Movie Collaborative Filtering System Based on Low-rank Matrix Factorization.

Zhengtian Zhu

Institute of Statistics & Big Data
Renmin University of China

Introduction

- When you buy a product online, most websites automatically recommend other products that you may like. **Recommender systems** look at patterns of activities between different users and different products to produce these recommendations.
- **Collaborative filtering systems**, one of the recommendation systems, are used to predict future ratings of items by users based on their past ratings.
- We implement and compare two kinds of recommender systems, item-based collaborative filtering (IBCF) system and user-based collaborative filtering (UBCF) system.

MovieLens Data

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. This data set consists of:

- $100,000$ ratings from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (age, gender, occupation, zip)

The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set.

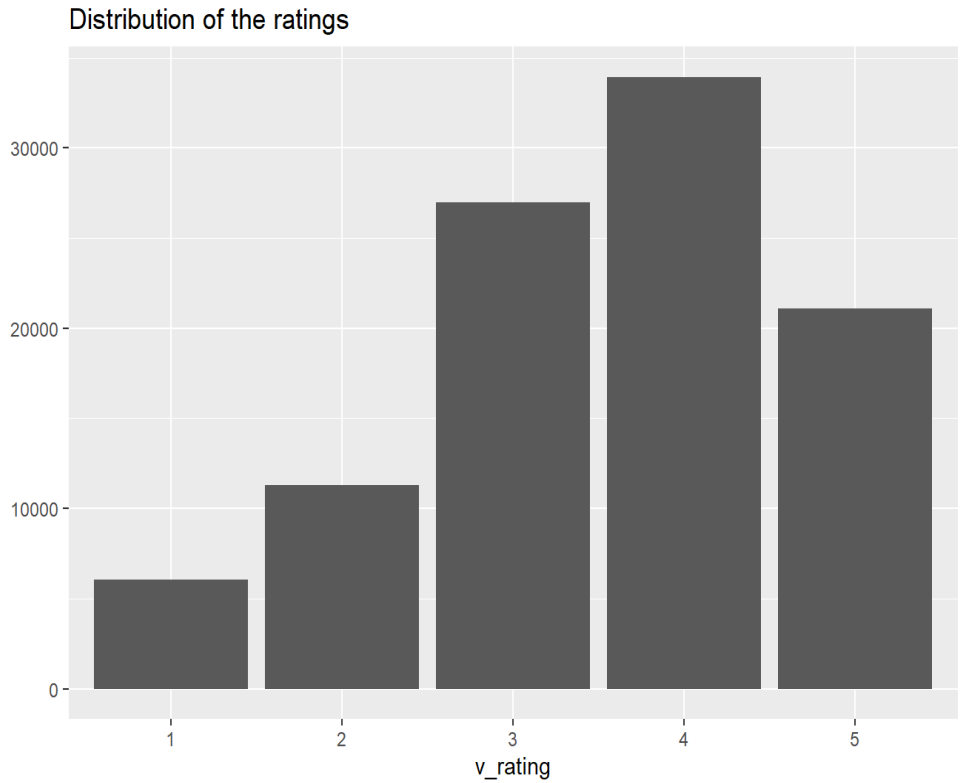
The data can be downloaded at
`https://grouplens.org/datasets/movielens/`

EDA

The data includes 100004 ratings, each rating has 7 features:

- The 1st feature is movieID which ranges from 1 to 1682.
- The 2nd feature is the title of movie. Notice that the title may not be unique.
- The 3rd feature is the time of first roadshow of a movie.
- The 4th feature identifies the genre(s) of a movie. Notice that some genres may have overlapping. This kind of classification is not very explicit in this dataset.
- The 5th feature is the userID which ranges from 1 to 943. The data is arranges by the order of userID originally.
- The 6th feature is the rating which ranges from 0.5 to 5 and has interval 0.5.
- The last feature is the timestamp of a user rating a movie.

