# 데이터 사이언스

Programming Assignment (Long term project): Recommender

**학생**: 최윤석

**학번**: 2015005169

**작성일자**: 2022.05.22

**교수님**: 김상욱 교수님

## # 실행 환경

- Window
- Python 3.10.1
- numpy 1.22.3
- pandas 1.4.2

## # Summary of my program ( +algorithm)

► `if __name__ == "main"`:

프로그램을 실행할 때 git bash를 사용하여

`python recommender.py u1.base u1.test` 와 같은 형태로 실행하였다.

실행 파일일 경우 argparse를 사용하여 train_filename과 test_filename을 받을 수 있도록 하였다.

그 후 이 argument들을 main함수에 넘겨준다.

► `def main(train_file_name, test_file_name)`:

넘겨받은 파일 이름을 read_file함수에 넘겨준다.

read_file로부터 rating matrix(행이 user이고 열이 movie이며 각 요소는 rating인 행렬)과

test_data(test_file에서 rating과 timestamp를 제외하고 사용자 명과 영화 이름만 있는 데이터)를 받아

학습을 진행할 준비를 한다.

rating matrix에 Matrix Factorization을 적용시키기 위해 MF에 rating matrix를 넘겨주고

train을 하고, test data에 대한 예측을 수행한다.

예측이 끝나면 output file 형식에 맞게 결과를 출력해준다.

(output file 이름은 training_data file이름에 _prediction.txt를 붙여 저장한다)

► def read_file(train_file_name, test_file_name):

train_file_name과 test_file_name을 pandas를 통해 data frame형태로 변형한다.

그 후 사용하지 않는 time_stamp정보를 제거하고

현재 사용자 id, 영화 id, rating 으로 되어있는 데이터를

행이 사용자, 열이 영화 그리고 데이터는 rating이 될 수 있게

data frame의 pivot_table함수를 사용하여 변형시킨다.

이 때 비어 있는 값은 0으로 채운다.

► class MF

▹ def __init__(self, rating_matrix):

rating_array, dim_of_latent(matrix factorization에서 MxN을 Mxf Nxf로 나눌 때 사용하는 f값)

epochs, learning_rate(bias를 업데이트 할 때 얼마나 업데이트 할 것인지), reg_param, num_of_users, num_of_movies등 값을 초기화한다.

이어서 matrix를 쪼개기 위해 np.random.normal을 사용하여

user_latent_matrix와 movie_latent_matrix를 생성한다.

학습에 사용할 bias들도 초기화 해주고 전체 평균은 np.mean을 통해 구한다.

마지막으로 test data에 학습에서 나온 적이 없던 movie_id나 user_id가 나오는 경우를 대비하여

user_id와 movie_id를 dictionary형태로 저장해둔다.

▹ def train(self):

training을 진행하는 함수이다.

__init__에서 설정한 epoch만큼 학습을 반복하며 rating matrix에 대해 optimize를 수행한다.

▹ def optimize(self, i, j):

rating array에 저장되어 있는 실제 값과, user_latent_matrix와 movie_latent_matrix를 내적해서 구한 예측 값의 차이를 줄이는데 이 때 영화와 사용자에 따라 다른 편차를 조정하기 위해 bias도 함께 계산한다.

즉 실제값 – laten matrix로 계산된 값 – 사용자 편차 – 영화 편차 – 평균 이 0에 가까워지게 학습을 진행하는 것이다.

이 차이를 error라고 하며

error가 구해졌다면 regularization parameter와 편향값 을 곱한 걸 err에서 빼 주고 이 값을 bais를 업데이트 하는데 사용하는데 얼마나 반영할 것인지는 learning_rate에 의해 결정된다.

(이 프로그램에서 사용되는 hyperparameter값들은 여러 시행을 통해 제일 나은 결과가 나온 parameter 값들로 세팅되어있다.)

▹ `def test(self, test_data):`

test_data에서 user_id와 movie_id를 따로 list형태로 저장한다.

그 후 bias들과 latent matrix를 사용하여 calculated_rate을 구한다.

test_user_id와 test_movie_id에 들어있는 user_id와 movie_id를 순회하며

만약 사용자 – 영화 둘 다 학습 데이터에 없던 정보라면 평균 bias값을,

사용자만 없는 데이터일 경우 평균 bias + 해당 영화의 bias값을

영화만 없는 데이터일 경우 평균 bias + 해당 사용자의 bias값을 결과에 넣어준다.

사용자와 영화 모두 알고 있는 정보일 경우

calculated_rates에 저장되어 있는 값을 result_rate_list에 넣어준다.

## # 실행 결과

- u1.base u1.test

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 /d/programming/Github/tmp
$ PA4.exe u1
the number of ratings that didn't be predicted: 0
the number of ratings that were unproperly predicted [ex. >=10, <0, NaN, or format errors]: 0
If the counted number is large, please check your codes again.

The bigger value means that the ratings are predicted more incorrectly
RMSE: 1.054572
```

- u2.base u2.test

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 /d/programming/Github/tmp
$ PA4.exe u2
the number of ratings that didn't be predicted: 0
the number of ratings that were unproperly predicted [ex. >=10, <0, NaN, or format errors]: 0
If the counted number is large, please check your codes again.

The bigger value means that the ratings are predicted more incorrectly
RMSE: 1.024049
```

- u3.base u3.test

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 /d/programming/Github/tmp
$ PA4.exe u3
the number of ratings that didn't be predicted: 0
the number of ratings that were unproperly predicted [ex. >=10, <0, NaN, or format errors]: 0
If the counted number is large, please check your codes again.

The bigger value means that the ratings are predicted more incorrectly
RMSE: 1.023621
```

- u4.base u4.test

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 /d/programming/Github/tmp
$ PA4.exe u4
the number of ratings that didn't be predicted: 0
the number of ratings that were unproperly predicted [ex. >=10, <0, NaN, or format errors]: 0
If the counted number is large, please check your codes again.

The bigger value means that the ratings are predicted more incorrectly
RMSE: 1.011035
```

- u5.base u5.test

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 /d/programming/Github/tmp
$ PA4.exe u5
the number of ratings that didn't be predicted: 0
the number of ratings that were unproperly predicted [ex. >=10, <0, NaN, or format errors]: 0
If the counted number is large, please check your codes again.

The bigger value means that the ratings are predicted more incorrectly
RMSE: 1.036047
```

## # 주의 사항

- 실행 전 numpy와 pandas를 설치해주세요

- recommender.py가 있는 폴더에 train 파일과 test파일이 같이 위치해야 합니다.

- *_prediction.txt파일은 recommender.py가 있는 폴더에 생성됩니다.