Review of Recommendation System using Filtering based Concepts

Nikitha Shankar[1], Sharath H Nagaraj[2], S R Swamy[3]

[1]*Student, Dept. of Computer Science Engineering, R.V. College of Engineering, Karnataka, India*
[2]*Student, Dept. of Computer Science Engineering, R.V. College of Engineering, Karnataka, India*
[3]*Professor, Dept. of Computer Science Engineering, R.V. College of Engineering, Karnataka, India*

Abstract - *Recommendation systems which use filtering-based concepts such as collaborative filtering and content-based methods are forecasting new items of interest for a user. These methods have their own benefits but fail to provide a strong impact when used individually. A good Recommendation System involves integrating the components from both methods which gives rise to a Hybrid Recommendation System. A model incorporating content-based filtering and collaborative filtering may benefit from both content representation and user similarities. A hybrid approach blends these types of information, whereas the recommendations of the two filtering techniques can also be used independently. These fundamental filtering techniques are used to extract data from different users and use this information to construct an effective recommendation system. The paper provides an overview of recommendation systems which integrates a recommender system's concept of collaborative filtering, content-based filtering, and hybrid approach.*

***Key Words*: Collaborative filtering, Content-based Methods, Recommendation Systems, Hybrid Approach**

1. INTRODUCTION

Recommendation Systems are software tools that provide necessary suggestions to meet user requirements. To increase a company's overall sales volume, recommender systems are a common addition to web applications. Recommendations are a new form of understanding the behaviour of a system, can be an individual or a company. These recommendations are done based on the data accumulated over time. Most commercials websites selling products have recommendation systems where the users get to view the other products that are also generally bought along with the current selected product. The purpose of these systems are also to introduce users to new similar products that they might be interested in purchasing, maximizing customer satisfaction and company earnings. These systems are information filtering systems henceforth work in a way to predict the next best preference or the rating of the current selection.

This paper describes the system in four basic parts: the model, product, user, means of evaluating the values (i.e. ratings, quantity of a product, or another product itself). Thus, all recommendation models revolve around these factors for computation of recommendations. Before getting started with the aspects of the model, understanding the concepts of the various filtering techniques is important. The ability to deal with filters requires knowledge of mathematics, system modelling techniques, design of the filter and how it is implemented.

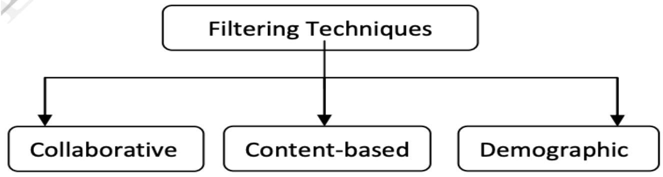


Fig -1: Different Types of Filtering Techniques used

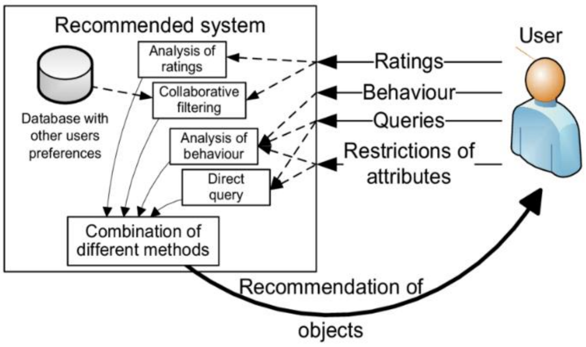

Recommended system
Analysis of ratings
Collaborative filtering
Database with other users preferences
Analysis of behaviour
Direct query
Combination of different methods
User
Ratings
Behaviour
Queries
Restrictions of attributes
Recommendation of objects

Fig -2: High Level Design of a Recommendation System

The Filtering techniques that are being used are Content-based filtering, Collaborative filtering and Demographic filtering. In short, the simpler and outer image of these types are, Content Based wherein the type of filtering which is based on the product (i.e. item-to-item), Collaborative is user defined and Demographic is the one which involves grouping of users based on similarities in behaviour into demo graphs and then evaluating their behaviour.

