RESEARCH ARTICLE OPEN ACCESS

Study and Analysis of Clustering Based Scalable Recommender Systems for E-Commerce

Partha Sarathi Chakraborty, Dr. Sunil Karforma
Assistant Professor, University Institute of Technology, The University of Burdwan, Burdwan
Reader, Dept. of Computer Science, The University of Burdwan , Burdwan

Abstract
Collaborative filtering based recommender systems help online users in choosing the right products based on the purchase history of the user and his most similar users. Scalability is one of the major issues in designing effective recommender system. In this paper, we have studied different ways of increasing scalability by applying clustering algorithms on three types collaborative filtering algorithms-user based, item based and slope one. Finally we have analyzed the relationship between scalability and accuracy for different number of clusters and neighborhoods.

I. Introduction

Recommender systems help customers in purchasing products from e-commerce websites. Mainly three techniques have been employed in designing the recommender systems. They are content based approach, collaborative filtering based approach and hybrid approach. In content based approaches the similarity among product features are considered in generating recommendations. The collaborative filtering algorithms, in turn, take into account the opinions of the like-minded customers i.e. the customers who have rated similarly.

The major challenges in designing collaborative filtering based recommender systems are sparsity, security, trust and scalability. As a very large number of items are offered by the e-commerce sites and customers are rated a small fraction of them, the user-item rating matrix which is the "black box" of the collaborative filtering based recommender systems, becomes very sparse. On the other hand, with the increase of the number of users and items, the computation process of generating recommendations takes more and more time. So designing scalable recommender systems is a major challenge.

In this paper, we have studied different collaborative filtering algorithms and analyzed their scalability. The rest of the paper is organized as follows. In section II related studies have been mentioned. In section III, three main collaborative filtering algorithms namely user-based, item-based and slope one have been discussed. In section IV, we have discussed the ways of making the solution scalable under the heading of memory and model based techniques. Section V concentrates on the use of clustering algorithms to collaborative filtering algorithms for increasing scalability. Empirical analysis has been done in section VI. Finally, we conclude this paper in section VII.

II. Related Studies

Survey on collaborative filtering algorithms has been done by many researchers. Authors of papers [Lee et al. 2012], [Sachan and Richariya 2013] and [Revankar and Haribhakta 2015] have discussed different memory-based, model based, and hybrid CF algorithms in general. They have discussed briefly the scalability issue as one of the main challenges of designing effective recommender systems. [Su and Taghi 2009] have discussed few clustering based approaches in their work. Application of k-means,fuzzy c-means and genetic clustering algorithms in improving scalability of recommender system have been discussed in paper [Darvishi-mirshekarlou et al. 2013]. In addition to introducing CLUSTKNN algorithm, [Rashid et al. 2007] also discussed efficiency of some other collaborative filtering algorithms like SVD-based, Plsa-based algorithms. In this paper we have concentrated on clustering based approach for increasing scalability in recommender system.

III. Different CF Mechanisms

A. Notation

The basic collaborative filtering algorithm deals with a set of N users, $U: \{U_1, U_2, \dots, U_N\}$ and a set of M items $I: \{I_1, I_2, \dots, I_M\}$. Each user $U_i \in U$ rates some of the items from I. The rating of item I_j by the user U_i is denoted by $r_{i,j}$. The matrix $R: \{r_{i,j}\}$ contains the ratings of all items given by all users in the system. The user profile, P_{U_i} of user U_i contains all the ratings given by that user to different items. The user for whom the recommendation is generated is called the active user and denoted by U_A.

B. User-based and item-based CF algorithms

Prediction generation using User-based [Resnick et al. 1994; Shardanand 1994] and item-based [Sarwar 2001] algorithms consist of two broad steps- Neighborhood generation (NG) and Prediction calculation (PC).

Neighborhood generation (NG)

The neighborhood generation phase can be further subdivided into similarity calculation (SC) and neighborhood selection (NS). In case of user-based algorithms, the similarity between the users are calculated and top N most similar users are chosen from the neighborhood of the active user, U_A. Two most common approaches for similarity calculation are Pearson correlation coefficient and cosine similarity.

Pearson correlation coefficient

[Resnick et al. 1994] first proposed to use Pearson Correlation coefficient to compute user similarity using (2). Here before the scalar product between two vectors is computed, ratings are normalized as the difference between real ratings and average rating of user.

$$s(a,b)=\frac{\sum_{i\in R(a,b)}(r_{a,i}-\bar{r}_a)\times(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i\in R(a,b)}(r_{a,i}-\bar{r}_a)^2}\times\sqrt{\sum_{i\in R(a,b)}(r_{b,i}-\bar{r}_b)^2}} \quad (1)$$

in which, $\bar{r}_x$ is the average rating of user x.

Two important variations of Pearson Correlation coefficient have been proposed by [Shardanand and Maes 1995] and [Herlocker et al. 1999]. [Shardanand and Maes 1995] have replaced average rating of the active user, U_A and the user with which similarity is calculated by the central rating value of the rating scale. Their constrained Pearson Correlation coefficient becomes

$$s(a,b)=\frac{\sum_{i\in R(a,b)}(r_{a,i}-r_C)\times(r_{b,i}-r_C)}{\sqrt{\sum_{i\in R(a,b)}(r_{a,i}-r_C)^2}\times\sqrt{\sum_{i\in R(a,b)}(r_{b,i}-r_C)^2}} \quad (2)$$

Where r_C is the central rating value of the rating scale (e.g. 3 in the rating scale of 1 to 5). By this change in the Pearson Correlation coefficient formula, the difference between the positive and negative ratings (with respect to central rating value in the rating scale) has been considered.

