Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 1, January 2017, pg.158 – 162

Using Data Mining to Determine User-Specific Movie Ratings

Harsh Mehta[1], Darshan Doshi[2]

[1]Department of Information Technology, NMIMS University, Mumbai, India
[2]Department of Information Technology, NMIMS University, Mumbai, India
[1] harshsmehta1@gmail.com, [2] darshandoshi45@gmail.com

Abstract — With the rise of various streaming services like Netflix and Amazon Prime, and the rise of movie collections offered by a single provider, the need for determining user-specific movie ratings increases. It is highly crucial for a company to know, and recommend the type of movies liked by users to increase customer retention and improve user experience. In this paper, we are going to use data mining techniques to analyse user preferences and determine user-specific movie ratings through the help of data mining techniques. We will use a movie database from IMDB and determine user specific ratings for each of them. The analysis of attributes of these movies will help us identify the decisive factors and identify user preferences accurately.

Keywords— Data Mining, Decisive Factors, User Experience, Recommend, User Preference

I. INTRODUCTION

Today, millions of movies are streamed online every-day. As the content providers are glutted with new movies and TV shows on a daily basis, it becomes arduous to explore movies that a user likes. This becomes an issue with companies that provide a large collection of movies and shows.

User preferences can vary immensely. One user may like horror movies, while another user may prefer comedy. It is almost impossible to identify a user's preference. [1] However, with the help of data mining we can find out a user's preference. Companies like Netflix and Amazon study a customer's profile and analyse the feedback that the customer provides, to recommend movies and other items to them. These systems are known as recommendation systems. The goal of recommendation systems is to suggest items to a particular user. [2]

We are using an IMDB movie database that we obtained from kaggle. Here, we will collect and analyse various attributes that contribute to the user-specific movie ratings. We will identify different attributes that can help us determine the user preferences and compare this finding with the existing attributes in the movie database to determine movie-ratings. Furthermore, we will use Apriori to identify frequent patterns and determine user similarity to further help us rate movies effectively.

II. APPROACH

The approach to determine user-specific movie ratings can be divided into 3 steps:

1. Identifying User Preferences and Similarity.
2. Normalization and Factorization
3. Collaborative Filtering

A. Identifying User Preferences and Similarity

To identify user preferences and similarity, our first step is to apply the Apriori algorithm to a (User x Item) matrix to generate association rules. Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It is an algorithm for efficient association rule discovery. We use the Apriori algorithm to generate a set of association rules. [3]

To implement our proposed procedure, we need to first obtain association rules via Apriori algorithm. We receive transaction file, minimum support and minimum confidence as an input from the algorithm. The transaction file is the rating matrix in our context as shown in the Table 1.

TABLE I
THE RATINGS MATRIX

$User/Item$	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$	.....	$item_n$
$user_1$	1	1	0	0	1	.....	1
$user_2$	0	1	0	1	1	.....	1
$user_3$	0	0	1	0	0	.....	0
.....	.....	.....	.....	.....	.....	.....	.....
$user_m$	1	0	0	0	1	.....	0

In the above table, 0 means the $user_m$ doesn't like $item_n$. 1 means the $user_m$ likes $item_n$. After running the Apriori algorithm, and based on the minimum support and minimum confidence, a list of strong association rules is obtained. For example, a list of association rules is shown in the Fig. 1.

With the help of strong association rules, we can identify similar users and successfully rate the movies similar users have watched and liked higher than the one's they haven't. [4]

$$item_1 \rightarrow item_3$$

$$item_1, item_4 \rightarrow item_5$$

$$item_2 \rightarrow item_3$$

Fig. 1. Example of Association Rules

We determine a user's movie preferences based on the ratings provided as feedback through various mediums. In our case, we will be using a sample data set that will include various attributes that will help us identify a user's preference. To determine a user's preference, we individually consider various attributes as shown in Table 2.

TABLE II
USER PREFERENCE ATTRIBUTES

Attributes	**Description**
Rating	The ratings given by the user to the movie.
Actor/Actress	The user may have a preference for certain actors and actresses, and thus it is a crucial factor to consider.
Genre	Genre determines the style or category of the movie. Whether the movie is based on drama, crime, action, love or a combination of various styles.
Running time	The length or the duration of the movie.
Director	The director who has produced the movie plays a significant role in determining the like-ability of the movie.

Social Media likes	Social media drives the opinion of the common man. Thus, the number of likes on a particular movie's fan page may reflect its likability. Furthermore, we can access the user's likes to identify which movies he/she likes.
Language	The language spoken in the movie is a driving factor for the audience. Example, a user may only watch Bollywood movies because he/she cannot understand English.

