MOVIE RECOMMENDATION SYSTEMS

Gaurav Patil, Rutuja Kadam, Neha Gore

SKNSIT, Savitribai Phule Pune University, Pune, Maharashtra, India

ABSTRACT

Now a day's recommendation system has changed the style of searching the things of our interest. This is information filtering approach that is used to predict the preference of that user. The most popular areas where recommender system is applied are books, news, articles, music, videos, movies etc. In this paper we have proposed a movie recommendation system named MOVREC. It is based on collaborative filtering approach that makes use of the information provided by users, analyzes them and then recommends the movies that is best suited to the user at that time. The recommended movie list is sorted according to the given to these movies by previous users and it uses K-means algorithm for this purpose. MOVREC also help users to find the movies of their choices based on the movie experience of other users in efficient and effective manner without wasting much time in useless browsing. This system has been developed in PHP using Dreamweaver 6.0 and Apache Server 2.0. The presented recommender system generates recommendations using various types of knowledge and data about users, the available items, and previous transactions stored in customized databases. The user can then browse the recommendations easily and find a movie of their choice.

Keywords— Recommender System, Machine Learning, Collaborative Filtering, Content based Filtering

With the rapid development of Internet technology[1], today'ssociety has entered the era of Web 2, information overload hasbecome a reality. How to find the required information in themass of data has become a hot research topic. Movie is one ofthe main spiritual entertainment, also has the problem of information overload. In order to solve this problem, this paper put forward a proposal of personalized movierecommendation system[1,2].

Personalized recommendation try to know the characteristics and preferences of the user by collecting and analysing historical behavior to know what kind of person theuser is, what kind of behavior preference the user has, what kind of things the user like to share and so on[3.4.5], and finally understand that user characteristics and preferences based on the rules of the platform and recommend the information and goods which the user interested[6.7]. Personalized recommendation system is a kind of information filtering technology. It is an integrated system which is a combination of a variety of data mining algorithmsand user related information, to meet the interests or potentialinterests of users. The common recommendation system is categorized as content based recommendation system, collaborative filtering recommendation system, and hybrid recommendation system[9,10]. Each recommendation algorithm has different use range

and use condition, it results in the use of different recommendation algorithm for the same information recommendation. In the actual application of recommendation system, the system tends to be a hybrid recommendation system. That is, to mix the advantage of

LITERATURE SURVEY

each recommendation algorithm to the recommended process to effectively improve the recommendation effect.

In this paper, the key research contents is to help users to obtain user-interested movie automatically in the massive movie information data using KNN algorithm and collaborative filtering algorithm, and to develop a prototype of movie recommendation system based on KNN collaborative filtering algorithm

KNN algorithm

KNN algorithm is called K nearest neighbor classification algorithm. The core idea of the KNN algorithm is: if the majority of the k most similar neighbors of sample in the feature space belongs to a certain category, then the sample is considered to belong to this category[8]. As shown in Figure 1, the majority of w's nearest neighbors belong to the x category, w belongs to the X category.

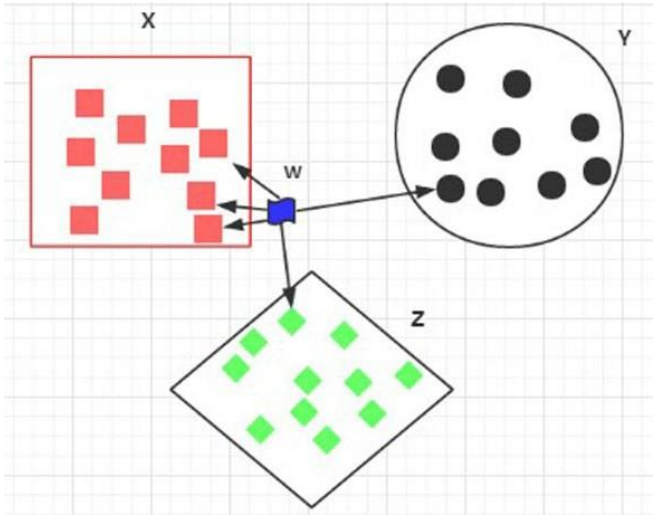


Fig. 1. example of KNN algorithm

A. COLLABORATIVE FILTERING

Collaborative filtering system recommends items based on similarity measures between users and/or items. The system recommends those items that are preferred by similar kind of users. Collaborative filtering has many advantages 1. It is content-independent i.e. it relies on connections only 2. Since in CF people makes explicit ratings so real quality assessment of items are done. 3. It provides serendipitous recommendations because recommendations are base on user's similarity rather than item's similarity.

B. Content-based Filtering

Content-based filtering is based on the 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 user's preferred liked or dislikes. In other words CBF algorithms recommend those items or similar to those items that were liked in the past

It examines previously rated items and recommends best matching item. There are various approaches proposed in various research papers listed below. These approaches are often combined in Hybrid Recommender Systems. An earlier study by Eyjolfsdottir et. al for the recommendation of movies through MOVIEGEN had certain drawbacks such as , it asks a series of questions to users which was time taking . On the other hand it was not user friendly for the fact that it proved to be stressful to a certain extent. Keeping in mind these shortcomings, we have developed MovieREC, a movie recommendation system that recommends movies to users based on the information provided by the users themselves. In the present study, a user is given the option to select his choices from a set of attributes which include actor, director, genre, year and rating etc. We predict the users choices based on the choices of the previous visited history of users. The system has been developed in PHP and currently uses a simple console based interface.