1.1 CONTENT-BASED FILTERING

This model is based on recommendations that are in close relation to the product type and features. The products which are correlated to each other are recommended by the system. The items are associated based on the description analysis made by a particular used based on his/her interests. They also can be correlated based on the user profile created which has a collective information of user likes or dislikes. In short, people tend to buy objects that complement each other. For instance people who buy milk cartons are more likely to buy biscuits and bread along with it. As we can see there is a similarity and correlation between milk and bread. Content based filtering looks for these similarities and acts on it. One way to find the similarity is through "cosine similarity". Cosine similarity means finding the cosine of the angle between the user profile and the product vector.

$$\sin(A.B) = \cos\theta = \frac{A.B}{|A|\,|B|} \quad (1)$$

Where in Equation (1):
A is the *profile vector* and B is the *product vector.*
The resulting values are grouped in descending order and are based on the needs of the users and then recommended for these items.
There are other ways to calculate the similarities for content based filtering, like:

1.1.1 Pearson's correlation:

A statistical test to find a value called Pearson's correlation coefficient (PCC) which determines the relationship between the profiles of the products. It's considered to be the best known association rule as it uses covariance.

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j} - \bar{R}_j)^2}} \quad (2)$$

The PCC value sim (i, j) is calculated in Equation (2), where:
- i and j are the products whose similarity is to be found.
- u representing each user at a time from the whole users U.
- Ri and Rj being the mean of the values.

1.1.2 Euclidean distance:

Most simple way of finding the similarity and classifying it is by plotting the points on a plane. The points close to each other are correlated, the Euclidean distance between the points are calculated in Equation (3):

$$Euclidean\ Dis\tan ce = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2} \quad (3)$$

The key drawback with content-based filtering is that it is item-centric and takes no account of user-related aspects. The content based filtering works when the products have the features noted down. The job of maintaining the record and documenting the features is a tedious job. They are literally document classifiers and won't work when put up with the task of recommending movies, restaurants etc. The aspect of users' likeness is kept in the dark. Consider a case in which the person buying milk is allergic to bread wheat, the system doesn't take into the fact that our user is allergic to wheat and nor does it have the idea of composition of the bread to draw up conclusions. Another aspect is that this system is too restrictive, implying addition of important products later is not meaningful in the system point of view as it doesn't have any previous record.

1.2 DEMOGRAPHIC FILTERING

It's a type of filtering where the user information is used to classify them into groups. To begin with the information can be age, gender and location. For instance, all the information that we provide to third party clients while ordering pizza (i.e. the details while filling the form when placing an order) . This particular data about a person is used to classify them into groups and are then targeted based on the vendor's needs. The behaviour of a group is categorized and used, like in tagging a group of a particular kind with a distinctive stereotypical label. Demographic filters are in line with collaborative filtering as it takes into consideration the "people-to-people" commerce compared to the content based mentioned earlier where the objective focused mainly on the item. But demographic is not entirely similar to collaborative, this filtering type does not take into consideration the historic data related to the transactions. When a new user enters into the system, the filter first classifies the user based on the features of categorizing and then recommends products to that user. Demographic is a generalized stereotypical user classification filter.

