IJCTA, 9(27), 2016, pp. 127-136 ISSN: 0974-5572
© International Science Press

A Study and Analysis of Collaborative Filtering Algorithms for Recommender Systems

M. Anbazhagan* and Michael Arock**

ABSTRACT

In the E-commerce world, recommender systems are becoming increasingly popular. Collaborative Filtering, one of the most successful approaches in building Recommender Systems uses ratings/preferences expressed by a group of users who are similar to the target user, based on some agreement. The main objective of using their ratings is to predict ratings for items not seen by the target user and to recommend items that the target user is likely to buy. In this paper, we present a brief discussion on how recommendations/predictions are generated in Collaborative Filteringand its challenges. We then present two types of Collaborative Filtering techniques namely Memory-based CF and Model-based CF, and two efficient representative algorithms for each type. The algorithms presented for Memory-based CF are User-based and Item-based and for Model-based CF are Tendency-based CF and Regularised Singular Value Decomposition. Finally, we attempt to present a discussion on comparison of all these algorithms based on their prediction accuracy, computational efficiency, and ability to tackle challenges such as data sparsity, scalability, cold-start problem, and so on.

Keywords: Collaborative Filtering, Memory-based CF, Model-based CF, User-based CF, Item-based CF, Tendencies-based CF Method, Regularised Singular Value Decomposition.

1. INTRODUCTION

Recommender systems have gained importance ever since E-commerce sites came into existence. They implement the natural social process of giving recommendations from other people by word-of-mouth. Literally speaking, a system which enthusiastically recommends a product to purchase, a service to subscribe, or a movie to watch and so forth can be regarded as a recommender system.They are classified into the following categories.

Content-Based Filtering stores feature description about each item to be recommended. This information will be used to recommend items similar to those previously viewed or purchased by the user, based on how similar certain items are to each other or the similarity with respect to user preferences[1].

Collaborative Filtering predicts preferences that a user is likely to give based on preferenceinformation from many similar users. The fundamental assumption that the collaborative filtering works with is, if two persons P and Q have same interest about an item, then P is more likely to have the same interest as Q about a different item x than to have the interest of a randomly chosen person.

Hybrid Recommendationapproaches combine Content-based filtering and Collaborative filtering, using the users' preferences, user and item information. Such Hybrid systems have better prediction accuracy than Content-based filtering and Collaborative filtering systems taken alone.

In this paper, we present two subtypes of Collaborative Filtering Approach such as Memory based CF and Model Based CF. Representative algorithms for each type are also presented along with their advantages and

* Department of Computer Applications, NIT-Trichy

** Department of Computer Applications, NIT-Trichy

disadvantages.The paper is structured as follows. Section 2 briefly describes the formulation of CF problem, types of CF, challenges faced by CF, and metrics used for evaluating the performance of CF. In Section 3, we present how predictions are generated in Memory-based CF, and the algorithms for User-based and Item-based CF. Section 4 focuses on Model-based CF and its two efficient representative algorithms namely Tendency Based Collaborative Filtering Method and Regularised Singular Value Decomposition. The heart of this paper lies in Section 5, because it attempts to present a discussion on comparison of all four algorithms presented in Section 3 and 4 in terms of their prediction accuracy, computational efficiency, and ability to tackle challenges such as data sparsity, scalability, cold-start problem, and so on.

2. COLLABORATIVE FILTERING

CF approaches use a large collectionof ratings or preferences for items given by users to predict what products a new user would like. In a typical CF approach, there is a list of *m* users $\{u_1, u_2, \ldots, u_m\}$ and a list of *n* items $\{i_1, i_2, \ldots, i_n\}$, and each user, u_i, has a list of items, Iu_i, rated by him,or their ratings/preferences have been obtained through their behaviours. The preferences about items can be either explicitly collected from users (numerical rating on a scale 1-5 or binary rating as like/dislike),or implicitly derived from data sources such as purchase records or weblogs, thereby making use of the data collected for other purposes [2].

