Movie Recommender System

Arezoo Rajabi
Oregon State University
rajabia@ond.orst.edu

Abstract

Today, recommender systems are one of the main part of e-commerce business. These systems offer right items/products to right costumers. These systems try to find similar items to ones that the target costumer was interested before based on the given rates to them. While, because of high number of items and users, the search space is huge. In addition, the similar rates for two items can not guarantee that these items are similar in real worlds. In this paper, we propose a method to decrease search space. Moreover, we show that we can not forecast the users' interest for items that were rated by few users.

1 Introduction

Due to spread of the Internet, lots of business and trade opportunities appear so that internet has become an inseparable part of human's life. The most popular among these businesses is E-commerce. E- commerce is type of online trading in which services or products are provided over internet. By growing internet usage, many challenges raised due to the high volume of information. For example, the people who are looking for a certain product or services, encounter with a high volume of information. Also, there are lots of items which users do not have any information about their content. As a result, finding the best options is really difficult for them, even in some cases it is impossible. To solve these issues, recommender systems have been introduced to purge information that is accordance with the users' personality and interests.
For instance Amazon [1] and Movilens [2] are of those commercial systems which employ recommender systems for suggesting items to users. Also social networks like Facebook [3] employs recommender systems to suggest friends.
Recommender systems can estimate the target users' interest in items which have not seen yet, by analyzing the users' rates for other items. There are three traditional category of Recommender systems to estimate the rate which include content-based, collaborative filtering and hybrid recommendation [1].

1. Content-based: in this method the recommended items are those which are similar to the users' previously interested items. Usually textual information is employed for recommendations [2].
2. Collaborative filtering: in this method two major trends are employed including user-based and item-based method[3].
   (a) User-based :In this method users who are similar to the target user are identified according to similarities in their given rates to the same items, and then by the rates that similar users have given, the rates of other items that have not ranked by target users would be estimated.

[1] https://amazon.com
[2] https://movielens.org/
[3] https://facebook.com

(b) Item-based: In this method the similarity of items are calculated according to the users' rates. To estimate the rate of those items which have not ranked by target user, rates of similar items is considered .

3. Hybrid method: each of the above mentioned methods have some deficiencies. The hybrid method by combining these methods tries to overcome their limitations [1].

In all the methods we mentioned here, the forecasting operation is done according to the users' and items' information or the given rates to items by users. The space of recommendation system is:

$$R : User \times Item \rightarrow Rating \tag{1}$$

In which User is the set of users, and Item is the set of items. Also Rating is the estimated rates for items that users did not rate them. To present the rates of users to items, these systems use user-item matrix(M) in which the element of M_{ij} shows the rate of user i gave to item j.
Recently, another category of recommender systems have been proposed which are known as context-aware recommender systems. Methods of this category in addition to the users' and items' information use another kind of information called context information to offer personalized recommendation [5].

$$R : User \times Item \times Context \rightarrow Rating \tag{2}$$

The context information identify the user's situation. Information such as time and location of the users, their mood while interacting with the system, people around them, and etc. which can be used as context information.
Since the interests and ideals of the users in different situations can be widely varied, systems which use the context information can offer more precise and efficient recommendations compared to those systems which do not use these kinds of information.
As we mentioned, the main idea in these systems is based on finding similar users or items. While, there are a lot of users and items which makes systems to search in huge domain. Moreover, the users only have bought a few numbers of items and ranked them. So, finding similar users is hard. In this regard, in this project we try to induce search space and having a global view instead of local view.
Our contributions in this paper are briefly listed here:

1. Decreasing searching space by categorizing items
2. Having global view instead of local view
3. Finding relation between different categories

The rest of this paper organized as follows. In section 2, we will review traditional method of collaborative filtering, a well-know and popular approach that used in most recommender systems. In section 3 we suggest our method to decrease the search space and in section 4 we introduce the evaluation criteria and evaluate the proposed method. At the end, in section 5 the conclusion will be provided.

