A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System

IOP Conf. Series: Journal of Physics: Conf. Series **1000** (2018) 012101 doi:10.1088/1742-6596/1000/1/012101

A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System

Geetha G[1],Safa M[2] , Fancy C[3] , Saranya D[4]

[1, 2 , 3]Department of Information Technology, SRM Institute of Science and Technology

[4] Department of Electronics and Communication Engineering, Arasu Engineering College

E-mail : geetha.g@ktr.srmuniv.ac.in, safa.m@ktr.srmuniv.ac.in

Abstract- In today's digital world, it has become an irksome task to find the content of one's liking in an endless variety of content that are being consumed like books, videos, articles, movies, etc. On the other hand there has been an emerging growth among the digital content providers who want to engage as many users on their service as possible for the maximum time. This gave birth to the recommender system comes wherein the content providers recommend users the content according to the users' taste and liking. In this paper we have proposed a movie recommendation system. A movie recommendation is important in our social life due to its features such as suggesting a set of movies to users based on their interest, or the popularities of the movies. In this paper we are proposing a movie recommendation system that has the ability to recommend movies to a new user as well as the other existing users. It mines movie databases to collect all the important information, such as, popularity and attractiveness, which are required for recommendation. We use content-based and collaborative filtering and also hybrid filtering, which is a combination of the results of these two techniques, to construct a system that provides more precise recommendations concerning movies.

I. Introduction

In today's world where internet has become an important part of human life, the users are facing problems of choosing due to the wide variety of collection. Searching from a motel to good investment options, there is too much information available over the internet. To help the users cope with this information explosion, companies have deployed recommendation systems for guiding their users. The research in this area of recommendation systems has been going on for quite a long time but the interest still remains high because of the abundance of practical applications and the problem rich domain.

Recommender systems are used for providing personalized recommendations based on the user profile and previous behaviour. Recommender systems such as Amazon, Netflix, and YouTube are widely used in the Internet Industry. Recommendation systems help the users to find and select items (e.g., books, movies, restaurants) from the wide collection available on the web or in other electronic information sources. Among a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Similarly, a movie recommendation system provides a level of comfort and

personalization that helps the user to interact better with the system and watch the movies that best matches to his needs. The main purpose of our system is to recommend movies to its users based on their viewing history and ratings that they provide. The system will also recommend their products to specific customers based on the genre of movies they prefer. Collaborative filtering and content based filtering are the prime approaches in providing recommendation to the users. Both of them are best applicable in specific scenarios because of their respective properties. In this paper a mixed approach has been used such that both the algorithms complement each other thereby improving performance and accuracy to our system.

2. Literature survey

MOVREC [1] is a movie recommendation system presented by D.K. Yadav et al. based on collaborative filtering approach. Collaborative filtering makes use of information provided by user. That information is analyzed and a movie is recommended to the users which are arranged with the movie with highest rating first. The system also has a provision for user to select attributes on which he wants the movie to be recommended.

Luis M Capos et al. [2] has analyzed two traditional recommender systems i.e. content based filtering and collaborative filtering. As both of them have their own drawbacks he proposed a new system which is a combination of Bayesian network and collaborative filtering. The proposed system is optimized for the given problem and provides probability distributions to make useful inferences.

A hybrid system has been presented by Harpreet Kaur et al. [3]. The system uses a mix of content as well as collaborative filtering algorithm. The context of the movies is also considered while recommending. The user - user relationship as well as user - item relationship plays a role in the recommendation.

The user specific information or item specific information is clubbed to form a cluster by Utkarsh Gupta et al. [4] using chameleon. This is an efficient technique based on Hierarchical clustering for recommender system. To predict the rating of an item voting system is used. The proposed system has lower error and has better clustering of similar items.

Urszula Kużelewska et al. [5] proposed clustering as a way to deal with recommender systems. Two methods of computing cluster representatives were presented and evaluated. Centroid-based solution and memory-based collaborative filtering methods were used as a basis for comparing effectiveness of the proposed two methods. The result was a significant increase in the accuracy of the generated recommendations when compared to just centroid-based method.

Costin-Gabriel Chiru et al. [6] proposed Movie Recommender, a system which uses the information known about the user to provide movie recommendations. This system attempts to solve the problem of unique recommendations which results from ignoring the data specific to the user. The psychological profile of the user, their watching history and the data involving movie scores from other websites is collected. They are based on aggregate similarity calculation. The system is a hybrid model which uses both content based filtering and collaborative filtering.