As a first step in CF, the list of users and the items they rated can be converted into a user-item ratings matrix (Table 1), in which user U_4 is the active user to whom we want to make recommendations. There are missing values in the matrix where users did not give their preferences for certain items. Here, the problem of CF can be formulated as the problem of predicting missing values in user-item matrix. Sometimes CF can also recommend Top-N items to the active user i.e., a set of N top-ranked items that will be of interest to the active user.

Table 1
User-Item Matrix

	I_1	I_2	I_3	I_4
U_1	5	2	5	4
U_2	2	5	-	3
U_3	2	2	4	2
U_4	5	1	5	?

CF approaches are expected to be capable of dealing with highly sparse datasets, scaling with the extended numbers of users and items, making accurate recommendations in a short span of time, and dealing with other problems like synonymy, shilling attacks, data noise, and privacy protection problems [3].

Memory-based CF methods use the user rating data to determine the similarity between usersor items(neighbourhood based methods) and make predictions or recommendations according to similarity values determined [4]. This memory-based CF is widely deployed into commercial systems because of their ease of implementation and high effectiveness. But, memory-based CF behaves inefficiently when the dataset is sparse. Model-base CF methods were introduced in order to overcome this shortcoming of Memory-based CF methods. Unlike Memory-based CF methods, model-based CF methods use a part of the data as a training set to build a model and then the built model is used to make the predictions [5]. The Taxonomy of Collaborative Filtering Approach is presented in Figure 1.

2.1. Challenges faced by Collaborative Filtering

Generally, a recommender system giving high quality recommendations will attract the customers' interests and bring benefits to companies. Providing high quality and accurate recommendations heavily depends on how CF addresses certain challenges.

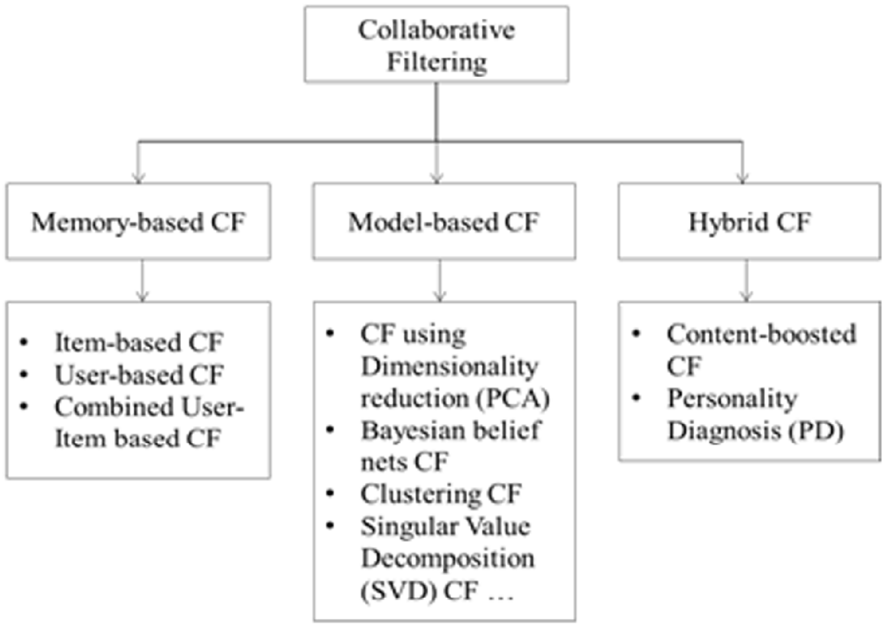

Figure 1: Collaborative Filtering Taxonomy

a. Data Sparsity: The user-item matrix used inCF systemsisexceedingly sparse. Hence,producing high quality predictions or recommendations is a challenging task. Data sparsity appears when a new user or item has just entered the system. In this occasion, it is difficult to find the similarity between newly entered user/item and existing ones. This situation is referred to as *Cold-Start* problem. The situation in which, the number of users' ratings is very less compared to the number of items is termed as *reduced-coverage* problem.