[Herlocker et al. 1999] considered the number of rating comparisons in calculating the similarity between two users. The idea is to give more importance to those neighbors who have put similar rating values for a large number of items. After adding the number of rating comparisons as weight to Pearson Correlation coefficient, their weighted Pearson Correlation coefficient becomes-

$$s(a,b)=\frac{\sum_{i\in R(a,b)}(r_{a,i}-\bar{r}_a)\times(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i\in R(a,b)}(r_{a,i}-\bar{r}_a)^2}\times\sqrt{\sum_{i\in R(a,b)}(r_{b,i}-\bar{r}_b)^2}}\times\frac{no_C}{T} \quad (3)$$

where $no_C < T$. no_C is the number of the common items rated by U_A and the user with which similarity is calculated, T is the threshold value which is determined experimentally. When $no_C \geq T$, the simple Correlation coefficient is used.

Cosine similarity

Vector Cosine method computes user similarity as the scalar product of rating vectors:

$$s(a,b)=\frac{\sum_{i\in R(a,b)} r_{a,i}\times r_{b,i}}{\sqrt{\sum_{i\in R(a,b)} r_{a,i}^2}\times\sqrt{\sum_{i\in R(a,b)} r_{b,i}^2}} \quad (4)$$

in which, $s(a,b)$ is the degree of similarity between user a and user b, $R(a,b)$ is the set of items rated by both user a and user b, $r_{x,y}$ is rating that user x gives to items y.

Prediction calculation(PC)

In this phase, recommendation is made considering the item ratings of the neighbors of U_A using formula (1).

$$\hat{r}_{a,j}=\bar{P}_{U_a}+\left(\sum_{i\_rates\_j,i\in NU}(r_{i,j}-\bar{P}_{NU_i})*\omega_{a,i}\right)*\left(\sum_i|\omega_{a,i}|\right)^{-1} \quad (5)$$

where $\bar{P}_{U_a}$ denotes the average ratings of active user U_A, $r_{i,j}$ is the actual rating of neighbor U_i on product I_j, $\bar{P}_{NU_i}$ denotes the average ratings of neighbor NU_i and $\omega_{a,i}$ denotes the correlation between the active user, U_A and its i^{th} neighbor, NU_i.

Item-based algorithms [Sarwar and Karypis 2001] follow the similar steps- instead of calculating the similarity between the users, the similarity between items are calculated and prediction is calculated considering the ratings for the most similar items of the target item for which the recommendation is made. In the similarity calculation(SC) phase, either pearson correlation

coefficient or cosine similarity measure mentioned above can be used. One particular difference in calculating similarity between users and between items is-in case of item similarity computation, each pair of rating values under comparison corresponds to a different user and rating pattern of different users are different. In order to take into this fact under consideration, Adjusted Cosine Similarity is introduced by [Sarwar and Karypis 2001] shown in formula (6).

Adjusted Cosine Similarity

$$s(a,b)=\frac{\sum_{i\in R(a,b)}\left(r_{a,i}-\overline{P_i}\right)\times\left(r_{b,i}-\overline{P_i}\right)}{\sqrt{\sum_{i\in R(a,b)}\left(r_{a,i}-\overline{P_i}\right)^2}\times\sqrt{\sum_{i\in R(a,b)}\left(r_{b,i}-\overline{P_i}\right)^2}} \quad (6)$$

Here, before comparing a rating pair, the average rating value of the corresponding user is deducted from each rating value.

In prediction calculation(PC) phase, prediction is calculated by formula ()

$$\hat{r}_{a,j}=\frac{\sum_{i,i\in NI}\ r_{a,i}*\omega_{i,j}}{\sum_{i,i\in NI}\ \omega_{i,j}} \quad (7)$$

Where *NI* is the set of similar items to item I_j and $\omega_{i,j}$ is the degree of similarity between I_i and I_j.

The time complexity of the basic user-based and item-based algorithms is O(nm) for N users and M items present in the system in addition to the space complexity of the order O(nm) for storing the entire rating matrix, R in memory.

C. Slope One

Slope one [Lemire and Maclachlan 2005] is an item-based algorithm for generating recommendations which is much simpler than the traditional collaborative filtering algorithm. The name "slope one" has been proposed due to the form of the predictor used in this scheme- $f(x)=x+b$. For obtaining the best predictor, the value of the parameter b is estimated as $b=\frac{\sum_i m_i-n_i}{n}$ by minimizing $\sum_i(m_i+b-n_i)^2$ where m_i and n_i are two evaluation arrays and $i=1,2,\dots,n$. So, the best value of the parameter b is the average difference between the two arrays m and n. The steps of slope one algorithm for predicting recommendation of item I_m for user U_k are as follows-

1. Identify the set of items, I_k, rated by user U_k.
2. Compute the mean difference, D_i, $i\in I_k$ between item I_i and item to predict, I_m considering only the commonly rated items.
3. Predicted rating of item I_m for user U_k is calculated as

$$P_{k,m}=\sum_{i\in I_k}\left(r_{k,i}+D_i\right)/\left|I_k\right| \quad (8)$$

In the above algorithm, when computing the mean difference, D_i, $i\in I_k$ between item I_i and item to predict I_m the number of users, $\left|U_{i,m}\right|$ who rated both the items I_i and I_m is not considered. When the value of $\left|U_{i,m}\right|$ is high, prediction quality will be better. To take into account this fact, [Lemire and Maclachlan 2005] had introduced weighted slope one algorithm where prediction is calculated as

$$P_{k,m}=\sum\left(\left(R_{k,i}+D_i\right)*\left|U_{i,m}\right|\right)/\left|I_k\right| \quad (9)$$

IV. Scalability through Model-based techniques

As we have seen in the previous section, the process of recommendation generation using collaborative filtering is a two stage process-neighborhood formation and prediction generation. In the memory-based CF approaches, both the phases are done online i.e. both the phases starts execution after a customer asks for a recommendation for a particular product.

Two characteristics of memory-based CF approaches are notable here from the point of view of scalability issue. Firstly, the whole user-item rating matrix is used in the first phase for finding k-nearest neighbors and secondly, predictions are made in the second phase directly using the rating matrix. The user-based and item-based collaborative filtering techniques that we have seen in the previous section are example of memory based techniques. Again, between the user and item based collaborative filtering techniques, the second one is more scalable compared to the first because the item-item similarity can be computed offline based on the assumption that new items are added less frequently to the system. So, the cost of deriving k-nearest neighbor items becomes less at online.