EDA

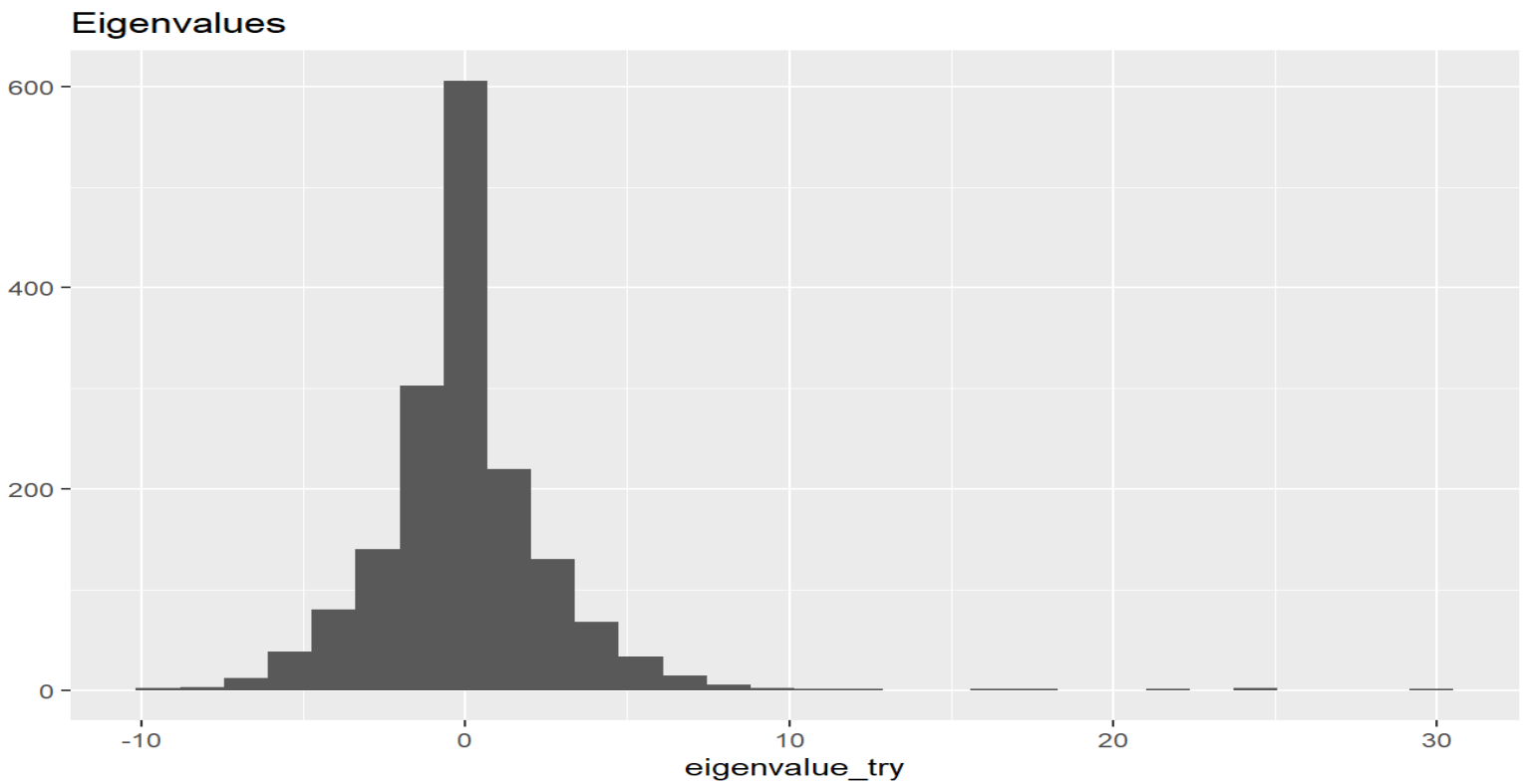


We then plot the histogram of the ratings. From the plot, we can see most films have rating 3 and 4. Low ratings films $(1-2)$ are less than high rating films $(3-5)$. The distribution of the rating seems normal and reasonable.

