A New Algorithm for Recommender System by clustering Items based on Stability of User Similar

Sajad Manteghi[1], Zakiye Bozorgvari[2]

[1] Islamic Azad University, Yasooj Branch, MA Student in Sciences and Researches Branch, Teacher of Education in Kohgiluyeh and Boyer-Ahmad Province
[2] Employee in Education Office of Kohgiluyeh and Boyer-Ahmad Province
manteghisajjad@yahoo.com

Abstract: Recommendation systems can help people to find interesting things and they are widely used with the development of electronic commerce. Many recommendation systems employ the collaborative filtering technology, which is proved to be one of the most successful techniques in recommender systems in recent years. Gradual increase of customers and products in E-commerce systems, the time consuming nearest neighbor collaborative filtering search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system. At the same time, it suffers from its poor quality when the number of the records in the user database increases. Sparsity of source data set is the major reason causing the poor quality. To solve the problems of scalability and sparsity in the collaborative filtering, we proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on their ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of the target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.
[Sajad Manteghi, Zakiye Bozorgvar. **A New Algorithm for Recommender System by clustering Items based on Stability of User Similar.** *Rep Opinion* 2019;11(6):51-57]. ISSN 1553-9873 (print); ISSN 2375-7205 (online). http://www.sciencepub.net/report. 8. doi:10.7537/marsroj110619.08.

Keywords: recommender systems, collaborative filtering, user clustering, item clustering

1. Introduction

With the development of the internet, intranet and electronic commerce systems, there are amounts of information that we can hardly deal with. Therefore, personalized recommendation services have been developed to provide us the useful data employing some information filtering technologies. Information filtering has two main methods. One is the content- based filtering and the other one is the collaborative filtering. Collaborative filtering (CF) has proved to be one of the most effective for its simplicity in both theory and implementation [Breese J, Hecherman D, Kadie, 1998; Chong-Ben Huang, Song-Jie Gong, 2008].

Many researchers have proposed various types of CF technologies to make a quality recommendation. All of them make a recommendation based on the same data structure as user-item matrix having users and items including their rating scores. There are two methods in CF as user based collaborative filtering and item based collaborative filtering [Sarwar B, Karypis G, Konstan J, Riedl J, 2001; Manos Papagelis, Dimitris Plexousakis, 2005]. User based CF assumes that, a good way to find a certain user's interesting items is to find other users who have a similar interest. Therefore, at first, it tries to find the user's neighbors based on user similarities and then combine the neighbor rating scores of the users, which have previously been expressed, by similarity weighted averaging. Item based CF fundamentally has the same scheme with user based CF. It looks into a set of items; the target user has already rated and computes how similar they are to the target item under recommendation. After that, it also combines his previous preferences based on these item similarities.

The challenge of these two CF as following [Hyung Jun Ahn, 2008; SongJie Gong, 2008]: Sparsity: Even as users are very active, there are a few rating of the total number of items available in a user-item ratings database. Since the majority of the collaborative filtering algorithms are based on similarity measures computed over the co-rated set of items, large levels of sparsity can lead to less accuracy.

Scalability: Collaborative filtering algorithms seem to be efficient in filtering the items that are interesting to users. However, they require computations that are very expensive and grow non-linearly with the number of users and items in a database.

Cold-start: An item cannot be recommended unless it has been rated by a number of users. This problem applies to new items and is particularly detrimental to users with eclectic interest. Likewise, a new user has to rate a sufficient number of items before the CF algorithm becomes able to provide accurate recommendations.

To solve these problems in the collaborative filtering, in this paper, we proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and accurate than the traditional one.

L.H. Ungar et al. presented a formal statistical model for collaborative filtering and compare different algorithms for estimating the model parameters including variations of K-means clustering and Gibbs Sampling. This formal model is easily extended to handle clustering of objects with multiple attributes. And it is better than the traditional one.

