Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 3; January-March, 2015 pp. 252-257
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Recommender System-A Close Look at Collaborative Filtering

Shivani Agrawal[1], PankhuriRastogi[2] and Shubhi Agarwal[3]

[1]B.Tech, Department of Information Technology JSSATE C-20/1, Sector 62 Noida, (U.P.)
[2]B.Tech, Department of Computer Science and Engg. JSSATE C-20/1, Sector 62 Noida, (U.P.)
[3]B.Tech, Department of Information Technology JSSATE C-20/1, Sector 62 Noida, (U.P.)
E-mail: [1]shivaniagr94@gmail.com, [2]pankhurirastogi06@gmail.com, [3]shubhiksj95@gmail.com

Abstract—*Recommender Systems form the major and integral part of ecommerce.It helps in providing the best suited and most favorable items from myriad of items. Its application ranges from recommending what to buy on amazon to what to watch on Netflix. With the increased use of web, problem of information overloading is coming up. It is important to look for and obtain the relevant information rather than going through all the information. This is where recommender systems come into the picture. They provide the list of items depending upon the taste of the user and choices of people who share common interest and taste with the user. From time to time several techniques for recommender system have been discovered and used. The paper covers one such important technique of collaborative filtering. The paper covers the other techniques and problems faced by them and then the evolution of collaborative filtering as major and efficient tool for reliable and accurate recommendations. Different types of collaborative filtering have been discussed and compared. With the changing time and scenario collaborative filtering approach has been modified to meet the requirements of cross domain and context awareness. Several techniques for similarity measurement have been discussed and evaluated. Problems faced by the collaborative filtering have been pondered. The paper contains all the rudimentary details and deeper insight into collaborative filtering. It can be utilized by students and other researchers who are looking forward in the research area of Recommender systems.*

Keywords: *Collaborative filtering, Context-aware collaborative filtering, Cross-domain collaborative filtering, Cosine Similarity, Incremental collaborative filtering, Item based collaborative filtering, Hybrid Recommender system, Pearson Correlation.*

1. INTRODUCTION

Recommender systems provide personalized services in e-commerce domain by providing individual user with set of items best suited and matches the taste of the user. A recommender system is used to predict ratings or preferences so that a few top items or services can be selected and recommended to potential users. With the help of machine learning concepts and other algorithms, recommender systems can judge if a particular product will be liked by the user or not. For example, youtube recommends new videos to the active user on the basis of his browsing history. There have been several approaches for implementation of recommender systems, namely content based filtering, collaborative filtering and hybrid filtering. Content based recommendation system works on the basis of the historical records of purchase and ratings by the user. The items that are mostly similar to the items positively rated by the user are recommended to him. For example, if a user has purchased books of c programming in the past, then he will be recommended other books of programming language. It faces few challenges and has disadvantages. Next comes the Collaborative filtering. It is a method of making of predictions about interest of the users by collecting preferences or taste information from many users. The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviors (e.g., buying, watching, listening), and hence will rate or act on other items similarly. Even this approach suffers from several shortcomings of sparsity, scaling problem, cold start problem, gray sheep problem and synonymy problem. To overcome them, researchers came up with other new approaches. Hybrid approach of filtering combines both the features of content based filtering and collaborative filtering. Analysis of these approaches has been done by great researchers. Through this paper, our aim is to give the general idea about recommendersystem and to deal in detail about the collaborative filtering technique, its various problems and the approaches to solve them.

Collaborative Filtering

It recommends the items to a user based on the ratings provided by the users that share same interest as that user. User item rating matrix is used for it. According to Xiaoyuan Su and Taghi M. Khoshgoftaar, if there are m users and n items, then a user–item matrix of (m×n) is used to store the ratings for an item by a user(see Fig. 1).

Let kui be a cell where uth user rates the ith item [1]. These ratings can be obtained explicitly by asking users about their preferences or implicitly by observing the items being clicked, viewed or purchased. Collaborative filtering approach takes care of the people's interest and taste. It assumes that their taste remains constant or changes at very slow rate. For example, consider a case in which three users rated the different types of books. The table is given below:

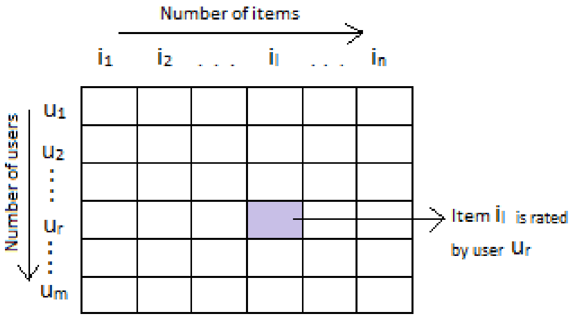
Number of items
i_1 i_2 . . . i_l . . . i_n
Number of users
u_1
u_2
u_r
u_m
Item i_l is rated by user u_r

Fig. 1: Representing User-Item Matrix.