Sparsity Level

- The data set is converted into a user-item matrix A that had 943 rows (i.e., 943 users) and 1682 columns (i.e., 1682 movies that were rated by at least one of the users).
- For our project, we take sparsity level of data sets into consideration. For a data matrix R This is defined as $1 - \frac{\text{nonzero entries}}{\text{total entries}}$. The sparsity level of the Movie data set is, therefore,$1 - \frac{100{,}000}{943 \times 1682}$, which is 0.9369.
- We decide to use collaborative filtering based system because of high sparsity level. We consider two kinds of system, item-based system and user-based system.

EDA



We check the correlation matrix of film by using all the data. From the plot, we know that the matrix has many 0 eigenvalues. The data is highly sparsed.

Overview of the Collaborative Filtering Process

- The goal of a collaborative filtering algorithm is to suggest new items or to predict the utility of a certain item for a particular user based on the user's previous likings and the opinions of other like-minded users.
- In a typical CF scenario, there is a list of m users $U = \{u_1, u_2, \ldots, u_m\}$ and a list of n items $I = \{i_1, i_2, \ldots, i_n\}$ Each user u_i has a list of items I_{ui} , which the user has expressed his/her opinions about. Opinions can be explicitly given by the user as a rating score.
- Recommendation is a list of N items, $I_r \subset I$, that the active user will like the most. Note that the recommended list must be on items not already purchased by the active user, i.e.,This interface of CF algorithms is also known as **Top-N** recommendation.

User-Based Collaborative Filtering System

User-based collaborative filtering system utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as **neighbors**, that have a history of agreeing with the target user (i.e., they either rate different items similarly or they tend to buy similar set of items). Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or top-N recommendation for the active user. The techniques, also known as nearest-neighbor or **memory-based** collaborative filtering, are more popular and widely used in practice.

An Example of User-User Similarity Matrix

Here is a table of user's rating of item.

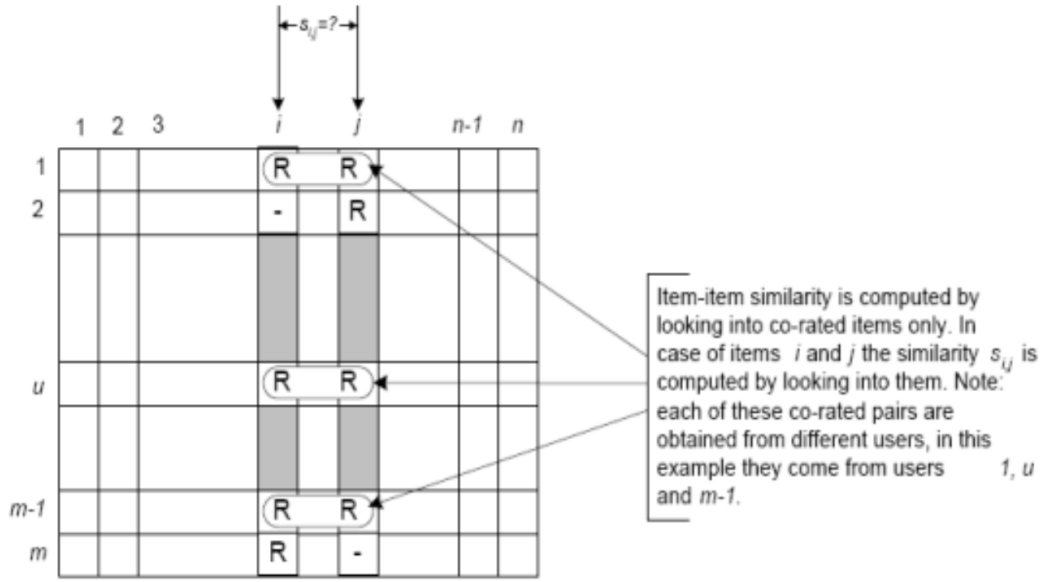
User or Item	A	B	C	D	E	F	G
1	4	-	-	-	3	2	4
2	3	3	2	1	2	-	-
3	-	5	4	3	4	5	3

The user-user similarity matrix is

$$\begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

Item-Based Collaborative Filtering System

Item-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. The techniques are also known as or **model-based** collaborative filtering.