b. Scalability: A CF algorithm with tens of millions of customers (m) and millions of distinct items (n) possesses the complexity of O(n), which is already too large. Many systems need to immediately make recommendations for all users irrespective of their purchase activities, which demands a highly scalable CF system [6].

c. Synonymy: Synonymy refers to a situation of a number of similar items to have different names or entries. CF systems are unable to find the relationship between them and hence treat them differently. For example, "children movie" and "kids' movie" are in fact same but seems to be treated differently by CF.

d. Gray Sheepand Black Sheep: Gray sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering [7].Black sheep are the opposite group whose idiosyncratic tastes make recommendations nearly impossible [7].

e. Shilling Attacks: In a Recommender system where anyone can provide recommendations irrespective of his/her purchase behaviour, people may give good number of positive recommendations for their own products and a lot of negative recommendations for their competing products.CF systems should be cautious enough to prevent this kind of phenomenon from occurring.

2.2. Evaluation Measures

One of the most important evaluation metrics is accuracy. With accuracy we can measure how well a recommender system gives predictions/recommendations. Accuracy is measured by means of Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE).

$$MAE = \frac{1}{n}\sum_{u,i}\left|p_{u,i} - r_{u,i}\right|$$

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

where $p_{u,i}$ and $r_{u,i}$ are the predicted and observed rating for user u and item i, respectively.

Other evaluation metrics are *Coverage* (A measure of percentage of item for which a Recommender system can give predictions/recommendations) and *ROC-Sensitivity* (A measure of the diagnosticpower of a Recommender system).

3. MEMORY-BASED COLLABORATIVE FILTERING ALGORITHMS

CF algorithms that use the entire or a sample of the user-item database to generate a prediction/recommendation are called Memory-based CF algorithms. For these algorithms to work every user should be a part of a group of people with similar interests. Prediction/Recommendation for an active user can be generated by identifying the so-called neighbours. This neighbourhood-based CF algorithm uses the following steps:

Step 1: Similarity Computation

This step is to calculate the similarity or weight, $w_{i,j}$, which reflects correlation, or weight, between two users or two items, i and j. Different methods to compute similarity are given below.

For User-based CF, Pearson correlation betweentwo users u and v is

$$W_{u,v} = \frac{\sum_{i\in I}\left(r_{u,i} - \bar{r}_u\right)\left(r_{v,i} - \bar{r}_v\right)}{\sqrt{\sum_{i\in I}\left(r_{u,i} - \bar{r}_u\right)^2}\sqrt{\sum_{i\in I}\left(r_{v,i} - \bar{r}_v\right)^2}}$$

where the $i \in I$ summation is ranging over the items that both the users u and v have rated and $\bar{r}$ is the average rating of the co-rated items of the u^{th} user.

For Item-based CF, the set of users $u \in U$ who rated both items i and j, then the Pearson Correlation will be

$$W_{i,j} = \frac{\sum_{u\in U}\left(r_{u,i} - \bar{r}_i\right)\left(r_{u,j} - \bar{r}_j\right)}{\sqrt{\sum_{u\in U}\left(r_{u,i} - \bar{r}_i\right)^2}\sqrt{\sum_{u\in U}\left(r_{u,j} - \bar{r}_j\right)^2}}$$

where $r_{u,i}$ is the rating of user u on item i, $r_{\underline{i}}$ is the average rating of the i^{th} item by those users.

If R is the $n \times m$ user-item matrix, then the similarity between two items, i and j, is defined as the cosine of the n dimensional vectors corresponding to the ith and j^{th} column of matrix R.

Vector cosine similarity between items i and j is given by

$$W_{i,j} = \cos\left(\vec{i}, \vec{j}\right) = \frac{\vec{i} \bullet \vec{j}}{\left\|\vec{i}\right\| * \left\|\vec{j}\right\|}$$

where "•" denotes the dot-product of the two vectors.