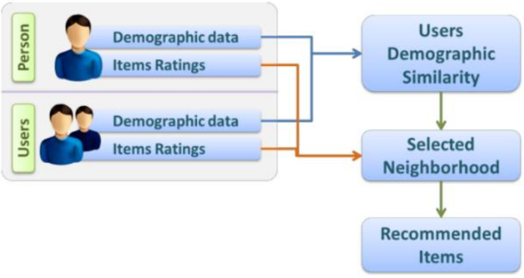


Fig -3: Demographic Based Approach Used

1.3 COLLABORATIVE FILTERING

This is the most widely used approach to design a recommendation system. This filtering method plays a significant role in the recommendation process and is used along with other techniques like content-based and knowledge-based filtering. Collaborative filtering is usually user based and performs analysis based on historical data. It is mainly established on gathering large amounts of user data and anticipating similarities among different users. It covers the "people-to-people '' aspect that isn't present in content based filtering. This approach is the oldest technique present out there but works just fine with all kinds of data. There are two parts to any filtering technique, the prediction part and the recommendation aspect. Recommendation is comparatively easy when it comes to collaborative filters, as each historic data can be considered as a unit and then compared in whole over a period of time. But prediction requires information regarding the transaction values of any given problem. To summarize, Collaborative Filtering is used in Recommendation systems where the recommended objects are selected on the basis of past evaluation of a large group of users.

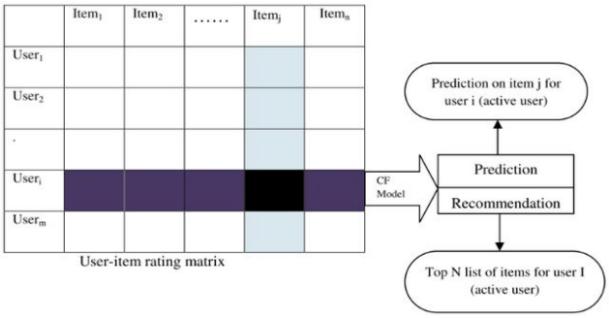


Fig -4: Overview of Collaborative Filtering Process

1.3.1 User Based Collaborative Filtering:

User Based collaborative filtering as the name indicates finds similarities between users and then makes recommendations. Consider three users X, Y and Z buy items a, b and c respectively. User based filtering calculates the similarities between user X and Y, X and Z, Y and Z by comparing the items they bought. Here as we saw earlier, similarity can be found using Pearson's or any of the methods. Prediction of the value is done by using the formula:

$$P_{x,a} = \frac{\sum_y (R_{y,a} * S_{x,y})}{\sum_y S_{x,y}} \quad (4)$$

In Equation (4), Px,a is the prediction variable where Ry,a is the rating of item a by user Y and Sx,y being similarity between user X and Y.

Users with high similarity will have high value of recommendations and predictions. Problem with user based is that when a larger number of users are in the picture, it becomes illogical to calculate similarities between each one of them.

1.3.2 Item Based Collaborative Filtering:

Item-based collaborative filtering is similar to user based but instead of finding similarities between users, here similarities between the items are computed. For the instance considering the above scenario similarities are calculated for items a and b, b and c, c and a by the trend in the way these are bought by various customers.

$$P_{x,a} = \frac{\sum_N (S_{x,N} * R_{y,N})}{\sum_N (|S_{x,N}|)} \quad (5)$$

Where in Equation (5): Px,a is the prediction of similarities between items and it's like a weighted sum of items neighborhood.

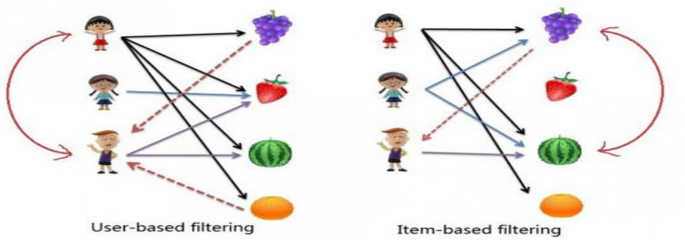


Fig -5: Types of Collaborative Filtering Process

2. EXPERIMENTAL RESULTS AND DISCUSSION

To summarize, Collaborative Filtering is used in Recommendation systems where the recommended objects are selected on the basis of past evaluation of a large group of users. The paper further contains the methodology on how Collaborative filtering is used in line with the project which aims to build an efficient Recommendation System.

For a Company which produces say N number of various products which is bought by M various users, the transaction data is obtained from the order ledger which notes the shipping details. The data is bulk and in order to speed up the analysis process, data cleansing is required. Every shipping order contains factors like:

Account ID, Account holder's Name, Product ID, Product Name, Date of Shipping, Quantity of the product, Price of the product, Date when the order was placed, Country where the product needs to be shipped, Country from which it is shipped, Application and Industry in which the user does business.