M.O. Conner reported on work related to applying data clustering algorithms to ratings data in collaborative filtering. They use existing data partitioning and clustering algorithms to partition the set of items based on user rating data. Predictions are then computed independently within each partition. Ideally, partitioning will improve the quality of collaborative filtering predictions and increase the scalability of collaborative filtering systems. They report preliminary results that suggest that partitioning algorithms can greatly increase scalability, but they have mixed results on improving accuracy. However, partitioning based on ratings data does result in more accurate predictions than random partitioning, and the results are similar to those when the data is partitioned based on a known content classification.

Lee, WS et al. have been studied two online clustering methods for collaborative filtering. In the first method, they assume that each user is equally likely to belong to one of m clusters of users and that the user's rating for each item is generated randomly according to a distribution that depends on the item and the cluster that the user belongs to. In the second method, they assume that each user is equally likely to belong to one of m clusters of users while each item is equally likely to belong to one of n clusters of items. And the result is that the proposed methods are good in some way.

S.H.S. Chee et al. developed an efficient collaborative filtering method, called RecTree that addresses the scalability problem with a divide-and conquer approach. The method first performs an efficient k-means-like clustering to group data and creates neighborhood of similar users, and then performs subsequent clustering based on smaller, partitioned databases. Since the progressive partitioning reduces the search space dramatically, the search for an advisory clique will be faster than scanning the entire database of users. Moreover, the partitions contain users that are more similar to each other than those in other partitions. This characteristic allows RecTree to avoid the dilution of opinions from good advisors by a multitude of poor advisors and therefore yielding a higher overall accuracy. Based on the experiments and performance study, RecTree outperforms the well-known user based collaborative filtering, in both execution time and accuracy. Particularly, RecTree's execution time scales by O (nlog2 (n)) with the dataset size while the traditional user based collaborative filtering recommendation scales in a quadratic manner.

George, T. et al. considered a novel collaborative filtering approach based on a recently proposed weighted co-clustering algorithm that involves simultaneous clustering of users and items. They design incremental and parallel versions of the co-clustering algorithm and use it to build an efficient real-time collaborative filtering framework. Their empirical evaluation of the proposed approach on large movie and book rating datasets demonstrates that it is possible to obtain accuracy comparable to that of the correlation and matrix factorization based approaches at a much lesser computational cost.

Rashid, A.M. et al. have proposed ClustKnn, a simple and intuitive algorithm that is appropriate for large data sets. The proposed method first compresses data tremendously by building a straightforward but efficient clustering model. Recommendations are then generated quickly by using a simple Nearest Neighbor-based approach. They demonstrated the feasibility of ClustKnn both analytically and empirically. They also show, by comparing with a number of other popular collaborative filtering algorithms that, apart from being highly scalable and intuitive, ClustKnn provides very good recommender accuracy as well.

2. Material and Methods

2.1. Rating smoothing based on user clustering

2.1.1. User clustering

User clustering techniques work via identifying groups of users who appear to have similar ratings. As soon as the clusters are created, predictions for a target user can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. After that user clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller [B. Sarwar, G. Karypis, J. Konstan and J. Riedl, 2002].

The idea is to divide the users of a collaborative filtering system using user clustering algorithm and use the divide as neighborhoods, as Figure 1 show. The clustering algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate a requested number of partitions of varying size.

	Item-1	Item-2	...	Item-n
User-1	R11	R12		R1n
User-2	R21	R22		R2n
...				
User-m	Rm1	Rm2		Rmn

user clustering

	Item-1	Item-2	...	Item-n
center1	a11	a12		a1n
center2	a21	a22		a2n
...				
centerc	ac1	ac2		acn

Figure 1: Collaborative filtering based on user clustering

Where,

Rij: is the rating of the user i to the item i,

Aij: is the average rating of the user center i to the item i,

m: is the number of all users,

n: is the number of all items,

c: is the number of user centers.