2 Collaborative Filtering

As mentioned before, Collaborative filtering (CF) method is the first and common method that is used in many recommendation systems. This method is based on measuring similarity between users or items. In other words, there are two approach for computing similarity:

1. User-User CF: This method is the first generation of CF methods. This method try to find similar users to a target user based on the past ratings of users. After finding the similar users to target users, this method tries to predict the unranked items based on ranks that the similar users have given to these items. A critical part in these methods is computing similarity between users. There are two well-known criteria for computing similarity:
   (a) Pearson Correlation: This measure computes statistical correlation between users' ratings and it is computed as below:

$$S_{(u,v)} = \frac{\sum_{i \in I_u \cup I_v} (r_{u,i} - Avg(r_u))(r_{v,i} - Avg(r_v))}{\sqrt{\sum_{i \in I_u} (r_{u,i} - Avg(r_u))^2} \sqrt{\sum_{i \in I_v} (r_{v,i} - Avg(r_v))^2}} \tag{3}$$

Where $r_{(u,i)}$ is the rate that user u has given to item i and r_u is the sum of rates of user u. This criterion gives high similarity score to users that have rated only some few items.

(b) Cosine similarity: This criterion is defined based on linear algebra approach and assign each user an I-dimension vector. The ith element of this vector shows the rate of the user to item i. This criterion is computed as below:

$$S(u,v) = \frac{r_u.r_v}{||r_u||_2.||r_v||_2} \tag{4}$$

The main the defect of these criteria is that they consider zero value for unranked items.

2. Item- Item: The user-user CF suffers from searching in high space of users which is growing continuously. While, the scalability in recommender systems is necessary. So, item-item CF has been proposed. In this approach, the similar items to items that the target user was interested in would be proposed. Similar items are defined based on the similar rates that users have given to them. In other words, two items are similar if the users gave same rates to both them. There are two main approach to compute the similarity between items:

(a) Cosine similarity: This criterion is the most popular measure for computing similarity and it's the as same as the criterion defined in user-user CF approach. This criterion assign a vector to each item in which the ith of it's elements shows the rate of user i has given to this Item[2].

$$S(i,j) = \frac{r_i.r_j}{||r_i||_2.||r_j||_2} \tag{5}$$

(b) Conditional probability: This criterion is based on conditional probability and computes the probability of buying the item i if the user has bought the the item j before. the formula for computing this criterion is:

$$S(i,j) = \frac{Freq(i,j)}{Freq(i)(Freq(j))^{\alpha}} \tag{6}$$

In which α serves as dumping factor for items that are so popular between users. These items usually have a high probability of being purchased. So α reduce probability of selecting only most popular items as candidate[5].

As you can see, in user-user CF approach, the methods should search whole dataset for finding similar users. To solve this problem of the item-item CF approach has been proposed. However, the number of items is growing too. Moreover, using users' rates to compute similarity of items is not good idea. Because, the items can be totally different and have different application while users gave them same rates. Moreover, by increasing the number of items, the matrix of user-item that shows the rates of the users to items, become so sparse and most its elements have the value of zero that is meaningless and fake. So, to decrease the sparsity and searching space we decide to find similarity and relation between items' categories instead of looking for similar items. In next section we explain our method and goals in details.

3 Proposed Method

As aforementioned, recommender systems are based on finding similar users or items and they suffer from sparsity. The number of users and items are growing continuously, while the number items that each user has purchased is few compared to total number of items. To solve this problem, we combine the ratings of items in same category and try to find the relation between them. In other words, our goal is decreasing the sparsity by having global view instead of local one. In this sake, we investigate Movielens recommender systems which offer movies to users. In this system, each user ranks the movies that he has watched. Each rank is in the range of $[1-5]$. In this dataset[4], the users allocated rank of 1 or 2 to a movie when they have not been satisfied by the Movie and they gave the rank of 4 or 5 when they have liked that. The rank 3 means that they do not like it so much.