Table 1: Example of Collaborative Filtering

BOOKUSER	ALCHEMIST	ANGELS AND DEMONS	WINGS OF FIRE	THE ZAHIR
A	9	8	-	-
B	8	8	5	10
C	8	4	10	9

In Table I, the user A hasn’t rated the book “The Zahir”, which probably means that he hasn’t read it yet. As the other users rated it positively, he will get this item recommended.The recommendation process can be divided into two parts.

1.1. Similarity Computation

There are a number of ways to compute the similarity between items. Here we describe two ways.

1.1.1. Cosine similarity

Cosine similarity is a vector space approach using linear algebra and not being a statistical approach. Users are represented as n-dimensional vectors and the cosine distance between two rating vectors is used to compute similarity. The similarity values between two users lie in the range 0 and 1, where 0 represents the lowest degree and 1 represents the highest degree of similarity. Let u and u_i be two n-dimensional vectors representing the ratings given by the user previously for item d_i where i can be from 1 to n. Similarity is inversely proportional to the angle between them. This distance can be measured using dot product between two vectors and divided by Euclidean norm of both vectors. This can be calculated using equation (1).

$$sim(\vec{A},\vec{B}) = \cos(\vec{A},\vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} \qquad (1)$$

where$\vec{A} \bullet \vec{B}$represents dot product of these two vectors.

1.1.2. Pearson correlation.

Pearson relation is used to compute the similarity between the users in user item matrix based on the associated rows between users. It measures the linear dependence between two users as a function of their attribute.it is a static approach.it does notmeasure over the entire population of users.The similarity can be measured by Pearson correlation constant with the help of equation (2). Positive correlation implies users have rated the item in similar way, and vice versa.

$$W_{a,u} = \frac{\sum_{i \epsilon l}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \epsilon l}(r_{a,i} - \bar{r}_a)^2 \sum_{i \epsilon l}(r_{u,i} - \bar{r}_u)^2}} \qquad (2)$$

1.2. Prediction Computation

After the similarity calculation has been done, the next step is to give the output in terms of predictions. Different techniques are used for it. Here we describe two such techniques.

1.2.1. Weighted sum

This can be computed by calculating the sum of the ratings given by user u on the items similar to the target item i on which prediction is to be made. As given by [2], prediction $P_{u,i}$ can be denoted as given in equation (3).

$$W_{a,u} = \frac{\sum_{i \epsilon l}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \epsilon l}(r_{a,i} - \bar{r}_a)^2 \sum_{i \epsilon l}(r_{u,i} - \bar{r}_u)^2}} \qquad (3)$$

Where $s_{i,N}$ represents the similarity between i and N and $R_{u,N}$ repressents the rating of item i by user u.

1.2.2. Regression

Unlike Weighted sum technique which uses the ratings of similar items, regression model uses the approximated values of ratings of similar items. The formula used is same as the weighted sum method.

2. Techniques of collaborative filtering

The two techniques used for collaborative filtering are-

2.1. Memory-based collaborative filtering

Memory based CF uses the entire database for recommendations. This technique finds a set of users that have a similar history with the current user. After the neighborhood is formed, different algorithms are used to combine the preferences of neighbors and give top-N recommendations or predictions [3]. Two similarity measurements are used. The first is Pearson correlation coefficient and the other is cosine similarity(stated above). The advantages of this technique are that the algorithm is very simple to implement and it is very simple to update the database. It has several limitations like it cannot recommend for new items and new users, it is very slow as the entire database is used and if the data is sparse, then the performance is decreased.

2.2. Model-based collaborative filtering

Memory based CF is slow in case of large datasets, so Model based CF is used. Instead of using the whole database, it builds a model on the basis of dataset of ratings. The benefits of speed and scalability are offered. The disadvantages of Model-based CF is that it is less accurate than memory based CF, model building is expensive and adding data to such systems is difficult, so it is inflexible.

2. TYPES OF COLLABORATIVE FILTERING

The two major types of collaborative filtering are-

2.1 User based collaborative filtering

In this approach, the people having similar preferences are joined into one group. Recommendation is given to a user on the basis of the ratings of the other user of the same group to whom he belongs. It is expensive in terms of computation. Its running time is of the order mn i.e., O(mn) where m is the number of users and n is the number of items, this is the worst case scenario. Since the user rates very few items or we can say the user vector is sparse hence the average time complexity tends to O(m+n). Time required for traversal of all users is O(m). The users who rate significantly or purchase excessively leads to processing time of O(n)[4].

