Recommender System Using Collaborative Filtering

COMP9417 Machine Learning Project

Group Name: E.Z.L
Group Members:
Zhao Wang(z5124945), Bowen Zhou(z5127532) and Lili Yu(z5040787)

1. Introduction

The recommendation system has widely employed in online service. It provides the guidance for users and helps them make choices. Although people have the different selection, they tend to choose similar things that they liked and choices by who has the homogeneous preference. The recommendation system can collect the similar features and helps to predict potentially favourite items for customers. The goal of this project is to use the method of collaborative-based recommendation, obtain a set of features from the training dataset to make a prediction for the users about the movie rating.

2. Related Work

Recommender systems refer to a system that is competent of predicting the future preference of a set of items for a user and recommend the top items. Furthermore, Collaborative Filtering (CF), as a technique of recommendation based on users' past behaviour, is used in Recommender systems[1]. Basic user-based CF, one of memory based methods, is to measure the similarity between target users and other users, and to make a prediction for the target user by calculating a weighted average of the ratings of the selected users.

One common problem with many collaborative filtering models is that different users adopt different criteria in determining the ratings of items, resulted in assigning different ratings for the same item through users with similar interests. In an early study of collaborative filtering by Resnick et al., this problem is successfully solved by a normalization method[2].

When the rating density is low, using the Singular-value decomposition(SVD) rather than the original matrix is to avoid difficulty generating accurate recommendations[3]. Moreover, SVD of a matrix in computations has the advantage of being more robust to numerical error.

For evaluation of recommender system, average RMSE is one of major metrics, because it can adjust for not only balanced but also unbalanced test sets. For example, when there is an unbalanced items distribution in the test set, the RMSE value might be heavily affected by the error on a few very frequent items. However, the approach which is to compute the RMSE for each item and then take the average over all items is representative of the prediction error on any item[4]. Similarly, computing a per-user average RMSE could probably solve not only the prediction error of a randomly selected user but also the test set with an unbalanced user distribution.

3. Implementation

3.1 Data

The original datasets are from a collection collected by the *GroupLens* research group. The first dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies, each user has rated at least 20 movies. The second dataset consists of 1 million ratings (1-5) from 6000 users on 4000 movies. [5]

3.2 Data Processing

In this project, we use the method of collaborative filtering (CF). The CF algorithm systems have many forms, but the common systems that we use have two steps [6]:
1. Look for users who share the same rating patterns with active user (the user whom the prediction is for)
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.
Thus, in our project, in the data processing step, we will build a user-movie matrix from the original dataset.

The user-movie matrix example is like below: (Assume we have m users as row, and n movies as column)

x_{11}	x_{12}	…	
x_{21}			
…			
			x_{mn}

Table 1. the User-Movie Matrix

For example, x_{11} represents rating from user 1 for the movie 1 and so on.

Then, we separate the whole data matrix into four quadrants. We take 80% data from the whole user-matrix to be the training data and 20% data's remaining 20% ratings to be the testing data. First, we calculated the similarity between the training users (Quadrant 1) and testing users (Quadrant 3) for the ratings of 80% movies. Next, we use the similarity that calculated in the previous step and the 20% rating items in training data (Quadrant 2) to make the prediction for the testing users about their ratings for the remaining 20% movies (Quadrant 4).

