Recommendation System using Cosine Similarity of Content Based Collaborative Filtering

Vanshaj Jain, UtkarshGoel, VipulTyagi

ABES Engineering College

[1]Anmol Jain
ABES Engineering College

Abstract

The recommendation system plays an important role in the digital world and is used in many applications like shopping electronics gadgets like smartphones, TV. The recommendation system is also required in the field of entertainment like movies, online games, etc. A recommendation system predicts a bit of advice to the user based on its previous experience of recorded statistics. This recommendation system can be implemented various techniques like HMM, Bayesian and decision tree. In this paper, we represent details of the approaches and techniques used in a basic recommendation system. A Recommendation system is broadly categorized into two categories: (i) Collaborative Filtering, (ii) Content-based Filtering. In this paper, we have implemented an efficient model to perform recommendations using cosine similarity. The paper elaborates on content-based filtering and content-based filtering. The results of the recommendation system are generated to predict the movie.

Keywords: *Recommender system, collaborative-based filtering, content-based filtering, hybrid filtering, evaluation, HMM, Cosine similarity*

1. Introduction

The recommendation system is a system used to filter information. Its work is to observe choices of customers. And then make recommendations worked upon their choices. They have become quite useful in recent times and hence are generally used in online resources [1]. Applications for recommendation systems include a wide range of preference recommendations. Usually, known platforms are pictures, books, music, videos, movies, friends on the internet. Its application moves to other areas such as retail websites, to people on dating and professional websites. It also helps in recommending results on Search Engines. Some more examples are, Instagram and WhatsApp. We can draw a pattern of user's interaction with their posts on their social media account and hence, learn what types of posts they like. It is very useful because recommendation system's main purpose is to gather data of the particular user and take observe them so that it can be used in the future to better the suggestions. There are instances in which the recommender systems make changes that are reflected by the searches made by the users. This can be understood by an example: if Flipkart takes note that many people purchasing the T-shirts also purchase trousers and pants. From this, they can prefer pants and trousers to new users based on past searches. As these recommender systems are advancing, the purchasing users are expecting fresh results. If the result are not useful enough for the users than it can be very dangerous for the firms. For example, retailer website is not able to recommend the user preferences, then the customer will just give up on that website [1]. Seeing this there has been great investments by the technology giant to improve their suggestion systems. But, the problem is a lot complex than it appears on a theory. It is true that every customer has his/her own choices and preferences. It is also true that the preferences of one customer can change very frequently. This depends on a number of elements, for example state of mind, activity the customer, time of the year. This can be proved with an example, the type of films a user would watch with close ones will be different from the one they would witness with mates. The first approach is content-based filtering, the other is collaborative filtering. Content based filtering is based on analysis of one user while collaborative is used for many users.

2. Background

There are different methods and ways to give suggestions that can be rating or content information; however other suffers with same constraints. There have been many attempts to overcome the constraints using collaborative-based filtering and content-based filtering together as a hybrid approach that can combine ratings and content information.

The following methods can be used:

a. Hybrid system

b. Content based system

c. Collaborative Filtering system

The difference between this three can be stated as: in collaborative based approach, matching elements of the customer from the past can be suggested to another customer, while in the content based approach [2], elements that likely people with similar likings can be suggested. There are certain limitations in these two which can be overcome by the hybrid approach which is the combination of both content-based filtering and collaborative based filtering.

3. Algorithm

The Algorithms used for this project are the two most widely used algorithms: Collaborative-based filtering and Content-based filtering.

1. Collaborative-Based Recommendations

This method provides suggestions to the customer which was rated by other users. Main concept behind this is that the users which have rated something similar elements, then the elements that one customer has liked can be suggested to other user. Collaborative filtering method is seen on various social media such as, Facebook, Netflix, and Flipkart. The two sub-methods that can be utilized are as follows:

a. Nearest Neighbors Collaborative Filtering: This method is based on its parent method i.e. collaborative filtering. The main concept is to detect users having like rating habits. To detect customers who have similar tastes. The process of this method is as follows. First it calculates the similarity between customers by using the row vector in the rating matrix.

b. Latent Factor Methods: This factoring method or algorithm decomposes the rating matrix R into two matrices Q and P.