2.2 Item-based collaborative filtering

To deal with large data, Amazon came up with Item-to-Item Collaborative filtering. In this approach items are compared with the items user has already rated or purchased. The similar ones are recommended to the user. It involves formation of similarity table where each product is compared with every other product and the similarity metrics for the product pairs are obtained [4]. Its time complexity is O(n^2). It is inefficient when it comes to memory usage. It suffers from Item cold start problem i.e., we cannot predict similar items until we have ratings for the item to be compared.

3. CHALLENGES FACED BY COLLABORATIVE FILTERING

3.1 Scalability

The approach suffers from the scalability problem when either the number of users or number of data items is large. Different solutions have been tried out from time to time namely-

3.1.1 By taking the subsets of users

It is done by clustering of users. This technique works by identifying groups of people who have similar preferences. Recommendation is given to the user on the basis of the preferences of the other users in that cluster. Clustering technique usually reduces accuracy but it increases the performance since the size of the group to be analyzed becomes much smaller. A collaborative filtering algorithm using this approach first divides a database A into n partitions on the basis of preferences of users. Then the whole partition A_i to whom the active customer belongs, works as a neighborhood for that customer c_i [5].

3.1.2 By reducing dimensionality

Reduction in dimensionality is achieved by Singular Value Decomposition (Sarwar et al, 2000). SVD reduces the time complexity of item based collaborative filtering significantly. Time Complexity is O(mn^2). SVD reduces the dimension to k which is much less than m, hence reducing the complexity to O(kn^2).The reduction in complexity helps in dealing scalability issue and improves the performance of recommender system[6].

By adopting these approaches we will be ignoring certain potential items. Another approach called *Incremental Collaborative Filtering* is given [7].Incremental collaborative Filtering deals with scalability problem without affecting recommendation quality. In this, the similarity value between active user and all other users need to be recomputed whenever the active user submits the new rating or changes the existing rating. The aim is to express the new computed similarity in terms of old similarity values.As in [7], i_x is the subset of items that have been co-rated by user u_x and u_y. Here x can be from 1 to n' where n' can be less than or equal to the total number of items n.

To find the similarity between user u_x and u_y using Pearson correlation equation (4) is used.

$$sim(u_x, u_y) = \frac{\sum_{h=1}^{n'}(r_{u_x,i_h} - \bar{r}_{u_x})(r_{u_y,i_h} - \bar{r}_{u_y})}{\sqrt{\sum_{h=1}^{n'}(r_{u_x,i_h} - \bar{r}_{u_x})^2}\sqrt{\sum_{h=1}^{n'}(r_{u_y,i_h} - \bar{r}_{u_y})^2}} \quad (4)$$

wherer_{u_x,i_h} denotes the rating of item i_h given by user u_x and the average rating of user u_x, u_y is represented by $\bar{r}_x, \bar{r}_y$ respectively.

This relation can be written in following form in equations (5),(6),(7).

$$sim(u_x, u_y) = A = \frac{B}{\sqrt{C}\sqrt{D}} \quad (5)$$

$$B = \sum_{h=1}^{n'}(r_{u_x,i_h} - \bar{r}_{u_x})\,(r_{u_y,i_h} - \bar{r}_{u_y}) \quad (6)$$

$$C = \sum_{h=1}^{n'}(r_{u_x,i_h} - \bar{r}_{u_x})^2\,, D = \sum_{h=1}^{n'}(r_{u_y,i_h} - \bar{r}_{u_y})^2 \quad (7)$$

This measure is split into three factors i.e., B, C, D. The new values of factor B', C', D' (equation (9)) results in the new value of similarity measure

$$A' = \frac{B'}{\sqrt{C'}\sqrt{D'}} => A' = \frac{B+e}{\sqrt{C+f}\sqrt{D+g}}(\quad 8)$$

$$B^{'} = B + e, C^{'} = C + f, D^{'} = D + g \qquad (9)$$

where e, f, g are the increments that are computed taking into consideration that whether the submission is new or the updation of existing one.

Classic CF depends on the size of the database. Accuracy is directly proportional to size but response time increases on the increment of size.The complexity of CF can be calculated in two parts- first is the time maintaining the user similarities matrices and other for calculating a single recommendation to active user. It uses user-to-user similarities. This can be achieved by computing offline and then feeding back the updated information to database periodically. The worst case complexity of user similarity matrix is O(m^2n) where m_x and m_y are the users, n are the no. of items co-rated by users. It is not accurate since the ratings given between two offline computations are not updated. If these are not pre-computed i.e., computation is done at the time of recommendation then its complexity is of order O(mn). For a single recommendation of active user, the complexity is O(n) when computed offline and O(mn) when not pre-computed. This problem can be solved using ICF which is highly scalable than the CF approach as its response time is considerable for large data also. In ICF, algorithm is computed at the time of rating. Complexity is O(mn) when not pre- computed and is of order O(n) when it is pre-computed.