Step 2: Producing Predictions

Predictions/Recommendations for the active user is produced by taking the weighted average of all the ratings of the user on a certain item, or using a simple weighted average [8]. The top-N recommendations can be generated by finding the k most similar users or items after finding the similarities between items or users, and then aggregation of the neighbours is done to get the top-N recommendations.

To generate a recommendation for the active user, a, on a certain item, *i*, we can take a weighted average of all the ratings on that item as follows [9]:

$$P_{a,i} = \bar{r}_a \frac{\sum_{u \in u}\left(r_{u,i} - r_u\right) \cdot W_{a,u}}{\sum_{u \in u}\left|w_{a,u}\right|}$$

The simple weighted average can be used to predict the rating, $P_{u,i}$, for user *u* on item *i* is given below [8]

$$P_{u,i} = \bar{r}_a \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N}\left|w_{i,n}\right|}$$

where the summation is ranging over all other rated items $n \in N$ for user *u*, $w_{i,n}$ is the weight between items *i* and *n*, $r_{u,n}$ is the rating for user *u* on item *n*.

The following section presents two representative algorithms for User-base CF and Item-based CF each for one.

3.1. User-based CF Algorithm

User-based algorithms find other users whose past preferences are similar to that of the active user and use their preferences on other items, to predict what the active user would like. The algorithm for User-based CF is given below.

Algorithm 1 User-based Collaborative Filtering

Input: User-Item Rating matrix *R*

Output: Prediction of an Item that the Active user u would like

Const *v*: Maximum number of users in *N(u)*, the neighbours of user *u*

For each user *u* Do

  Set *N(u)* to the *v* users most similar to user *u*

  For each item *i* that user *u* has not rated Do

    Calculate the Weighted Combination of ratings given to item *i* by neighbours $N_i(u)$

  End

  Recommend to user u the item with the highest predicted rating P_{ui}

End

Similar users are obtained by using a similarity function as described in Section 3

3.2. Item-based CF Algorithm

Item-based CF is one of the most widely deployed Collaborative Filtering techniques today. As its name determines, Item-based CF uses similarities between the rating patterns of items, instead of using similarity between users. Item-based CF algorithms will only recommend those items that are similar to the items that the active user have purchased or viewed in past.

Algorithm 2 Item-based Collaborative Filtering

Input: User-Item Rating matrix *R*
Output: Prediction of an Item that the Active user u would like
Const *j* : Maximum number of items in *N(i)*, the neighbours of item *i*
For each item *i* Do
    Set *N(i)* to the *j* items most similar to item *i*
    For each user *u* that has no rating for item *i* Do
        Calculate the Weighted Combination of ratings of user u in neighbours $N_u(i)$
    End
    Recommend to user *u* an item with the highest predicted rating *Pui*
End

4. MODEL-BASED COLLABORATIVE FILTERING ALGORITHMS

In the context of real-time recommendations operating on very large data-sets, the Memory-based CF approaches are not fast andnot as scalable as how we would like them to be. In Model-based CF approaches, a model is designed and developed to use the training dataset (a part of the dataset) to produce the predictions. The developed model is then used to make intelligent predictions for the Collaborative Filtering tasks for test data or real-world data. A lot of approaches can be made use of to build the model. Some examples are: Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Latent Semantic Analysis (LSA), Latent Dirichlet Analysis (LDA), Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS),Bayesian Networks, Clustering methods and Association Rule-based methods [3].

Amongst all the Model-based CF approaches mentioned above only two models are taken into considerationin this paper. The selection of those two algorithms is purely based on the ability to deal with different challenges highlighted in Section 2.1, the quality of prediction/recommendation and the computational complexity. The first Model-based approach chosen is Tendencies-Based Collaborative Filtering Method(TBCFM). This model was chosen because its errors are less visible to the user than those committed by other algorithms, as it provides high accuracy for the relevant items and the items it recommends are likely to be purchased. The computational time complexity of Tendencies-based CF approach is better than other Model-based CF approaches [10,11]. The second Model-Based approach chosen for this paper is Regularised Singular Value Decomposition (RSVD). RSVD approach presents better results under sparse conditions and it clearly outperforms the accuracy and precision of all memory-basedapproaches[11].