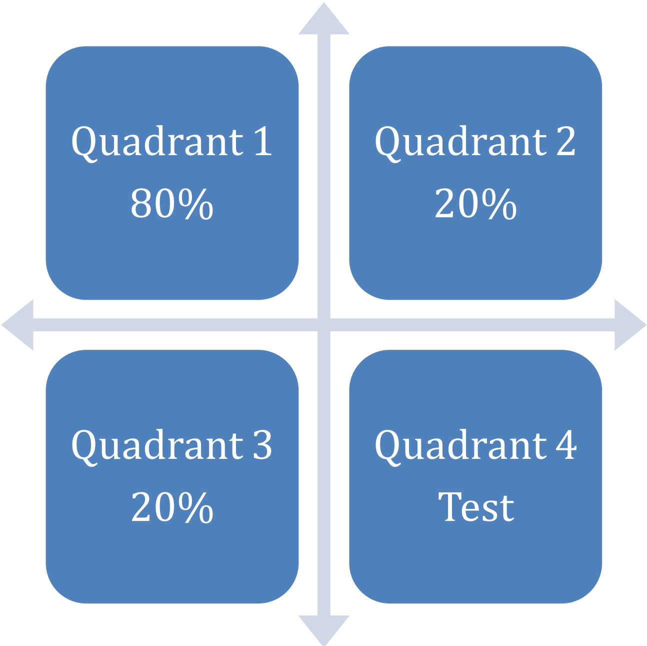

Figure 1. Methods of separating training and testing data

3.3 Memory-Based Collaborative Filtering (CF)

The memory-based CF algorithm used the in our project is based on the similarity between users and users. The value of rating user 'u' gives to item 'i' is calculated as an aggregation of some similar users' rating of the item: [6]

$$r_{u,i} = aggr_{u^{'} \in U} r_{u^{'},i}$$

where 'U' donates the set of top 'N' users that are most similar to user 'u' who rated item 'i'.

The aggregation function we use is:

$$r_{u,i} = k \sum_{u' \in U} simil(u, u')\, r_{u',i}$$

where k is just a normalizing factor and the similarity calculating function we use is the cosine distance, which is:

$$similarity = \cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In order to compare the influence of the different top-N users, we will choose different k values to do the experiments and analyse the corresponding results.

3.4 Normalized Rating Method Using on CF

For the user-movie matrix, we try to use normalized rating method [7] to do the corrections. In this method, we normalize ratings by subtracting from each rating the average rating of that user. By doing this, we will turn low ratings into negative numbers and high ratings into positive number. Thus, when we observe the cosine distance, we could find that users with opposite views of the movies they viewed in common will have opposite directions of vectors. Besides, the users with similar options about the movies rated in common will have a relatively small angle between them.

3.5 Hybrid Recommender System Using SVD

In this project, we also try to solve the problem by using the method of hybrid recommender system, which is the combination of memory-based and model-based. Singular value decomposition is a one way in Hybrid Recommender System.

Suppose M is a m × n matrix, then there exists a factorization, called a singular value decomposition of M, of the form:

$$M = U\Sigma V^{*}M = U\Sigma V^{*},$$

where U is an m × m unitary matrix over K,

Σ is a diagonal m × n matrix with non-negative real numbers on the diagonal,

V is an n × n unitary matrix over K, and V∗ is the conjugate transpose of V. [8]

By using the method of SVD, one advantage is that it solves the sparsity of the original data matrix. We also make comparison with non-SVD results in the experiments.

3.6 Evaluation Method

We use Root-Mean-Square Error to do the evaluation for the results. The formulas are: [9]

$$RMSE(\hat{\theta}) = \sqrt{MSE(\theta)} = \sqrt{E((\hat{\theta} - \theta)^2)}$$

4. Results

By running the program based on top N similar users' preference in the 100-k datasets and 1 million datasets, we can get a group of results by comparing the training data and test data, we measure the results by using RMSE. The graph below shows the results. Here we choose N values for 5, 10, 20, 40, 100 and training user numbers respectively.

