Knowledge Discovery in the Movie Database

Sainan He
School of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Cananda
Email: s66he@uwaterloo.ca

Yuzhou Wang
School of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Cananda
Email: y2345wan@uwaterloo.ca

***Abstract*—Data mining has become a popular topic both in academic and induristrial fields. With mining techniques which aim to discover potential information and obtain knowledge, people are able to analyze links between attributes in real world data sets. For example salesmen can boost their sales by puting associated or similar products together, or by recommending products the customers will most likely be interested in. The abundance of movie data in terms of review, rating or even detail information in the internet has encouraged many researches to formulate techniques to analyze the pattern in movie data, invovlving in discovering factors which will influence the success of movies and developping recommendation systems of movie according to user reviews. In this paper, we will apply several data mining techniques including classification, clutering, association rules, and prediction on two Internet movie datasets. We will implement different algorithms and evaluate their performances and analyze the results.**

I. Introduction

Nowadays, we live in a world with vast amounts of data collected everyday. Analyzing such data is an urgent need, thus, data mining has become a popular topic and a fast-growing field. Data mining techniques are already widely deployed in many essential area such as business, society, science, medicine, engineering and almost everywhere[1].

Data mining involves different knowledge discovery such as classification, clustering, association analysis[2]. Association rule learning is a method for discovering interesting relations between variables in large databases. Clustering is an unsupervised data mining technique for discovering interesting patterns from a given database. Classification is a supervised approach classifying records in a data set into predefined classes or even defining classes on the go.

It is very likely that a user will give a movie similar rating with another user who has the same taste. Such an approach that making predictions based on the interests of a user by collecting preference or tastes information from other users is Collaborative Filtering(CF) which is the most popular method in recommendation systems.

The objective of this paper is firstly, to provide a suitable approach along with necessary factors that are to be considered for association rules, clustering and classification using Internet Movie Database (IMDB) data. Performing different classification, a comparison is based on the evaluation the results. Lastly, apply collaborative filtering method to predict users' rating to movies using MovieLens dataset.

The organisation of the paper is as follows: Section 2 provides the literature review about the problem domain. Section 3 and 4 introduces the two dataset that we used in this paper and the data processing approaches. Section 5 gives an overview of the techniques we use to perform our analysis. Section 6 describes the actual analysis performed, and then presents the results and a discussion thereof. Section 7 gives the conclusions reached and a note about possible further work.

II. Literature Review

A. Data mining for the internet of things: literature review and challenges[3]

This is a review article surveying data mining through 3 perspectives. From the knowledge view, it illustrated classification, clustering, association analysis, time series analysis, outlier analysis and related realization such as SVM, Bayesian networks, SVD, etc from technique view. Then it introduced application of these theory and techniques in different fields. It also discussed challenges for data mining, like extraction useful data from large quantity of data with low quality. Finally, based on the former analysis on data mining algorithms and application, the author proposed a system architecture for big data mining system.

B. A Case Study in a Recommender System Based on Purchase Data[4]

This paper present three kinds of collaborative filtering algorithms- memory based approach, matrix factorization and bigram matrix method- on a real-world dataset for recommending items to customer according customers purchase histories. The author established models for the three methods and applied them on different settings, comparing their results. They also proposed the multidimensional model for contextual analysis, but they didnt talk about it in detail due to the space limit. The research mainly drew the conclusion that: (1) the algorithm based on bigram association rules obtained the best performances; (2) the performance of these algorithms has slight difference compared to that of introducing contextual analysis.

C. Algorithms and Methods in Recommender Systems[5]

In this report, the researcher have described traditional and modern recommender approaches with giving concrete

examples and presenting their problems. The traditional ones which work with profiles of users, Content-based filtering measuring similarities and Collaborative filtering building neighbourhood, are usually combined to avoid some limitations and problems for better results. Some modern methods are an extension of collaborative filtering, such as Context-aware, Semantic-based and Cross-domain based approaches, which outperform original one. Nevertheless, obtaining context information and creating new text mining techniques are some of the problems remaining to be solved. Others like Peer-to-Peer and Cross-lingual approaches are briefly introduced by the researcher.

III. Data Collection

In this paper, we will implement association rules, clustering, classification and rating prediction. We choose two widely used datasets about moives- IMDB and Movielens. IMDB dataset is used for the former three data mining methods and MovieLens is for rating predicting.

A. IMDB

The Internet Movie Database (IMDb) is a comprehensive online database having information about movies, actors, television shows, production, etc. The IMDb web site[6] provides more than 50 text files in ad-hoc format (called lists) containing different characteristics about movies (e.g. actors.list or running-times.list). Given the large scale of the data and the degree of interactions between the people, IMDb is a fertile source of data mining problems.