2.1.2. Smoothing

In present study, we use the k means clustering algorithm to cluster the users into some groups as clustering centers. Specific algorithm as follows:

Input: clustering number k,
user − item rating matrix
Output: smoothing rating matrix
Begin
Select user set U = {U1, U2 ... Um};
Select item set I = {I1, I2 ... In};
Choose the top k rating users
as the clustering
CU = {CU1, CU2 ... CUk};
The k clustering center
is null as c = {c1, c2 ... ck};
do
for each user Ui ∈ U
for each cluster center CUj ∈ CU
calculate the sim (Ui, CUj);
end for
sim (Ui, CUm) = max {sim (Ui, CU1),
sim (Ui, CU2), ..., sim (Ui, CUk);
cm = cm ∪ Ui
end for
for each cluster ci ∈ c
for each user Uj ∈ U
CUi = average (ci, Uj);
end for
end for
while (C is not change)
End

2.1.3. New ratings

One of the challenges of the collaborative filtering is the data sparsity. To predict the vacant values in user-item rating dataset, we make explicit use of item clusters as prediction mechanisms. Based on the item clustering, we apply the prediction strategies to the vacant rating data as follows:

$$R_{ij} = \begin{cases} R_{ij} \text{ if user i rate the item j} \\ C_j \text{ else} \end{cases}$$

Where cj denotes the prediction value for user i rating towards an item j and cj have calculated in above specific algorithm.

2.2. Using the item clustering method to produce Recommendations

Through the calculating the vacant user's rating by user clustering algorithm, we obtained the dense users' ratings. After that, to generate prediction of a user's rating, we use the item clustering based collaborative filtering algorithms.

2.2.1. The dense user-item matrix

After we used the user clustering algorithm, we obtained the dense ratings of the users to the items. Therefore, the original sparse user-item rating matrix is now becoming the dense user-item matrix.

2.2.2. Item clustering

Item clustering technique works by identifying groups of items who appeared to have similar ratings. After the clusters are created, predictions for a target item can be made by averaging the opinions of the other items in that cluster. Some clustering techniques represent each item with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Once the item clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller.

The idea is to divide the items of a collaborative filtering system with item clustering algorithm and use the divide as neighborhoods, as Figure 2 show. The clustering algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate a requested number of partitions of varying size.

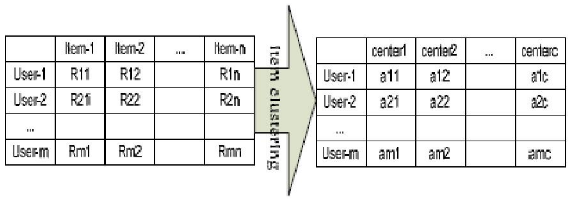

Figure 2: Collaborative filtering based on item clustering

Where,

Rij: is the rating of the user i to the item i,

aij: the average rating of the user i to the item center j,

m: is the number of all users,

n: is the number of all items,

c: is the number of item centers.

2.2.3. Algorithm

There exist many algorithms that can be used to create item clustering. In this work, we choose the k means algorithm as the basic clustering algorithm. The number k is an input to the algorithm that specifies the desired number of clusters. First, the algorithm takes the first k items as the centers of k unique clusters. Each of the remaining items is then compared to the closest center. In the following, the cluster centers are re-computed based on cluster centers formed in the previous pass and the cluster membership is re-evaluated. Specific algorithm as follows:

Input: clustering number k,
user − item rating matrix
Output: item − center matrix
Begin
Select user set U = {U1, U2 ... Um};
Select item set I = {I1, I2 ... In};
Choose the top k rating items
as the clustering
CI = {CI1, CI2 ... CIk};
The k clustering center
is null as c = {c1, c2 ... ck};
do
for each item Ii ∈ I
for each cluster center CIj ∈ CI
calculate the sim (Ii, CIj);
end for
sim (Ii, CIx) = max {sim (Ii, CI1), sim (Ii, CI2), ... , sim (Ii, CIk);
cx = cx ∪ Ii
end for
for each cluster ci ∈ c
for each user Ij ∈ I
CIi = average (ci, Ij);
end for
end for
while (CU and c is not change)
End

We used Pearson's correlation, as following formula, to measure the linear correlation between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r) = \frac{\sum_{i=1}^{m} (R_{it} - A_t)(R_{ir} - A_r)}{\sqrt{\sum_{i=1}^{m} (R_{it} - A_t)^2 \sum_{i=1}^{m} (R_{ir} - A_r)^2}}$$

Where,

Rit: is the rating of the target item t by user i,

Rir: is the rating of the remaining item r by user i,

At: is the average rating of the target item t for all the co-rated users,

Ar: is the average rating of the remaining item r for all the co-rated users,

m; is the number of all rating users to the item t and item r.