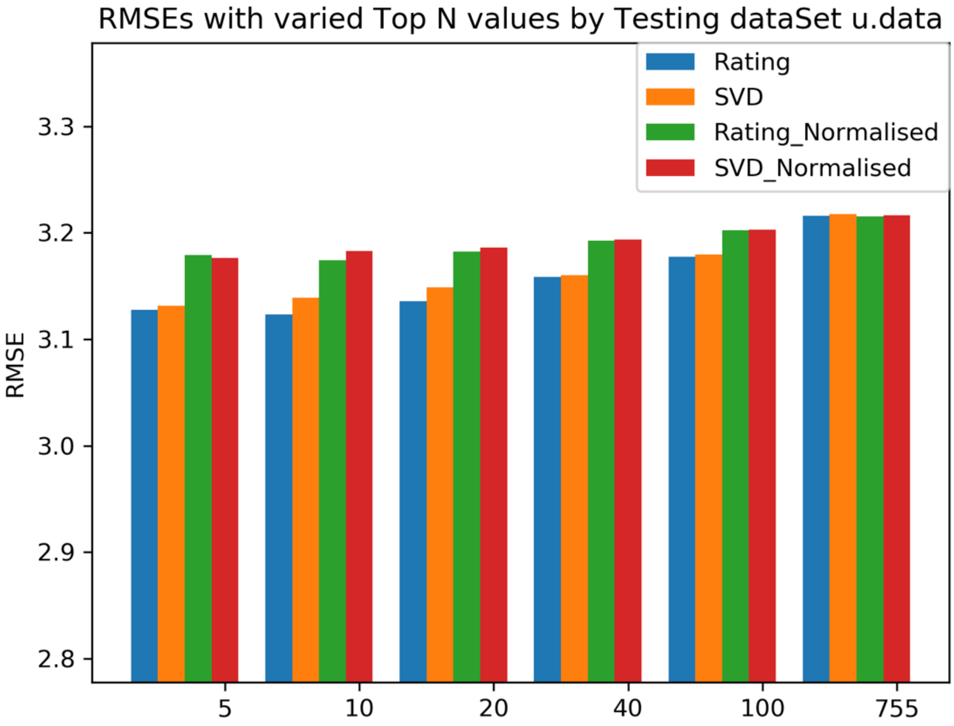

Figure 2. The dataset 100K

N\RMSE	Rating	Rating with SVD	Normalized	Normalized with SVD
5	3.1274	3.1317	3.1790	3.1765
10	3.1236	3.1392	3.1745	3.1830
20	3.1357	3.1489	3.1825	3.1861
50	3.1586	3.1602	3.1925	3.1940
100	3.1776	3.1799	3.2026	3.2027
755	3.2162	3.2177	3.2156	3.2163