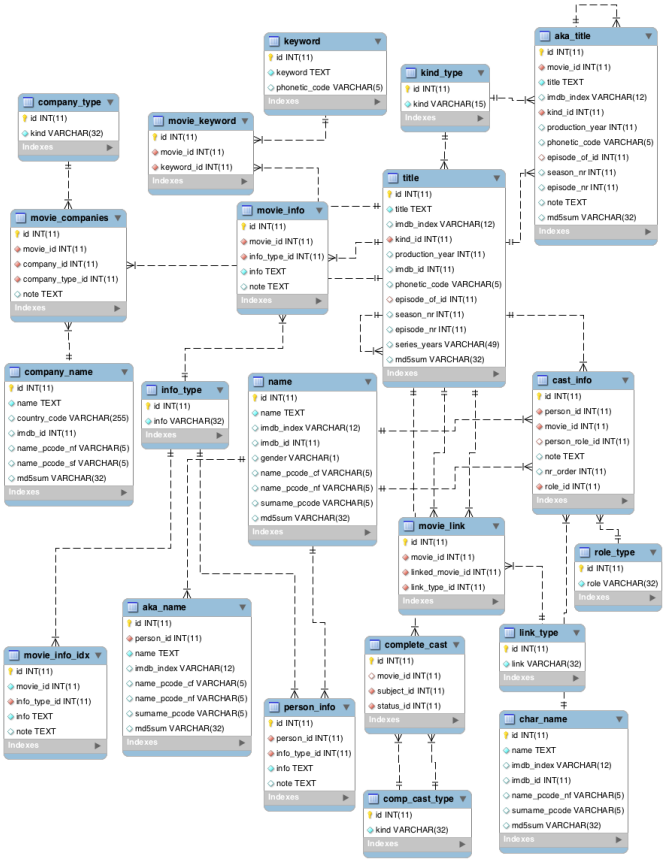

Fig. 1. IMDB Database Relational Schema

B. MovieLens

MovieLens is a movie recommender project, developed by the Department of Computer Science and Engineering at the University of Minnesota. MovieLens is a typical collaborative filtering system that collects movie preferences from users and then groups users with similar tastes. Based on the movie ratings expressed by all the users in a group it attempts to predict for each individual their opinion on movies they have not yet seen. Two data sets are available at the MovieLens web site[7]. In this paper, we will use the MovieLens 100K dataset which consists of 100,000 ratings for 1682 movies by 943 users.

IV. Data Preparation

A. Reconstruct Data in MySQL Database

At first we tried to store all the data in IMDB into MySQL. As mentioned above, the original dataset in IMDB is in text files which is a list format within natural language. This provided the insight that raw IMDB data are unsuitable for data mining unless they are processed through some natural language processing tool. To minimize the effort spent on parsing all the text files and then converting each one to a table in a database, a third party tool name IMDBPY as obtained from[8] is used. It is an alternative way to navigate through movie information. IMDBPY automatically imports the list files and creates a MySQL database with tables and populates the tables with required data.

The dataset contains abundant information about movies which makes it a very large amount of records. It takes more than 3 hours to convert all the files and import the data into MySQL with IMDBPY. Then we add relevant indexes and foreign keys in the database. Finally, we have 21 tables. The basic information of actor/actress is stored in name table. The movie basic information is in title table. Table cast_info links the actor/actress, crews and movies. Table movie_info has the detailed information about a movie such as budget, gross, rating, runtime, etc.

Figure 1 shows the relational schema of the IMDB database. We can see that this structure makes query in the database complicated because many additional joins are needed.

We tried several queries on the built database, such as finding the movie with the largest amount of budget or gross. The results are not always correct. For example, the values of budget are strings like ”£998,852” or ”$102,437 (USA)” which makes us unable to find the movie with the largest amount of budget with simple SQL query. Besides, the database doesn’t contain rating information of movies which is a crucial factor for our analysis. So in the next round, we tried to load the files directly into R.

Action	Adventure	Animation	Comedy	Crime	Documentary	Drama
136	18	196	744	42	763	752
History	Horror	Music	Mystery	Romance	ScienceFiction	Thriller
3	236	337	18	41	47	104
Adventure	Animation	Comedy	Crime	Documentary	Drama	Family
25	16	128	70	41	455	254
Horror	Kids	Music	Mystery	Other	Romance	ScienceFiction
176	3	187	94	3775	830	308
Western						
131						