In case of model based techniques, a model is built from the rating matrix in the first phase and in the second phase prediction is made using that model without using the user-item rating matrix directly. The model building is done offline i.e. before any request is made for recommendation.

Different statistical and data mining methods have been used by the researchers for building the model. Some of the important model-based approaches proposed by them are discussed bellow.

Bayesian network

[Breese et al. 1998] explained collaborative filtering task from the probabilistic point of view

where recommendation generation is viewed as calculation of an expected value of a vote given some known votes for some other items in the system. They proposed construction of a bayesian network where each node corresponds to an item. For each rating value in the rating scale, a different state is assigned to every node. Through learning of the bayesian network dependencies for each node is searched and in the resulting network, each item will have a set of parent nodes which are the best predictors of its votes.

Neural Networks

In their work, [Billsus and Pazzani 1998] build a neural network based model for generating recommendations. They have also used Singular Value Decomposition on the user-item rating matrix as a feature extraction technique. For training of the model, the sparse user-item rating matrix is first converted into Boolean feature vectors from which a binary matrix containing zero and one is obtained. Then Singular Value Decomposition is performed on this training data and the resulting singular vectors are reduced by choosing the value of k, the desired number of dimensions to retain. The neural network is then trained by the singular vectors scaled by the singular values.

Using this model the prediction is made as follows. First, the target item's user ratings are converted into a Boolean feature vector and scaled into the k-dimensional space. Then this vector is fed into the neural network for getting the prediction.

Clustering

[Kohrs and M´erialdo 1999] proposed a model where hierarchical clusters of both the users and the rated items are built. The root node of the user cluster hierarchy contains all users and other nodes in the hierarchy contain users of the subsequent nodes along the hierarchy. Each user appears in only one leaf node. The cluster hierarchy is formed in such a way that the similarity between users of a cluster increases as one travels from root node to a leaf node along the hierarchy.

Following the similar logic, the item hierarchy is also formed. Once both the user and the item cluster hierarchies are formed, the prediction can be calculated as the weighted sum of the defined centers of all nodes in the cluster hierarchies on the paths from the root node to particular leaves.

Other than the works mentioned above, some other significant works in this direction are done by [Billsus and Pazzani 1998; Sarwar et al. 2000] (singular Value Decomposition based), [Rennie and Srebro 2005] (matrix factorization based), [Canny 2002](factor analysis based), [Goldberg et al. 2001](PCA based) and [Aggarwal et al. 1999](graph based).

Memory based approaches are much simpler than their model-based counterparts. But they greatly suffer from the scalability problem. In the model-based techniques, models that are built from the rating matrix are relatively complex in nature and though they are derived offline, building them is obviously expensive.

Combined approach

The model-based and memory-based approaches can be combined in many ways. [Pennock et al. 2000] have proposed a collaborative filtering method which is a combination of memory- and model-based approaches. They named it as personality diagnosis (PD) as the rating preferences given for different items by a user is seen as his or her underlying personality type. User ratings are assumed to contain Gaussian errors. The probability that active user has the same personality type as every other user is computed from the known rating of that active user. Then the probability of liking a new product is computed. In order to reduce the space and time complexity, expected value of information(VOI), as they name, is computed from their probabilistic model.

In their approach, [Koren 2008] devised a collaborative filtering algorithm which combines both the latent factor model and the nearest neighbor-based algorithm. Their proposed algorithm exploits strong points from both the sides-very localized relationship identification from nearest neighbor-based algorithm and effective estimation of the overall structure that relates simultaneously to most or all items from latent factor models. Instead of combining model-based and memory-based approaches, we can improve the scalability of the memory-based algorithms by applying clustering algorithms to them which will be discussed in detail in the next section.

V. Scalability through clustering

Clustering techniques have been used in different ways to memory-based collaborative filtering algorithms for making the solution scalable. We group clustering based collaborative filtering algorithms into three categories.

Clustering users

Based on their profiles, users are grouped into different user clusters. The general framework is as follows-

Step 1: apply clustering algorithm to user-item rating matrix to group users into multiple clusters.

Formally, the data set A is partitioned in $A_1, A_2, \ldots, A_p$, where $A_i \cap A_j = \varphi$, for $1 \leq i, j \leq p$; and $A_1 \cup A_2 \cup \ldots \cup A_p = A$.

Step 2: when the active user,U_A asks for recommendation, the cluster to which he belongs, say A_m is selected.
Step 3: k-nearest neighbors of that active user is selected from the selected cluster, A_m.
Step 4: prediction is calculated from the ratings of the nearest neighbors.
Offline time-complexity becomes $O(mn)$. Online time-complexity becomes $O(mn/k)$. If the similarity between each user and the members of the cluster to which he belongs is calculated offline (step 3) then online time complexity will be $O(m)$ but additional $k(n/k)^2$ similarity values have to be stored. So space complexity will be $O(n^2/k)$.

Some important contributions where user clustering has been proposed are discussed bellow. In his paper, [Sarwar et al. 2002] considered the entire cluster to which the active user belongs as the neighborhood of that user. Then prediction calculation is done using the standard collaborative filtering algorithm. [Rashid et al. 2007] proposed CLUSTKNN algorithm where user profiles are clustered using bisecting k-means algorithm. For recommendation generation, k most similar cluster centroids (they name as surrogate model users) to the active user's profile are selected as neighbors. Offlline time-complexity becomes $O(mn)$ and online time-complexity becomes $O(mn/k)$.
Some other papers where clustering of user profiles have been performed are [Rana and Jain xxxx], [Kelleher and Bridge 2003], [Yang 2009] , [Zhang et al. 2006], [Moghaddam and Selamat 2011] , [Xiaohui and Murata 2012].