To predict the difficulty level of each case for each trainee Hongli LIn et al. proposed a method called content boosted collaborative filtering (CBCF).The algorithm is divided into two stages, First being the content-based filtering that improves the existing trainee case ratings data and the second being collaborative filtering that provides the final predictions.

3. Proposed system

Recommendation algorithms mainly follow collaborative filtering, content-based filtering, demographics-based filtering and hybrid approaches.

Collaborative filtering:-It recommends items based on the similarity measures between users and items. The system recommends those items that are preferred by similar category of users. Collaborative filtering has many advantages

1. It is content-independent
2. In CF people makes explicit ratings so real quality assessment of items is done.
3. It provides effective recommendations because it is based on user's similarity rather than item's similarity.

Content based filtering:-It is based on profile of the user's preference and the item's description. In CBF, to describe items we use keywords apart from user's profile to indicate users preferred likes or dislikes. In other words CBF algorithm recommend items or similar to those items that were liked in past. It examines previously rated items and recommends best matching item.

Demographic: It provides recommendation based on the demographic (like age, profession) profile of the user. Recommended products can be produced for different demographic niches, by combining ratings of users in those niches.

Knowledge-based: It suggests products based on inferences about user's needs and preferences, item selection and its basis for recommendation.

Hybrid recommender: Hybrid recommender system is the one that combines multiple recommendation techniques together to produce the output. If one compares hybrid recommender systems with collaborative or content-based systems, the recommendation accuracy is usually higher in hybrid systems. The reason is the lack of information about the domain dependencies in collaborative filtering, and about the people's preferences in content-based system. The combination of both leads to common knowledge increase, which contributes to better recommendations. The knowledge increase makes it especially promising to explore new ways to extend underlying collaborative filtering algorithms with content data and content-based algorithms with the user behavior data.

Step1: Use content-based predictor to calculate the pseudo user-rating vector 'v' for every user 'u' in the
Database.

$vu, = ru,i$ □ is user u rated item i

$vu, = ru,i$ □ otherwise

Step2: Weight all users with respect to similarity with the active user.

· Similarity between users is measured as the Pearson correlation between their ratings vectors.

Step3: Select n users that have the highest similarity with the active user.

· These users form the neighborhood.

Step4: Compute a prediction from a weighted combination of the selected neighbors' ratings.

In step 2, the similarity between two users is computed using the Pearson correlation coefficient, defined below:

$$P_{a,u} = \frac{\sum_{i=1}^{m}\left(r_{a,i} - \bar{r}_a\right) \times \left(r_{u,i} - \bar{r}_u\right)}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - \bar{r}_a\right)^2 \times \sum_{i=1}^{m}\left(r_{u,i} - \bar{r}_u\right)^2}}$$

Where, ra, is the rating given to item i by user a ;
$\bar{r}$
a
is the mean rating given by user a ; m is the total number of items .
In step 4, predictions are computed as the weighted averages of deviations from the neighbor's mean:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}}$$

Where, pa, is the prediction for the active user a for item i ;
Pa, is the similarity between users a and u ;
n is the number of users in the neighborhood .

4. Implementation

The Basic K-means Algorithm

The original K-means algorithm was proposed by MacQueen. The ISODATA algorithm by Ball and Hall was an early but sophisticated version of k-means. Clustering divides the objects into meaningful groups. Clustering is unsupervised learning. Document clustering is automatic document organization. In K-means clustering technique we choose K initial centroids, where K is the desired number of clusters. Each point is then assigned to the cluster with nearest mean i.e. the centroid of the cluster. Then we update the centroid of each cluster based on the points that are assigned to the cluster. We repeat the process until there is no change in the cluster center (centroid). Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function where, k is the number of clusters, n is the number of cases is a chosen distance measure between a data point and the cluster centre is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of the following steps:

1. Select K points as initial centroids.
2. Repeat
3. From k clusters by assigning each point to its closest centroid.
4. Re-compute the centroid of each cluster.
5. Until Centroid do not change.