Fig. 2. All Genre

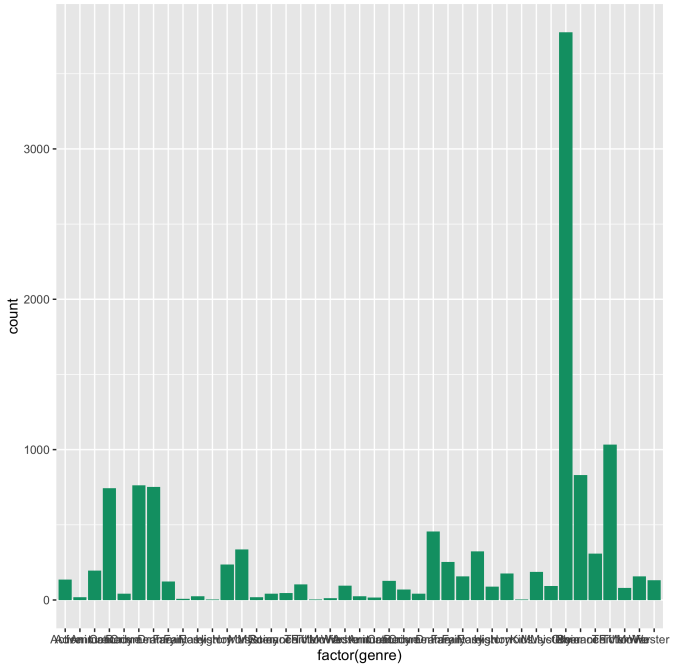

Fig. 3. Genre Distribution

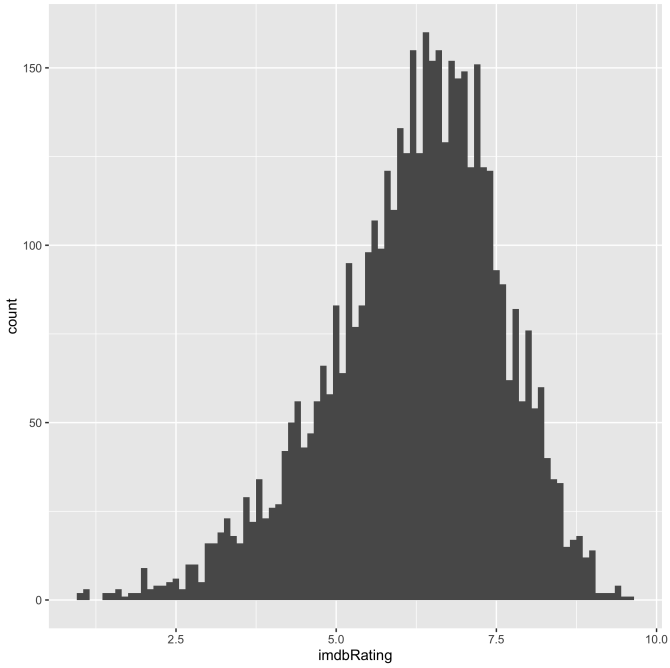

Fig. 4. Rating Distribution in IMDB

B. Parsing Data in R

The downloaded original files were too large to effectively load in R. The computation time and complexity are also a big challenge that our computer cannot handle. So we decide to build a reasonable sample dataset. A python script which can make a unique http request and retrieve data sample randomly is used. As our classification and clustering are based on the genre of movies, there are many attributes that are useless for us, such as seriesID, season, Episode, and etc. We choose to drop these columns during data cleaning process. After processing, the movie dataframe contains 11142 rows of records and 31 attributes for each record. There are 29 genres in total in our dataset as shown in Figure 2. Except for the genre of "Other", the most popular genres are "Romance", "Documentary", "Drama", "Comedy".

There is one crucial problem that will influence our analysis in the following parts is that we noticed there are many missing values in the data. Almost every rows have one or more N/A value in different attributes. It is not possible to delete rows with missing value since it results in deleting almost the whole data. It is also impossible to replace the missing value because the values are strings, stratagies in data cleaning dealing with missing values are not suitable for this case.

We count the number of each genre in the sample set and plot the histogram as Figure 3. The rating range in IMDB is from 1 to 10. From Figure 4, we can see that more than 90% of the movies have rating between 5 to 7.5. The maximum rating is 9.6 while the minimum rating is 1.

Figure 5 shows the movie numbers in each year from 1891 to 2018. There is a clear and exponential increase in the number of movies with the increase of the year, especially during the latest 20 year. This is reasonable as it is consistant with reality which can be inferred to correspond to the growth of the movie industry over time.

The IMDB provided a ranking list of top250 movies. The ranking is based on both the rating and voting counts of movies. So we plot a figure of voting numbers against the ratings. The scatter plot is shown in Figure 6. We can see that the distribution is similar to the rating distribution in Figure 4.

V. Methodologys

A. Association Rules

Association rules are information gathered from datasets. Data mining are interested in discovering relations between attributes that always occur together for prediction and recommendation.

Formulation of the Association Rules Mining problem[9]: Given a dataset of n samples $S = \{s_1, s_2, ..., s_n\}$, where each sample has k items $s_i = \{I_{i1}, ..., I_{ik}\}$, the total possible items $I = \{I_1, ..., I_m\}(k \leq m)$. An association rule is an implication in the form of $x \Rightarrow y$, where x, y are subsets of I and disjoint, also known as itemsets. Following are some definition of concepts in the Association Rules Mining.

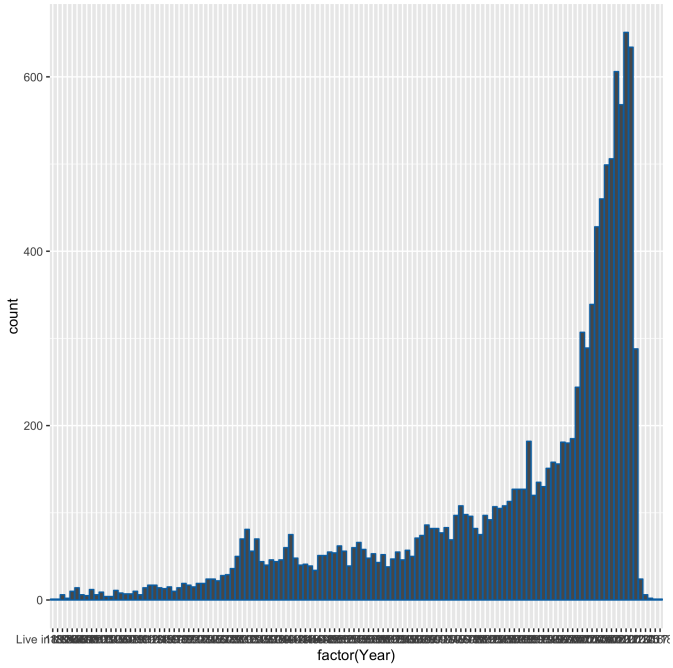

Fig. 5. Movie count by Year

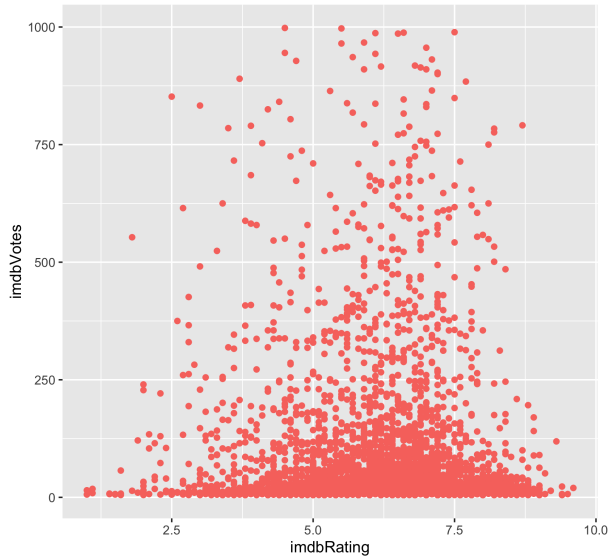

Fig. 6. Vote Count by Rating

Support. The support of an itemset x with respect to a dataset is the proportion of samples in which x occurs in the dataset

$$support(x) = \frac{thenumberofsamplescontainx}{thenumberoftotalsamples}.$$

The support of an association rule $x \Rightarrow y$ with respect to a dataset is the proportion of samples in which itemsets x and y occur together in the dataset

$$support(x \Rightarrow y) = \frac{thenumberifsamplescontainxandy}{thenumberoftotalsamples}.$$