Item clustering

For improving scalability, items can also be clustered instead of users. The main intuition behind clustering items is to consider only rating of the similar type of items for which the recommendation is going to be generated. The general framework is as follows-
Step 1: apply clustering algorithm to user-item rating matrix to group items into multiple item clusters.
Step 2: when a user asks for a recommendation, the cluster to which the item belongs is selected.
Step 3: k-nearest neighbors of that item is selected from that item cluster.
Step 4: prediction is calculated from the ratings of the nearest neighbors.
[Quan and Khanh 2006] proposes clustering of items in such a way that inside a cluster, user similarity remains stable i.e. similarity between users does not change significantly. At the time of predicting rating for an item, ratings of users who have high similarity degree with that user inside the cluster to which that item belongs are considered.
[Li and Kim 2003], in their work, has applied fuzzy clustering algorithm to create group-rating matrix. Then they have calculated the sub-similarity of group-rating matrix and item-rating matrix. Finally, the prediction is generated from the total similarity calculated as linear combination of two subsimilarities obtained before.
[Birtolo et al. 2011] has also applied fuzzy clustering algorithm to group similar items. The k-nearest neighbor items of the target item have been selected from different clusters. The number of items that has been selected from a specific cluster depends on the degree of membership with which the target item belongs to that cluster.

VI. Empirical Analysis

In this section we present our experimental results of analyzing the effect of applying clustering to different collaborative filtering algorithms namely user-based, item-based and Slope-One algorithm.

A. Datasets

In our experiment, we have used MovieLens dataset (movielens.umn.edu). The data set used contained 100,000 ratings from 943 users and 1682 movies (items), with each user rated at least 20 items. The item sparsity is easily computed as 0.9369. The ratings in the MovieLens dataset are integers ranging from 1 to 5.

B. Evaluation Metrics

Mean Absolute Error (MAE)
MAE is defined as

$$MAE = \frac{1}{n}\sum_{(i,j)}\left|\hat{r}_{i,j} - r_{i,j}\right| \tag{10}$$

where $\hat{r}_{i,j}$ and $r_{i,j}$ are the predicted and actual ratings of the jth product for the ith user and n is the total number of recommendations made by the system. This statistical metric measures the extent of deviation of the scores predicted by the recommender system from the actual user rating values. Smaller MAE indicates better recommendation.

Root Mean Square Error (RMSE)
RMSE imposes more emphasis on large errors. It is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{(i,j)}(\hat{r}_{i,j} - r_{i,j})^2} \tag{11}$$

Neighborhood Stability
For identifying change in the neighborhood of the active user due to clustering, we measure Neighborhood Stability (NS) as

$$NS = \frac{\left|NH_{WOUTC} \cup NH_{WC}\right|}{\left|NH_{Size}\right|} \tag{12}$$

where NH_{WOUTC} is the neighborhood of the active user without clustering and NH_{WC} is the neighborhood of the active user with clustering.

Speed Up

Speed Up(S_p) is defined as

$$S_p = \frac{|T_{WC} - T_{WOUTC}|}{T_{WOUTC}} \times 100 \quad (13)$$

where T_{WOUTC} is the average time required for generating a single recommendation without clustering and T_{WC} is the average time required for generating a single recommendation with clustering.

c. Results

For all the experiments, neighborhood size has been taken as 10,20,30,40 and 50. Number of clusters has been considered as 10,15,20,30 and 40. All experiments have been done in a computer with Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz processor (6 cores) and 16 GB of primary memory. Figure 1(a) and 1(b) shows the MAE and RMSE values of user-based collaborative filtering algorithms for different neighborhood size and for different number of clusters. Pearson correlation coefficient has been used as a metric for calculating the distance between user profiles. In case of figure 2(a) and 2(b), cosine similarity has been used as distance metric. In all the experiments, K-means algorithm has been used for clustering users and items. Results of figure 1(a) and 1(b)

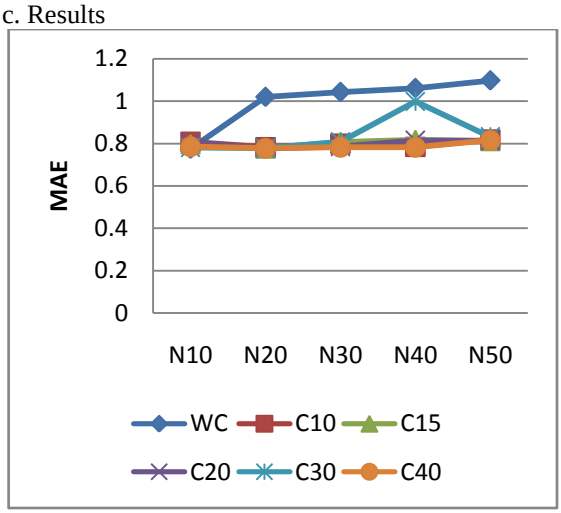


Figure 1(a): MAE for User-based CF (PCC)

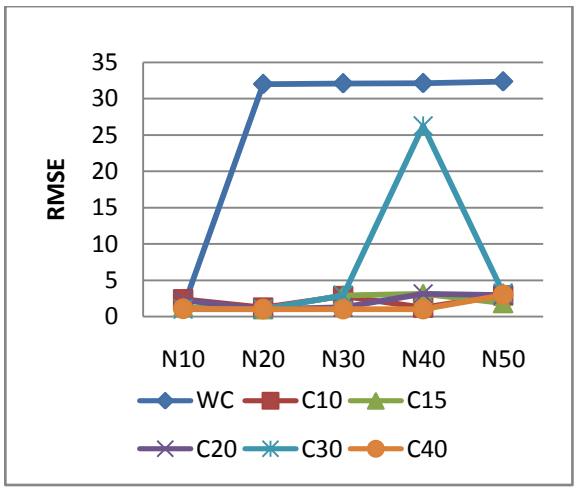


Figure 1(b): RMSE for User-based CF (PCC)

show significant improvement in accuracy of user-based collaborative filtering algorithm when pearson correlation coefficient(PCC) is used. When cosine similarity is used, accuracy suffers significantly due to clustering. When comparing reduction in average recommendation time as a result of applying clustering to user profiles, user-based collaborative filtering using cosine similarity outperforms user-based collaborative filtering using pearson correlation coefficient. Change in the neighborhood of the active user is less (i.e. stability of the neighborhood is high) when using cosine similarity.