[4]downloaded from http://grouplens.org/datasets/movielens/

Because, there are many movies and each users only watched the few ones, sparsity is main problem in this dataset too.
In addition to the users' ranks, we have the gender, age and occupation of the users. In this data set, the gender is denoted by a "M" for male and "F" for female, and age is chosen from the following ranges:

- 1: "Under 18"
- 18: "18-24"
- 25: "25-34"
- 35: "35-44"
- 45: "45-49"
- 50: "50-55"
- 56: "56+"

Furthermore, the occupation of the users are available and their occupation categorized in 20 categories [5].
Moreover,the movilens recommender system assigns one or more Genre label[6] to each movie which shows the type of that movie.
As discussed, the recommendation systems use matrix of user-item as input. Because of sparsity of dataset, most of the elements of this matrix is zero. In this paper, we tried to decrease the number of zero in our data set. In this regard, we used the average of rates that each user has given to each genre instead of rates of users to each movie and constructed a different type of matrix which is allowed to have non-numeric values. As shown in table 1, our input is a type of low-dimension matrix. And, we are looking for a relation between genres' average rate.
Moreover, we used another type of input that it have higher dimension in cost of adding zero rates.

Table 1: The first proposed type of data which we used in our simulation

UserID	Gender	Occupation	Age	GenreName	Avg(Genre)
1238	M	Student	18	Action	4.3
4235	F	Lawyer	25	drama	2.6

The new dataset have the same information of the previous one have. The only difference of this new dataset with previous one is that we assign a vector to each Genre in which the element i shows the average rates of user i that has been given to that Genre. To find the relation between different

Table 2: The second proposed type of data which we used in our simulation

UserID	Gender	Occupation	Age	Genre1	...	Genre n
1238	M	Student	18	0	...	4.3
4235	F	Lawyer	25	2.1	...	2.6

Genres we used different well-known machine learning methods. In this sake, we used four different algorithms which are fast in building model and can work with non-numeric data:

- M5P decision Tree:
- REPtree
- Random Forest Tree
- Decision Table

[5]{ other or not specified, academic/educator, artist, clerical/admin, college/grad student,customer service, doctor/health care, executive/managerial, farmer, homemaker, K-12 student, lawyer, programmer, retired, sales/marketing, scientist, self-employed, technician/engineer, tradesman/craftsman, unemployed, writer}

[6]{ Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western }

In next section, we apply the mentioned algorithms on our new types of data set and looked for a similarity of between different categories. In other words, we want to examine whether there is correlation between different categories of different items or not. If there is, we can restrict our search space into categories that are more similar to the category of target items.

4 Evaluation metrics and Results

To evaluate the result we used four criteria:

1. Mean absolute error: this criterion is used to compute how much our prediction is close to real values[6].
$$MAE = \frac{1}{n}\sum_{i=1}|y' - y| \tag{7}$$
In which y is real value and y' is predicted value for y.
2. Root mean absolute error: This measure is like MAE but it assumes errors follow normal distribution. This measure provide a good picture of error distribution
$$RMAE = \sqrt{\frac{1}{n}\sum_{i=1}(y' - y)^2} \tag{8}$$
3. Relative squared error: This metric measures the difference of using a simple predictor (such as using average of values for prediction) and the used method. In this regard, the relative squared error normalizes the total squared error by dividing it to the total squared error of the simple predictor[7].
$$RSE = \frac{\sum_j (P_{ij} - T_j)^2}{\sum_{i=j}(T_j - \bar{T})^2} \tag{9}$$
Where, P_{ij} is the predicted value of individual program i for sample case j and T_j is the real value for this case.
4. Relative absolute error: This measure is similar to the RSE but in this case the error is computes by total absolute errors while in RSE the error is computed by total squared errors[7].
$$RAE = \frac{\sum_j |P_{ij} - T_j|}{\sum_{i=j} |T_j - \bar{T}|} \tag{10}$$

As discussed, we use two different type of datasets. To apply algorithm we used weka package. And in each simulation we separate data into parts, 66% of data for training and the rest for test. For rest of this section, we illustrate each dataset and the main results.