Confidence. The confidence of an association rule $x \Rightarrow y$ with respect to a dataset is the ratio of number of samples in which itemsets x and y occur together to the number of samples in which as long as itemset x occurs in the dataset

$$confidence(x \Rightarrow y) = \frac{thenumberofsamplescontainxandy}{thenumberofsamplescontainx}.$$

The confidence of $x \Rightarrow y$ can also be presented by the ratio of the support of $x \Rightarrow y$ to the support of itemset x

$$confidence(x \Rightarrow y) = \frac{support(x \Rightarrow y)}{support(x)}.$$

Lift. The lift of an association rule $x \Rightarrow y$ with respect to a dataset is defined as the ratio of the support of $x \Rightarrow y$ to the product of the support of itemset x and the support of itemset y

$$lift(x \Rightarrow y) = \frac{support(x \Rightarrow y)}{support(x) \cdot support(y)}.$$

As many association rules are inadequate and useless due to some items occur together by chance, it is necessary to set thresholds for support and confidence respectively for reducing such association rules.

In conclusion, the Association Rule Mining problem is to find out those association rules of which support and confidence above given support and confidence thresholds.

B. Classification

As we have got the main genre for every movie sample after preprocessing the original dataset, we will carry on a genre prediction using some classification methods. Classification is a supervised approach in the field of data mining, and it discovers a knowledge pattern from a training set of data of which the class membership is already labeled, then use this pattern to identify every entry of a new set of data, call testing set, belong to which class. There are a lot of specific methods of classification, such as K-Nearest Neighbours(K-NN), Support Vector Machine(SVM), Naive Bayes classifier, Decision Tree, Neural Network. In this project, these classification algorithms as following are used to predict the genre of a movie sample: K-Nearest Neighbours, Naive Bayes classifier, C4.5, RIPPER, Oblique Tree. It is worthy to notice that the last three methods belong to Decision Tree.

1) K-Nearest Neighbours: K-NN is a similarity based classifier, the label of a testing entry is associated with the label of its neighbours. When given a testing set of data, first calculate each entrys K nearest neighbours in the training set defined by the closeness between them, usually on the basis of a specific distance function or other metrics. After generating K nearest neighbours, the algorithm finds out their labels and take a vote. In another word, an entry is assigned to the most common class among its K nearest neighbours. In practice, the value of K should be chosen not too large or too small[10]. Though large value of K reduce the effect of noise, it makes less distinct boundaries between classes and leads to a underfitting problem. In the contrast, small value of K will cause overfitting. Therefore, we choose a proper K under the

guideline that it should be odd and where n is the number of samples.

2) Naive Bayes classifier: Naive Bayes classifier is a probabilistic approach based on Bayes theorem and make a strong assumption that the value of every attribute of a sample is independent with each other[11]. Bayes Formula is as follows:

$$P(c_i|s) = \frac{P(s|c_i) \cdot P(c_i)}{P(s)} \tag{1}$$

c_i is class i and s is a sample where $P(c_i|s)$ is called posterior, $P(s|c_i)$ is called likelyhood, $P(c_i)$ is called prior, $P(s)$ is called evidence.

Bayes decision rule:

if $P(c_1|s) > P(c_2|s)$, s belongs to class 1;

else if $P(c_1|s) < P(c_2|s)$, s belongs to class 2.

Naive Bayes classifier assumes a sample has d discrete-valued attributes, $s = (a_1, ..., a_d)$. Therefore,

$$P(s|c_i) = P(a_1, ..., a_d|c_i) = \prod_1^d P(a_j|c_i).$$

We can learn likelihood and prior from the training dataset, as $P(a_j|c_i)$ can be evaluated by calculating the fraction of samples in class i share the same feature values, prior $P(c_i)$ is equals to number of samples in class i / number of all samples, moreover, evidence $P(s)$ is identical for all samples. Thus, it is possible to assign each new sample a class label based on the learning of training dataset,

$$P(c_i|s^new) = \frac{\prod_1^d P(a_j|c_i) \cdot P(c_i)}{P(s)}.$$

3) C4.5: Iterative Dischotomiser 3 (ID3), which is used to generate a decision tree from the training dataset. This algorithm regards the training dataset as a root node at the beginning. During each iteration, it uses every attribute to split the dataset and calculates the Entropy and Information Gain of the selected attribute, then choose the attribute which leads to a largest Information Gain as the partition attribute and generate subsets of the original dataset. Recursion will be applied to each subset considering only nonpartition attribute. The recursion will be stopped in some cases: in the subset, every sample belongs to the same class, then the node is assigned with the class label and becomes a leaf node; there are no more nonpartition attribute, then the node is assigned with the most common class label and becomes a leaf node; there no samples correspond to the condition, then the node is assigned with the most common class label in its parent node and becomes a leaf node[12].

C4.5 makes some improvements to ID3. First, it handles both continuous and discrete attributes with a threshold metric. Then, once a decision tree is created, it is able to replace branches not helping with leaf nodes, which is known as pruning. Finally, it can deal with missing attribute values in training dataset by ignoring them in the process of calculating Entropy and Information Gain, which is also a main reason for us to use C4.5 instead of ID3 as the dataset has some missing values of attributes.

C. Clustering

Compared with classification, clustering is an unsupervised approach of data mining. Thus, we create a new dataset without attribute genre, and there is no need to divide the dataset into training and testing dataset. However, after applying clustering algorithm on the dataset, we will use the attribute genre in the original dataset to measure how good the performance of the method is.

As there are both numeric and nominal values of attributes in the dataset, we will not use traditional approaches, such as KMeans, Fuzzy CMeans and DBSCAN, which are based on distance metric. Instead, we choose Robust Clustering using links (ROCK) to cluster the dataset into clusters within the number of genres.