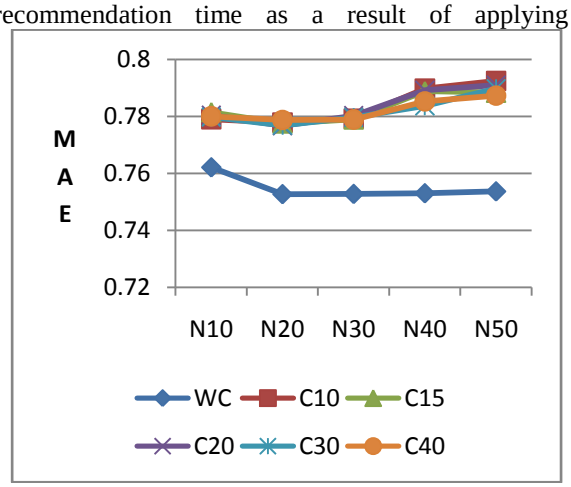


Figure 2(a): MAE for User-based CF (COSINE)

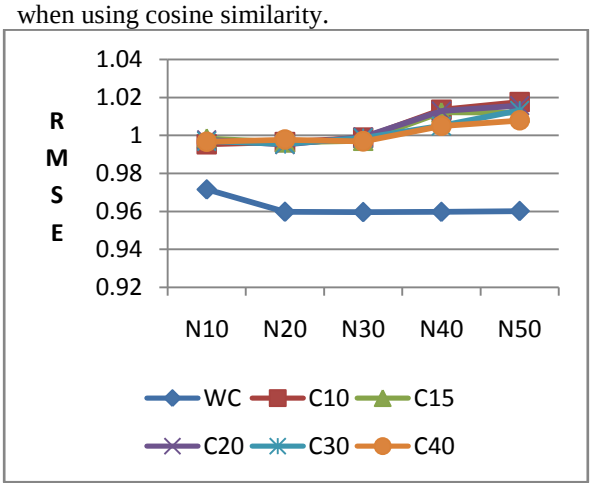


Figure 2(b): RMSE for User-based CF (COSINE)

Reduction in average execution time for generating a single recommendation is more while using cosine similarity metric. From figure 3(a) and 3(b) it can be seen that maximum speed up is for neighborhood size 10 (for different number of clusters) and it is 4% higher in case of cosine based similarity than in case of PCC based similarity.

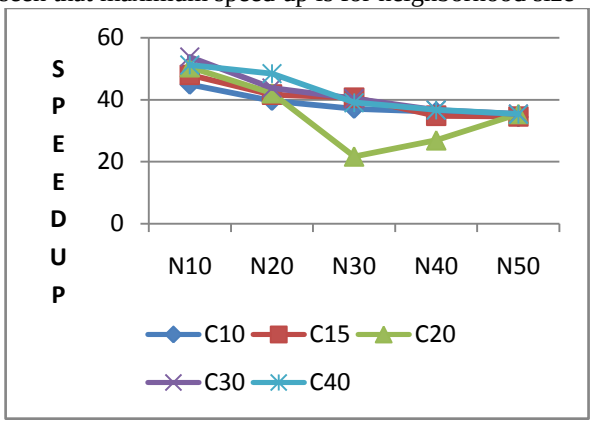


Fig 3(a). Speed Up User-based CF (PCC Distance measure)

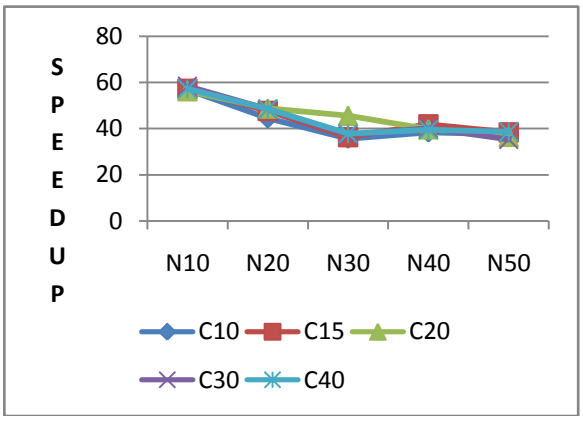
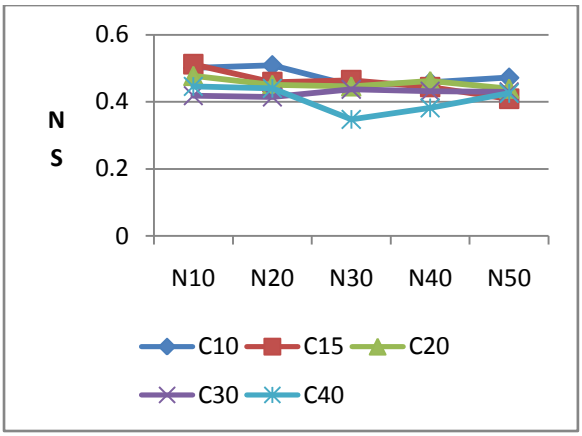


Fig 3(b). Speed Up User-based CF (Cosine Distance measure)



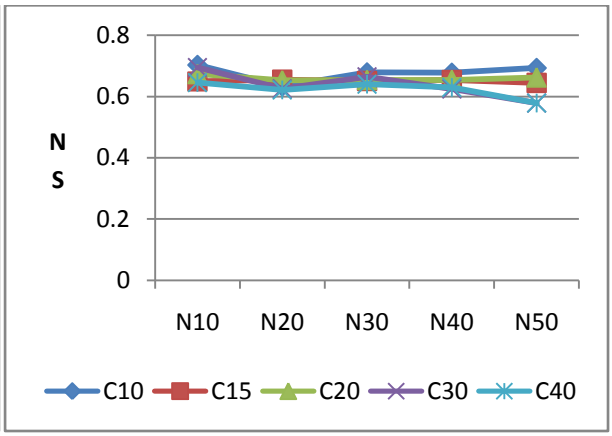


Fig 4(a). Neighborhood Stability(NS) for User-based CF (PCC Distance measure) Fig 4(b). Neighborhood Stability(NS) for User-based CF (Cosine Distance measure)