4.1 Fist Approach for Organizing Data

To solve the sparsity of dataset, we decided to use different type of dataset. As we discussed, since, each user only rates few movies and most of elements of user-item matrix are zero, we used the average of rates per user for each genre instead of movie rates. To remove zero values, we used a different type of input in which having non-numeric values is allowed. Table 1 indicate a simple schema of this dataset. But this new dataset includes a few number of features. As shown in table

Table 3: The simulation result for first type of dataset

Algorithm	Correlation Coefficient	MAE	RMSE	RAE(%)	RRSE (%)
M5P	0.2336	0.5209	0.6886	96.3103	97.2419
REPTree	0.18	0.5331	0.7043	98.5676	99.4541
Random Forest Tree	0.1082	0.6144	0.806	113.599	113.828
Decision Table	0.2347	0.5207	0.6885	0.806	97.2232

3, the errors of all algorithms for this dataset are high. The main reason of that is low correlation between target feature (average of Genres) and other features. So, we organized our data as shown in table 2. The obtained result for new type of dataset is demonstrate in next section.

4.2 Second Approach for Organizing Data

In last section, we observed that because of low dimension of data, the errors increased. So, we organized the average of each genre in separate column. By new type of dataset, the number of features for each sample increases. However, because users have not watched all genres, the zero valued appeared in new data set. To find out which algorithm works better on new datasets, we tested all algorithms on this dataset by choosing different goal feature . The all observations were same and M5P works better than the other algorithms. In table 4, the results for chosen action genre as a target feature is shown. In table 5 the results of applying M5P on dataset by choosing different

Table 4: The results of applying different Algorithms on new dataset by choosing action genre as target feature

Algorithm	Correlation Coefficient	MAE	RMSE	RAE(%)	RRSE (%)
M5P	0.7287	0.2455	0.4384	52.9649	68.5294
Random Forest Tree	0.721	0.2544	0.4434	54.8865	69.3056
REPTree	0.6944	0.2759	0.4612	59.5364	72.0889
Decision Table	0.6623	0.2847	0.4799	61.4267	75.0133

target features are shown. As you can see, for some features, the obtained errors are so small such as Drama but for the other are so big such as Documentary. It's because the correlation coefficient

Table 5: The M5P algorithm's results

Target Feature	Correlation Coefficient	MAE	RMSE	RAE(%)	RRSE (%)
Action	0.7287	0.2455	0.4384	52.9649	68.5294
Documentary	0.818	0.6394	1.1079	35.5224	57.4763
Crime	0.5876	0.5333	0.9067	71.1203	80.9141
Comedy	0.671	0.2598	0.3791	66.9328	74.3845
Children	0.666	0.6663	1.0289	64.5464	74.6084
animation	0.6138	0.8923	1.2919	66.7254	78.9787
Adventure	0.6295	0.3421	0.6113	62.4949	78.2095
Drama	0.9932	0.0155	0.0926	2.5505	11.6543
Romance	0.5286	0.3351	0.5677	72.4879	85.219
Sci Fi	0.6394	0.342	0.6152	61.9196	76.8941

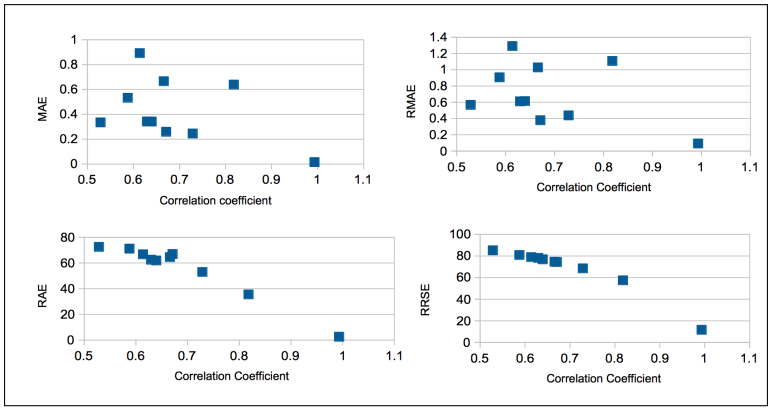

Figure 1: Relation of correlation coefficient and errors

of Documentary is small which indicates the rates of this genre are independent. As shown in figure 1, the correlation coefficient and errors have reverse relation. Moreover, as shown in figure 2 the number of nonzero average rates is few. In other words, this genre has not been watched by many

users. While, Drama movies is so popular between users and most of users at least watch a drama movie. So, predicting the rate for this genre based on the other genres is harder.

