Non-negative Matrix Factorization

YoungWoong Cho

August 2020

In this project, non-negative matrix factorization is used to train the recommendation algorithm.

Dataset : Movie Recommendation Data Sets

Source : http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

0.1 Preliminary works

1. Import libraries

```
[ ]: pip install scikit_surprise
```

```
[ ]: import pandas as pd
     from surprise import NMF, Dataset, Reader
     from surprise.model_selection import train_test_split, cross_validate
```

2. Get a dataset

```
[ ]: from google.colab import files
     data_to_load = files.upload()
```

```
<IPython.core.display.HTML object>


Saving ratings.csv to ratings.csv
```

```
[ ]: df = pd.read_csv('ratings.csv')
     del df['timestamp']
     df.head()
```

```
[ ]:    userId  movieId  rating
     0       1        1     4.0
     1       1        3     4.0
     2       1        6     4.0
     3       1       47     5.0
     4       1       50     5.0
```

0.2 Non-negative Matrix Factorization algorithm

```
[ ]: reader = Reader(rating_scale=(1.0, 5.0))
     data = Dataset.load_from_df(df, reader)
```

```
[ ]: # Fit the algorithm to the full dataset
     train = data.build_full_trainset()

     algo = NMF()
     algo.fit(train)
```

```
[ ]: <surprise.prediction_algorithms.matrix_factorization.NMF at 0x7f74c8cf6518>
```

```
[ ]: #Run 5 fold CV
     cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm NMF on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9240  0.9248  0.9175  0.9235  0.9097  0.9199  0.0057
MAE (testset)     0.7094  0.7083  0.7031  0.7074  0.6970  0.7050  0.0045
Fit time          6.29    6.54    6.67    6.62    6.30    6.48    0.16
Test time         0.23    0.26    0.11    0.11    0.25    0.19    0.07
```

```
[ ]: {'fit_time': (6.285089015960693,
       6.536697149276733,
       6.674468755722046,
       6.62369966506958,
       6.298094272613525),
      'test_mae': array([0.70939679, 0.70830663, 0.70307781, 0.7074144 ,
     0.69702027]),
      'test_rmse': array([0.92404159, 0.92479641, 0.91750888, 0.92347584,
     0.90965239]),
      'test_time': (0.2251298427581787,
       0.2599964141845703,
       0.11418485641479492,
       0.11135745048522949,
       0.24715781211853027)}
```

```
[ ]: # Recommendation for a user or two as testing

     pred_1 = algo.predict(378, 78499, r_ui=4, verbose=True)
     pred_2 = algo.predict(453, 2324, r_ui=5, verbose=True)
```

```
user: 378        item: 78499      r_ui = 4.00   est = 3.75   {'was_impossible':
False}
user: 453        item: 2324       r_ui = 5.00   est = 4.19   {'was_impossible':
False}
```