ROCK evaluates the similarity between samples with the number of shared neighbours rather than distance, then a hierarchical clustering scheme is used to cluster the dataset. Thus, it can handle nominal attribute values with this special link and perform better than other traditional algorithms[13].

D. Collaborative Filtering

Collaborative filtering filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see. The above is the basic idea of our implementation is this part.

Typically, collaborative filtering adopts the neighborhood-based technique. There are two kinds of approach: user-based and item-based.

User-based collaborative filtering, also know as *k-NN collaborative*,was the first of the automated CF methods. It find other users whose past rating behavior is similar to that of current user and use their ratings on that item to predict what the current user will like.

Item-based Collaborative Filtering is similar to User-based CF, it uses similarity between the rating patterns of items. If two items tend to have the same users like and dislike them,then they are similar and users are expected to have similar preferences for similar items.

1) Computing Predictions: To compute predictions or recommendations for a user u, user-user CF firstly needs to determine the number N of neighbors will be used to generate the result. Then computing the weighted average of the chosen neighboring users' rating i by using similarity as weights[14]. The formula is given as below:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} s(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u,u')|} \tag{2}$$

In order to eliminate the differences in users's use of the rating scale, subtracting the user's mean rating $\bar{r}_{u'}$ to

compensate is necessary. The parameter $p_{u,i}$ is predicated rating on item i for user u. $\bar{r}_{u'}$ is average rating on all items rated by user u.The parameter $r_{u',i}$ indicates the rating of user u' on item i.$s(u,u')$ is similarity between user u and u'. N is the number of neighbors chosen for user u.

2) Measure of Similarity: An critical parameter used to calculate predications is similarity function. One of the most common and typically measurements is the cosine similarity.

In Cosine Similarity model, users are represented as $|I|$-dimensional vectors of rating on $|I|$ items. Similarity is measured by the cosine distance between two rating vectors. The formula is given below indicating how to calculate the Cosine Similarity between user u and v [15].

$$s(u,v) = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2}\sqrt{\sum_i r_{v,i}^2}} \tag{3}$$

r_u is rating vector of user u.

3) Evaluation Metrics: Our goal is to predict the rating a user would give to a restaurant. We predict the rating that user has not rated in the training dataset, but the true rating is stored in the test dataset. We use the root-mean-square error and mean-absolute error for evaluation.

$$RMSE = \sqrt{\frac{\sum \left(r'_{u,i} - r_{u,i}\right)^2}{N}} \tag{4}$$

$$MAE = \sqrt{\frac{\sum \left|r'_{u,i} - r_{u,i}\right|}{N}} \tag{5}$$

Here $r'_{u,i}$ is the predicted rating from user u on item i and $r_{u,i}$ is the true rating; N is the size of test dataset.

Another evaluation method is precision-recall: precision tells us how good the predictions are. In other words, how many were a hit; recall tells us how many of the hits were accounted for, or the coverage of the desirable outcome.

$$precision = \frac{|\{relevantdocuments\} \bigcap \{retrieveddocuments\}|}{|\{retrieveddocuments\}|} \tag{6}$$

$$recall = \frac{|\{relevantdocuments\} \bigcap \{retrieveddocuments\}|}{|\{retrieveddocuments\}|} \tag{7}$$

VI. Experimental Evaluation

A. Classification

After preprocessing the original dataset by deleting non-movie, approximately twothirds samples were dropped.Then, we ignore the useless attributes for data mining and divide the original dataset into two new dataset: training dataset with attribute genre and testing dataset without attribute genre. In addition, because there are a several hundred of genre in the original dataset, we choose 29 genres having sufficient support samples. Furthermore, many samples have more than one genre, in order to evaluate the performance of methods, we only take into account a main genre for these samples.These new datasets provide abundant information for genre prediction. We apply classification algorithms on training dataset to generate a learning model and then use it on testing dataset to predict the genres of movie samples.

TABLE I
Comparison of Different classification methods

Methods	Accuracy	95% CI	No Information Rate
K-Nearest Neighbours	0.0573	(0.0324, 0.0927)	0.2366
C4.5	0.1714	(0.1581, 0.1854)	0.2686
Naive Bayes classifiere	0.3429	(0.3259, 0.3602)	0.2686

1) KNearest Neighbours: It is already mentioned in section 5 the reason for choosing a proper value of k. However, even if we choose the best value for parameter k, due to many samples contain null values and every sample has both nonminal and numeric attribute values, the performance of K-NN is poor as it is a distance based approach. The accuracy of K-NN is 5.73%.

Accuracy : 0.0573
95% CI : (0.0324, 0.0927)
No Information Rate : 0.2366
P-Value [Acc > NIR] : 1
Kappa : -0.0534
Mcnemar's Test P-Value : NA

2) C4.5: Although the classification performance of C4.5 is not good enough, it is still better than K-NN. One reason for this is that C4.5, one of Decision Tree algorithm, has the ability of dealing with nonminal attribute values, which K-NN is lack of. The accuracy of C4.5 is 17.14%.

Accuracy : 0.1714
95% CI : (0.1581, 0.1854)
No Information Rate : 0.2686
P-Value [Acc > NIR] : 1
Kappa : 1e08
Mcnemar's Test P-Value : NA

3) Naive Bayes classifier: This classification approach shows the best performance of the three methods as it has the highest accuracy. This is likely due to that each attribute is considered as independent and make a contribute to the probability of the sample belonging to a specific class. The accuary of NB is 34.29%.

Accuracy : 0.3429
95% CI : (0.3259, 0.3602)
No Information Rate : 0.2686
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.2377
Mcnemar's Test P-Value : NA

The classification accuracy of the three methods is compared in table1.

