Recommender system using collaborative filtering

COMP9417, Assignment 2

**Introduction**

Recommender system is a very popular and practical model, which can help people to find the most likely movie for themselves. Collaborative filtering produces recommendations based on the knowledge of users’ attitude to items, that is it uses the “wisdom of the crowd” to recommend items.

During the processing of this project our group will use 20M and 100K MovieLens Data Sets to train and test the algorithm of collaborative filtering, which includes content-based and model-based.

**Related work**

Before starting this project, our group discusses the topic and search some relative materials to find that movie recommender system mainly focuses on the algorithm of KNN, SVD and deep learning.

20M MovieLens is so large that we have to try different ways to reduce the size of data because of memory limit.

Model-based collaborative filtering is based on matrix factorization, which means that we assume that there are some latent features for users and movies. Through this method we can get low-rank matrices. Then multiplying these low-rank matrices together can generate a new matrix which fills the missing rating value of original matrix.

**Method**

Model-based collaborative filtering:

Step 1: generating training sample

Firstly, use dataframe.sample() to select some small parts from rating.csv file of 20M data and save them as sample\_\*.csv file;

Step 2: generating a pivot table

Read each sample\_\*.csv to get a dataframe in memory and only choose the columns of userId, movieId and rating as rating\_matrix;

Use pivot\_table function of pandas to convert the rating\_matrix into a new matrix(pivot\_matrix), which uses userId as index, movieId as column and rating as value. In order to increase the precision of training the pivot\_matrix should be normalized by minus each userId’s mean for their evaluation of movies to get a normalized pivot\_matrix;

Step 3: matrix factorization

For the normalized pivot\_table it can be decomposed by a module called svds from scipy.sparse.linalg to generate three low-rank matrices, respectively user-feature matrix, weights matrix, and movie-feature matrix;

During the processing of decomposing the latent feature number can be set as 20 and weights matrix should be converted into the diagonal form, such as the format:



where x is the number of latent feature, m is the total numbers of users and n is the total numbers of movies

Step 4: generating a predicted matrix

Multiply the three low-rank matrices to get a new matrix and then add the each user’s mean of rating for movies to get the predicted matrix;

To make the predicted matrix same with the pivot\_matrix, add the index and column name into the predicted matrix in pandas function;

Step 5: testing the predicted matrix in RMSE

Now there are two matrices, one is original pivot\_matrix losing some ratin value and another is predicted matrix filling the missing rating value;

Choose the users and movies which have corresponding rating value in original pivot\_matrix;

Compare the actual rating value with predicted rating value between the two matrix,so we use RMSE as the evaluation standard, such as that:



where i and j are the index of user and movie, N is the total numbers of actual rating value.

Then each sample file has a value of RMSE and get a average error in all samples.

**Results**

Model-based collaborative filtering:

sample\_0.csv has 0.1% of the rating.csv and RMSE is 3.551560927145061;

sample\_1.csv has 0.2% of the rating.csv and RMSE is 3.5476378896337515;

sample\_2.csv has 0.3% of the rating.csv and RMSE is 3.542448714317859;

**Discussion**

**Conclusions**

**References**

James L 2018, The 4 Recommendation Engines That Can Predict Your Movie Tastes, accessed 17 July 2019,<https://medium.com/@james\_aka\_yale/the-4-recommendation-

engines-that-can-predict-your-movie-tastes-bbec857b8223>

Jae DS 2018, My Notes for Singular Value Decomposition with Interactive Code (feat Peter Mills), accessed 16 July 2019, <https://towardsdatascience.com/my-notes-for-singular-value-

decomposition-with-interactive-code-feat-peter-mills-7584f4f2930a>