By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

***Abstract*--Recommender systems apply machine learning techniques for filtering unseen information and can predict whether a user would like a given resource. There are 3 main types of recommender systems: collaborative filtering, content-based filtering, and demographic recommender systems. For the purpose of this project we will be covering two important recommender systems namely collaborative and content based systems. Collaborative filtering recommender systems recommend items by taking into account the ratings of a product (in terms of preferences of items) of users, under the assumption that users will be interested in items that users similar to them have rated highly. Content-based filtering recommender systems recommend items based on the textual information of an item, under the assumption that users will like similar items to the ones they liked before. These systems suffer from scalability, data sparsity, and cold-start problems resulting in poor quality recommendations and reduced coverage. In this paper, we propose a unique cascading hybrid recommendation approach by combining the rating and feature.**

I. INTRODUCTION

With a substantial increase in number of buyers on ecommerce websites like Amazon and EBay, the data for each user goes on increasing exponentially. As a result of this overwhelming amount of data it is difficult to determine the importance of data for satisfying the buying needs of each user separately. For example, suggesting similar items to a user based on his recent search history, likes or ratings. This problem highlights the significance of filtering irrelevant information from the data. This filtering can be achieved with the help of recommender systems. The recommender systems are used vastly for predicting items for a particular user that he may be interested in based on his buying information. Ecommerce services rely heavily on these systems to get maximum possible buyers

Let $M = \{ m1, m2, \ldots, mx \}$ be the set of all users, $N = \{ n1, n2, \ldots , ny \}$ be the set of all possible items that can be recommended, and ($\boldsymbol{r}_{mi,nj}$) be the rating of user *mi* on item *nj*. Let $\boldsymbol{u}$ be a utility function that measures the utility of item *nj* to user *mi*, i.e., $u : M \times N \rightarrow R$, where R is a totally ordered set. Now for each user $mi \in M$, the aim of a recommender system is to choose that item $n'j \in N$ which maximizes the user's utility. We can specify this as follows: $n'j\ mi$= argmax $u(mi, nj) : \forall mi \in M,$.

Our Content based filtering algorithms will consider only the textual features whereas Collaborative algorithms that are used in this project are memory based algorithms. Memory-based approaches make a prediction by taking into account the entire collection of previously rated items by a user; examples include user-based CF algorithms. There is one more type of collaborative system that is model based. Model-based approaches learn a model from collection of ratings and use this model for making prediction; examples include item-based CF. But for this project, we will only use memory based collaborative system.

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

There are two potential problems with the recommender systems. One is the *scalability*, which is how quickly a recommender system can generate recommendation, and the second is to ameliorate the *quality* of the recommendation for a customer. Pure CF recommender systems produces high quality recommendation than those of pure content-based and demographic recommender systems, however, due to the *sparsity*, they cannot find similar items or users using rating correlation, resulting in poor quality predictions and reduced *coverage*. Collaborative filters are said to be more reliable and accurate than content-based systems. However, both the algorithms have their own pros and cons. Hence in this paper, we propose a hybrid scheme which is capable of producing accurate and practical recommendations. Our proposed scheme is based on a cascading hybrid recommendation technique that builds item models by using textual features of content based system. The predicted utility matrix derived from this algorithm is then fed to collaborative filtering algorithm to produce more accurate and better results. We evaluate our model on product data-sets of Amazon.

II. BACKGROUND

Table 1: A SAMPLE OF UTILITY MATRIX

	Item1	Item2	…	ItemN
User1	R11	R12	…	R1n
User2	R21	R22		R2n
. . .	. . .	. . .	. . .	. . .
UserM	Rm1	Rm2	…	Rmn

where M is the number of users and 1..m ∈ *M*, *N* is the number of items and 1..n ∈ *N* whereas R_{ij} is the rating given by a user i to item j and i ∈ *M* and j ∈ *N*

A. Content-based recommender systems:

Content-based recommender systems recommend items based on the textual information of an item. In these systems, an item of interest is defined by its associated features, for instance, reviews of the Amazon product dataset for each movie are used as features. Using these features to calculate frequencies, we created a TF-IDF matrix. With product of TF-IDF matrix and original utility matrix, we obtained a different set of matrix that is a prediction of ratings

B. Item Based Collaborative filtering recommender system:

Item Based Collaborative filtering recommender system build a model of item similarities using gradient descent method. The steps are as follows:

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

This model uses matrix factorization method to compute similarities between k users. The steps are as follows:

1. Utility matrix is derived by utility function: u : M × N R
2. Gradient descent method: A random matrix U of length equal to the number of users and a random I matrix equal to the number of items is then generated for k most similar items. K=2 for our algorithms.
3. For given number of iterations, matrix multiplications of R, U and I are executed to ultimately predict the ratings for each item by that each user.

C. User – Based Collaborative Filtering

User-based collaborative filtering predicts a test user's interest in a test item based on rating information from similar user profiles. Ratings by more similar users contribute more to predicting the test item rating. For prediction, we have used factorization method. Cosine similarity and Pearson's correlation are popular similarity measures in collaborative filtering. This paper adopts new similarity measure known as boosted similarity [1] for measuring the similar user profiles.

III. EXPERIMENTAL EVALUATION

A. Dataset

We used Amazon product dataset [4] for evaluating our algorithm. This dataset contains 3 million unique users, 4 million unique items, and 5 million ratings within the scale of 1 (bad) to 5 (excellent). To increase the scalability of data set, we filtered the dataset to contain only the users that have rated at least 30 items and also filtered the items that have been rated by at least 2 users.

