International Journal of Electrical, Electronics and Computer Engineering **7**(2): 106-109(2018)

ISSN No. (Online): 2277-2626

Content-Based Filtering in Movie Recommendation

Arun Kumar Sharma

[1]*Department of Computer Science & Engineering, NIT Hamirpur, India.*

(Corresponding author: Arun Kumar Sharma)

(Received 24 September, 2018 Accepted 03 December, 2018)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: The information and items/data that needs to be found on very large web sites may be difficult and sometimes even time-consuming. The suggestions can be personalised by using the Recommender systems. In this paper the movie recommender system is defined which basically make use of techniques involving content-based filtering so that personalised recommendations could be obtained. The implication of such domain suggests that the user model needs to be very dynamic and learned only from positive feedback.

Keywords: Recommender, Filtering, Encryption, Content, User, Field, Cost, Optimized.

I. INTRODUCTION

In today's era World Wide Web has grown and is further growing at rate that is exponential, the complexity and size of many web sites has grown with it. For any user that pertains to the website it becomes very difficult and time consuming to search the information that is being looked for. To help users find information relevant to their interests the web site can be customized Recommendation programs can improve a website for individual users by vigorously adding hyperlinks. The main aim of these dynamic hyper links is to enable user to find interesting items i.e the items which they like and similar to the items they are searching for and further improving the communication between the system and the user.

When users browse the website they often search for those items which they find interesting. Interested items can have many items. For example, text and data details can be viewed as objects of interest or a reference to a particular topic may be something the user wants. Another example, which applies to a web retailer, is to view purchased products as items of interest. Whatever content it contains, a website can be viewed as a collection of these interest items which can be acquired easily by the help of Recommender Systems [4, 7].

II. RELATED WORK

A recommender system is a special type of data filter. Data sorting works by the delivery of selected items in a large collection that the user may find interesting or useful and may be seen as a sorting function [2]. Depending on the training details the user model is developed which enables the filter system to distinguish intangible objects into positive c (user-friendly) or negative-c (non-user-active) category. The training set contains features that the user will find interesting. These factors create all the necessary training conditions. This attribute specifies the category of the item according to user rating or explicit proof. In principle, an object is defined as vector $X=(x_1, x_2,\ldots, x_n)$ of n objects. Items can have binary, fiction or numerical symbols and are found in the content of the material or in information about user preferences. The learning method function to select a task based on a set of input vegetable training m can separate any object in the collection. Activity h (X) will be able to distinguish something that does not appear to be right or wrong at the same time by multiplying the binary value or multiplying the numerical value. In that case the limit can be used to determine if the item is important or not applicable to the user.

A Japanese video service provider's recommendation program has been proposed that uses the actors and keywords information of user films [1, 8]. They also look at the time of day when users watch TV. They use the average number of times a user watches a movie with a particular feature (such as an actor, keyword) in the number of times that feature is seen in all movies. This rating is evaluated for every character and keyword features. Then count each movie, counting the movie rating features. They use recall, clarification and F-Measure as test measures. It is different from our test method, they also measure performance of the system based on the feedback provide by the user after the recommendations.

FIT system recommend tv programs. They ask for the genre from each household. They also store time of the day they turn on television. Thus, maintaining each household profile. In the recommendations section, the FIT system first guesses which house turned on the TV using the date information period.

III. CONTENT BASED FILTERING IN MOVIE RECOMMENDATION

In case of movie recommendation the major problem that arises in collaborative filtering is the lack of user-item matrix [3, 5]. The lack of good quantum of ratings by the users to fairly large range of movies exists in the system. Since, scarcity of the movies which are rated commonly by two users, it becomes a great deal to find users who are in a location nearby. Moreover, the recommendation of movie cannot occur if no recommendation is done by any user (cold-start problem).

But in movie recommendation using content-based filtering involves moving information and viewing profile of the user. In a content-based approach each user is unique and the interests of the user are not the same as the other user's as in the collaborative methods [6, 9]. In this method the features and their values related to the movie items are already given. For ex. Let there are only two features and these are comedy and romance. A movie can be a comedy movie or a romantic movie or the combination of both the type of movie. Depending on how much a movie is comedy or romantic or both, the feature value for that movie is being provided. Take an ex. Of '500 days of summer', this movie is romantic and little bit comedy movie, it's romance feature value will be 0.7 and the value of comedy feature will be 0.3 out of 1.