Here, we consider various attributes and a combination of them to determine a user's preference. We apply a counter to the above attributes and determine the decisive factors for a user to like a particular movie. [5] For example, if a user has watched 10 movies with the actor Tom Cruise starring in it, it is safe to assume that the user has a preference for Tom Cruise movies. Same for genre, if a person has watched 10 comedy movies out of the total 15, it is safe to assume that the person has a preference for Comedy movies.

Thus, when we determine the decisive factors to identify the user preference, we can move forward with the proposed procedure. In the next step, we move forward to normalization and factorization.

B. Normalization and Factorization

Retrieved content features and obtained rating information from IMDB are normalized separately, before they are combined to a single rating matrix. This extended matrix is then passed through the factorization process to reduce the dimension of the data. [6] Finally, collaborative filtering is applied on the reduced data to predict the missing ratings.

In our approach, we are mainly interested in the Movie and User entities, and their relations to any other available features. Possible movie features are actor, country and genre, as well as users that gave ratings on these items. From the perspective of a user, we have the features gender, age and occupation, plus items that were rated by these users. [7] Our goal is to combine the original rating matrix with all extracted feature information in a single model. After constructing an extended matrix, we can apply collaborative filtering techniques to estimate missing user-item ratings. Inflating the original rating matrix with content-based features is expected to improve the performance for rating prediction.

Normalization

When we compare the items contained in our original IMDB ratings matrix with the entries of our generated feature matrices, we will notice that both exhibit a different range of values. Where movie ratings can range from 1 to 5 (zero if non-rated), content-features are either existent or not (1 or 0). Hence, rating matrix and feature matrices both need to be normalized in a different way.

In our case, we make use of subtractive normalization. User-item ratings usually exhibit different kinds of global effects. [8] For instance, some users always tend to give higher ratings on items than other users, and some items at an average receive more positive user feedback than other items. In order to compute accurate rating predictions, global effects need to be removed from our data before applying any collaborative filtering techniques. [9] Typically, a weighted combination of user-, item- and overall-average rating values is subtracted from the original entries to remove individual user preferences and item popularity effects. Typically, all normalized rating have a value around zero, which enables a comparison of the single ratings entries. The general equation for the subtractive normalization is formulated in the following:

$$\tilde{r}_{u,i} = r_{u,i} - \alpha\overline{r} - \beta\overline{r}_u - \gamma\overline{r}_i$$

$$\overline{r} = \frac{\sum_x \sum_y r_{xy}}{\#ratings(r)} \qquad \overline{r}_u = \frac{\sum_y r_{uy}}{\#ratings(u)} \qquad \overline{r}_i = \frac{\sum_x r_{xi}}{\#ratings(i)}$$

The parameters, and determine the influence of the observed effects on the final normalization result. Our purpose is to find a single parameter configuration, which gives optimal normalization results for all entries of our original rating matrix.

Matrix Factorization

Besides analysing diverse movie and user features, our research is also concerned with matrix factorization, and its impact on prediction accuracy. Typically, matrix factorization techniques are employed to reduce the dimension of the item space and/or to retrieve latent relations between items of the observed dataset.

In our case, we employ the well-known Singular Value Decomposition (SVD) method, which factorizes the original rating matrix into three low-dimensional matrices containing the left-singular vectors, the singular values and right-singular vectors respectively (R = U · S · V′). [10] The resulting matrices can be utilized for rating prediction in several different ways. Because the new generated matrices can be considered as an approximation of the original rating matrix, they can be employed to directly estimate missing rating values. For a particular matrix entry $r_{u,i}$, this is done in the following way:

$$\hat{r}_{u,i} = X(u) \cdot Y(i) \qquad (X = U_k S_k^{1/2} \ \& \ Y = S_k^{1/2} V_k')$$

The compound matrices X and Y represent user and item concepts respectively. Unknown user-item ratings are predicted by computing the dot-product between the appropriate user and item concept.

C. Collaborative Filtering

Collaborative filtering algorithm usually works by searching a large group of people and finding a smaller set with tastes similar to that of a particular user. It looks at other things they like and combines them to create a ranked list of suggestions. In our case, we employ the item-based collaborative filtering algorithm (IBCF).

In order to make the rating predictions for target item B by user A, the first step is to determine a set S of items that are most similar to target item B. The ratings in item set S, which are specified by A, are used to predict whether the user A will like item B. [11] Therefore, considering the users in the previous example, user X's ratings on similar science fiction movies like Alien and Predator can be used to predict his rating on Terminator and subsequently make a recommendation.