Moreover, finding amount of a user's interest in a documentary movie based on other movies in

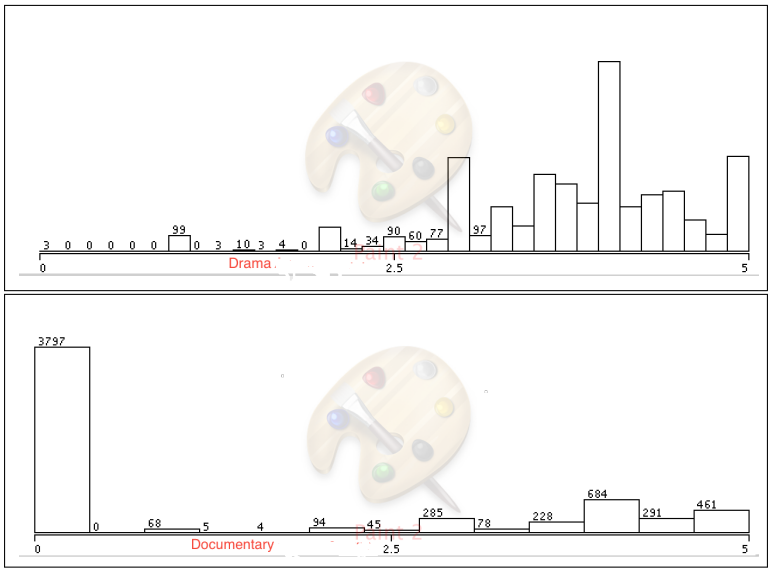

Figure 2: Distribution of rates for Drama and Documentary Genres

different categories is not good idea. Also, for the new released movies that have not been rated by many users, probably their rates vectors are similar to low ranked movies in documentary. While these movies could be a popular in future and they are not in documentary category. In addition, as we observed the rate of many categories have relation with few categories. As a result, finding relation between categories can help us to find which movies are good candidates for being similar items to target movie.

5 Conclusion

By invent of internet, e-shopping has become one of main part of human's life so that many people prefer buying their needs online. However, by growing the number of internet's users and products, the amount of knowledge incaseses too. And it makes finding the right products for customer to be hard. To solve this problem, recommender systems are proposed. These systems try to estimate amount of users' interest in each product and offer the right product to right costumer .

These systems try to find the similar items to items that the target user was interested in. In this regard, these systems find the items that their given rates to, are similar to target items(items that were interesting for the target user) and they search whole item space to find that. This space is so big and moreover items with similar rates maybe are not similar in real world. In this paper, we used average of users' rates that have been given to different categories of items and found out for some categories we can predict the amount users' interest in them. Furthermore, the results show that the average ratings of some category have a low correlation coefficient which means the ratings of these category are independent to other categories' rating. Therefore, predicting the the rate of movies in this category based on other categories is not good idea.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Adomavicius, G., & Tuzhilin, A. (2005). *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.*Knowledge and Data Engineering, IEEE Transactions on,17(6), 734-749.

[2] Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). *A literature review and classification of recommender systems research*.Expert Systems with Applications,39(11), 10059-10072.

[3] Lu, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). *Recommender systems.* Physics Reports,519(1), 1-49.

[4] Adomavicius, G., & Tuzhilin, A. (2011). *Context-aware recommender systems.* InRecommender systems handbook(pp. 217-253). Springer US.

[5] Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). *Collaborative filtering recommender systems.* Foundations and Trends in Human-Computer Interaction, 4(2), 81-173.

[6] http://en.wikipedia.org/

[7]Chai, T., & Draxler, R. R. (2014). *Root mean square error (RMSE) or mean absolute error (MAE)?Arguments against avoiding RMSE in the literature.*