Figure 4(a) and 4(b) shows the Neighborhood Stability (NS) of user-based algorithms for PCC based and Cosine based distant metric respectively. For all combinations of clustering size and neighborhood size, CF algorithm using Cosine similarity shows better Neighborhood Stability (NS) than their PCC based counterpart. This result is also supported in Figure 1(a)-2(a) and 1(b)-2(b). The difference of MAE and RMSE values between clustering-based CF and CF without clustering is much higher in case of CF using PCC similarity metric than their cosine based counterpart.

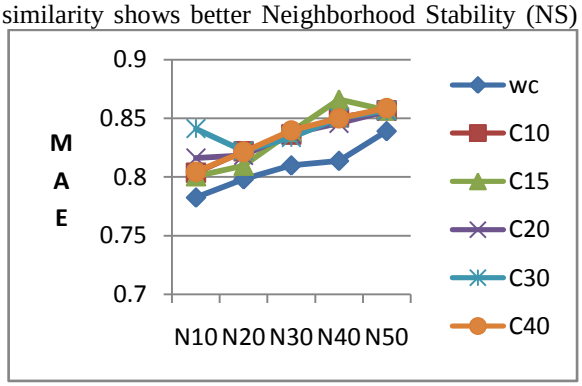


Figure 5(a): MAE for Item-based CF (PCC Distance measure)

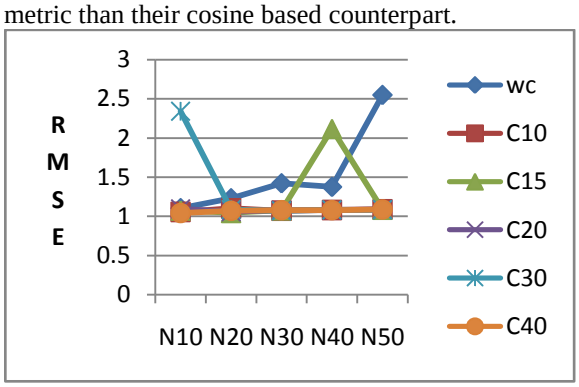


Figure 5(b): RMSE for Item-based CF (PCC Distance measure)

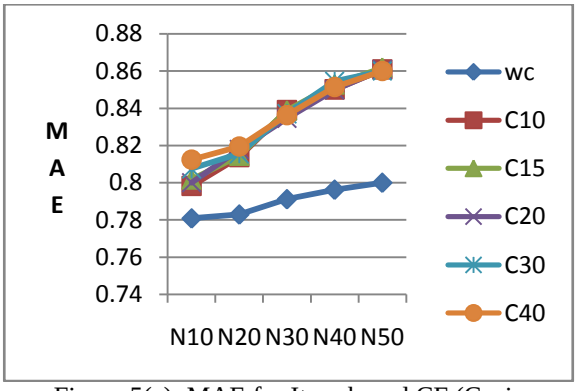


Figure 5(a): MAE for Item-based CF (Cosine Distance measure)

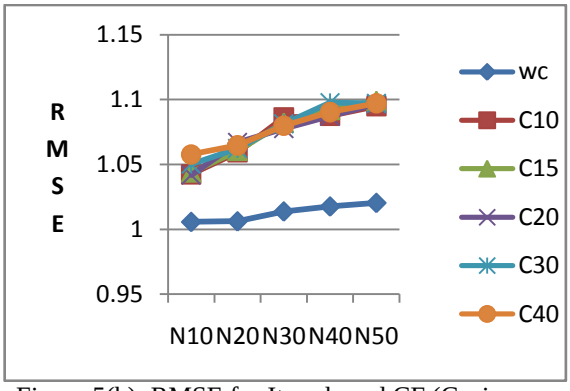


Figure 5(b): RMSE for Item-based CF (Cosine Distance measure)

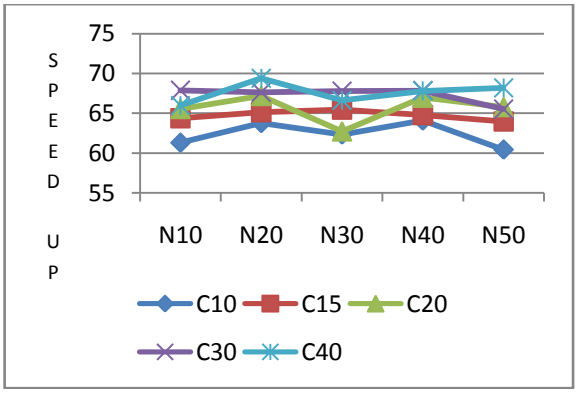


Fig 6(a). Speed Up Item-based CF (PCC Distance measure)

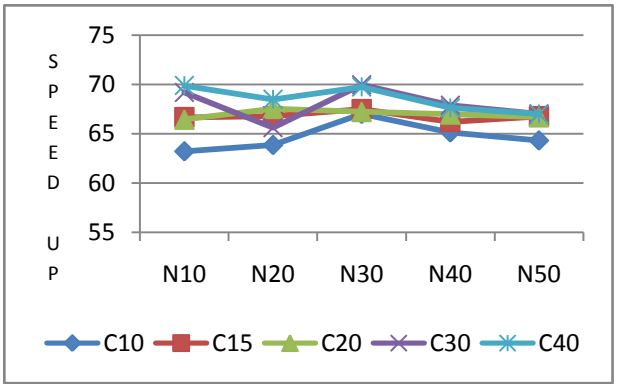


Fig 6(b). Speed Up Item-based CF (Cosine Distance measure)

Comparison of figures 6(a) and 6(b) with figures 3(a) and 3(b) shows that higher scalability is achieved in item-based CF algorithms over user-based CF algorithms while applying k-means clustering algorithm. Like user-based CF algorithm, item-based CF algorithm also shows better speed up while using cosine similarity metric. From figure 7(a) and 7(b), it can be observed that in case of item-based CF algorithm like its user-based counterpart, use of cosine similarity shows better network stability than PCC-based approach. For both user-based and item-based algorithms, consistently good network stability is shown for 10 numbers of clusters.