EXISTING METHOD

Collaborative filtering algorithm is categorized as user-based collaborative filtering algorithm[4] and project-based collaborative filtering. The basic principles of the two is quite similar, and this section mainly introduces the user-based collaborative filtering recommendation algorithm. The basic idea of collaborative filtering recommendation algorithm is to introduce the information of similar-interest users to object users[7].]. As shown in figure 2.

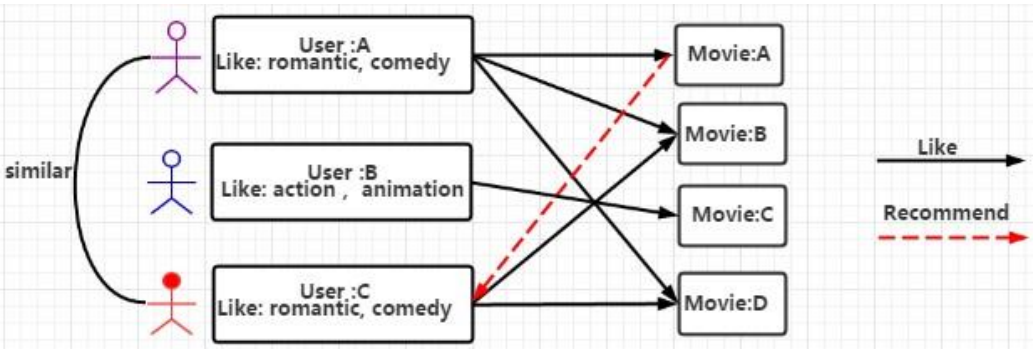
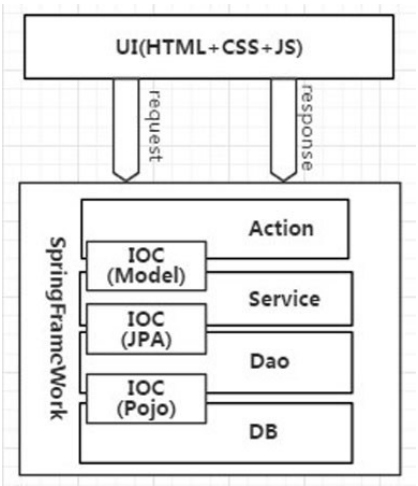


Fig. 2. example of UserCF algorithm

User A loves movie A, B, C, and user C likes movie B, D, so we can conclude that the preferences of user A and user C are very similar. Since user A loves movie D as well, so we can infer that the user A may also love item D, therefore item D would be recommended to the user. The basic idea of the algorithm is based on records of history score of user. Find the neighbor user as u` who has the similar interest with target user u, and then recommend the items which the neighbor user u` loved to target user u, the predict score which target user u may give on the item is obtained by the score calculation of neighbor user u` on the item. The algorithm consists of three basic steps: user similarity calculation, nearest neighbor selection and prediction score calculation.

SYSTEM ARCHITECTURE DESIGN



A. Front view is implemented using HTML, CSS, JAVASCRIPT, the back end uses Struts2, Spring and Hibernate, the database uses MySQL for storage. The system is object-oriented to guarantee system of high cohesion and improve development efficiency using the SSH protocol[17].Besides, it enhances the maintainability and scalability by separating Controller layer and View layer to reduce the degree of coupling between them, making it easier to maintain and modify the WEB application.

B. Data Description

In proposed model we use a pre filter before applying Kmeans algorithm. The attributes used to calculate distance of each point from centroid are Genre, Actor, Director ,Year ,Rating Different attributes have different weights. In our research we have found that the most appropriate recommendations that can be generated should be based on the ratings given to the movies by previous users, therefore we have given more importance to the rating attribute than other attributes. These ratings have been taken from www.imdb.com because perhaps it has the largest collection of movies along with the rating given to these movies by a large number of different users from different parts of the world. Another important parameter in our proposed model is total number of votes received by a particular movie. We have divided number of votes in to three categories that is less than or equal to 1000, more than 1000 but less than or equal to 10,000 and greater than 10,000. International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015 9 Wm= Wr + Wa +Wd +Wg +Wy In our research we have found that as the number of vote's increases the weight of rating should also increase respectively. There fore we have used ratios of 1:1, 1:2, and 1:3 depending on total number of votes received by a movie. we have also found that the movies which have rating less than 5 are the ones which are least suitable for recommendation, and are least desirable by users. Users generally want to see a good movie and higher rating ensures that our predicted movie set are of those movies which are liked by a large number of users. Weights assigned to other attributes are generally based on the average of total movies associated with that particular attribute to the total number of movies in our data set.