Table 2. the Result from different methods with 100k data Set

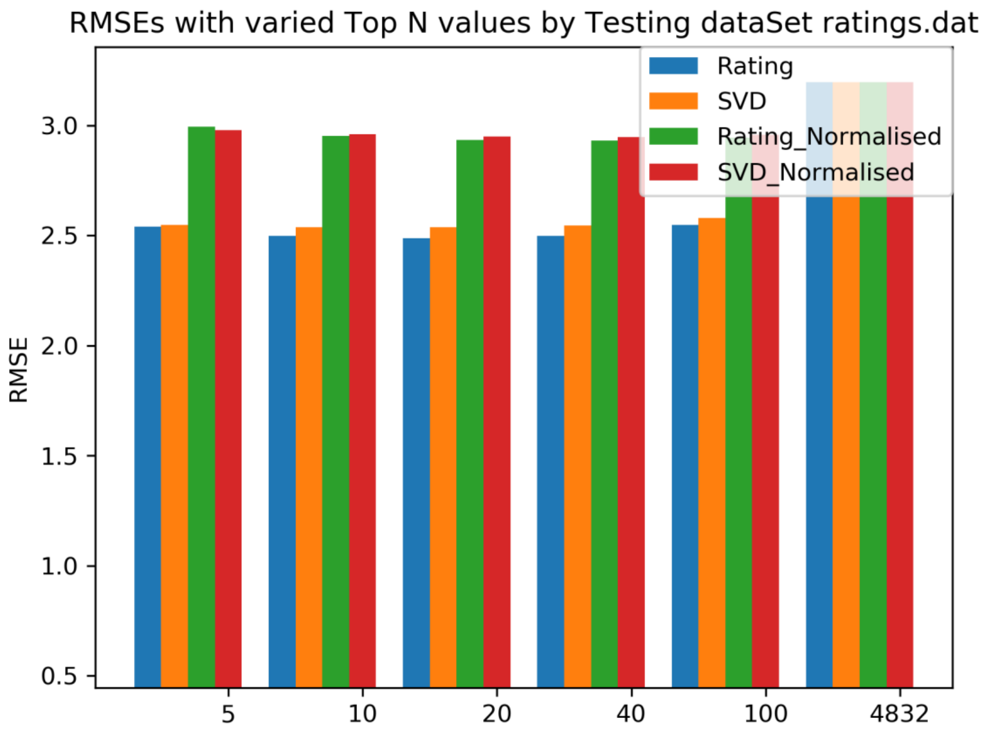

Figure 3. The dataset 1 million

N\RMSE	Rating	Rating with SVD	Normalized	Normalized with SVD
5	2.5401	2.5472	2.9953	2.9783
10	2.4987	2.5384	2.9532	2.9593
20	2.4884	2.5374	2.9351	2.9505
50	2.4990	2.5452	2.9311	2.9474
100	2.5487	2.5809	2.9451	2.9582
755	3.1967	3.1964	3.1959	3.1975

Table 3. the Result from different methods with 1M data Set

5. Discussion

From the bar charts, we will discuss three points.

The first one is that the RMSEs of Rating Values are generally lower than the Normalised Rating results, which mean clustering and prediction by pure rating score are better than normalised correlations in this case. It infers that the data normalising processing in some cases does not always improve the accuracy.

Secondly, when enlarging the data size, the prediction accuracy is significantly increasing. The RMSEs of N from 5 to 100 with 100 thousand record datasets are around 3.0, while the RMSEs with same parameters but 1 million records are round 2.5. A possible reason is that for some prediction in the smaller data set, the algorithm cannot find a high relevant record as the user-based data, so the prediction is according to other low similarity users and leads to the high bias. However, by given more data, the probability of this situation is getting lower, and the predicted values are much closer to the real values.

The third finding is that SVD method reduces the RMSE when N is 5 as well as more other cases when we change the percentage of training and testing. To some extent, we can conclude that the SVD can increase the predicting accuracy.

6. Conclusion

From the results, it can be concluded that based on the method of CF algorithm, comparing with the normalized rating method, SVD could lead to increase the accuracy on this task. We also find that with the increasing of datasets, the CF algorithm could performance better. Due to the time and equipment limit, we think this algorithm could still be improved by testing larger datasets and apply the normalized method in different datasets in the future study.

7. References

[1] X. Su, T. M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques, Hindawi Publishing Corporation Advances in Artificial Intelligence Volume(2009) 19.

[2] R. Jin, L. Si, A Study of Methods for Normalizing User Ratings in Collaborative Filtering, accessed 28 May 2018, <https://www.cs.purdue.edu/homes/lsi/sigir04-cf-norm.pdf>.

[3] G. Shani, A. Gunawardana, Evaluating Recommendation Systems, Microsoft Research(2010).

[4] M. Polato, F. Aiolli, Exploiting sparsity to build efficient kernel based collaborative filtering for top-N item recommendation, Neurocomputing 268 (2017) 17–26.

[5] MovieLens 100K, 1M Datasets(2018), accessed 28 May 2018, <https://grouplens.org/datasets/movielens/100k/>.

[6] Collaborative filtering(2018), accessed 28 May 2018, <https://en.wikipedia.org/wiki/Collaborative_filtering>.

[7] Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman and Jeffrey D.Ullman, pg323-324.

[8] Singular-value_decomposition(2018), accessed 28 May 2018, <https://en.wikipedia.org/wiki/Singular-value_decomposition>.

[9] Root-mean-square_deviation, Mean_absolute_error (2018), accessed 28 May, <https://en.wikipedia.org/wiki/Root-mean-square_deviation>, <https://en.wikipedia.org/wiki/Mean_absolute_error>.