Knowledge Discovery in the Movie Database

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

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

***Abstract*—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. With these techniques, poeple are able to analyse links between attributes in real data sets so that salesperson can boost their sales by puting associated or similar products together, or by recommending products the customers will most likely be interested in. In this paper, we apply different data mining techniques including association rules, clutering, classification and prediction on two Internet movie datasets.**

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.

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

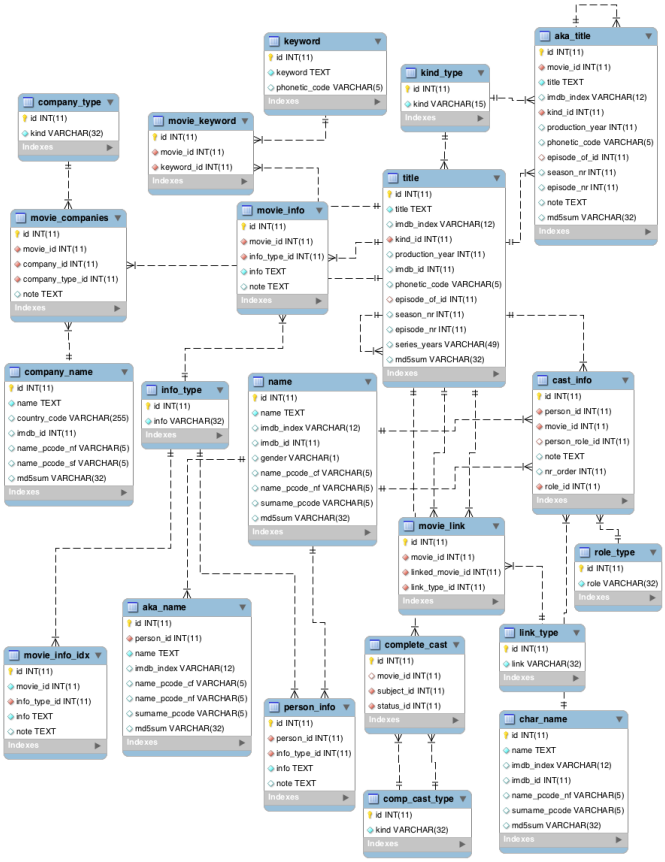

Fig. 1. IMDB Database Relational Schema

MySQL with IMDBPY. Then we add relevant indexes and foreign keys in the database. Finally, we have 21 tables. 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 next, we tried to load the files directly into R.

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 the movie, there are many attributes that are useless, such as seriesID, season, Episode, and etc. We choose to drop these columns during data cleaning process. After processing,