B. Clustering

We perform a similar preprocess on the original dataset as the way we do for classification. The different is that after eliminating non-movie samples, useless attributes and choosing the main genres of samples, we keep a copy but

delete the attribute genre for clustering, and use the one with attribute genre to measure the performance.

ROCK clustering algorithm can deal with nonminal attribute values, which is a significant strength from other traditional clustering algorithm. First, the link value between each pair of samples are computed using the number of shared neighbours between them to reflect the dissimilarities. Then, with the dissimilarity and a objective function which is to maximize the shared neighbours, an agglomerative hierarchical clustering is applied on the dataset. Finally, those samples have not assigned cluster labels will be put in the clusters found.

After applying ROCK clustering algorithm on the dataset, it is divided into 723 clusters. This number is pretty larger than the number of genres that we choose, 29. A reason for this is that maybe some movie samples with a hybrid genre have been considered in distinct clusters. It is reasonable and consistent with dataset preprocessing, where we choose the main genre of each this kind of sample. Another reason is that though some movie samples share the same genre, their other attributes are not such similar that the algorithm can not differentiate them into the same cluster. As clustering is an unsupervised approach, it is easy to understand that its limitations are more than classification.

C. Association Rules Mining

The Brute Force method of mining association rules is that list all possible rules on the basis of subset of itemset I, and select those whose support and confidence are larger than given thresholds. However, since our dataset has a large amount of samples, it is not practical to apply this method. In addition, two observations can be discovered from the definition of association rules: those rules with large support consist of itemsets with large support, which is also known as large itemsets; any subset of a large itemset is also large, in contrast, if a large itemset contains a non-large subset, it is not large either.

As above indicates, we choose Apriori Algorithm[16] to implement association rules mining.In the first pass, it examines every item in the itemset I, determine which part are large itemsets of size 1, and prunes those non-large itemsets. Then, it permutates 1-size large itemsets to 2-size itemsets which are candidate large itemsets. Again, supports and confidences are calculated and compared with the specified thresholds to decide whether those candidates are large itemsets and prune non-large ones. It makes a recursion on the candidates until there is no more candidate.

Owing to that in the original dataset, there are too many attributes and we are not interested in the relationships between some attributes, we produce two new datasets. One is comprised of leading roles, their two most significant supporting roles and three high-ranking crews got from each movie they participated in. Another includes crews and movies genres.

After performing Apriori Algorithm on our new dataset and deleting those redundant association rules, we set thresholds for support and confidence as 0.0002 and 0.9, then select top 10 and 10 rules respectively. The rules are shown in the table 2 and table 3.

TABLE II
THE USUAL LISTS OF ACTORS

antecedent	consequent
{supporting role1 = Divine, supporting role2 = David Lochary}	{leading role = Mary Vivian Pearce}
{supporting role1 = Divine, supporting role2 = Mary Vivian Pearce}	{leading role = David Lochary}
{supporting role1 = Divine, supporting role2 = David Lochary, crew = John Waters}	{leading role = Mary Vivian Pearce}
{supporting role1 = Divine, supporting role2 = Mary Vivian Pearce, crew = John Waters}	{leading role = David Lochary}
{supporting role1 = David Lochary}	{leading role = Divine}
{supporting role1 = David Lochary}	{leading role = Mary Vivian Pearce}
{supporting role1 = David Lochary, supporting role2 = Mary Vivian Pearce}	{leading role = Divine}
{supporting role = David Lochar, crew = John Waters}	{leading role = Divine}
{supporting role1 = David Lochary, supporting role2 = Mary Vivian Pearce, crew = John Waters}	{leading role = Divine}
{supporting role1 = Joe Pennyr, supporting role2 = William R. Moses}	{leading role = Lea Thompson}

We regard leading roles as primary key in this new dataset, and search the original dataset with every leading role to find out the movies they take part in, then extract the supporting roles and crews in the movies, count the number and make a ranking to choose the top two supporting roles and three crews for each leading role as his attribute in the dataset. We produce this dataset for the reason that it is typical that some actors always cooperate together in the movies of a specified genre.

The Apriori algorithm generates thousands of rules and we choose 10 samples of rules to analyse. However, these 10 rules are highly related with the filmmaker John Waters and his coworkers Divine, David Lochary, Mary Vivian Pearce. It can be deduced that John Waters always make low-budget movies as himself play different roles in the same movie such as writer, director, producer and his actors are only several ones.

TABLE III
THE CREWS AND GENRES

antecedent	consequent
{crew = Walt Disney}	{genre = Animation}
{crew = Jules White}	{genre = Comedy}
{actor = Curly Howard}	{genre = Comedy}
{actor = Mel Blanc}	{genre = Family}
{actor = Vijayakanth}	{genre = Drama}
{crew = Ilayaraja}	{genre = Drama}
{crew = Robby D.}	{genre Other}
{crew = Dave Fleischer}	{genre = Other}
{crew = Mel Blanc}	{genre = Other}
{genre = Animation}	{actor = Clarence Nash}

For this new dataset, we discover each movies genre and the actors and crews participating in, since some actors almost