Figure 1. K-means Algorithm

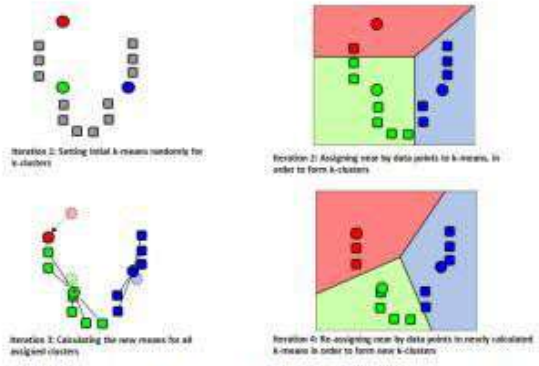
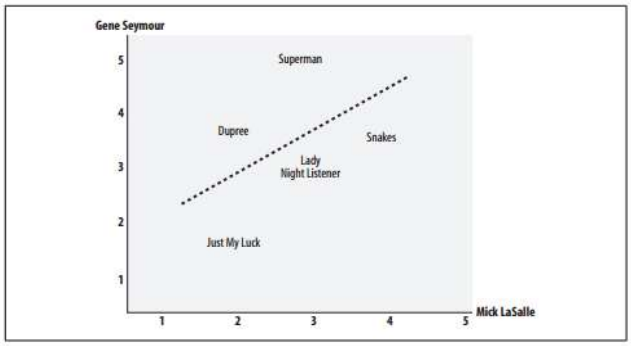


Figure 2: The 5 steps of the K-Means algorithm [3]

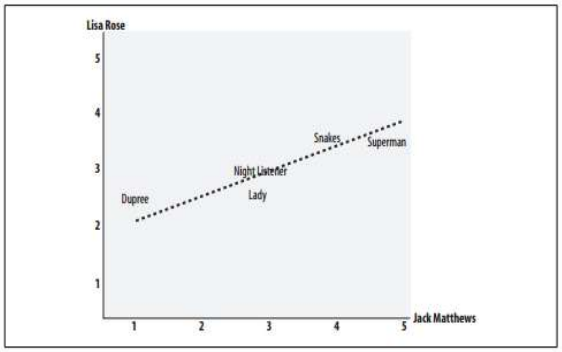
Pearson Correlation Score

A slightly more sophisticated way to determine the similarity between people's interests is to use a Pearson correlation coefficient. The correlation coefficient is a measure of how well two sets of data fit on a straight line. The formula for this is more complicated than the Euclidean distance score, but it tends to give better results in situations where the data isn't well normalized—for example, if critics' movie rankings are routinely more harsh than average. To visualize this method, We can plot the ratings of two of the critics on a chart, as shown in figure below. Superman was rated 3 by Mick LaSalle and 5 by Gene Seymour, so it is placed at (3,5) on the chart.



Comparing two movie critics on a scatter plot

We can also see a straight line on the chart. This is called the best-fit line because it comes as close to all the items on the chart as possible. If the two critics had identical ratings for every movie, this line would be diagonal and would touch every item in the chart, giving a perfect



correlation score of 1. In the case

Two critics with a high correlation score

illustrated, the critics disagree on a few movies, so the correlation score is about 0.4. The above figure shows an example of a much higher correlation, one of about 0.75.

One interesting aspect of using the Pearson score, which we can see in the figure, is that it corrects for grade inflation. In this figure, Jack Matthews tends to give higher scores than Lisa Rose, but the line still fits because they have relatively similar preferences. If one critic is inclined to give higher scores than the other, there can still be perfect correlation if the difference between their scores is consistent. The Euclidean distance score described earlier will say that two critics are dissimilar because one is consistently harsher than the other, even if their tastes are very similar. Depending on your particular application, this behavior may or may not

be what you want. The code for the Pearson correlation score first finds the items rated by both critics. It then 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, shown in the code below. Unlike the distance metric, this formula is not very intuitive, but it does tell you how much the variables change together divided by the product of how much they vary individually. To use this formula, create a new function with the same signature as the sin_ distance function in recommendations.py:

```
 # Returns the Pearson correlation coefficient for x and y def sin_pearson (pref,x,y):
# Get the list of mutually rated items si= { }
for item in pref[x]:
if item in pref[y]: si[item]=1
# Find the number of elements n=len (si)
# if they are no ratings in common, return 0 if n==0: return 0
# Add up all the preferences
sum1=sum ([pref[x][it] for it in si]) sum2=sum([pref[y][it] for it in si])
# Sum up the squares sum1Sq=sum([pow(pref[x][it],2) for it in si])
sum2Sq=sum([pow(pref[y][it],2) for it in si])
# Sum up the products pSum=sum ([pref[x][it]*pref[y][it] for it in si])
# Calculate Pearson score num=pSum-(sum1*sum2/n)
den=sqrt((sum1Sq-pow(sum1,2)/n)*(sum2Sq- pow(sum2,2)/n))
if den==0: return 0 r=num/den
return r
```

