International Journal of Recent Trends in Engineering & Research
ISSN (ONLINE) : 2455 – 1457
IMPACT FACTOR : 4.101

Analysis of Movie Recommender System using Collaborative Filtering

Debani Prasad Mishra[1], Subhodeep Mukherjee[2], Subhendu Mahapatra[3], Antara Mehta[4]
[1]*Assistant Professor, IIIT Bhubaneswar*
[2,3,4]*Btech,IIIT, Bhubaneswar,Odisha*

Abstract—A collaborative filtering algorithm works by finding a smaller subset of the data from a huge dataset by matching to your preferences. It combines these preferences to create a list of required suggestions. There are several different ways of deciding which people are similar and combining their choices to make a list. Collaborative filtering algorithms have been in use for Recommender systems for a long time now. They have been successful in solving many issues of the systems which are in use in the market. Neighborhood based methods have shown promise in predicting user ratings. In this article we present some of more potential areas of working on Collaborative Filtering technique also we examine approaches for combining multiple algorithms for predicting user ratings and also discuss some results from the analysis of various strategies used by prior researchers and find solutions to them.

Keywords-Collaborative Filtering, Movie Ratings Recommender Systems,Similarity.

I. INTRODUCTION

The low-tech ways to get recommendations for song,movies, and e-commerce websites have been known already. We know that some of the peers have better "taste" than others and this can be learnt simply by observing things around . As technologies advanced,the people choices differed and it became difficult to group their opinion on the basis of age,country,religion or sex. So we needed a powerful technique called Collaborative Filtering to concentrate on a large group of people. The potential of Collaborative Filtering to increase the prediction rate has been satisfying. The advantages of Collaborative Filtering are-- The notable advantage is that CF systems can produce personalized recommendations, because they consider other people's experience and recommendations are based on that experience. Another notable advantage is that the CF recommender systems can suggest serendipitous items by observing similar-minded people's behavior.

In this article,we will focus on Nearest Neighbor algorithms ,Euclidean Distance,Pearson Correlation Score and ranking and present ways to implement them on dataset for recommending movies in a Movie Recommender System.

II. PROBLEM STATEMENT

A. *Collecting Preferences*

The first thing we would need to collect is the preferences of the different people. These preferences will possibly be the ratings (as in case of a Movie Recommender System) of the critics who have rated some movies in the past. The ratings will be on a scale of 1 to 5 where 1 depicts strong dislike and 5 suggest a strong liking for the movie. A rating of 3-4 would mean average opinion on that movie. The value of rating will change for a shopping site where we need to use a value of 1 to indicate that a user has bought that item in the past and a value of 0 if the item is not bought by the user. For a voting site,-1 means 'down-vote',0 means 'did not vote' and 1 means 'up-vote'.Use of databases for storing the prior records of users or critics is recommended for large datasets.

	Air Lift	Neerja	MsDhoni	3 idiots
Komal Nahta	5	2	5	3
Nangagopal	2	4	3	4
Rajeev Masand	2		2	3
Deepa Gahlot	5	1	?	3

Table 1:The prediction algorithm should try to predict the ratings on MsDhoni for Deepa Gahlot.

B. Finding similar users

After the data has been collected,a technique has to be devised to present similarities between different users and generate a better prediction for new users. A Similarity score is calculated by finding similar choices between one users to every other user.

III. IMPLEMENTING TRADITIONAL ALGORITHMS.

The most promising approaches to find the similarity scores between two users are Euclidean Distance and Pearson Correlation .

A. Euclidean Distance Score

Calculation of the similarity between two people who have rated similar items can be done with Euclidean Distance Score.

It takes two common items rated by all the users(critics) into two axes and plot them on the chart .

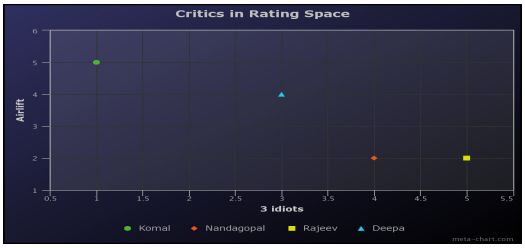


Fig 1: Critics in Rating Space

Rajeev has been plotted at 5 on the 3 Idiots axis and 2 on the Airlift axis. The closer two critics are on the rating space,the similar are their ratings. This means the two critics share a similar taste of movies. The distance between Rajeev and Deepa can be calculated by taking the square of the difference in each axes by adding them together and finally doing the square root of the sum. This can be done by the formula-

$$\text{dis}(x,y)= \sqrt{(xn - yn)2}$$

This formula has been tested and found to be unsuccessful in dealing with larger datasets since it gives small similarity values for people who are similar in nature. Therefore a new way to get the higher similarity values for similar critics have been proposed by finding inverse of the distance function by adding 1 to it.[**2**] A value of 1 would suggest that two critics share similar taste of movies and a value near to 0 would show how two critics are dissimilar in their tastes of movies.