This chunk of data needs to be refined to our needs based on things we require in order to recommend products to users. Analyzing the data will provide an idea of what aspects play a role in recommending products. For instance, of the two dates that mentioned above, only one can be used i.e. either the date of the item being shipped or the date of the order placed. Similarly, country to be shipped to or country from which it is shipped, only one of this is sufficient. The data is provided by the company in an excel file which is then filtered according to user requirements in order to make suggestions.

The few necessary variables required are described in brief below:

Account ID — a unique reference ID given to every user, which makes the analysis process more efficient instead of referring to Account Holder's Name

Product ID — a unique reference ID given to every product, used for detailed analysis for each product

Order Date — shipping date varies when compared to Order Date. A same product may be shipped early for a certain user and late for another based on the company's reserved stock. So shipping date has variables affecting its value.

Quantity — this factor plays a key role in making recommendations to the customers

Application, industry, country are important factors required for classification while recommending the products. Like in demographic filtering, these classifiers can be used as a basic for easier representation of the recommended products.

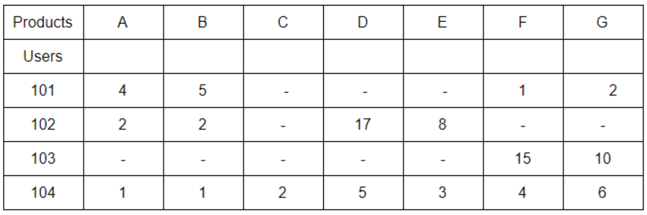
Now that the data has been cleaned according to the requirements with only the necessary columns present, there are few changes to be made from coding point of view. The Account ID may be alphanumeric or huge numeric figures which might be a hard nut while coding. So just numbering it in a simpler form with a linking to account id is a suitable change. There are two ways of selecting a factor for recommendation, one is by considering the frequency of transactions with regard to a user and a particular item irrespective of the amount or quantity of purchase. The matrix corresponding to this will look like:

Each of the transactions being unique, that is if an order for the same material was made twice a day, each being recorded as separate ones.

This chunk of data needs to be refined to our needs based on the factors we need, to recommend products to users. Analyzing the data will give an idea of what aspects play a role in recommending products and what doesn't. For instance, of the two dates that we have, only one can be used i.e either date being shipped or date of the order. Similarly, country to be shipped or country from which it is shipped, only one of these is enough. The data that we are dealing with is usually in an excel format with columns being the various elements we mentioned above. On filtering the data to the columns that we require, we end up with, after cleaning the data according to the requirements, with only the necessary columns present, there are few changes to be made accordingly to the code to be executed. The Account ID may be alphanumeric or huge numeric figures which might be a hard nut while coding. So just numbering it in a simpler form with a linking to account id is a suitable change. There are two ways of selecting a factor for recommendation, one is by considering the frequency of transactions with regard to a user and a particular item irrespective of the amount or quantity of purchase. Table I shows a pivot table corresponding to the Python code on which the Recommendation System was executed:

```
dataframe.pivotTable(values='X', rows='Y', cols='Z', aggfunc=lambda x: len(x.unique()))
```

Table -1: Pivot Table Showing Count of Products



Products	A	B	C	D	E	F	G
Users							
101	4	5	-	-	-	1	2
102	2	2	-	17	8	-	-
103	-	-	-	-	-	15	10
104	1	1	2	5	3	4	6

Here the pivot table with products as columns and user ids as rows and entries being the count of transactions of a particular type of product.