The user provides another term which is called weight parameter. Depending on the user, that which kind of movie he liked the weight value is find for that particular user. These weight parameters are different for different users and their taste.

Above Table 1, indicates the relationship between the movies and the user. The '?' field represents that user in that column has not assigned any movie rating in that row. But, the rating ranges between 0-5A rating of 0 represents that user did not like that movie and a rating of 0 represents that user has disliked that movie. Unique ids are used to represent each movie and each user.
Above matrix in Table 2, shows the relationship between the movies and the users, known as binary matrix, where the field marked with 0 denotes that movie in the row has not been yet rated by user in that column.

Table 1.

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

Table 2.

1	1	1	1
1	0	0	0
0	1	1	0
1	1	1	1
1	1	1	0

Table 3.

Movie	X$_1$ (romance)	X$_2$ (action)
Love at last	0.9	0
Romance forever	1.0	0.01
Cute puppies of love	0.99	0
Nonstop car chases	0.1	1.0
Swords vs. karate	0	0.9

The above Table 3 shows the feature table which is representing the relation between the movies and their features. Take ex of movie 'Love at last' feature value x1(romance) for this movie is 0.9 while feature value x2(action) is 0 which shows that this movie is romantic not action movie while movie' Swords vs. Karate' is totally action movie.

IV. METHODOLOGY

1. Finding Parameters theta

Theta mainly denotes the weight parameters vector which enables us to determine as to which movie is liked by which user For eg. theta value for user1(Alice) is [0 5 0] , it represents that Alice like romantic movies so much , but she does not like action movies. Theta^{1}denotes weight parameter vector for user1(Alice), which is further represented by [theta$_0^1$, theta$_1^1$, theta$_2^1$]. Here theta$_0^1$determines weight adjustment value.

Initially if we can give a movie max 5 rating then the Theta1 will be initialised to [5, 5, 5].
Now we have to find the optimal val of theta vector for each user so that cost can be minimised. Here cost is another term which is used to minimise the difference between actual movie rating given by particular user and the movie predicted by this model. For this we use linear regression. Using the linear regression, for each user j, learn a parameter theta$^{(j)}$. Predict user j as rating movie i with (Theta$^{(j)}$)^{T}X$^{(i)}$ stars.
In short it can be written as.
Theta$^{(j)}$ = Parameter vector for user j.(This is basically column vector)
X$^{(i)}$ = Feature vector for movie i.
For user j and movie i predicted movie is (Theta$^{(j)}$)^{T}X$^{(i)}$
m(j) = No. Of movies rated by user j to learn Theta$^{(j)}$
To learn $\boldsymbol{\theta}^{(j)}$ (parameter for user j):

$$\min_{\theta^{(j)}} \tfrac{1}{2}\textstyle\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \tfrac{\lambda}{2}\textstyle\sum_{k=1}^{n}(\theta_k^{(j)})^2 \qquad (1)$$

To learn $\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(2)},\ldots.,\boldsymbol{\theta}^{(n_u)}$:

$$\min_{\theta^{(1)},\ldots\ldots,\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2 \qquad (2)$$

Here, lambda is regularization term and r(i,j)=1 represents that use the value of those positions in movie-user to a movie is not equal to '?'. First equation is the cost for user j and the second equation is the cost of overall users. Using above algo or say equations we can optimize the value of theta for each user j that will minimize the cost functions.

2. Predicted Ratings

$$\begin{bmatrix} ((\theta^{(1)})^T(x^{(1)}) & ((\theta^{(2)})^T(x^{(1)}) & . & . & ((\theta^{(n_u)})^T(x^{(1)}) \\ ((\theta^{(1)})^T(x^{(2)}) & ((\theta^{(2)})^T(x^{(2)}) & . & . & ((\theta^{(n_u)})^T(x^{(2)}) \\ . & . & . & . & . \\ . & . & . & . & . \\ ((\theta^{(1)})^T(x^{(n_m)}) & ((\theta^{(2)})^T(x^{(n_m)}) & . & . & ((\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

Here n_m = no. of movies = 5, n_u = no. of users = 4

From predicted rating we can calculate the unknown ratings for a movie for any user. If there are m movies we can find the k movies with high rating for recommendation for that particular user.