This function will return a value between −1 and 1. A value of 1 means that the two people have exactly the same ratings for every item. Unlike with the distance metric, we don't need to change this value to get it to the right scale. Now we can try getting the correlation score:

```
reload(recommendations)
print recommendations.sim_pearson(recommendations.critics,
... 'Lisa Rose','Gene Seymour') 0.396059017191
```

Conclusion

In this paper we have introduced a recommender system for movie recommendation. It allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes and using K-means algorithm. By the nature of our system, it is not an easy task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinions. Based on informal evaluations that we carried out over a small set of users we got a positive response from them. We would like to have a larger data set that will enable more meaningful results using our system. Additionally we would like to incorporate different machine learning and clustering algorithms and study the comparative results.

A hybrid approach is taken between context based filtering and collaborative filtering to implement the system. This approach overcomes drawbacks of each individual algorithm and improves the performance of the system. Techniques like Clustering, Similarity and Classification are used to get better recommendations thus increasing precision and accuracy. In future we can work on hybrid recommender using clustering and similarity for better performance. Our approach can be further extended to other domains to recommend songs, video, venue, news, books, tourism and e-commerce sites, etc.

References

[1]Manoj Kumar, D.KYadav, Ankur Singh, Vijay Kr. Gupta," A Movie Recommender System: MOVREC" International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015.
[2] Luis M. de Campos, Juan M. Fernández-Luna *, Juan F. Huete, Miguel A. Rueda-Morales; "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks", International Journal of Approximate Reasoning, revised 2010.
[3] Harpreet Kaur Virk, Er. Maninder Singh," Analysis and Design of Hybrid Online Movie Recommender System "International Journal of Innovations in Engineering and Technology (IJIET) Volume 5 Issue 2,April 2015.
[4] Utkarsh Gupta1 and Dr Nagamma Patil2," Recommender System Based on Hierarchical Clustering Algorithm Chameleon" 2015 IEEE International Advance Computing Conference (IACC).
[5] Urszula Kuzelewska; "Clustering Algorithms in Hybrid Recommender System on MovieLens Data", Studies in Logic, Grammar and Rhetoric, 2014.
[6] Costin-Gabriel Chiru, Vladimir-Nicolae Dinu , Ctlina Preda, Matei Macri ; "Movie Recommender System Using the User's Psychological Profile" in IEEE International Conference on ICCP, 2015.
[7] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online- product recommendation system: Combining implicit rating- based collaborative filtering and sequential pattern analysis," Electron. Commer. Res. Appl., vol. 11, no. 4, pp. 309–317, Jul. 2012.
[8] R. Burke, "Hybrid Web Recommender Systems," Springer Berlin Heidelberg, pp. 377–408, 2007.
[9] G. Groh and C. Ehmig, "Recommendations in Taste related Domains: Collaborative Filtering vs. Social Filtering," In Proceedings of GROUP '07, pp. 127–136, 2007. ACM.
[10] A. Said, E. W. De Luca, and S. Albayrak, "How Social Relationships Affect User Similarities," In Proceedings of the 2010 Workshop on Social Recommender Systems, pp. 1–4, 2010.
[11] H. Lee, H. Kim, "Improving Collaborative Filtering with Rating Prediction Based on Taste Space," Journal of Korean Institute of Information Scientists and Engineers, Vol.34, No.5, pp.389- 395, 2007.
[12] P. Li, and S. Yamada, "A Movie Recommender System Based on Inductive Learning," IEEE Conf. on Cybernetics and Intelligent System, pp.318-323, 2004.
[13] W. Woerndl and J. Schlichter, "Introducing Context into Recommender Systems," Muenchen, Germany: Technische Universitaet Muenchen, pp. 138-140.
[14] G. Adomavicius and A. Tuzhilin, "Context-aware Recommender Systems," in Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.
[15] T. Bogers, "Movie recommendation using random walks over the contextual graph," in Proc. of the 2nd Workshop on Context-Aware Recommender Systems, 2010.
[16] Tang, T. Y., & McCalla, "A multi-dimensional paper recommender: Experiments and evaluations," IEEE Internet Computing, 13(4),34–41, 2009.
[17] Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, "Item-based collaborative filtering recommendation algorithms," In: Proceedings of the 10th international World Wide Web conference, pp. 285–295, 2001.