4.1. Tendencies-Based CF Method Algorithm

Calculating the similarity between items/users requires a great amount of information as it is a rather complex task [11]. As a consequence, similarity based algorithms face serious problems. The Tendencies-based CF algorithm doesn't look for relations between users or items but looks at the difference between them. Considering the fact that the users evaluate the items differently, this algorithm captures the tendency of the user. The concept of tendenciesrefers to whethera user evaluates an item positively or negatively. The algorithm for Tendencies-based CF approach is given below [10].

Algorithm 3 Tendencies-based Collaborative Filtering

Input: User-Item Rating matrix R

Output: Prediction of rating p_{ui} that user u would give for an item i

For each user u Do

  Calculate the tendency (ub_u) of user u using $ub_u = \frac{\sum_{i \in I_u}(v_{ui} - \bar{v}_i)}{|I_u|}$

End

For each item i Do

  Calculate the tendency(ib_i) of an item i using $ib_i = \frac{\sum_{u \in U_i}(v_{ui} - \bar{v}_u)}{|U_i|}$

End

For each user u Do

  For each item i Do

    If $ub_u >= 0$ && $ib_i >= 0$

$p_{ui} = \max(\bar{v}_u + ib_i, \bar{v}_i + ub_u)$

    Else-If $ub_u < 0$ && $ib_i < 0$

$p_{ui} = \min(\bar{v}_u + ib_i, \bar{v}_i + ub_u)$

    Else-If $ub_u < 0$ && $ib_i >= 0$

$p_{ui} = \min[\max(\bar{v}_u, (\bar{v}_i + ub_u)\alpha + (\bar{v}_u + ib_i)(1 - \alpha)), \bar{v}_i)]$

    Else-If $ub_u >= 0$ && $ib_i < 0$

$p_{ui} = \min(\bar{v}_u + ib_i, \bar{v}_i + ub_u)$

End

  Assign the rating p_{ui} to item i for user u

End

4.2. Regularized Singular Value Decomposition CF Algorithm

Singular Value Decomposition is one of thefactorization algorithms for Collaborative Filtering. This type of algorithm tries to find the features of users and item, and makes predictions based on these features. SVD doesn't have restrictions on any feature value and it is easy to implement.[12] Given an input rating matrix M of size m*n which consists of ratings of m users and n items. Low-rank matrix approximation of M using singular value decomposition gives two feature matrices corresponding to users and movies. User feature matrix P is of size m*k represents the associativity of a user with k features. Movie feature matrix Q is of size k*n represents the associativity of a movie with k features. To obtain P and Q, matrix M is decomposed into three matrices U, S, V. U is a m*m matrix, S is a m*n diagonal matrix and V is a n*n matrix. Now only the k left most columns are taken from U, k top most rows are taken from V and only k singular values are taken from S. Now P and Q are calculated as follows,

$$P = U^* \sqrt{S}\text{, where dimension of U is m*k and S is k*k}$$

$$Q = \sqrt{S}\ ^*V\text{, where dimension of S is k*k and V is k*n}$$

After obtaining P and Q, rating of user i for movie j is calculated as follows,

$$\text{Pred(i,j) = dot product of } P_i \text{ and } Q_j$$

where P_i is user feature matrix for user i, Q_j is movie feature matrix for movie j.

Regularized SVD is a technique used for collaborative Filtering proposed by Simon Funk which includes regularization constants along with learning rate [13]. Unlike SVD, RSVD uses different objective function and negative gradients.Objective function and negative gradients are used to update the feature matrices P and Q. The RSVD CF Algorithm is given below.