C. Simulation of MOVREC

When any user enters our system MOVREC he has a couple of options. He /she can search a particular movie or see upcoming movies list or can go to our recommendation page. On recommendation page he is given the choice to select/input values for different attributes. On the basis of these input values, we search our search our database and prepare an array of suitable movies. Movies included in the array are those whose even one attribute value matches with the input value of the user. We then calculate the number of movies in our array with the help of a counter. If the counter value is less than or equal to twenty we display the movie list sorted according to ratings associated with the movies. If number of movies is greater than twenty then we apply a pre filter and select top twenty movies according to rating.

D. If two movies have same rating then priority is given to the movie having a large number of votes. After filtering the movie list we match the attributes value to their respective weights and compute the total weight of each movie. Once we have calculated the total weight of each movie we apply K-means clustering algorithm on these group of movies. In our research we have also found that generally a user prefer a list with five movies so we assume K equal to be 4 so that an average every K has five movies, where K is the number of cluster to be formed. For each cluster k1, k2 , k3, k4 we assume initial centroid c1, c2, c3, c4 which corresponds to the first, sixth, eleventh, and sixteenth movie in the movie array. After defining the initial centroid we compute the distance of all the other data points from each centroid and assign the remaining data points (movies) to closest centroid and form clusters. The distance measure we have used to calculate the distance between data points and centroid is the Euclidean Distance. After forming initial clusters we take one cluster at a time. We again calculate centroids but this time each centroid corresponds to mean of the points in that cluster. After recalculating centroids we compute the distance of all data points with respect to these newly formed centroids and reassign them to form clusters. We repeat this process till there is no change in centroids. This ensures that the clusters finally formed are optimized and no further grouping is possible. Once final cluster are formed we compute the average rating of all points belonging to that cluster i.e. cluster rating, then according to the input user query we display the cluster having highest cluster rating

Weightage and matching of attributes

1. Actor (Wa) Wa= No. of movies of Actor(a) in data set Total no. of movies in data set D. Proposed Algorithm

1) Input: a number of movies: m

2) Output: a number of clusters: K

a) Step 1 Select n movies from m movies n20 then select top 20 movies from n movies based on ratings. Else display the output movies sorted by rating.

b) Step 3 If rating of movies x, y are equal i.e. If Rx= Ry Then select those movies which have greater number of user votes. Step 4 Assume K=4

.c) Step 5 REPEAT (6, 7)

d) Step 6 Chose initial centroid C1, C2, C3, C4.

e) Step 7 Calculate Euclidean distance of all data points w.r.t. C1, C2, C3, C4 and re-compute the centroid of each cluster.

f) Step 8 UNTILL centroid does not change. Where, m: Total number of movies in database n: Number of movies after user query x, y: Two random movies Rx , Ry : Rating of movies x, y K: Number of cluster C1, C2, C3, C4: Initial Centroid.

FUTURE SCOPE

Cosine similarity calculation do not work well when we don't have enough rating for movie or when user's rating for some movie is exceptionally either high or low.As an improvement on this project some other methods such as adjusted cosine similarity can be used to compute similarity.

Adjusted cosine similarity, which is similar to cosine similarity, is measured by normalizing the user vectors Ux and Uy and computing the cosine of the angle between them. However, unlike cosine similarity, when computing the dot product of the two user vectors, adjusted cosine similarity uses the deviation between each of the user's item ratings, denoted Ru, and their average item rating, denoted ¯Ru, in place of the user's raw item rating.

CONCLUSION

In this paper we have introduced MovieREC, 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. Eventually we would like to implement a web based user interface that has a user database, and has the learning model tailored to each user

ACKNOWLEDGMENT

We would like to express our sincere gratitude towards our guide Prof. G.M.Kadam for his in-valuable guidance and supervision that helped us in our research. He has always encouraged us to explore new concepts and pursue newer researchproblems. We credit our project contribution to him. Collectively, we would also like to thank our principal Prof. M. S. Rohokale for their time, suggestions, and for always making themselves available. We cannot thank them enough.

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] Han J., Kamber M., “Data Mining: Concepts and Techniques”, Morgan Kaufmann (Elsevier), 2006.

[2] Ricci and F. Del Missier, “Supporting Travel Decision making Through Personalized Recommendation,” Design Personalized User Experience for e-commerce, pp. 221-251, 2004.

[3] Steinbach M., P Tan, Kumar V., “Introduction to Data Mining.” Pearson, 2007.

[4] Jha N K, Kumar M, Kumar A, Gupta V K “Customer classification in retail marketing by data mining” International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014 ISSN 2229- 5518

[6] Farheen Naaz, Farheen Siddiqui, modified n-gram based model for identifying and filtering near-duplicate documents detection,International Journal of Advanced Computational Engineering and Networking, ISSN: 2320- 2106, Volume-5, Issue-10, Oct.-2017

[7] N-gram Accuracy Analysis in the Method of Chatbot Response, International Journal of Engineering & Technology. (2018)

[8] Shukla, V.K, Verma, A, "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language", 2019 International Conference on Automation, Computational and Technology Management (ICACTM)