B. Metrics

Our specific task in this paper is to predict scores for items that already have been rated by actual users, and to check how well this prediction helps users in selecting high quality items. Keeping this into account, we used *Mean Absolute Error(MAE) and Root Mean Squared Error(RMSE)*. *MAE* measures the average absolute deviation between a recommender system's predicted rating and a true rating assigned by the user. RMSE measures the quality of the predicted rating by comparing with the true rating.

It is computed as follows:

$$MAE = \sum \frac{|rpi - rti|}{N} \, , \; RMSE = \sum \frac{|rpi - rti|^2}{N}$$

where *rpi* and *tai* are the predicted and true values of a rating respectively, and N is the total number of items that have been rated.

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

IV. PROPOSED ALGORITHM

The proposed algorithm can be summarized as follows:

Step-1: Use Content based system features i.e the user profile obtained from TF-IDF matrix and previous utility matrix to get predictions

Step-2: Compute the similarity between items of utility matrix using gradient descent method.

Generating Recommendation

$fmax = \text{argmax}\ f \in F\ u(mi, nj) : \forall mi \in MT\ ,\ \forall nj \in NT\ .$

This equation tells us to choose that function which maximizes the utility (i.e. reduces the *MAE*) of all users (***MT***) over set of items (***NT***) in the training set. Table I gives the different combination of functions checked over the training set with their respective lowest *MAE* observed. It shows that a cascading hybrid setting in which rating and demographic correlation are applied over the candidate neighbor items found after applying the feature correlation gives the minimum error.

V. IMPLEMENTATION DETAILS

A. Design Issues

We have used Python scripting for this assignment. For this assignment, we have imported numpy, math, json, sklearn packages to solve the proposed algorithm for the given data set. For algorithm implementation, we have used all the math functions available under python. Only very problems encountered during the implementation. We have used 10-fold cross validation for testing the performance of each and every model designed.

Initially, it took time to find out and extract the required data set. Once we extracted, faced some difficulties while filtering and preprocessing the data because of sparsity problem. While working content-based filtering, it took some time to learn about tokenization of reviews. With punctuation and special characters, the results obtained were absurd. Later we used regular expressions to ignore punctuations and special characters.

B. Instructions

1) Three. ipynb files created for each neighborhood sizes (k=20, 30, 40) respectively.
2) We have used Amazon product data set [4] for this implementation.

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

V. RESULTS AND DISCUSSION

We randomly divided the data set into 80% training and 20% for testing. We have conducted 10-fold cross validation by randomly selecting different training and testing set each time for evaluating the performance. We have varied the number of neighbors for an active user and validated the performance of our algorithm. We also compared the algorithm with traditional item-based CF using Pearson similarity and content - based recommendation based on textual information (reviews). We also have provided top item recommendations to each user based on the ratings predicted.

Table 2
COMPARISON OF PROPOSED ALGORITHM, METRICS

Neighborhood Size(k=30)		
Algorithm	**MAE**	**RMSE**
Item - Based CF	5.9361	0.224422
User - Based CF	3.016697	0.12363
Content - Based	29.633	0.8874
Hybrid	1.5721602	0.0652
Neighborhood Size(k=40)		
Item - Based CF	5.273872	0.32513
User - Based CF	2.53644	0.15759
Content - Based	25.344	1.156790
Hybrid	1.33780	0.08067
Neighborhood Size(k=20)		
Item - Based CF	7.216365	0.14768
User - Based CF	3.84303	0.09077
Content - Based	34.29	0.58557
Hybrid	1.982702	0.046869

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

From the above table, it is very clear that the hybrid algorithm outperforms the other algorithms in terms of MAE. This algorithm overcomes the problem faced in item-based and content based. It also overcomes the misleading similarities between the items by calculating boosted similarity. It also shows that the size of neighborhood does affect the quality of the prediction.

We also tried recommending top items to each user based on the predicted ratings obtained via our algorithm. The top recommendations can be obtained by calculating the similarities between items rated by similar users. After calculating similarities, we have recommended items to a user which are not rated yet by that user based on the predicted ratings.
Sample output is shown below:

User: A1MVH1WLYDHZ49

Top 5 item recommendations:
item-B00JBIVXGC
item-B00GTSM8FW
item-B00CPWTRGE
item-B00BZ4X3ZE
item-B009ZX8ZJG

Performance evaluation shows that the proposed algorithm provides better quality of prediction than the other algorithms. We also showed top recommendations based on the rating predictions obtained.

V. CONCLUSION

In this paper, to overcome the cons and integrate pros of each system we have built a hybrid recommendation system by combining content based and collaborative algorithm together. We also showed that our approach works better than the traditional algorithms.

VI. ACKNOWLEDGMENT

The work reported in this paper is inspired from a published IEEE paper [1].

By-
Pavithra Kuttiyandi Chandrakasu(A20353191)
Mamta Kanvinde(A20345211)

VII. REFERENCES

[1] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, "A Scalable, Accurate Hybrid Recommender System", in 2010 Third International Conference on Knowledge Discovery and Data Mining.

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

[3] D. Pennock, E. Horvitz, S. Lawrence, and C. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach," in Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 473–480

[4] http://jmcauley.ucsd.edu/data/amazon/links.html

[5] http://www.quuxlabs.com/blog/2010/09/

[6] http://recommender-systems.org/hybrid-recommender-systems