Algorithm 4 Regularised SVD Collaborative Filtering

1 Input: User-Item Rating matrix R

2 Output: Prediction matrix P

3 Create the input matrix $A \in R^{m*n}$ from the given dataset

4 Find out the indicator matrix $I \in, \{0, 1\}^{m*n}$ that indicates which movies are rated by users

5 A is given as input to SVD to get the feature matrices $U \in R^{k*m}$ and $M \in R^{k*n}$, where k is number of features

6 Calculate the prediction matrix as follows,

$$\begin{array}{l} \text{a if } U_i^T M_j < 0 \\ p(U_i, M_j) = \\ \text{b if } U_i^T M_j > \end{array} \left[\begin{array}{l} \\ a + U_i^T M_j \text{ if } 0 <= U_i^T M_j <= b - a \\ b - a \end{array} \right.$$

where p is the prediction function whose arguments are U_i, M_j(feature vectors). It computes the prediction value which lies in the range of (a, b).

7 Calculate the RMSE from the obtained prediction matrix P

8 To optimize the error, use the partial derivative of the squared error with respect to each parameter U_{ki} and M_{kj}

$$U_{ki(t+1)} = U_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * M_{kit} - \beta * U_{kit})$$
$$M_{ki(t+1)} = M_{kit} + \alpha * (2 * (A_{ij} - P_{ij}) * U_{kit} - \beta * M_{kit})$$

where α is the learning rate and α is regularization coefficient.

9 Repeat from step4 until the RMSE is minimum

5. DISCUSSION

This paper presents a totalof four algorithms; two for each Memory-based and Model-based Collaborative Filtering methods. This section presents a comparative study of all these algorithms based on their ability to tackle the challenges of Collaborative Filtering, accuracy in making predictions/recommendations under various constraints, and their computational complexity.

Memory-based CF algorithms are really simple to implement for any situation, and they are able to produce reasonably accurate recommendations. It is easy to update the database, while using memory-based CF algorithms, because the entire database is used every time they make predictions. However, they present serious scalability problems given that the algorithm has to process all the data to compute a single prediction [11]. These algorithms are not appropriate for real time recommendation systems with a large number of users. Furthermore, compared to model-based algorithms these algorithms are more sensitive to common problems of recommender systems and very slow in making predictions. These algorithms cannot also succeed, if similarity doesn't exist between items/users.

On the other hand, Model-based CF algorithms can obtain the underlying characteristics of dataset and thereby extract more information [11]. Constructing a model for Model-based CF approach requires considerable time, but once the model is built, it tends to be faster in making prediction. However, model-based algorithms also present a lot of problems. Many models are awfully complex, as they have to estimate a mass of parameters, and they are too sensitive to changes in data. Sometimes, the developed model may not be able to fit the real data, thus leading to wrong recommendations. Many theoretical models cannot be practically applied to real data. Moreover, construction of a model and updating it in order to reflect the newly added data are time and resource-consuming tasks.

The computational efficiencies of the algorithms are given below in Table 2.

Table 2
Computational Efficiencies of CF Algorithms Studied

Algorithm	*Training*	*Prediction*
	Memory-based CF Algorithms	
User-based CF	-	$O(mn)$
Item-based CF	$O(mn^2)$	$O(n)$
	Model-based CF Algorithms	
Tendencies-based	$O(mn)$	$O(1)$
RSVD	$O(mnk)$	$O(1)$

m – number of users, n – number of items, k – number of features(only in RSVD)

In Table 2, Complexity of CF algorithms has been separated into two parts, training part is corresponding to the building of the model using the training dataset and prediction is corresponding to making a single prediction. Creation of a model will be performed only once, while large number predictions will be made. Generally speaking model-based algorithms are more efficient when computing a prediction, despite the fact that the construction of the model is considerably complex [11]. Among many model-based CF algorithms Tendencies-based CF algorithm is the most efficient, with a training complexity of $O(mn)$ and prediction complexity of $O(1)$. The time required to make predictions is also much better than memory-based approaches such as User-based and item-based CF algorithms. In Memory-based CF methods, Item-based CF algorithm performs better than User-based CF algorithm.Although Item-based CF requires a complexity of $O(mn^2)$ for training/constructing a model, the time complexity for making prediction is only $O(n)$.