Training the Model

We split the data into training set and test set. 80% of the data are training set, while 20% are test set. After spliting the data, we mainly train four models:

- Item-based collaborative filtering system using cosine-based similarity;
- Item-based collaborative filtering system using correlation-based similarity;
- User-based collaborative filtering system using cosine-based similarity;
- User-based collaborative filtering system using correlation-based similarity.

The cosine-based similarity is

$$sim(i,j) = \cos(i,j) = \frac{i \cdot j}{\|i\|_2 * \|j\|_2}.$$

The correlation-based similarity is

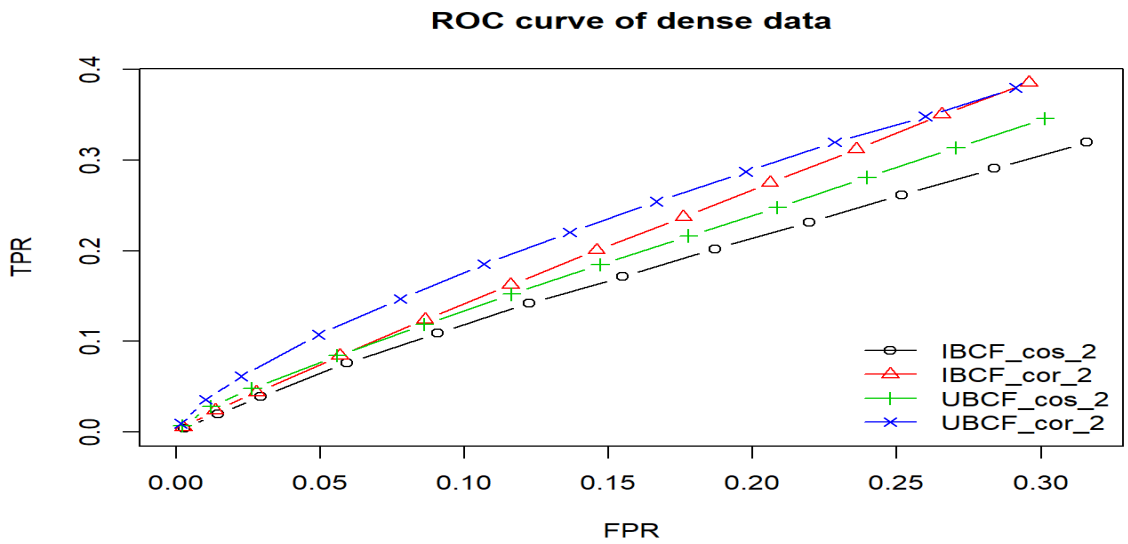
$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}}.$$

Result

```
 [1,] "Toy Story (1995)"
 [2,] "GoldenEye (1995)"
 [3,] "Get Shorty (1995)"
 [4,] "Twelve Monkeys (1995)"
 [5,] "Babe (1995)"
 [6,] "Dead Man Walking (1995)"
 [7,] "Seven (Se7en) (1995)"
 [8,] "Usual Suspects, The (1995)"
 [9,] "Mighty Aphrodite (1995)"
10,] "Postino, Il (1994)"
11,] "Mr. Holland's Opus (1995)"
12,] "Braveheart (1995)"
13,] "Taxi Driver (1976)"
14,] "Rumble in the Bronx (1995)"
15,] "Birdcage, The (1996)"
16,] "Apollo 13 (1995)"
17,] "Batman Forever (1995)"
```