$$\text{New-dis}(x,y)= \frac{1}{1+dis(x,\,y)}$$

B. Pearson Correlation

This method looks at the degree to which things vary together as a relation to how much they vary individually. The similarity between items can also be given by their correlation which measures the

linear relationship between objects or the correlation coefficient is a measure of how well two sets of data fit on a straight line. Traditional Pearson correlation coefficient does not consider the size of the set of common users. To overcome this problem, weighted Pearson correlation coefficient has been proposed .It considers the idea of capturing the confidence which can be placed on the neighbor. The confidence will increase with the number of common rated items.

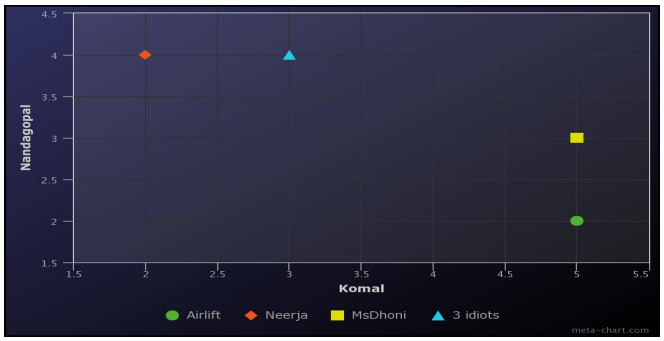


Fig 2:Peason Correlation

The formula for PCC(Pearson Correlation coefficient) is somewhat more complicated than Euclidean Distance score but in situations where data is not well normalized it worthwhile to deploy PCC rather than using Euclidean Distance[. Pearson correlation score first finds the items rated by both critics. Secondly, it calculates the sums and the sum of the squares of the ratings for the two critics, and calculates the sum of the products of their ratings. Finally, it uses these results to calculate the Pearson correlation coefficient.

Mathematical Formula for calculating similarity by Pearson Coefficient-

$$\text{Pearson}(x,y) = \frac{\sum (x, y)}{x \square y}$$

Depending upon how close are the items on the chart a best line can be generated. If the two critics had identical ratings for every movie, the line would be diagonal and would touch every item in the chart, giving a perfect correlation score of 1.

C. K-Nearest Neighbour

The two approaches discussed above fails considerably for larger datasets and therefore an approach is required for elaborate recommendations. K-Nearest Neighbor has a potential to store previous ratings and predict new recommendations for critics in unseen cases. This approach memories the entire record of the critics' past ratings and give recommendations on the attributes of the new movie that exactly match from the dataset.

Given a movie to be rated,this algorithm finds k closest critics who would rate the movie (nearest neighbors) from the records.

The underlying idea behind KNN approach is that if a dataset element falls in a particular neighborhood of a movie to be rated where the ratings from a critic is predominant ,it is likely to rated by that critic.It is always advisable to take least number of k-neighbors to reduce the dangers of noise in the data.

The important step that we need to take care of is to find similarity between two movies and this function can be calculated by using any of the similarity methods of Euclidean Distance Score and Pearson Coefficient .It has been found that k-NN algorithm is easy to implement[3].Although with some disadvantages,it has an advantage of not forming new training datasets every time new movie is added up. This function finds the distance between the new movie and every other movie in the dataset and puts the most relevant ones on the top.

Weighted kNN can be used as an alternative for kNN algorithm.Instead of calculating simple averages, this method uses weighted average method which is done by multiplying each movie's weight with its rating then adding them together and divided by the total of the weights.

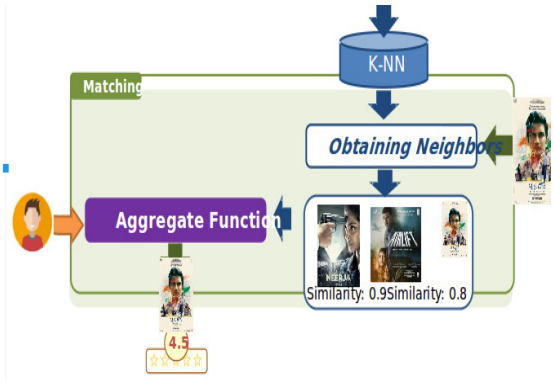


Fig 3:Obtaining Neighbors

IV. RECOMMENDING MOVIES

It is great to find a critic to rate a movie but a movie recommender system looks for a critic who shares a similar taste for movies with the new critic and suggest him a movie which is rated by previous critic to this new critic who hasn't rated that particular movie. But this approach would bring in critics who haven't rated some movies that the new critic likes. This becomes a potential problem for the system. For solving this issue,a method is proposed that will calculate how similar a critic is with all the critics specified. It will then iterate through every movie for which a critic has given a rating.