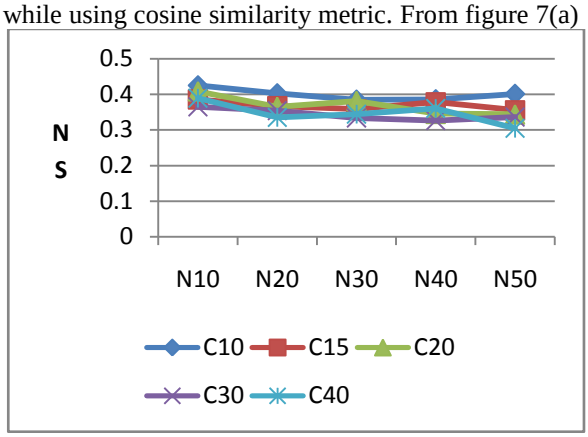


Fig 7(a). Neighborhood Stability(NS) for Item-based CF (PCC Distance measure)

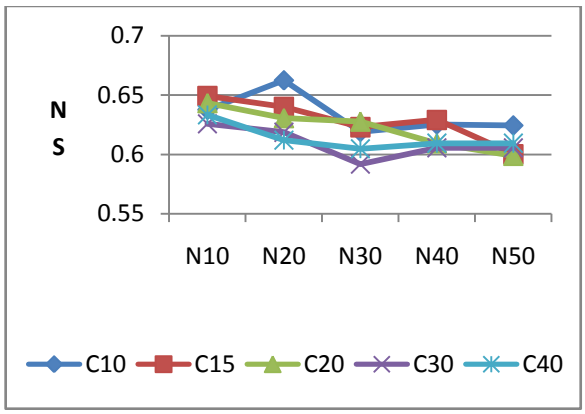


Fig 7(b). Neighborhood Stability(NS) for Item-based CF (Cosine Distance measure)

Figure 8(a), 8(b) and 8(c) shows MAE, RMSE and Speed Up of Slope One algorithm for different number of clusters. Application of clustering increases accuracy to a great extent (figure 8(a), 8(b)). Average time for recommendation generation also decreases significantly (figure 8(a)).

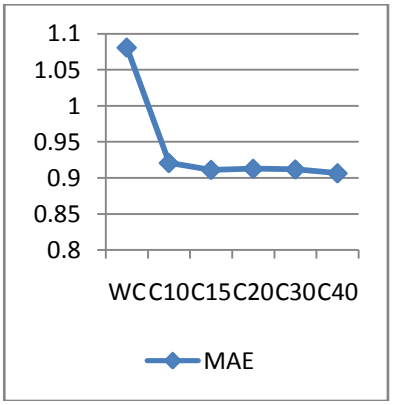


Fig 8(a). Slope One (MAE)

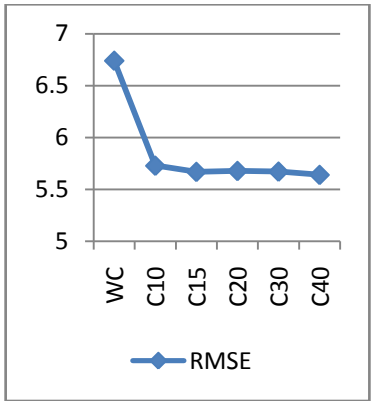


Fig 8(b). Slope One (RMSE)

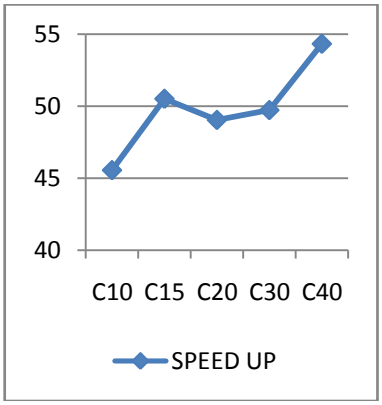


Fig 8(c). Slope One (Speed Up)

VII. Conclusion

In this paper, we have studied and analyzed three collaborative filtering algorithms namely user-based, item-based and slope-one algorithm. We have compared accuracy as well as speed up of these three algorithms for different neighborhood size and for different number of clusters. Two metrics have been used for calculating the similarity between the users or items. K-means algorithm has been applied in all the cases. In our future work, we like to apply other clustering algorithms and observe their effect on scalability of recommender systems.

References

[1.] [Aggarwal et al. 1999] Aggarwal, C. C., Wolf, J. L., Wu, K.-L., and Yu, P. S. 1999. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99). ACM, New York, NY, 201–212.

[2.] [Billsus and Pazzani 1998] Billsus, D. and Pazzani, M. J. 1998. Learning collaborative information filters. In Proceedings of the 15th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, 46–54.

[3.] [Birtolo et al. 2011] Birtolo, C., Ronca, D., Armenise, R., & Ascione, M. (2011). Personalized suggestions by means of collaborative filtering: A comparison of two different model-based techniques, In NaBIC, IEEE (pp. 444-450).

[4.] [Breese et al. 1998] Breese, J. S., Heckerman, D. & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, 43–52.

[5.] [Canny 2002] Canny, J. 2002. Collaborative filtering with privacy via factor analysis. In Proceedings of the 25th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval (SIGIR'02). ACM, New York, NY, 238–245.

[6.] [Chee et al. 2001] S. H. S. Chee, J. Han, and K. Wang, "RecTree: an efficient collaborative filtering method," in *Proceedings of the 3rd International Conference on DataWarehousing and Knowledge Discovery*, pp. 141–151, 2001.