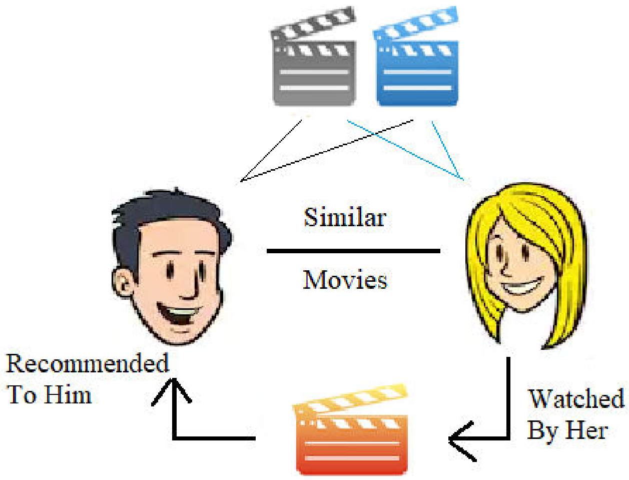

Figure 1. Collaborative Based Filtering

2. Content-Based Recommendations

The other method which will be used in this project is the Content-Based filtering method. It observes the likings and disliking's of the customer. It then makes a personal profile. To make an individual profile, the algorithm takes into account the item profiles and their corresponding user rating. Once the user profile is generated, we calculate the similarity of the user profile with all the items in the dataset, which is calculated using cosine similarity between the user profile and item profile **[1]**. Advantages of Content Based approach is that data of other users is not required and the recommender engine can recommend new items which are not rated currently, but the recommender algorithm doesn't recommend the items outside the category of items the user has rated**[1]**.

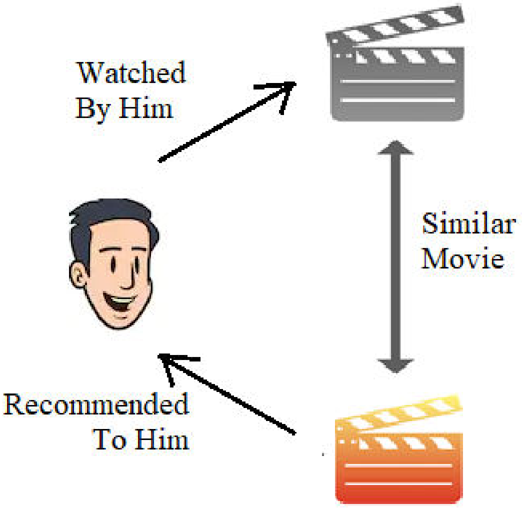

Figure 2. Content-Based Filtering

Advantages:

The advantages of the system are as follows:

•**Independent from user:** The biggest advantage is that the profile generated is of the individual and not that of a group. Only the single user choices are taken into account and not that of a group. This helps the individual to see recommendations that are solely based on his/her preferences.

•**Introduction of new Elements:** This system has the ability to suggest elements that are not even rated by any customer. But this problem is faced by collaborative method whose approach is based on group likings.

•**It is transparent:** This system is much more transparent than any other recommendation methods. In collaborative method the transparency levels are not quite clear because of the large data set of groups that have to calculate by the algorithm. Here only individual accounts are taken care of.

4. Cosine Similarity

Cosine similarity is a measurement of degree of similarity between two non-zero vectors that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval (0,π] radians. So, by using cosine similarity, we can assume that it is just an observation of orientation only and neglecting magnitude: vectors with same orientation have a cosine similarity as 1, while on the other hand two vectors at 90° to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. The Cosine Similarity is known as the Vector Similarity or Cosine coefficient **[4].** This metric assumes that common rating items of two users are two points in a vector space model, and then calculates cosϴ between the two points.

$$similarity = \cos(\theta) = \frac{A.B}{||A||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (1)$$

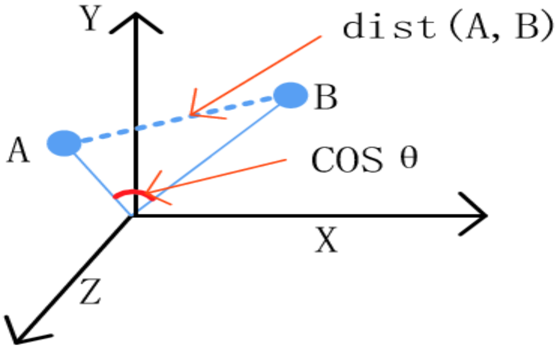

Figure 3. Cosine Similarity

5. Data Set

In the recommendation system we gather data from the following:

- GitHub
- IMDb
- Kaggle

- Manually

In total the dataset counts for over 13000+ movies. In which 9000+ are Bollywood movies and 4000+ are Hollywood movies.