The final score for every movie is calculated by multiplying the rating for each similiar movie that the critic has rated to the similarity of the two critics .Normalization is done by dividing each score by the sum of the similarity. The scores are then sorted in descending order to get a ranked list of movies. We can also guess the rating the each critic would give to that movie. Choosing a different similarity algorithm would give only a slight change in recommendations.

CRITIC	SIMILARITY	FAN	RECOM. FOR FAN	MERI PYAARI BINDU	RECOM. FOR MERI PYAARI BINDU	RAABTA	RECOM. FOR RAABTA
Komal Nahta	0.78	3.5	2.73	4	3.12	4	3.12
Nangagopal	0.64	4	2.56	3.5	2.24	4	2.56
Rajeev Masand	0.82	3.5	2.87	3.5	2.87	4	3.28
TOTAL	2.24		8.16		8.23		8.96
Final score			**3.6428**		**3.6741**		**4**

Table 2: User based recommendations

In this table we can see that we get a recommendation for some new movies Fan, Meri Pyaari Bindu and Raabta. First the similarities of the current critic was found out with other critics that have already rated similar movies. Now the similarities of the critics like for Komal Nahta(0.34) is multiplied to rating given to Fan by Komal Nahta(4) to get the recommendation for fan for the current critic from Komal Nahta. Following the same method the table is filled. To get the normalized result the sum of the similarities and the sum of the recommended ratings,which are

present in the total row, are divided to get the final score, in case of fan the total reomadation(8.16) is divided by the total similarity(2.24) to get the final score 3.6741.

Now that we have found the similar critics ,we recommend movies for different critics .To let the critics see what their peers are interested in,we can show the movies as well to the new critic.

Critics who rated this also rated:

Rang de Basanti,2006
Sultan,2016
Dangal,2016
Lagaan,2001
P.K,2014

Table 4 :Example of a simple movie recommender system

In this case, similarity can be found by showing who rated a particular movie that a critic would like to rate and what other movies the person had liked or disliked in past.

The other method that can be used here is flipping the critics and the movies. The movies and the critics are swapped to form a new method of similarity. This method yields good results but in many situations it will give interesting comparisons. In alternative recommender system, this strategy will help in marketing and clearing of some items.

V. ITEM BASED FILTERING

The critic similarity method can be used for a small number of critics in a class or a locality but when our datasets are as vast as a city or the nation then comparing one critic to another would be very difficult as there could be lot of critics. So to solve this problem we could compare plus critics from different areas might have a rated different movies like a critic from the south may have rated a lot of the movie from the south and vice a versa for a critic from north. This phenomena could effect the similarity of two critics hence another method of knowing the rating of a unknown movie was introduced[7].

The method that was used till now is called the user-based collaborative filtering. We saw its drawbacks so a new alternative for this is used which is called item based collaborative filtering which solves the above problems. This technique includes the elimination of other critics as it takes into the account the similarities of two movies and hence we get our recommendation for a movie with the help of the ratings we have given to the movies which are similar to it.

A.FINDING SIMILAR MOVIES

This technique includes to compute the list of the most similar movies of every movie using any of the similarity method mentioned earlier but the major difference here is that instead of comparing two critics and there similar movies they have rated, this technique would compare two movies and the similar critics those who have rated them. It will iterate through every movie for which a critic has given a rating and check for a similarity.

B. GETTING A RECOMMENDATION:

The final score for every movie is calculate by multiplying the movie that the critic has rated before to the similarity of that movie to the movie that the critic require a rating for. Normalization is done by dividing each score by the sum of the similarity. The scores are then sorted in descending order to get a ranked list of movies. We can also guess the rating the each critic would give to that movie. Choosing a different similarity algorithm would give only a slight change in recommendations.