3.2 Sparsity

It refers to situation when the users do not rate much items. The user Item rating matrix is found to be sparse. This adversely affects the accuracy of recommendations because we cannot compare users who have not rated. It reduces the quality of recommendations, one of the reasons is loss of neighbor transitivity which means even after having similar kind of interests, they cannot be identified as similar users because they have not rated the same item [1].For example user A and B are highly correlated and user B and C are highly correlated that does not mean that A and C will be highly correlated, because it may be so that A and C may not have rated the same items. Cold Start problem states that it is difficult to recommend the items to the new users which has empty profile that is they haven’t rated any item therefore very less information is available about them. *Cold start* can be conceptually considered as one of the special case of sparsity problem where either the row or column is completely empty [8]. There are three types of cold start problem namely new item problem, new system problem and new user problem. New items problem is when they are new to the system and no ratings are available for that item. Pure collaborative cannot help in this. Since there is no information available on the user’s choices, this challenges the attainment of various ways to fetch user data. According to [9], if a user is new to the system but has profile in another system then his external profile can be used to recommend possible items. For example, a user is new to youtube and already has a profile on facebook. A profile of the user can be made using the movies he has liked or posted on facebook which will be used to recommend the appropriate movies to him on youtube. Another solution to this as proposed by Halpin et al is the study of tags in the social networking site and then uses a model of collaborative tagging to evaluate the recommendations. In [10], a n item based probabilistic model is proposed. In this items are divided into groups and then predictions are made accordingly based on Gaussian distribution of user ratings.

From time to time various eminent researchers have worked on alleviating this problem by using approaches likes reducing dimensionality with the help of SVD technique (described above). It reduces the dimensionality by removing unrepresentative and insignificant users [3]. Another approach for reducing dimensionality is Principle Component Analysis, which is a mathematical technique that converts the correlated variables into completely different new set of uncorrelated variables. The PCA demog algorithm represents the data (including the user–items rating and demographic user content) into data tuples. PCA is then applied onto the data tuples, the data tuples whose projections are achieved in reduced space are used for making recommendations [11], combining content based and collaborative filtering into *hybrid filtering*.

3.3 Synonymy Problem

Collaborative filtering fails to understand the similarity between two items having similar properties but different names. This is called Synonymy problem. For example, one user rated fruit beverage high and other user rated fruit drink high, but these items will be taken as different items by collaborative filtering for computation of correlation. So it could not identify the association between them. This problem affects the performance of recommender systems and reduces the accuracy of recommendations. LSI along with truncated SVD can identify the synonyms as long as there exists a link between the two [12]. Cyrus Shahabi et al. formulated a method based on content based ranking and combined it with the collaborative filtering models, this identifies the associations between users (Collaborative filtering) and also association between items (Content based Filtering).

3.4 Gray Sheep problem

Gray sheep users are the users which have unusual taste in comparison to other users. As given in [13], a clustering solution is proposed which works in offline fashion to detect these users. This approach is useful since it reduces error rate for recommendation to gray sheep users. It uses K-Means++ concept over the users which rates large number of items.it uses a improved centroid selection approach for K means clustering algorithm .The probability of user m_i to be a power user m_p is given by equation (10).

$$P(m_i) = \frac{I_{m_i}}{I_{m_p}} \tag{10}$$

whereI_{mi} and I_{mp} denotes the number of items rated by user m_i and m_p. This equation finds the centroid that is at farthest distance from current chosen centroid and with the maximum probability of being a user which rates maximum.

4. COLLABORATIVE FILTERING WITH MODERN APPROACHES

4.1 Context-aware Collaborative Filtering

Considering only the ratings of the product provided by the user is not the ultimate criteria for recommendations. Ratings are highly affected by the environment or the situation in which the user is. There may be few items preferred by a user in particular situation and may not be preferred in other. Hence the knowledge of context i.e., the environment or situation of user plays a vital role in obtaining accurate results. With the developments in technology, increased consumption of mobile phones with high speed internet has enabled the usage of GPRS and other sensors to obtain the context of the user. Collaborative filtering has been modified to make it context aware.