2.2.4. Selecting clustering centers

An important step of item based collaborative filtering algorithm is looking up for neighbors of the target item. Traditional memory- based collaborative filtering searches the whole ratings database and it suffers from poor scalability when more and more users and items added into the database [Xue, G., Lin, C., & Yang, Q., et al, 2005].

When we cluster the items, we get the items centers. This center is represented as an average rating over all items in the cluster. So we can choose the target item neighbors in some of the item center clustering. We used Pearson's correlation for similarity between the target item and the items centers.

After calculating the similarity between the target item and the items centers, we take the items in the most similar centers as the candidates.

2.2.5. Selecting neighbors

After we selected the target item nearest clustering centers, we also needed to calculate the similarity between the target item and items in the selected clustering centers.

We have selected the Top K most similar items based on the cosine measure, as following formula, which looks at the angle between two vectors of ratings as the target item t and the remaining item r.

$$\mathrm{sim}(t,r) = \frac{\sum_{i=1}^{m} R_{it}R_{ir}}{\sqrt{\sum_{i=1}^{m} {R_{it}}^2 \sum_{i=1}^{m} {R_{ir}}^2}}$$

Where,

Rit: is the rating of the target item t by user i,

Rir: is the rating of the remaining item r by user i,

m: is the number of all rating users to the item t and item r.

2.2.6. Producing Recommendations

Since we have got the membership of item, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target item.

The rating of the target user u to the target item t is as following:

$$P_{ut} = \frac{\sum_{i=1}^{c} R_{ui} \times \mathrm{sim}\,(t,i)}{\sum_{i=1}^{c} \mathrm{sim}\,(t,i)}$$

Where,

Rui: is the rating of the target user u to the neighbor item i,

sim (t, i): is the similarity of the target item t and the neighbor it user i for all the co-rated items,

m: is the number of all rating users to the item t and item r.

3. Results

In this section, we will describe the dataset, metrics and methodology for the comparison between traditional and proposed collaborative filtering algorithm, and present the results of our experiments.

4.1. Data set

We have used MovieLens collaborative filtering data set to evaluate the performance of proposed algorithm. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota and MovieLens is a web-based research recommender system that debuted in 1997. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies [Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan, 2007; Sarwar B, Karypis G, Konstan J, Riedl J, 2001]. The site now has over 45000 users who have expressed opinions on 6600 different movies. We randomly selected enough users to obtain 100, 000 ratings from 1000 users on 1680 movies with every user having at least 20 ratings and simple demographic information for the users is included. The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence.

4.2. Performance measurement

Several metrics have been proposed for assessing the accuracy of collaborative filtering methods. They are divided into two main categories: statistical accuracy metrics and decision-support accuracy metrics. In this paper, we use the statistical accuracy metrics [Huang qin-hua, Ouyang wei-min, 2007; Songjie Gong, Chongben Huang, 2008].

Statistical accuracy metrics evaluate the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Some of them frequently used are mean absolute error (MAE), root mean squared error (RMSE) and correlation between ratings and predictions. All of the above metrics were computed on result data and generally provided the same conclusions. As statistical accuracy measure, mean absolute error is employed.

Formally, if n is the number of actual ratings in an item set, and then MAE is defined as the average absolute difference between the n pairs. Assume that p1, p2, p3... pn is the prediction of users' ratings, and the corresponding real ratings data set of users is q1, q2, q3... qn. See the MAE definition as following:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n}$$

The lower the MAE, the more accurate the predictions would be, allowing for better recommendations to be formulated. MAE has been computed for different prediction algorithms and for different levels of sparsity.