Based on the study of all presented CF algorithms we would like to conclude the following.

1. Memory-based CF algorithms work efficiently with relatively dense matrices, worsening significantly in presence of data sparsity.
2. Although Model-based CF algorithms are less accurate than Memory-based CF algorithms under ideal conditions, they behave better when the data is sparse.
3. Parallel and Distributed algorithms can be devised for Collaborative Filtering Techniques to withstand scalability issues.
4. For dealing with challenges such as Gray Sheep, Data Sparsity and Shilling Attack, users' review comments can also be considered in addition to their ratings. Advances on opinion mining and Aspect extractions will help us do it.

6. CONCLUSION AND FUTURE RESEARCH

Collaborative Filtering is a renowned and widely used successful Recommendation Technique. However, the current generation of CF techniques still requires further improvements to make predictions/recommendations more effectively with real-time data. Collaborative Filtering is being stressed by huge volume customer data in existing databases and even more stressed by increasing volume of information available on the web. Advent of new technologies to enhance the performance of Recommender Systems and their ability to take up certain challenges is greatly needed. In this paper, we presented the process of how predictions/recommendations are generated in Collaborative Filtering techniques. Four representative algorithms namely User-based CF, Item-based CF, Tendencies-based, and RSVD were studied in order to understand the pros and cons of Memory-based and Model-based CF approaches and their computational efficiency. The result of the study showed that Memory-based CF algorithms are simple, easy to implement and producing high quality predictions, but their performance will get worsened in case of data sparsity. On other hand, Model based CF algorithms work fine even with sparse data, but construction of a model is a time and resource consuming task.

Directions for future research include devising Parallel and Distributed algorithms for existing Collaborative Filtering techniques, incorporating users' review comments to elicit the implicit ratings given for different aspects of an item, introducing multi-criteria rating into existing CF techniques to overcome data sparsity problem, enhancing existing CF techniques to handle real-time data.

REFERENCES

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

[2] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, "An Algorithmic Frameworkfor performing Collaborative Filtering", *SIGIR '99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 230-237,1999,.

[3] Xiaoyuan Su and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques". *Advances in Artificial Intelligence*,January 2009.

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture or collaborative filtering of netnews," in Proceedings of the *ACM Conference on Computer Supported Cooperative Work*, 175–186, 1994.

[5] EfthaliaKarydi and Konstantinos G. Margaritis, "Parallel and Distributed Collaborative Filtering: A Survey", *Information Retrieval (cs.IR); Distributed, Parallel, and Cluster Computing* (cs.DC), arXiv:1409.2762 [cs.IR], 2014

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, **7 (1)**, 76–80, 2003.

[7] M. Claypool, A. Gokhale, T. Miranda, et al., "Combining content-based and collaborative filters in an online newspaper," in Proceedings of *the SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, Calif, USA, 1999.

[8] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Itembased collaborative filtering recommendation algorithms," in Proceedings of the *10th International Conference on World Wide Web (WWW '01)*,285–295, 2001.

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in Proceedings of *the ACM Conference on Computer Supported Cooperative Work*, 175–186, 1994.

[10] V Formoso, F Cacheda, V Carneiro, "Algorithms for Efficient Collaborative Filtering," In: *Efficiency Issues in Information Retrieval Workshop*, **17**, 2008.

[11] Fidel Cacheda, Victor Carneiro, Diego Lucas Fernández, VreixoFormoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems", *ACM Transactions on the Web*, **5 (1)**, 2011.

[12] Chih-Chao Ma., "A Guide to Singular Value Decomposition for Collaborative Filtering", Proceedings of *seventh IEEE International Conference on E-commerce Technology*, 1-7, 2005

[13] Paterek A., "Improving regularized Singular Value Decomposition for collaborative Filtering", *Proceedings of KDD Cup and workshop*, San Jose, California, USA, 5-8, 2007.