3. Finding Accuracy

The above model predict the rating for the unknown rating movie for particular user. We can see that this model again predict the rating for known rating which has already given by particular user. According to the different model used different algorithm predict diff rating for known as well as unknown rating. But how will we calculate that the particular algo predict the accurate rating. Two terms are used to calculate the accuracy of the model for used algorithms. These are MEAN ERROR and ROOT MEAN SQUARE ERROR. ME is the defined as the ratio of the total difference between the specified rating and the actual rating for each movie and the actual rating for a specific user. Similarly, RMS error is square root of the division of sum of the square of difference between predicted rating and actual rating for each movie and sum of actual rating for a user.

4. Performance

In order to measure the effectiveness of recommendation methods we have used precision, recall and F- Measure metrics. We made use of topN hit counts to evaluate the exactness of our recommendation system.

The recommendation of movies is done in accordance with the predicted ratings which are generated. All the best movies are sorted keeping in mind the ratings which were generated and the recommendation of top N=10 movies is done. Afterwards the counting process pertaining to the number of movies which are watched in the test set by the user out of top 10 recommendations is done and this quantity is named as the #hitCounts.

Precision is the measure of the system hitshow many movies in the top 10 movies:

$$Precision = \frac{\#hitCounts}{N}$$

Recall is defined as the ratio of the number of hits calculated out of top 10 recommendations and the size of test u which denotes number of movies that is watched by the user u in the test set:

$$Recall = \frac{hitCounts}{|I_u^{test}|}$$

For example, if the recommendation system hits 5 movies out of the 10 recommended movies which were watched, the precision value will be 0.5. If 30 movies corresponding to the test set are watched by the user, then the recall value will be 0.16. The usage of both precision and recall is done by the F-measure as follows:

$$F - measure = 2.\frac{Precision.Recall}{(Precision + Recall)}$$

5. Strengths and Weaknesses

Large amount of user data is not required by Content-based recommender systems. Only data pertaining to item is needed and the process of giving recommendations to user scan start. User data is not the sole criteria which lays the basis of recommendation, so, recommendations can be given to even your first customer as long as you have adequate data is available to build his user profile.

A well distribution of item data needs to be done. It won't be effective to have a content-based recommender if 80% of the movies involved are action movies. Also, the recommendations that you will get are likely to be direct substitutes, and not complements, of the item the user made use to interact with. Complements are more likely to be discovered through collaborative techniques.

V. CONCLUSION

In this paper the content based movie recommendation system has been described where genre is used as its feature. It uses the weight parameters and features value for movie rating prediction by adjusting the weight parameters and using these weight parameters minimizing the cost. The predicted rating is the rating of the unknown rated movies for a user.

REFERENCES

[1]. Aggarwal, C. C. (2016). Recommender systems (Vol. 1). Cham: Springer International Publishing.

[2]. Dunn, G., Wiersema, J., Ham, J., and Aroyo, L. (2009). Evaluating Interface Variants on Personality Acquisition for
Recommender Systems. In: Houben, G.J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) User Modeling, Adaptation,
and Personalization. LNCS, vol. **5535,** pp. 259-270. Springer, Heidelberg.

[3]. Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018, April). A hybrid approach using collaborative filtering and content based filtering for recommender system. In Journal of Physics: Conference Series (Vol. 1000, No. 1, p. 012101). IOP Publishing.

[4]. Ghauth, K. I., & Abdullah, N. A. (2010). Learning materials recommendation using good learners' ratings and content-based filtering. *Educational technology research and development,* **58**(6), 711-727.

[5]. Li, H., Cai, F., & Liao, Z. (2012, August). Content-based filtering recommendation algorithm using HMM. In *2012 Fourth International Conference on Computational and Information Sciences* (pp. 275-277). IEEE.

[6]. Pal, A., Parhi, P., & Aggarwal, M. (2017, August). An improved content based collaborative filtering algorithm for movie recommendations. In *2017 tenth international conference on contemporary computing* (IC3) (pp. 1-3). IEEE.

[7]. Resnick, P. and Varian, H.R. (1997). Recommender Systems. Communication. ACM 40, 56-58.

[8]. Soares, M., & Viana, P. (2015). Tuning metadata for better movie content-based recommendation systems. Multimedia Tools and Applications, **74**(17), 7015-7036.

[9]. Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, **89**, 404-412.