4.3. Sensitivity of different training-test ratio x

In order to determine the sensitivity of density of the dataset we carried out an experiment where we varied the value of x from 0.2 to 0.8 in an increment of 0.1. For each of these training-test ratio values we ran our experiments using our proposed algorithm and the traditional CF algorithm. The results are shown in Figure 3. We can see that, the quality of prediction increase as we increase x and our proposed CF is better than the traditional.

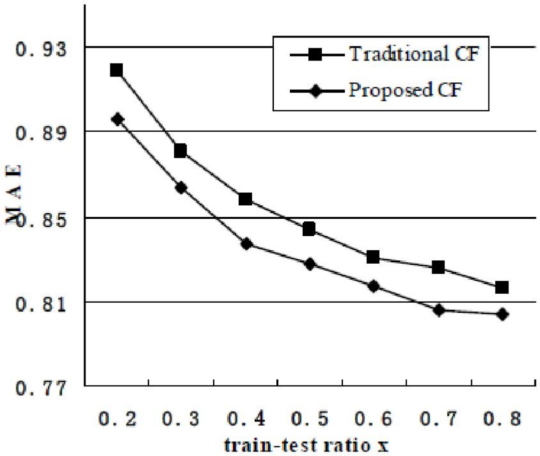

Figure 3: MAE of the different prediction algorithm with respect to train-test ratio x

4.4. Comparing with the traditional CF

We compare the proposed method combining user clustering and item clustering collaborative filtering with the traditional collaborative filtering. The size of the neighborhood has a significant impact on the prediction quality. In our experiments, we vary the number of neighbors and compute the MAE. The obvious conclusion from Figure 4, which includes the Mean Absolute Errors for the proposed algorithm and the traditional collaborative filtering as observed in relation to the different numbers of neighbors, is that our proposed algorithm is better.

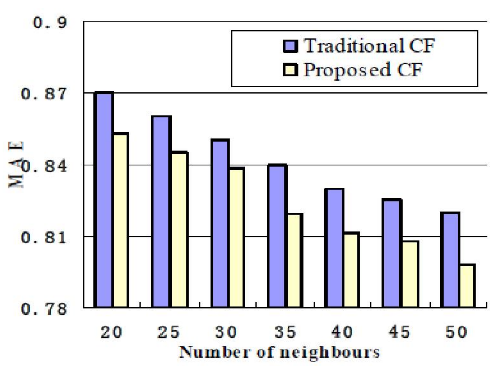

Figure 4: Comparing the proposed CF algorithm with the traditional CF algorithm

4. Discussions

Recommender systems can help people to find interesting things. They are widely used in our life with the development of electronic commerce. Many recommendation systems employ the collaborative filtering technology, which has been proved to be one of the most successful techniques in recommender systems in recent years. With the increase of customers and products in electronic commerce systems, the time consuming nearest neighbor collaborative filtering search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system. At the same time, it suffers from its poor quality when the number of the records in the user database increases. Sparsity of source data set is the major reason causing the poor quality. In order to solve the problems of scalability and sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each user cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.

Acknowledgements:

Foundation item: The National Project of India. Authors are grateful to the Department of Science and Technology, Government of India for financial support to carry out this work.

Corresponding Author:
Sajad Manteghi
Islamic Azad University, Yasooj Branch, MA Student in Sciences and Researches Branch, Teacher of Education in Kohgiluyeh and Boyer-Ahmad Province
manteghisajjad@yahoo.com