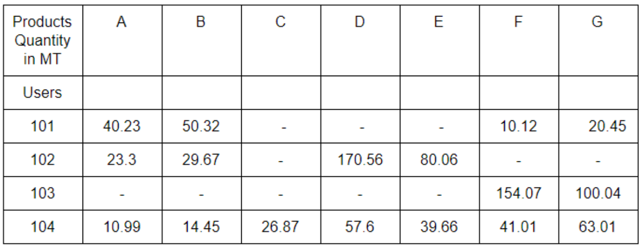
```
dataframe.pivotTable(values='X', rows='Y', cols='Z', aggfunc=lambda x: len(x.unique()))
```

This pivot table is used to calculate using collaborative filtering models. The problem with using frequency is that at the end the product with the highest frequency gets the prominence over another product which was bought least number of times. By doing so it is shadowing the fact that the product bought frequently may be in smaller amounts compared to the ones bought in large sums. When large sums are bought their frequency is comparatively low. This would draw up conclusions based on the fact that few products won't be recommended just because it was bought very less number of times for a certain period of time. The best way to quantify the recommendations are by using the quantity of the product bought. This would help predicting what quantity of a product would be bought the next time and what product would be bought as the sum of quantities over the entire period would also cover the view of the problem solved using frequency as seen before.

```
dataframe.isnull().sum()

dataframe['accountID'].nunique()

dataframe['productID'].nunique()

pivot_df = dataframe.pivot(index = 'accountID', columns ='productID', values = 'quantityMT').fillna(0)
```

All the empty entries are filled with “0”. The number of unique users and unique products are found out and a pivot table is formed as shown below. Here the values are the sum of a particular product bought by a customer over a given time (i.e. it is the sum of quantities from each purchase)

Table -2: Pivot Table Showing Quantity of Products in MT



Products Quantity in MT	A	B	C	D	E	F	G
Users							
101	40.23	50.32	-	-	-	10.12	20.45
102	23.3	29.67	-	170.56	80.06	-	-
103	-	-	-	-	-	154.07	100.04
104	10.99	14.45	26.87	57.6	39.66	41.01	63.01

Now that the data is ready, we use matrix factorization method to calculate the predictions and recommendations. Matrix factorization is basically splitting of a matrix into 3 different parts and multiplying them again to obtain a complete matrix with no empty fields. The pivot we obtained is usually a sparse matrix i.e. there is a lot of empty space in the matrix, this is because it's rarely a case when a user buys all the products present. So this sparse matrix as shown in Figure (6) is split into 3 parts being:

- AccountID matrix
- Latent feature matrix
- ProductID matrix

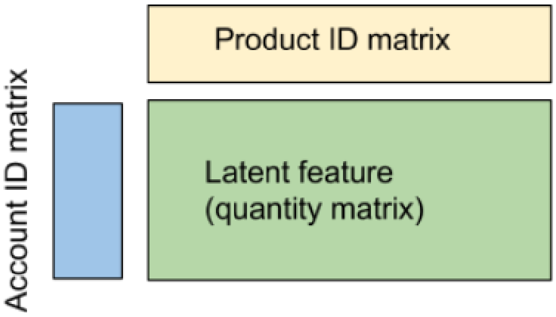


Fig -6: Sparse Matrix

The process of splitting the matrix into 3 parts is called SVD- Singular value decomposition.

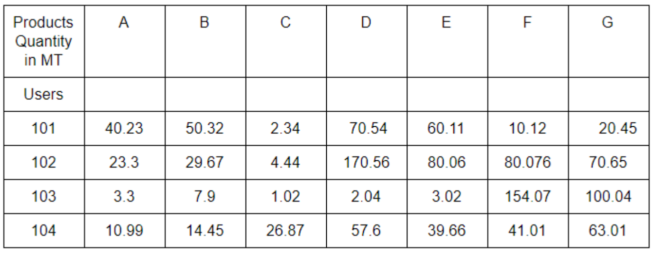
```
U, sigma, Vt = svds(pivot_df, k = number of
```

Where U and Vt are left and right singular matrices respectively. Sigma being the latent features matrix.
The code "svds" decomposes the matrix in python to the respective matrices. According to a lemma matrix A and B are equal if Av = Bv where v is a vector. In simpler terms, the decomposed matrices are vectors whose dot product is found to be the same as the starting matrix.

```
predicted_pivot = np.dot(np.dot(U, sigma), Vt)
```

This new predicted matrix is our result. The empty spaces are found to be filled with values.