In the following, we want to illustrate the algorithm of IBCF. Again, we first need to compute the similarities of the neighbourhood items, before their ratings can be employed for estimation.

TABLE III
RATING TABLE

	Forrest Gump	Pulp Fiction	Toy Story	Star Wars
Forrest Gump (FG)	-	0.7389	0.6301	0.7006
Pulp Fiction (PF)	0.7389	-	0.4523	0.7314
Toy Story (TS)	0.6301	0.4523	-	0.5307
Star Wars (SW)	0.7006	0.7314	0.5307	-

Employing the item-oriented approach, the final rating prediction for our tuple <A, Pulp Fiction> is quite different to what we calculated before.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} itemSim(i,j) \cdot r_{uj}}{\sum_{j \in ratedItems(u)} |itemSim(i,j)|}$$

$$pred(A,PF) = \frac{iSim(PF,FG) \cdot r_{A,FG} + iSim(PF,TS) \cdot r_{A,TS} + iSim(PF,SW) \cdot r_{A,SW}}{|iSim(PF,FG)| + |iSim(PF,TS)| + |iSim(PF,SW)|}$$

$$pred(A,PF) = \frac{0.7389 \cdot 3 + 0.4523 \cdot 4 + 0.7314 \cdot 2}{|0.7389| + |0.4523| + |0.7314|} \approx 2.8548$$

In summary, our approach is special in that we unify user-item ratings and content features in a single model/matrix (Feature Combination), which is exploited by nearest neighbourhood techniques (NNH-algorithm)

subsequently. [12] Moreover, our research gives attention to both user and item features, where the impact of single and joined features is analysed.

In order to decrease the computational runtime and memory usage of our system implementation, we furthermore employ matrix factorization (SVD) on the model. Although the resulting low-dimensional matrices are just an approximation of the original rating matrix ($R \approx U \cdot S \cdot V'$), they are less sparse and reveal hidden user (U) or rather item (V′) relations. [13] Besides that, we expect the injected features to have a positive influence on the explanatory power of the decomposition, because additional features contribute to a more precise differentiation of the single users and items respectively. The main purpose of our research is to find out which features are beneficial to rating prediction, and in what extend matrix factorization is advantageous to our approach.

III. CONCLUSIONS

This paper proposes and exhibits a method to determine user-specific movie ratings using various data mining techniques such as Apriori and Collaborative Filtering. Initially, the analysis of user attributes is performed to find out the user preferences and association rules are determined to identify user similarities. Next, we use procedures such as normalization and factorization to construct a normalized matrix on which we can perform collaborative filtering. Finally, we perform collaborative filtering to successfully determine user-specific movie ratings.

In the future, we plan to further refine our procedure to extend the use of this likeability rating process to more domains like e-commerce. We also intend to iterate our process to avoid any limiting factors and increase the precision of the user-specific movie ratings.

REFERENCES

[1] B. Amini, R.,Ibrahim, and M.S.,Othman (2011). Discovering the impact of knowledge in recommender systems: A comparative study. arXiv preprint arXiv:1109.0166.

[2] M. A. Ghazanfar, and A. Prugel-Bennett (2010, January). A scalable, accurate hybrid recommender system. In Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on (pp. 94-98). IEEE.

[3] T.,Tran, and R.,Cohen (2000, July). Hybrid recommender systems for electronic commerce. In Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press.

[4] R. Perego, S.,Orlando, and P.,Palmerini (2001). Enhancing the apriori algorithm for frequent set counting. Data Warehousing and Knowledge Discovery, 71-82.

[5] B. Sigurbjrnsson, and R.,Van Zwol (2008, April). Flickr tag recommendation based on collective knowledge. In Proceedings of the 17th international conference on World Wide Web (pp. 327-336). ACM.

[6] P.,Tan, M.,Steinbach, and V.,Kumar (2005). Introduction to data mining. Boston: Pearson Addison Wesley.

[7] http://kaggle.com, for IMDb datasets

[8] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (n.d.). Machine Learning Group at University of Waikato Retrieved November 18, 2012

[9] B.Sarwar, G.,Karypis, J.,Konstan,and J.,Riedl (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

[11] G. Shani, and A. Gunawardana (2011). Evaluating recommendation systems. Recommender Systems Handbook, 257-297.

[12] Y. Koren (2008, August). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434). ACM.

[13] Alsalama, Ahmed (2013). A Hybrid Recommendation System Based on Association Rules. Masters Theses and Specialist Projects. Paper 1250.

