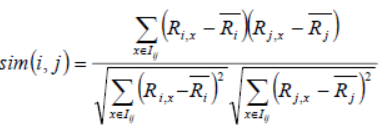
1. paradoxical
2. rank 320 & score 0.81

3. Approach

I tried both CF and content-based algorithm

User-Based Collaborative filtering

The main idea of ​​this type of algorithm is to use the scoring matrix of <user, item>, and use statistics to calculate the similarity between users and users, and between items and items. Then use similarity ranking, and finally get the recommendation result. The equation shows below:



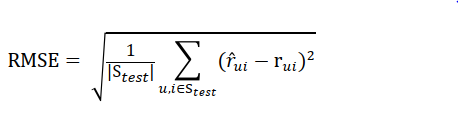
The formula is to calculate the similarity between user i and user j, I (ij) is the item that user i and user j have jointly evaluated, R (i, x) is the rating of user i on item x, R (i ) There is a bar on the head that represents the average score of all the ratings of user i. The reason why the average score is subtracted is because some users score strictly and loosely. Normalized user ratings avoid mutual influence.

Content-based recommendation

Because the collaborative filtering recommendation algorithm only recommends based on the user's rating of the product, there may be a problem of cold start. If you can make a more intuitive recommendation based on the characteristics of the item and the user's special preferences and other characteristic attributes, this cold start can be solved. The problem. Although the CB algorithm requires additional information that depends on items and user preferences, it does not require too many user ratings or group records, that is, only one user can complete the recommendation function and generate an item recommendation list. Three steps are item representation, profile learning and recommendation generation.

1. Implementation

The additional datasets are original from hetrec2011-movielens-2k, which is a subset and extension of MovieLens10M dataset. Movies are rated with score 1-5. The recommendation system will then suggest the highest rated movies from that algorithm. Prediction accuracy of the classifier is calculated and compared with the Root-Mean-Square Error of the assigned ratings compared to ratings in a test set. Root-Mean-Square Error (RMSE) was used for evaluation and it is a measure of the algorithm’s error in the predictions of users’ ratings of movies. The equation shows as follows:



User-Based Collaborative filtering

Dataset was loaded using python pandas into a matrix represented as dataframes, build a Trainset object to fit into the prediction algorithm of SVD. Predict ratings for each user and associated movie.

Content-based recommendation

Determine similarity between two movies by genres, director, actors, and tags, gather information from the movies\_ data files, these four files work on the implementation but I didn’t try others, then calculate the coefficient to see if this movie is more similar than current most similar movies.

The example predictor showed an RMSE = 1.52. The best achieved result with the approach described in this paper was RMSE = 0.81, which is an improvement of 46.7% compared with example. Running the program took 865 seconds on a ubuntu virtual machine with 4GB RAM 