CSE272 HW2

Jinghao Shen

May 22, 2022

1 Introduction

This homework works on recommendation system. The data set used in this homework is the 'Video Games' 5-core subset from Amazon product data. The raw data is split into training data and testing data for each user. Then, algorithms are used for rating prediction, i.e., to predict the ratings in the testing set as if we didn't know them. The predicted result is then evaluated by the MAE and RMSE.

Github: `https://github.com/jshen1s1/CSE272/tree/main/HW2`

2 Software Design

The program is written in Python. To achieve the basic functions mainly two libraries, sklearn and scipy, are used. Scipy is an open source python package that enables us to build sparse matrix. Sklearn is a free software machine learning library that features various evaluation, classification, regression and clustering algorithms.

2.1 Data process

The data from .json.gz file is stored into 4 lists, user_id, item_id, rating, and all_data, respectively. User_id and item_id are then processed to remove the duplicates, producing a unique list of user ID and item ID. All_data is then modified responding to the newly assigned index of each user. The data set contains 24303 unique users, 10672 unique items, and 231780 ratings in total. Each user and his/her reviews are group and randomly split into training data and testing data in ratio of 0.8:0.2. In this case, we have 185,424 training data and 46,356 testing data.

all_data is structured as {{userID, itemID, rating}}.

2.2 Sparse matrix and similarity matrix

To perform collaborative filtering, we need to build the user-item matrix for both training data and testing data and a similarity matrix based on the user-item matrix. We used 'csr_matrix' function from scipy to create a user-item matrix which returns a sparse matrix where the items users did not rate are score to 0. Then, with the created user-item matrix, we can calculate the similarity and create a similarity matrix. 'cosine_similarity' function from sklearn is used to calculate the cosine similarity.
User-based Similarity: measured by observing all the items that are rated by both users.
Item-based Similarity: measured by observing all the users who have rated both items.

2.3 User-based CF

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. After we have created the user-item matrix and users similarity matrix, we can make the prediction by applying the formula:

$$r_{u,p} = \overline{r_u} + \frac{\sum_{i \in users} sim(u,i) \cdot (r_{i,p} - \overline{r_i})}{\sum_{i \in users} |sim(u,i)|}$$

2.4 Item-based CF

Item-Based Collaborative Filtering explore the relationship between the pair of items. We find the missing rating with the help of the ratings given to the other items by the user. After we have created the user-item matrix and item similarity matrix, we can make the prediction by applying the formula:

$$r_{u,p} = \frac{\sum_{j \in items} r_{u,p} \cdot (sim(p,j))}{\sum_{j \in items} |sim(p,j)|}$$

3 Results

We evaluate our predictions on the testing data by MAE and RMSE. Be referring different measures and scores of 4 ranking algorithm, we can find and infer that:

- By looking at Figure 1, We found that user-based and item-based CF have close MAE and RMSE scores. It is as expected as they are both based on the similarity between users or items.
- The Figure 1 also shows the scores of MAE and RMSE are slightly above 4.0. This may cause by that model-based CF methods will suffer when users or items that don't have any or only small number of ratings. The data set contains a large amounts of reviews and each user is only a tiny

portion in it. Thus, the similarity between users or items is relevantly low. It is one of the drawbacks of memory-based CF when scaling real-world scenarios.

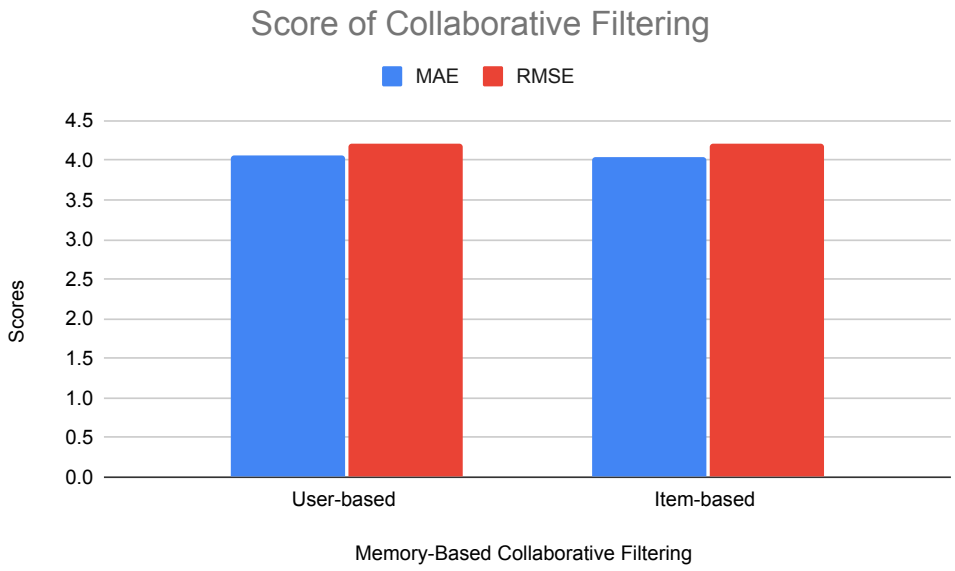

Figure 1: MAE RMSE for Collaborative Filtering