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

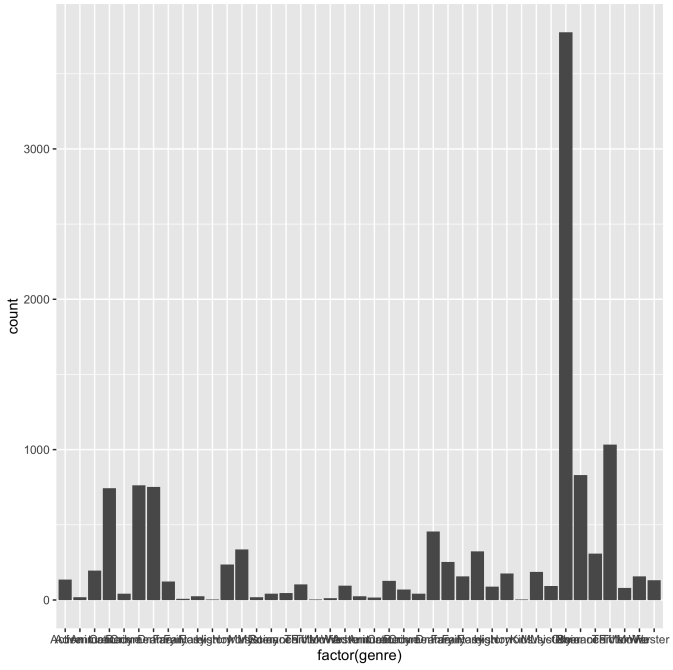

Fig. 3. Genre Distribution

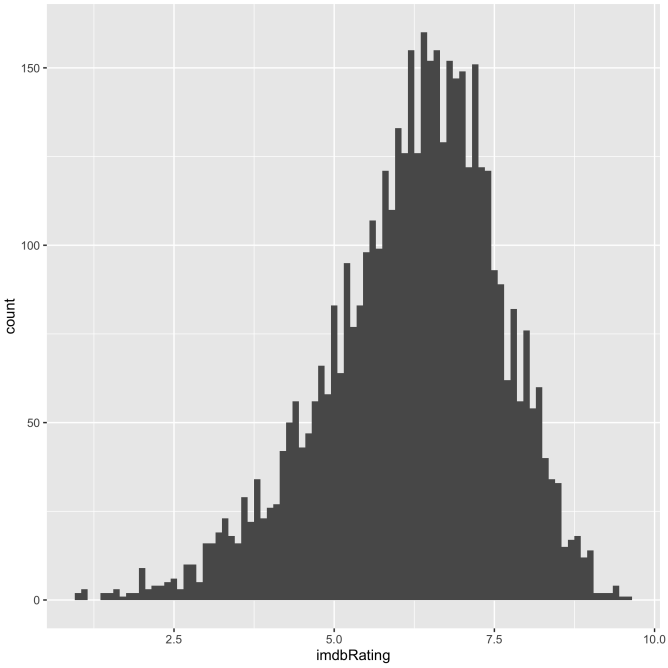

Fig. 4. Rating Distribution in IMDB

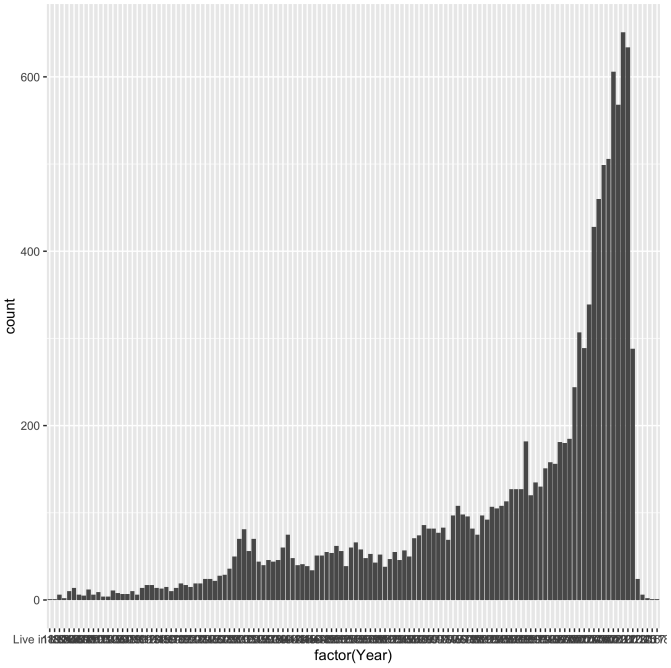

Fig. 5. Movie count by Year

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.

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.

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: Given a dataset of n samples , where each sample has k items , the total possible items (). An association rule is an implication in the form of , where x, y are subsets of I and disjoint, also known as itemsets. Following are some definition of concepts in the Association Rules Mining.

Support. The support of an itemset x with respect to a dataset is the proportion of samples in which x occurs in the dataset. The support of an association rule with respect to a dataset is the proportion of samples in which itemsets x and y occur together in the dataset.

Confidence. The confidence of an association rule 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. The confidence of can also be presented by the ratio of the support of to the support of itemset x.

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

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. 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.

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

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. title

B. title

C. 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 1 is the rating distribution of the raw data. The rating range is from 1 star to 5 stars. Then we use a z-score