MOVIE	RATING	FAN	RECOM. FOR FAN	MERI PYAARI BINDU	RECOM. FOR MERI PYAARI BINDU	RAABTA	RECOM. FOR RAABTA
DANGAL	4	0.34	1.36	0.45	1.8	0.81	3.24
LAAGAN	3	0.45	1.35	0.52	1.56	0.24	0.72
SULTAN	3.5	0.67	2.345	0.62	2.17	0.46	1.61
TOTAL		1.46	5.055	1.59	5.53	1.51	5.57
NORMALIZED			**3.4623**		**3.4780**		**3.6887**

Table 5: ITEM BASED FILTERING

Here in this table we can see that we get a recommendation for some new movies Fan, Meri Pyaari Bindu and Raabta. First the similarities of this movie were found out with movies that the critic has already rated in these case Dangal, Laagan and Sultan. The similarities of the two movies like for Fan and Dangal(0.34) is multiplied to rating given to Dangal by the critic(4) to get the recommendation for fan from Dangaal. Following the same way the other ratings are found out. Then the sum of the similarities are found out for a particular movie (1.46 for Fan) and the sum of the recommendation are then found out(5.055 for Fan). The normalized recommendation is then found out by dividing the two total ie. 5.055/1..46=3.4623 for Fan.

VII. ITEM OR USER BASED FILTERING

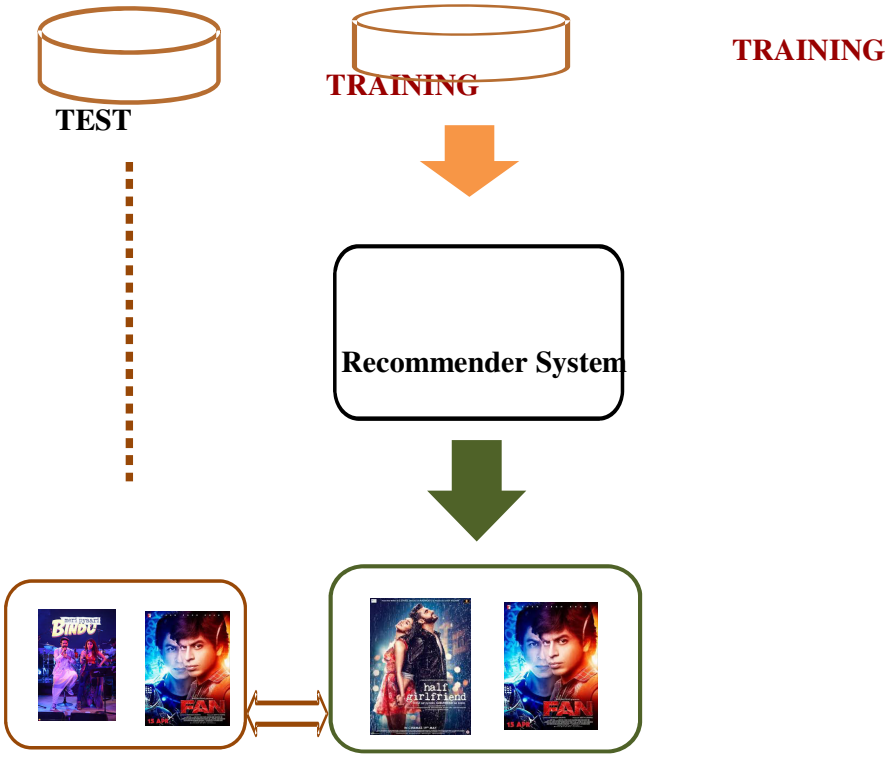
Item based filtering is significantly faster than user-based filtering when there is a need to get a list of recommendations for a large dataset ,but it requires an extra overhead of maintaining the similarity table. In addition to this ,there is a difference in accuracy which depends upon the “sparsity” of the dataset. In the movie recommender system, since every critic has rated nearly every movie,the dataset is dense(not very sparse). Item based filtering usually perform better than user-based filtering in sparse datasets, and perform almost equally in dense datasets. There are huge differences and similarities between the two genres of filtering as citied in the paper[2] by Sarwar.

User based filtering is simpler to implement and doesn't need more steps so it is beneficial for small datasets which are frequent to changes. Finally, in some applications like many shopping sites and recommending sites may require this method,which shows how two critics share similar tastes .

VIII. CROSS VALIDATION

Another approach to deal with large dataset is to apply a set of techniques to divide the dataset into *training sets* and *test sets.* The training set is given to the algorithm along with the right answers ,and it becomes the set to be used for making predictions. The algorithm is then asked to make predictions for each movie in the test set. The answers or the predictions it gives is matched with the correct answers. The overall score is calculated to know how well the algorithm has performed. Usually this procedure is performed many times,dividing the data differently in each case.

The test set will be a small portion ,perhaps a 5 per cent of all the data and remaining 95 percent consists of training data.



To start,you need to create a function that divides the dataset into two smaller dataset according to the ratio we provide. The next step is to test the recommender system algorithm by giving it a training set and calling it with each movie in the test set. The function should calculate the differences and combine them to create an aggregate score for how far off was in general by adding up the squares of all the differences.