The record is updated to the latest timeline.

6. Combine Column

There is more to the algorithm than just making suggestions just on movie names. The algorithm also uses a separate search for elements associated with the movie **[1].** These could be the actors, the directors, or the writers. Also the two searches are not dependent on each other. In fact using the elemental search, we will not get results for queries appearing in the movie plots, and visa-versa **[2].** So, in order to include the elemental search for ambiguous queries that could appear in either plot of in a name, such as Hill, we expanded the number of documents that are returned by the index servers **[3].**

7. Flow Diagram

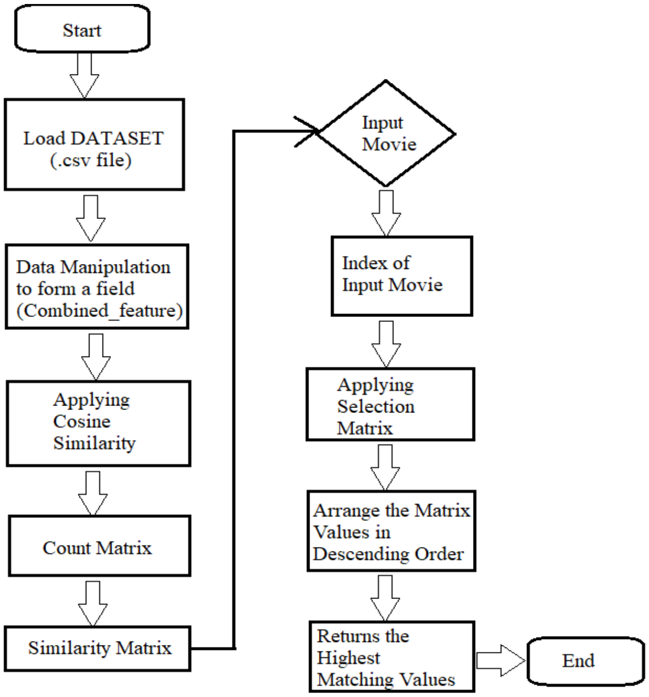
Start
Load DATASET (.csv file)
Data Manipulation to form a field (Combined_feature)
Applying Cosine Similarity
Count Matrix
Similarity Matrix
Input Movie
Index of Input Movie
Applying Selection Matrix
Arrange the Matrix Values in Descending Order
Returns the Highest Matching Values
End

Figure 4. Flow Diagram Procedure

Step 1. Start: The following process will depict the flow of the process from scratch till end.

Step 2. Load DATSET: The data set is loaded in the background so that when an input is given it can be processed in early phase.

Step 3. Data Manipulation: The data set is manipulated according to the need. It is based on this data set the similarity matrix is formed which form the core part of the recommendation system. It uses the combine column on which final processing is done. The combine column includes all the parameters on which cosine similarity feature will work. The combine column has movie name, director, year of release, genre, actor name etc.

Step 4. Cosine Similarity: Since the combine column is made, the raw data needs to be processed. So the cosine similarity is applied in the combined column to bring out results. As mentioned earlier, the cosine similarity is a mathematical approach to find similarity between movies, hence, this feature is already running in the background before user can input its requirements.

Step 5. Count Matrix: It is a part of cosine similarity. It is formed when cosine similarity is allowed to work on the combine column. It consists of the numeric pairs which can be represented graphically.

Step 6. Similarity Matrix: After the count feature works, it is time to convert them in cosine values. The similarity matrix has the same work. It converts the angle between the movies in cosine form. It is a symmetric matrix. Value ranges from -1 to 1.

Step 7. Input Movie: The backend processing is over till this step. Now the user can enter the movie that he/she needs to search. The interface takes in the input and presses search.

Step 8. Index of the Movie: The index of input movie is defined before further processing.

Step 9.Applying Selection Matrix: The selection matrix is applied on the input movie. The matrix that was formed earlier is now used to find similar movies.