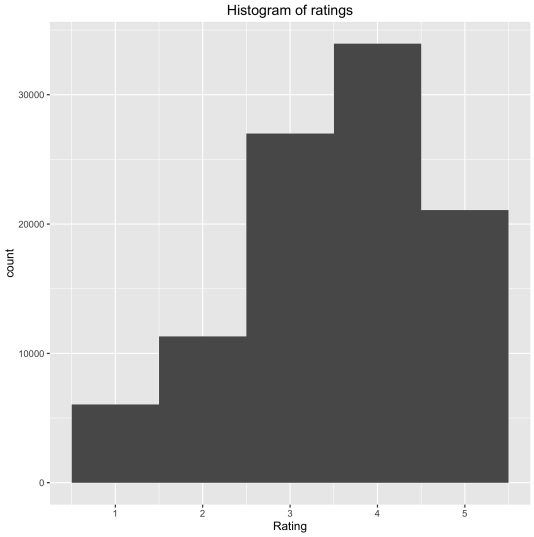

Fig. 6. Rating Distribution

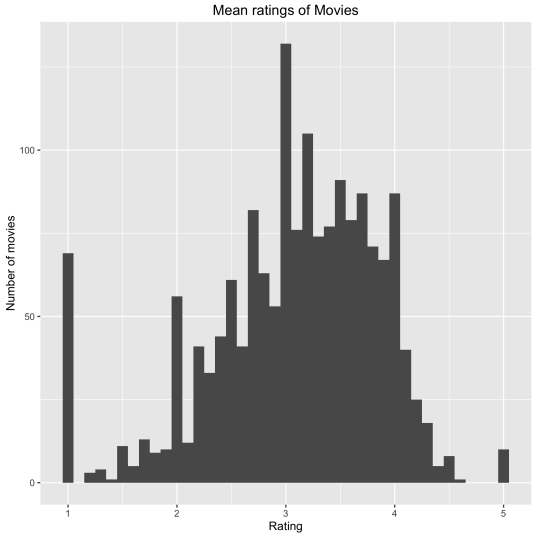

Fig. 8. Movie Average Rating Distribution

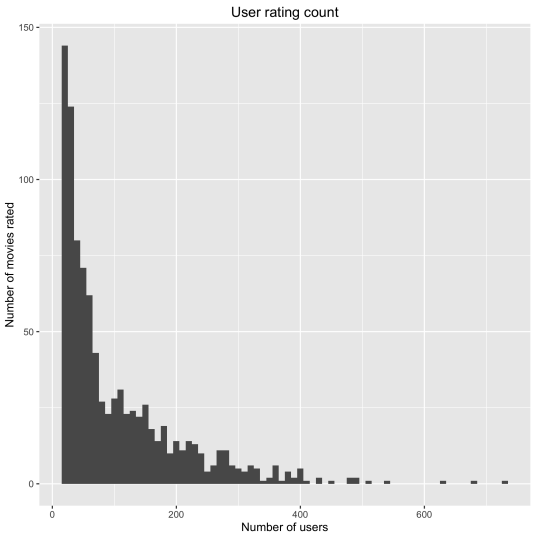

Fig. 7. User Rating Count Distribution

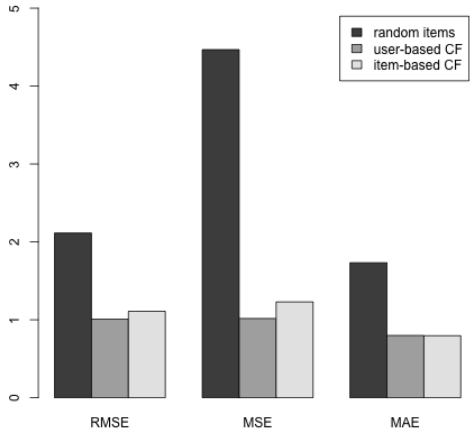

Fig. 9. RMSE

normalization to normalize the data for further analysis. Figure 2 shows the distribution of user's rating count. We can see that many people just reviews quite a few movies. Figure 3 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 for the baseline. Figure 5 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 is about when and how we generating recommendations. User-based CF saves the whole matrix and 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 recommendation lists. We can visualize the results by ploting ROC curves(Figure 10) and precision-recall curves (Figure11).

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.

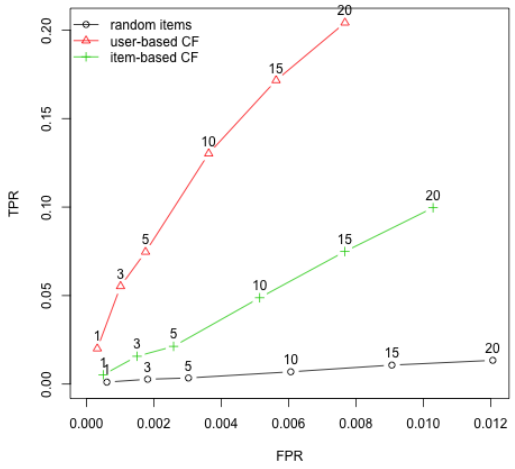

Fig. 10. ROC Curves

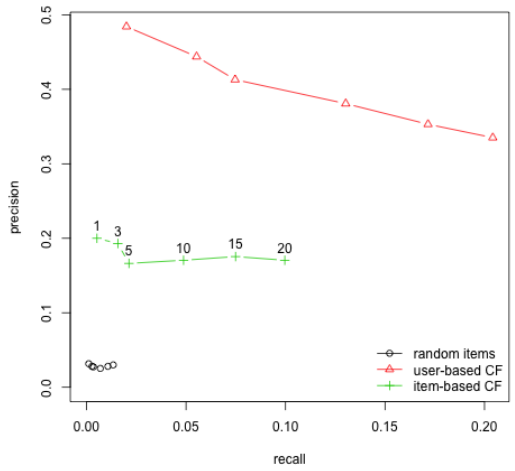

Fig. 11. Precision and Recall Curves

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 predcition time of each method. Table 1 shows the model time and prediction time of each applied algorithms.

TABLE I
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

Table 1 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.

VII. CONCLUSION

The conclusion goes here.

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] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.