Squaring the numbers is a common practice because it makes large differences visible and countable. This means an algorithm that is very close most of the time but often on occasion far off from vicinity will suffer more than the algorithm that is closer. This is often a required step as there might be situations in which a big error will be acceptable if the accuracy rate is very high the rest of the

time. When this is the case,we can bring changes in the function to just add up the absolute values of the differences[1].

The final step is to create a function that makes several different divisions of data and runs the test algorithm on each division and adding up all the result to get a final score.

IX. OPTIMIZING THE SCALE

In many cases ,it's not difficult to choose good parameters for scaling because you know in advance which variables are important. But in most cases ,you will be working on dataset that are not created by you and it will be difficult to choose which variables are important and which variables are unimportant so that we can remove them. Optimization focuses you to specify a range to choose the numbers,variables and a cost function. The only thing you need to do is wrap it so that it takes a list of inputs as arguments, normalize the data, and calculates the cross-validation error.

The domain is the range of weights for each dimension. In a movie recommender system , the lowest possible value is 0 because negative numbers will indicate an error in the rating .In theory, the weights can be as high as you want, but for practical purposes, ratings should be restricted to 5.

An advantage of optimizing the variable scales in this way is that you immediately find out the significant variables

and how their importance. In some cases, we can avoid the extra cost by finding out the unwanted data in the data which is difficult and expensive to find out before hand. In other cases, just knowing which variables are important—particularly in determining weights—may affect what you choose to emphasize as part of a marketing effort, or may reveal how products can be designed in a different manner to get the highest prices.

X. UNEVEN DISTRIBUTION

So far we have assumed that that if we take an average or weighted average of the data, we'll get a fairly good estimate of the final rating . In many cases accuracy level will be high, but in some situations there may be an insignificant variable that can cause big effect on the result.

To do this, you need a function that returns a value between 0 and 1 representing the probability. The function first calculates the weights of the neighbors within that

range, and then calculates the weights of all the neighbors. The probability is the sum of the neighbor weights within the range divided by the sum of all the weights. This function gives great results .The ranges that are away from the actual ratings have probabilities of 0, and the ones that attract the full possibilities are nearly 1. By breaking it down into smaller divisions , you can determine the actual ranges in which things tend to collect.

XI. CONCLUSION

Movie Recommender Systems are software tools that and techniques providing suggestions for movies to be rated by a critic. The suggestions provided are aimed at supporting

the critics in various decision-making processes, such as what movies to rate, what music from the movie users will like , or what new movies they have to rate. Recommender systems have proven to be necessary method for users to cope with the information bloom and have become most powerful and popular tools in e-commerce.

Simultaneously,various algorithms for recommendation systems have been proposed over last decade, many of them have also been successfully implemented in commercial markets and some still suffer from demerits and waiting to be corrected.

Recommender systems is a multi-disciplinary effort which involves expertise from various fields of Artificial intelligence, Human Computer Interaction, Information Technology, Data Mining, Statistics, Adaptive User Inter-

faces, Decision Support Systems, Marketing, or Consumer Behavior. The first part of this paper presents the most
popular and fundamental techniques used nowadays for building movie recommender systems that is Collaborative Filtering. The second part presents techniques that can be used to evaluate the quality of the movies recommendations.
The informative and new approaches will provide researchers, students and practitioners in industry with a comprehensive, yet concise and convenient reference source to movie recommender systems[6].

REFERENCES

[1] Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 230–237 (1999)
[2] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item–based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International World Wide Web Conference pp. 285–295 (2001)
[3] O'Mahony, M.P., Hurley, N.J., Silvestre, C.C.M.: An evaluation of neighbourhood formation on the performance of collaborative filtering. Artificial Intelligence Review 21(1), 215–228(2004)
[4] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW '94, Chapel Hill, NC.
[5] Ungar, L. H., and Foster, D. P. (1998) Clustering Methods for Collaborative Filtering. In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence.
[6] Ricci Francesco, Rokach Lior, Shapira Bracha, Kantor.B Paul Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners
[7] Segaran Toby:Programming Collective Intelligence Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
[8] Peng Xiao, Liangshan Shao, Xiuran Li (2013):Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie Recommendations In Intelligent Systems Design And Engineering Applications,2013 International Conference IEEE.
[9] Linden Greg, Smith Brent , and York Jeremy Industry Report on Amazon.com Recommendations Item-to-Item Collaborative Filtering
Ignatov I.Dimitri, Poelmans Jonas, Dedene Guido, Viaene Stijn (2012) A New Cross Validation Technique to Evaluate Quality Of Recommender Systems