[7.] [Darvishi-mirshekarlou et al. 2013] F.Darvishi-mirshekarlou, SH.Akbarpour, M.Feizi-Derakhshi, Reviewing Cluster Based Collaborative Filtering Approaches, International Journal of Computer Applications Technology and Research , ISSN: 2319–8656, Volume 2– Issue 6, 650 - 659, 2013.

[8.] [Goldberg et al. 2001] Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Inform. Retr. 4*, 2, 133–151.

[9.] [Herlocker et al. 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, NY, 230–237.

[10.] [Kelleher and Bridge 2003] Kelleher.J and D. Bridge.:" Rectree :An accurate, scalable collaborative recommender". In Procs. of the Fourteenth Irish conference on Artificial Intelligence and Cognitive Science, 2003 pages: 89–94.

[11.] [Kohrs *and* Merialdo *1999*] A. *Kohrs* and *B. Merialdo. Clustering* for *co llaborative filtering applications*.

In*Computational Intelligence* for *Modelling, Control* & *Automation 1999*, pp. 199–204, Amsterdam, (*1999*). *IOS Press*.

[12.] [Koren 2008] Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08). ACM, New York, NY, 426–434.

[13.] [Lee et al. 2012] J. Lee, M. Sun, and G. Lebanon. A comparative study of collaborative filtering algorithms. ArXiv Report 1205.3193, 2012b.

[14.] [Lemire And Maclachlan 2005] Lemire, D. and Maclachlan, A. 2005. Slope One Predictors for Online Rating-Based Collaborative Filtering, In SIAM Data Mining (SDM'05), Newport Beach, California, April 21-23, 2005.

[15.] [Li and Kim 2003] Qing Li, Byeong Man Kim. Clustering Approach for Hybrid Recommender System, In Proc. of the 2003 IEEE/WIC International Conference on Web Intelligence (WI 2003), pp. 33-38 (2003).

[16.] [Moghaddam and Selamat 2011] Siavash Ghodsi Moghaddam, Ali Selamat, "A Scalable Collaborative Recommender Algorithm Based on User DensityBased Clustering", 3rd international conference on Data Mining and Intelligent Information Technology Applications (ICMiA), IEEE 2011, pp. 246-249.

[17.] [Pennock et al. 2000] Pennock, D.,Horvitz, E., Lawrence, S., and Giles, C. L. 2000. Collaborative filtering by personality diagnosis:A hybrid memory- and model-based approach. In Proceedings of the 16th Conference on Uncertainty inArtificial Intelligence (UAI'2000). 473–480.

[18.] [Quan and Khanh 2006] Improving Accuracy of Recommender System by Clustering Items Based on Stability of User Similarity. Quan, Truong Khanh. Tokyo : s.n., 2006. Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on.

[19.] [Rana and Jain xxxx] Chhavi Rana and S K Jain "An extended evolutionary clustering algorithm for an adaptive recommender system" Social network Analysis and Mining, Volume 4, No. 1, Article: 164.

[20.] [Rashid et al. 2007] Al Mamunur Rashid, Shyong K. Lam, Adam LaPitz, George Karypis,and John Riedl, "Towards a Scalable kNN CF Algorithm:Exploring Effective Applications of Clustering," In:Advances in Web Mining and Web Usage Analysis (WebKDD 2006), Springer-Verlag Berlin Heidelberg, 2007, pp. 147-166, doi: 10.1007.

[21.] [Rennie and Srebro 2005] Rennie, J. D. M. and Srebro, N. 2005. Fast maximummarginmatrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*. ACM, New York, NY, 713–719.

[22.] [Resnick et al. 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported* Cooperative Work (CSCW'94). ACM, New York, NY, 175–186.

[23.] [Revankar and Haribhakta 2015] Omkar S. Revankar, Dr.Mrs. Y.V.Haribhakta Survey on Collaborative Filtering Technique in Recommendation System, ISSN. 2319 – 4847, Volume 4, Issue 3, March 2015.

[24.] [Sachan and Richariya 2013] Atisha Sachan, Vineet Richariya ―A Survey on Recommender Systems based on Collaborative Filtering Technique‖, International journal of Innovations in Engineering and technology (IJIET), ISSN. 2319- 1058, Volume 2 Issue 2, April 2013.

[25.] [Sarwar 2001] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web (WWW'01). ACM, New York, NY, 285–295.

[26.] [Sarwar et al. 2000] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000. Application of dimensionality reduction in recommender systems–a case study. In Proceedings of the ACM WebKDD Workshop.

[27.] [Sarwar et al. 2002] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the Fifth International Conference on Computer and Information Technology, pages158–167, 2002.

[28.] [Shardanand 1994] Shardanand, U. 1994. Social information filtering for music recommendation. M.S. thesis, Massachussets Institute of Technology.

[29.] [Shardanand and Maes 1995] Shardanand, U. and Maes, P. 1995. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, 210–217.

[30.] [Su and Taghi 2009] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in Artificial Intelligence 2009 (2009): 4.

[31.] [Takacs et al. 2009] Gabor Takacs, Istvan Pilaszy, Bottyan Nemeth, and Domonkos Tikk, "Scalable Collaborative Filtering Approaches for Large Recommender Systems," The Journal of Machine Learning Research, vol. 10, pp. 623-656, 2009.

[32.] [Xiaohui and Murata 2012] Xiaohui Li Murata, T., "Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity" Web Intelligence and Intelligent Agent Technology (WIIAT), 2012.

[33.] [Yang 2009] Wujian Yang, Zebing Waug and Mingyu You, An improved Collaborative Filtering Method for Recommendiations' Generation, In Proceedings of IEEE International Conierence on Systems, Man and Cybernetics, 2009.

[34.] [Zhang et al. 2006] Feng Zhang , Hui-you Chang, A Collaborative Filtering Algorithm Employing Genetic Clustering to Ameliorate the Scalability Issue, Proceedings of the IEEE International Conference on e-Business Engineering, p.331-338, October 24-26, 2006.