A scalable collaborative recommender algorithm based on user density-based clustering. In Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on (pp. 246-249). IEEE.

[45] Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-based systems, 24(8), 1310-1316.

[46] Sun, D., Luo, Z., & Zhang, F. (2011, October). A novel approach for collaborative filtering to alleviate the new item cold-start problem. In Communications and Information Technologies (ISCIT), 2011 11th International Symposium on (pp. 402-406). IEEE.

[47] WEN, J., & ZHOU, W. (2012). An Improved Item-based Collaborative Filtering Algorithm Based on Clustering Method. Journal of Computational Information Systems, 571-578.

[48] Xie, F., Xu, M., & Chen, Z. (2012, March). RBRA: A simple and efficient rating-based recommender algorithm to cope with sparsity in recommender systems. In Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on (pp. 306-311). IEEE.