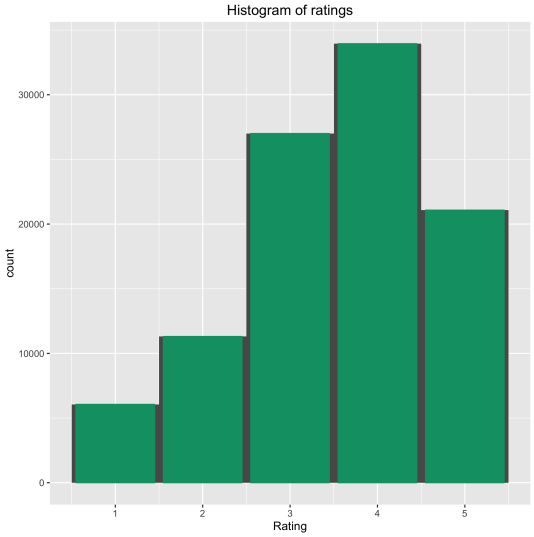

Fig. 7. Rating Distribution

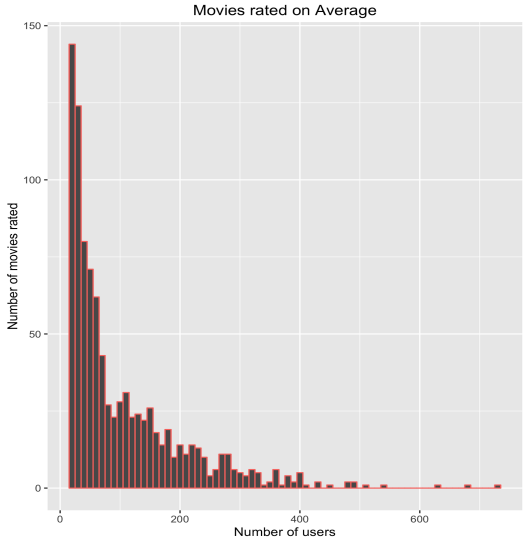

Fig. 8. User Rating Count Distribution

exclusively show in several genres of movie throughout their careers and some directors focus on only one specified genre in decades. And the above rules have proved our assumption. For example, Walt Disney was an American entrepreneur, animator, voice actor and film producer, who co-founded The Walt Disney Company with his brother. Thus, almost every film he played a role in belongs to animation. In addition, in the second rule shown above that if a movie belongs to animation, Clarence Nash always shows. It is a little absolute but not wrong. Clarence Nash was an American voice actor, best known as the Disney cartoon character Donald Duck as he provided the distinctive voice for nearly fifty years. Moverover, he also provided Tom in Tom and Jerry with the voice, which is a famous and popular cartoon.

D. Rating Prediction

1) Visualizing Data: Data for this part is from the MovieLens dataset which is a rich resource for recommendation. We firstly visualize the data by ploting several histograms. Figure 7 is the rating distribution of the raw data. The rating range is from 1 star to 5 stars. Then we use a z-score normalization to normalize the data for further analysis. Figure 8 shows the distribution of user's rating count. We can see that many people just reviews quite a few movies. Figure 9 is the distribution of each movie's average rating. It indicates that most movies received 3 to 4 stars.

2) Collaborative Filtering Results: We apply both user-based CF and item-based CF for rating predicting. Also, we use a random prediction as a baseline.

Figure 10 compares the prediction accuracy of the three methods. The error rate shows that use-based CF works best among these three methods with a RMSE error of less than 1. Item-based CF is not as good as use-based CF. Its RMSE error rate is slightly higher than user-based CF. One possible reason about this difference is when and how we generating recommendations. User-based CF saves the whole matrix and

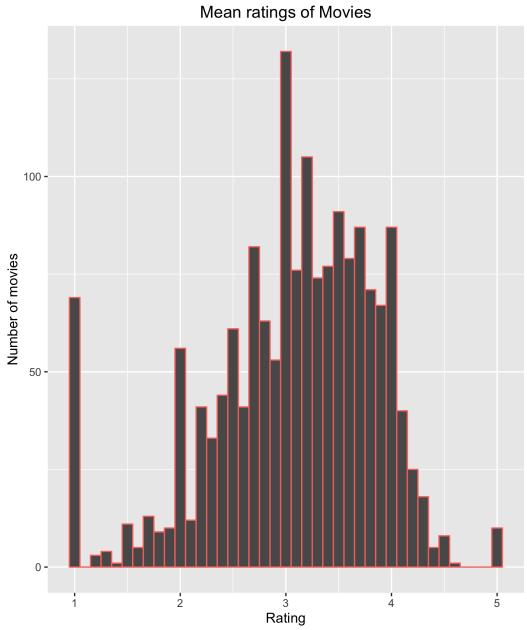

Fig. 9. Movie Average Rating Distribution

then generates the recommendation at predict by finding the closest user. While item-based CF saves only k closest items in the matrix and doesnt have to save everything. It is pre-calculated and predict simply reads off the closest items. We can see that the MAE of user-based CF and item-based CF are nearly the same. MAE is always smaller than RMSE because RMSE will enlarge the penalty on incorrect predictings. It is not surprise that Random approach is the worst because it only set a random rating between 1 to 5 for each movie by users.

We compared the performance of Random, user-based CF and item-based CF by changing the parameter n. That is we evaluate top-1, top-3, top-5, top-10, top-15 and top-20

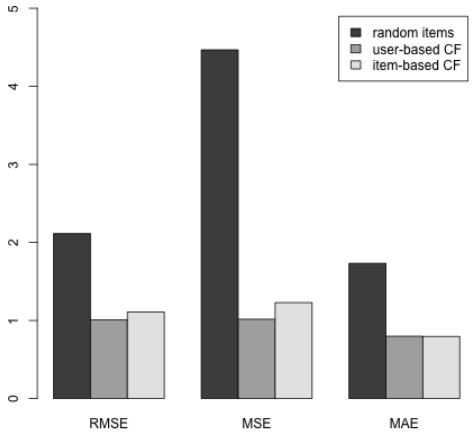

Fig. 10. RMSE

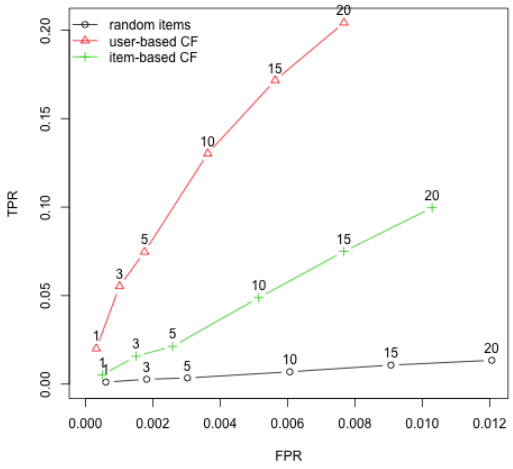

Fig. 11. ROC Curves

TABLE IV
RUNNING TIME OF THREE ALGORITHMS

Method	model time	prediction time
Random	0.025sec	0.062sec
User-based	0.054sec	0.859sec
Item-based	37.911sec	0.591sec