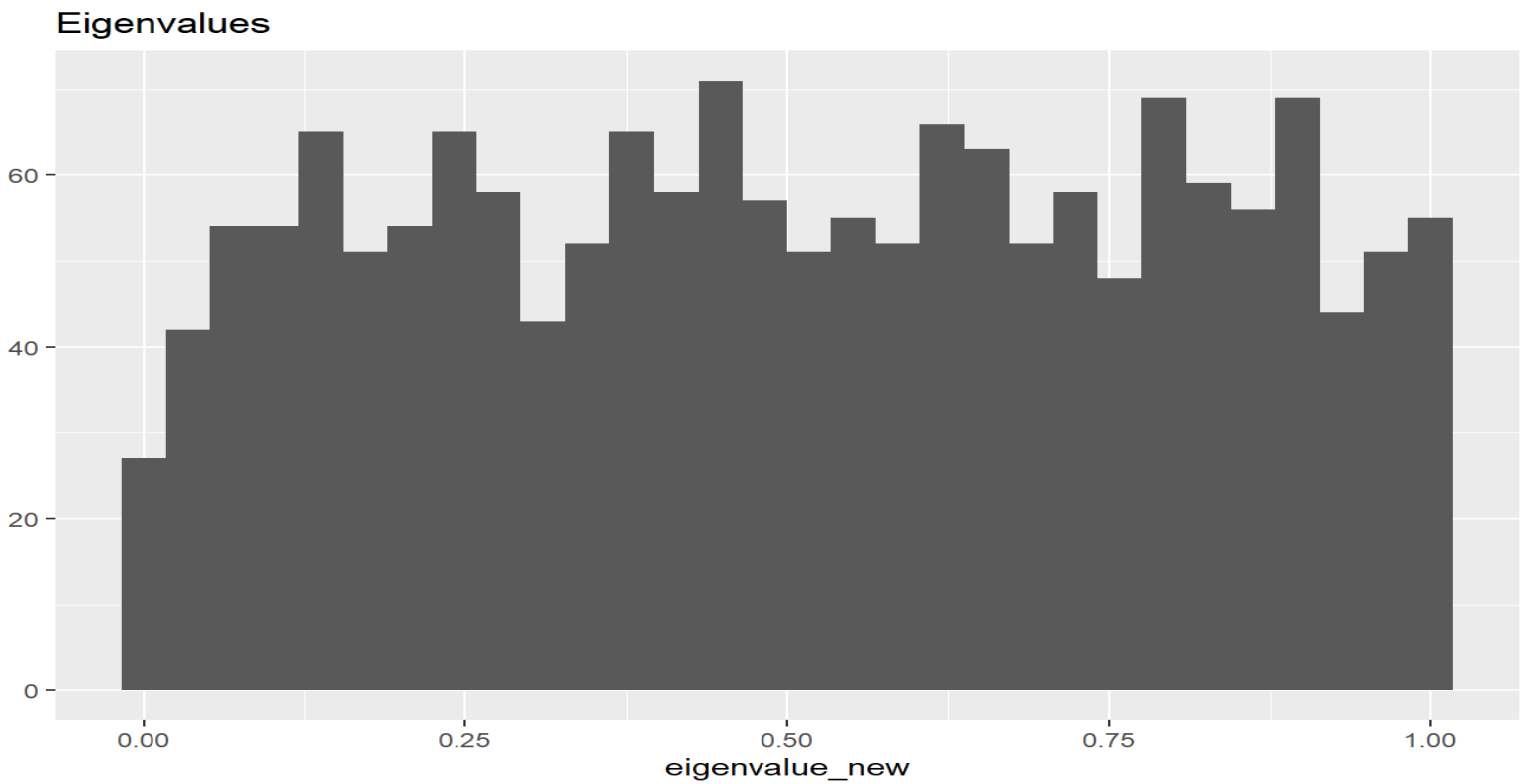
Here is the recommendation result of the first user. He or she might be more interested in action movie and science fiction movie.

ROC curve



We then plot the ROC curve of the four models. We can see the user-based collaborative filtering system using correlation-based similarity performs best. All models have a better performance than random recommendation.

Eigenvalue



We check the adjusted user-user similarity matrix. We can see that all the eigenvalues are nonnegative.

Conclusion

- When using the same similarity function, user-based system performs better than item-based system.
- When using the same system, correlation-based similarity function performs better than cosine-based similarity function.
- The precision of the system is relatively low. None of the implemented system is truly outperformed.
- For larger and more complicated dataset, how to implement fast and precise recommendation system is a challenage.