

Nearest Neighbour Collaborative Filtering

The experiments performed are an evaluation of user-based collaborative filtering and item-based collaborative filtering for varying neighbourhood sizes (with raw ratings and with mean normalized ratings), and an implementation of a user-based collaborative filtering system using raw ratings as well as Bayesian-normalized ratings. The code for the comparison between the item-based and user-based approach was originally authored by Sachit Tandukar (<https://github.com/sachittandukar>) and sourced from this report: https://www.academia.edu/48879942/Comparison_of_user_based_and_item_based_collaborative_filtering

The improved method is based on Herlocker et Al's 1999 paper and has been implemented in the same code with slight changes.

The concept of Bayesian normalization is borrowed from Jill Cates, from this tutorial: <https://github.com/topspinj/recommender-tutorial/blob/master/part-1-item-item-recommender.ipynb>

Files: Recommender_Evaluation_Original and Recommender_Evaluation_Improved

Dependencies and their installation:

scikit learn - pip install scikit-learn
Numpy - pip install numpy
Pandas - pip install pandas
scipy - pip install scipy

Differences in Implementation:

The data is downloaded from the Grouplens website and ratings are extracted to perform experiments. Recommender_Evaluation_Original.ipynb uses unnormalised/raw ratings to perform evaluation using cosine similarity and pearson correlation as the similarity metrics. Recommender_Evaluation_Improved.ipynb has an additional computation of mean user rating and mean movie rating which is used to adjust all the ratings (including zeroes). The other difference in implementation is the addition of the mean user rating to the rating prediction function in user based CF and adding the mean movie rating to the weighted average (rating prediction function) in item based CF.

Files: User-User based Collaborative Filtering Movie Recommender_without_normalisation.ipynb and User-User based Collaborative Filtering Movie Recommender_bayesian_normalisation.ipynb

Dependencies and how to install them:

scikit learn - pip install scikit-learn

Numpy - pip install numpy

Pandas - pip install pandas

scipy - pip install scipy

Differences in implementation:

The code builds on the majority borrowed from the beginner-friendly implementation by Rohith Kumar Poshala (<https://github.com/rposhala/Recommender-System-on-MovieLens-dataset>). However, the Bayesian normalization and its implementation in coding was inspired by Jill Cates' implementation of item-item recommenders in this tutorial - <https://github.com/topspinj/recommender-tutorial/blob/master/part-1-item-item-recommender.ipynb>

User based recommenders work on the same principle of weighted average of ratings given by k nearest neighbours of a given user. The similarity computation works based on a distance metric such as cosine or euclidean. The scikit library's unsupervised knn method `sklearn.neighbors.NearestNeighbors` has been used to compute the cosine distances between movies by passing a sparse matrix, ie, our user-item matrix which contains user ratings for all movies.

Given any input user, it computes the nearest neighbours of that user in the 943-dimensional space. Then the distances are used as weights for each of the neighbouring users to sort the top recommended in descending order. A post processing step removes the movies taht have already been watched by the user.

For the input user 5, movies recommended (using raw ratings) are:

```
'Rear Window (1954)',  
'Raising Arizona (1987)',  
'Brazil (1985)',  
'Contact (1997)',  
'Pulp Fiction (1994)',  
'Amadeus (1984)',
```

```
'Godfather, The (1972)',  
'Twelve Monkeys (1995)',  
'Babe (1995)',  
'Terminator, The (1984)'
```

For the input user 5, movies recommended (using bayesian normalised ratings) are:

```
'Pulp Fiction (1994)',  
'Contact (1997)',  
'Twelve Monkeys (1995)',  
'Terminator, The (1984)',  
'Seven (Se7en) (1995)',  
'Schindler's List (1993)',  
'Scream (1996)',  
'Braveheart (1995)',  
'Amadeus (1984)',  
'Babe (1995)'
```

The order of recommended movies is rearranged and some movies are replaced after normalization of ratings.