Table -3: Pivot Table after Matrix Decomposition



Products Quantity in MT	A	B	C	D	E	F	G
Users							
101	40.23	50.32	2.34	70.54	60.11	10.12	20.45
102	23.3	29.67	4.44	170.56	80.06	80.076	70.65
103	3.3	7.9	1.02	2.04	3.02	154.07	100.04
104	10.99	14.45	26.87	57.6	39.66	41.01	63.01

This new matrix has predicted values for all the products present. This table can be later filtered based on aspects we discussed before, End use, Industry etc. These features are present just to have a comparatively easy and meaningful looking system. "Group by" function in python is used for this purpose.

The Standardized Root Mean Square (NRMSE) is used to assess module accuracy. To understand and draw conclusions, only the RMSE value would be large data, so NRMSE that splits the whole thing with the average value.
Collaborative filtering is comparatively better suited for most cases than other filtering techniques but there are a few blind spots with this method. As the data bulk increases, the problem with calculations also increases, that is if there are too many users and products as many as them, the filtering will work and recommend products but the calculations aren't pretty solid. In the sense, the pivot table we saw in our instance will contain much more empty places than the ones filled with the value. This might lead to smaller values almost being generalized along with the zeros. This would reduce the reliability of the system. Selection of key features is problematic and always delusional. It's more of a trial error basis, which might become a tedious job. For a collaborative filter to work, it requires a large amount of historic data pertaining to a given user. So if large amounts of data is not available, then it becomes a light data in the matrix, generalizing things to zero.

3. CONCLUSIONS

A very noticeable growth has been observed over the last decade in the use of Recommendation Systems. Various businesses have been developed with the use of different filtering techniques. The advantage of collaborative filtering is that suggestions for a new user are based on the preferences of a group of customers who have similar tastes or interests. It can be concluded that Recommendation Systems are proving a very useful tool with multiple business advantages by providing the rightful suggestions to users. The Filtering techniques explained in this paper are used to enhance the recommendation accuracy in order to improve customer satisfaction. This accuracy is achieved by collaborating the various filtering techniques. The Quality of suggestions are also improved by integrating a Hybrid approach in building the model. Hybrid Algorithms are now being used to also incorporate location information in existing Recommendation Systems. In order to improve the efficiency of these recommendation systems, future researchers will be concentrating on progressing with the already existing techniques and algorithms. Novel lines of research will be formulated for following fields, such as on:

1. Integrating Security & Privacy in Recommender Systems
2. Efficient Frameworks that are designed for Machine controlled Analysis of heterogeneous data
3. Combination of different types of available information being used by the existing systems.

Despite the advancing growth in techniques, the current generation of recommender systems examined in this paper still seeks additional modifications to make recommendation procedures more effective in a wider range of applications.

ACKNOWLEDGEMENT

Any accomplishment, whether scholastic or otherwise, is not based solely on individual actions, but on scholars, elders and friends' support, encouragement and cooperation. A variety of personalities have helped us in carrying out this project work within their own capacities. We want to take this opportunity to thank every one of them. We express our sincere gratitude to our mentor Dr. S R Swamy, Professor at the Department of Computer Science and Engineering, R.V.C.E, Bengaluru for his skillful guidance, frequent source of motivation and support during this project. We would also like to commend Dr. Ramakanth, Head of Department, Computer Science and Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

REFERENCES

[1] Pazzani, M., Billsus, D." Content-based Recommendation Systems." In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Breese, J.S., Heckerman, D., Kadie, C." Empirical Analysis of Predictive Algorithms for Collaborative Filtering (UAI). (1998)

[2] Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J.:"Item-Based Collaborative Filtering Recommendation Algorithms". Proceedings Mangalindan, JP. 2012. "Amazon'S Recommendation Secret." Fortune. Linden, G., Smith, B., York, J."Amazon.Com Recommendations: Item-To-Item Collaborative Filtering".