Step 10.Arranging: Now since, the matrix is formed the values are arranged in descending order. The one with the highest cosine value is placed higher in the list and so on so forth.

Step 11.End: Finally the list of movies similar to the movie is printed.

8. Evaluation of Recommendation System

Accuracy is the measure of suggestion system to the extent it can predict those items that a user have already rated or interacted with. Thus suggestion systems which are able to optimize accuracy [9] will be the ones that will place those items at the top of a customer's list.

Following are the methods for determining the accuracy:

a. **Training Test**

- This is the simplest approach to evaluate our recommender system.
- Training data is 70-80%. While Test data is 20-30%.

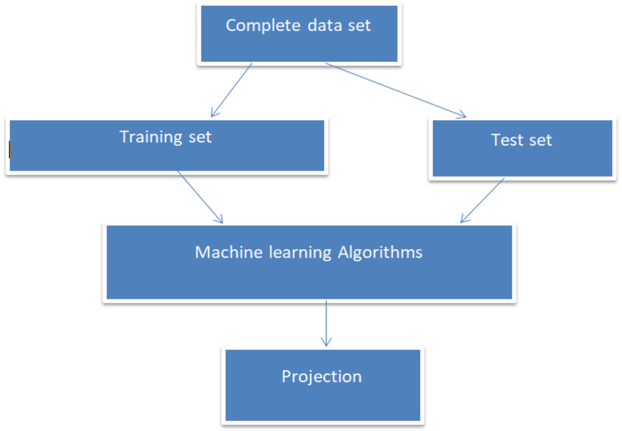

Figure 5. Pictorial Representation of Training Method

b. K-Fold Cross-Validation

- This method is quite similar to Training test method. The only difference is the training set is divided into k-equal parts.
- This decreases the monotonicity of the results.

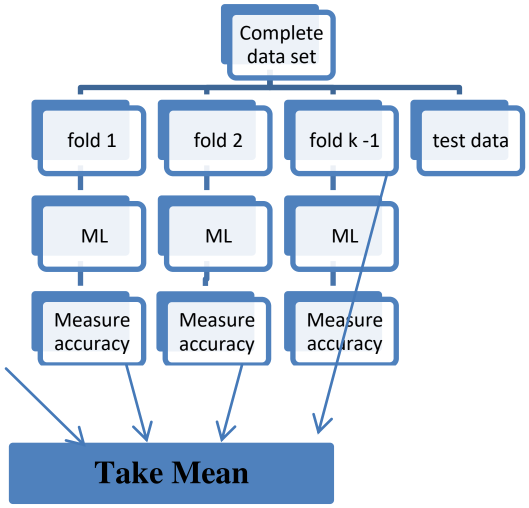

Figure 6. K-Fold Cross-Validation

c. Mean Absolute Error (MAE)

- In movie recommendation system, Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon.

$$M.A.E. = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{2}$$

d. Root Mean Square Error (RMSE)

In movie recommendation system, root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.

e. Hit Rate (HR)

Hit: If a recommendation by the algorithm matches with the movie rated by the user than it is referred to as a Hit. Higher the Hit Rate better is our recommendation algorithm.

$$hit\ rate = \frac{hits}{users} \tag{3}$$

f. Leave-One-Out Cross Validation

- This method is an improvement over the hit ratio method.
- We leave out one recommendation intentionally out of the list of N recommendations from the training set. The left item is given to the Test data.
- If the test data is able to recommend the left over item than algorithm is said to be working correct.
- The only drawback is that it only works with very large data sets.

g. Average Reciprocal Hit Ratio

- This method is reciprocal of Hit Rate.
- The difference is that we take up the sum of the reciprocal of the hits.
- This algorithm gets more credits for recommending movies in the upper slot than in the bottom slot.

$$A.R.H.R. = \frac{\sum_{i=1}^{n} \frac{1}{rank_i}}{Users} \tag{4}$$

h. Cumulative Hit Ratio

- This method is useful where fewer recommendations are to be made.
- In this method the recommendations with ratings below a particular threshold value are eliminated [10].
- It is more precise way of evaluating an algorithm.