References

1. AK Jain, and RC Dubes (1988): "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs NJ, USA.
2. Arnd Kohrs, and Bernard Merialdo (1999): "Clustering for collaborative filtering applications", in Proceedings of CIMCA'99. IOS Press.
3. Breese J, Hecherman D, Kadie C. (1988): Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98). 43-52.
4. CC Aggarwal, C. Procopiuc, JL Wolf, PS Yu, and JS Park (1999): "Fast algorithms for projected clustering", in Proceedings of ACM SIGMOD Conference on Management of Data, Philadephia, USA, pp. 61-72.
5. Chong-Ben Huang, Song-Jie Gong, (2008): Employing rough set theory to alleviate the

sparsity issue in recommender system, In: Proceeding of the Seventh International Conference on Machine Learning and Cybernetics (ICMLC2008), IEEE Press, 1610-1614.
6. Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan, (2007): Personalized Service System Based on Hybrid Filtering for Digital Library, Tsinghua Science and Technology, Volume 12, Number 1, February, 1-8.
7. Huang qin-hua, Ouyang wei-min, (2007): Fuzzy collaborative filtering with multiple agents, Journal of Shanghai University (English Edition), 11(3):290-295.
8. Hyung Jun Ahn, (2008): A new similarity measure for collaborative filtering to alleviate the new user cold starting problem, Information Sciences 178, 37-51.
9. Kasap, Seda; Tekbacak, Fatih, (2012): “Performance Comparison of Combined Collaborative Filtering Algorithms for Recommender Systems,” Computer Software and Applications Conference Workshops (COMPSACW), IEEE 36th Annual, PP. 284 – 289.
10. L. Ungar, and D. Foster, (1988): "Clustering methods for collaborative filtering", in Proceedings of the workshop on recommendation systems. Menlo Park, CA. AAAI Press.
11. Manos Papagelis, Dimitris Plexousakis, (2005): “Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents”, Engineering Application of Artificial Intelligence 18, 781-789.
12. Mark O’Connor, and Jon Herlocker, (1999): “Clustering Items for Collaborative Filtering”, in ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation.
13. Rashid, A.M.; Lam, S.K.; Karypis, G.; Riedl, J.; ClustKNN: (2006): “A Highly Scalable Hybrid Model- & Memory- Based CF Algorithm”. WEBKDD.
14. Sarwar B, Karypis G, Konstan J, Riedl J, (2001): “Item-Based collaborative filtering recommendation algorithms”. In: Proceedings of the 10th International World Wide Web Conference. 2001. 285-295.
15. Songjie Gong, Chongben Huang, (2008): “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 Academic Press, 449-454.
16. SongJie Gong, (2008): “The Collaborative Filtering Recommendation Based on Similar-Priority and Fuzzy Clustering”, In: Proceeding of 2008 Workshop on Power Electronics and Intelligent Transportation System (PEITS2008), IEEE Computer Society Press, 248-251.
17. Truong Khanh Quan, Ishikawa Fuyuki, Honiden Shinichi, (2006): “Improving Accuracy of Recommender System by Clustering Items Based on Stability of User Similarity,” International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, IEEE.
18. Xinjun Wang; Zhaohui Peng; Qiuyan Li**,** “Actively building collaborative filtering recommendation in clustered social data,” Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, PP. 693 – 697, 23-25 May 2012.
19. Xue, G., Lin, C., & Yang, Q., et al. Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the ACM SIGIR Conference 2005 pp.114–121.
20. Saxena AK, Pandey P, Singh JS. Biological Spectrum and other structural functional attributes of the vegetation of Kumaun Himalaya, Vegetatio 1982;49(1):111-9.
21. Mehrotra P. Adaptive significance of leaf in relation to other parts in oak forest herbs of Kumaun Himalaya, Ph. D. Thesis, Kumaun University, Nainital, India, 1988.
22. Moustafa AA. Environmental Gradient and Species Distribution on Sinai Mountains. Ph. D. Thesis, Botany Department, Faculty of Science, Suez Canal University, Egypt, 1990;115.
23. Tewari JC. Vegetational analysis along altitudinal gradients around Nainital, Ph. D. Thesis, Kumaun University, Nainital, 1982;570.
24. Pielou EC. Ecological Diversity. Wiley, New York, 1975;165.
25. Magurran AE. Ecological Diversity and Its Measurement. Princeton University Press, Princeton, New Jersey, 1988;179.

6/23/2019