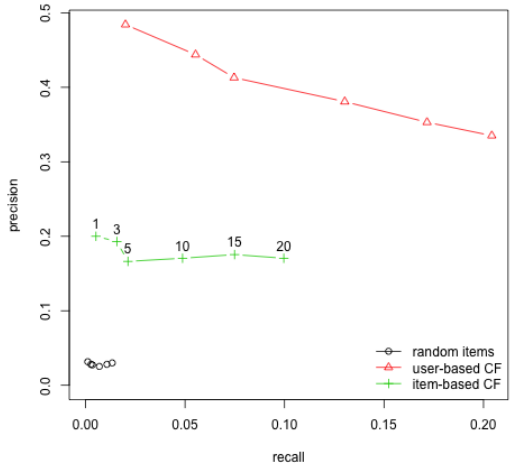

Fig. 12. Precision and Recall Curves

recommendation lists. We can visualize the results by ploting ROC curves(Figure 11) and precision-recall curves (Figure12).

ROC curve is a plot of the true positive rate against the false positive rate for the different possible cutpoints of a diagnostic test. The area under the curve is a measure of the accuracy. Thus the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The ROC curves show the same result with the former part. We can see that under a varying range of top-N list, the performance of user-based CF is always the best following by item-based CF. Random method is the worst. Besides, with the increasing of n, the differences of the TPR against FPR among the three becoming larger.

Different form ROC whose goal is to be in the upper-left-hand corner, the goal of precision-recall space is to be in the upper-right-hand corner, and the PR curves in Figure 11 show that there user-based CF works quite well while item-based CF still has vast room for improvement.

We also meatured the prediction time of each method. Table 4 shows the model time and prediction time of each applied algorithms.

Table 4 shows that Random approach is the fastest as it involves in little computation. Both user-based and item-based CF need to calculate similarities between users or items, which is a considerable computation. We noticed that item-based CF takes plenty of time on training model. This is because the number of ratings a movie receives is greater than the number of rating a user has rated. Therefore, computing similarity between items involving in larger data which apparently costs more time.

VII. CONCLUSION

Knowledge discovery in database is one of the most important tasks in database systems. In this project, we aim to study different data mining algorithms and their applications. We use the IMDB data and Movielens data as our dataset, applying four main techniques of data mining- classifacation, clustering, association rules and rating prediction. We first tried to create a database in MySQL, a third part tool IMDBPY was applied to utilize our work, and we got 21 tables. For the convenience of exercising different mining techniques, we adopt R for analysis. We extract a sample from the IMDB database, applying three classification algorithms including K-

NN, C4,5, and Naive Bayes, ROCK clustering and apriori algorithm. The compared results of classification shows that the Naive Bayes Classifier is the most suitable one for our dataset with an accuracy of 34%. The other two classifiers- C4.5, K-NN did not perform well, with only 17% and 5% accuracy respectively. After applying ROCK clustering, the dataset is divided into 723 clusters which is much larger than the original 29 types of genres. We also tried to find the relationships between actors and crews as well as between crews and genre. We usesd apriori algorithms, listing 10 potential rules. Finally, we used collaborative filtering method for rating prediction on MovieLens dataset. We exercised both user-based CF and item-based CF, compared their performance regarding RMSE error. The results show that user-based CF works better than item-based CF in our data.

REFERENCES

[1] Jiawei Han, Micheline Kamber, Jian Pei. *Data Mining Concepts and Techniques*, 3rd Edition, 2012.

[2] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. *From Data Mining to Knowledge Discovery in Databases*. AI Magazine Volume 17 Number 3, 1996.

[3] Feng Chen, Pan Deng, Jianfu Wan and etc. *Data Mning for the Internet of Things: Literature Review and cChallenges*. International Journal of Distributed Sensor Networks, 2015.

[4] Bruno Pradel, Savaneary Sean, Julien Delporte. *A Case Study in a Recommender System Based on Purchase Data*. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 377-385, 2011.

[5] Daniar Asanov. *Algorithms and Methods in Recommender Systems.* Berlin Institute of Technology, Berlin, Germany, 2011.

[6] http://www.imdb.com/

[7] http://movielens.umn.edu/

[8] http://imdbpy.sourceforge.net/

[9] Michael Hahsler *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules.* 2015.

[10] Everitt, B. S., Landau, S., Leese, M. and Stahl, D. *Miscellaneous Clustering Methods, in Cluster Analysis, 5th Edition.* John Wiley & Sons, Ltd, Chichester, UK, 2011.

[11] Narasimha Murty, M., Susheela Devi, V. *Pattern Recognition: An Algorithmic Approach.* 2011.

[12] Quinlan, J. R. *Induction of Decision Trees. Mach. Learn.* 1986.

[13] Guha S, Rastogi R, Shim K. *ROCK: A robust clustering algorithm for categorical attributes.* Data Engineering, 15th International Conference on. IEEE, 1999: 512-521.

[14] Xiaoyuan Su, Taghi M. Khoshgoftaar. *A Survey of Collaborative Filtering Techniques.* Advances in Artificial Intelligence (2009) 421425.

[15] Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, 22(1):5-53, 2004.

[16] Borgelt C, Kruse R. *Induction of association rules: Apriori implementation* Compstat. Physica-Verlag HD, 2002: 395-400.