Table 1. Example of Hit Ratio

HIT RANK	**Predicted Rank**
4	5.0
~~**2**~~	~~**3.0**~~
1	5.0
~~**10**~~	~~**2.0**~~

9. Result Set

Following accuracy measures (RMSE, MAE, Hit Rate, and ARHR) were calculated on our data set for movie recommendation system. The results found for most of the cases are very near to other algorithm. Hit rate predicts that higher the hit ratio, higher the accuracy for recommendation. In our study, it was found the hit rate is having better accuracy than the other existing algorithm. ARHR accuracy parameter is also in favor of the proposed algorithm. Therefore, recommendation system based on Cosine similarity based content based collaborating filtering is having good results for recommendation system.

TABLE 2. Observed Results

Method	**RMSE**	**MAE**	**Hit Rate**	**ARHR**
Risk Aware Recommender System	1.236	0.987	0.01356	0.0245
Multi Criteria Recommender System	1.78	0.874	0.01789	0.0237
Mobile Recommender System	1.098	0.901	0.0045	0.0279
Proposed Algorithm	1.389	0.6978	0.0298	0.0112

10. Issues Faced

a. Scalability Issues

The Scalability challenges that are to be faced working with the large dataset is that of memory restrictions. The problem is that the data because of its huge cannot be stored as a dense matrix **[8].** So the algorithm requires use of sparse matrix representations so that program works without memory problems. Going a little bit further, we see that middle results cannot be recognized or computed such as the user-user similarity matrix.

b. Broken links

Meta-data in this project was collected by scrapping out small elements from the Kaggle Website. The smaller dataset provided auto-generated links to the movies URL based on the movie’s title and release year **[1]**. This caused a large portion of the links to broken. Sometimes the movie names should ambiguous behavior leading to a search page with recommendations rather the movie page **[1]**. For others, there was some sort of error in the reference to the link **[1]**.

11. Future Scope

There is plenty of way to expand on the work done in this project. The first one being the content-based method. It can be elongated to include more areas to help classify the movies **[1].** Common areas or ideas are to add features to suggest movies with common actors, directors or writers. Also it can be included with movies released within the same time period **[1].** Similarly, the movies total gross could be used to identify a user’s taste in terms of whether he/she prefers large release blockbusters, or smaller indie films **[1].** However, the above ideas may lead to over fitting, given that a user’s taste can be highly varied, and we only have a guarantee that 20 movies (less than 0.2%) have been reviewed by the user[1]. Moreover, it can include hybrid method.

12. References

[1]Sappadla, P. &Sadhwani, Y. &Arora P. (2017). Movie Recommendation System: https://pdfs.semanticscholar.org/767e/ed55d61e3aba4e1d0e175d61f65ec0dd6c08.pdf.

[2]Choi, M. &Eom, H.(2010). A Smart Movie Recommendation System.Collaborative Filtering.

https://link.springer.com/chapter/10.1007/978-3-642-21793-7_63.

[3]Seyednezhad,S.M. & Nobuko Cozart, K. & Anthony Bowllan, J.(2018). A Review on Recommendation Systems: Context-aware to Social-based. Content-Based Filtering. https://arxiv.org/pdf/1811.11866.pdf.

[4]Bhatt, B. & J Patel, P.(2014). A Review Paper on Machine Learning Based Recommendation System.Cosine Similarity.https://www.ijedr.org/papers/IJEDR1404092.pdf.

[5]Hee Park, D. &Kyeong Kim, H. & Young Choi, Y.(2011). A Review and Classification of Recommender Systems Research.Data Collection.http://ipedr.com/vol5/no1/62-H00141.pdf.

[6]Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie Recommendations", 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, 2013.

[7]Munoz-Organero, Mario, Gustavo A. Ramíez-González, Pedro J. Munoz-Merino, and Carlos Delgado Kloos. "A Collaborative Recommender System Based on SpaceTime Similarities", IEEE Pervasive Computing, 2010.

[8]Al-Shamri, M.Y.H.. "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model", Expert Systems With Applications, 200810.

[9]Hu Jinming. "Application and research of collaborative filtering in e-commerce recommendation system", 2010 3rd International Conference on Computer Science and Information Technology, 07/2010.

[10]Yan, Bo, and Guanling Chen. "AppJoy : personalized mobile application discovery", Proceedings of the 9th international conference on Mobile systems applications and services - MobiSys 11 MobiSys 11, 2011.