In [14], researchers have used a dynamic vector for user called profile vector which includes the context information of their location, environment, time and user’s status. This context information was gathered with the help of sensors around the user also called object oriented implicit measures. Another way of achieving this is establishing a software to monitor user’s activities. This is called system oriented implicit measure. Usage of dynamic vector helped in improving accuracy and dealt with the cold start problem.

Cross-domain based Collaborative Filtering

Similarity of two users is evaluated on the basis of their appreciation of items. But similar preference in one domain does not guarantee that their preference about the other domains will also be similar. In Table II, the users have similar preferences for clothing brands but different for brands of watches.

Table 2: Example of Cross Domain Collaborative

ITEMS USERS	CLOTHING		WATCHES	
	LEVIS	MADAME	FASTRACK	SONATA
A	8	9	9	9
B	9	10	3	2

Standard or pure recommender system compares users without splitting them into different domains. In cross-domain recommender systems, local neighborhood is created for each user according to domains and then recommendations are provided on the basis of these neighborhoods. Thus the similarity computation is domain dependent [15].

5. CONCLUSION

Recommender system finds a wide range of applications in today’s world where web is loaded with information, filtering out correct predictions and recommendations are necessary. Collaborative approach is one of the most popular approaches for recommender systems. It recommends items to user on the basis of the items preferred by the users with similar taste. Broadly it can be divided into two classes, memory based and model based. Memory based which includes kth nearest neighbor technique. Model based technique deals with problem of large dataset faced by memory based technique. Providing recommendation involves following two steps. The first step involves finding similarity between users/items using cosine vector similarity and Pearson correlation. The next step is computation of predictions with the help of weighted sum and regression method. Collaborative filtering suffers from problems of scalability, sparsity, synonymy and gray sheep. For solution of the above problems various approaches like SVD, incremental collaborative filtering, clustering for scalability; PCA, hybrid filtering for sparsity; Latent Semantic Indexing along with SVD for synonymy and modified K-means++ clustering algorithm for gray sheep. No single approach is able to deal with all the issues. The paper also looked on the issues springing up in modern times and collaborative filtering being enhanced to context-aware,cross -domain and peer to peer in order to deal with them. At the end we conclude that recommender system is wide area of research, work is still going on in the field, new techniques and approaches are being employed in order to achieve an accurate, trust worthy and a personalized recommender system.

REFERENCES

[1] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, 2009.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." *In Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. ACM, 2001.

[3] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. "Application of dimensionality reduction in recommender system-a case study," No. TR-00-043. *Minnesota Univ Minneapolis Dept of Computer Science*, 2000.

[4] Greg Linden, Brent Smith, and Jeremy York," Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing*, IEEE vol. 7, no. 1, 2003.

[5] Badrul M. Sarwar, George Karypis, Joseph Konstan, and John 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*, vol. 1. 2002.

[6] Manolis G. Vozalis, and Konstantinos G. Margaritis. "Applying SVD on Generalized Item-based Filtering." *IJCSA* vol. 3, no. 3, pp. 27-51, 2006.

[7] Manos Papagelis, Ioannis Rousidis, Dimitris Plexousakis, and Elias Theoharopoulos. "Incremental collaborative filtering for highly-scalable recommendation algorithms," *In Foundations of Intelligent Systems*, pp. 553-561. Springer Berlin Heidelberg, 2005.

[8] Zan Huang, Hsinchun Chen, and Daniel Zeng. "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering." *ACM Transactions on Information Systems (TOIS)* vol. 22, no. 1, pp. 116-142, 2004.

[9] Shaghayegh Sahebi, and William W. Cohen. "Community-based recommendations: a solution to the cold start problem." *In Workshop on Recommender Systems and the Social Web, RSWEB*, 2011.

[10] Byeong Man Kim, and Qing Li. "Probabilistic model estimation for collaborative filtering based on items attributes." *In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 185-191. IEEE Computer Society, 2004.

[11] M. Vozalis, and K. Margaritis. "A recommender system using principal component analysis." *Current Trends in Informatics* vol. 1, pp. 271-283, 2007.

[12] Andri Mirzal. "The limitation of the SVD for latent semantic indexing." *In Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference*, pp. 413-416. IEEE, 2013.

[13] Mustansar Ghazanfar, and Adam Prugel-Bennett. "Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution." 2011.

[14] Hua Si, Yoshihiro Kawahara, Hisashi Kurasawa, Hiroyuki Morikawa, and Tomonory Aoyama. "A context-aware collaborative filtering algorithm for real world oriented content delivery service," *Proc. of ubiPCMM*, 2005.

[15] Daniar Asanov. "Algorithms and methods in recommender systems," *Berlin Institute of Technology, Berlin, Germany*, 2011.