Example Codes Guide

We offer two recommendation examples, one is Collaborative Filtering via Matrix Factorization method, another is content-based recommendation.

Requirements

- Python (>= 2.7)
- scikit-learn 0.20.2
- NumPy (>= 1.8.2)
- Pandas 0.23.0
- Matplotlib 2.2.3
- Seaborn 0.9.0

Directory Structure

This code assume the dataset (active1000), and code are not in the same folder. You can put the code in another folder like below.

- active1000/
- script
  -- project_example.py
  -- ExplicitMF.py

Exemple Usage

To run the main method you should type the following commands under *script* folder in a terminal:

```
python project_example.py
```

And you will get information like this:

```
>>> ../active1000
Basic statistics of the dataset...
Total number of events(front page incl.): 2207608
Total number of events(without front page): 788931
Total number of documents: 20344
Sparsity: 3.878%
project_example.py:67: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

which means you already run the code successfully. The code will dynamically run content-based RS and ExplicitMF for recommendations.

Recommendation with Collaborative Filtering (Matrix Factorization) based on this code:

```
if __name__ == '__main__':
    fpath = '../active1000'
    flist = traverse_dir(fpath)
    df = load_data(fpath, flist)

    ###### Get Statistics from dataset ############
    statistics(df)

    ###### Recommendations based on Collaborative Filtering (Matrix Factorization) #######
    collaborative_filtering(df)

    ###### Recommendations based on Content-based Method (Cosine Similarity) ############
    content_recommendation(df, k=20)
```

If you only want to run CF, you can quote statisdtics(df) and content_recommendation(df, k=20) with “#”.

```
if __name__ == '__main__':
    fpath = '../active1000'
    flist = traverse_dir(fpath)
    df = load_data(fpath, flist)

    ###### Get Statistics from dataset ############
    # statistics(df)

    ###### Recommendations based on Collaborative Filtering (Matrix Factorization) #######
    collaborative_filtering(df)

    ###### Recommendations based on Content-based Method (Cosine Similarity) ############
    # content_recommendation(df, k=20)
```

The same way to run recommendations based on Content-based Method (Cosine Similarity) only:

```
if __name__ == '__main__':
    fpath = '../active1000'
    flist = traverse_dir(fpath)
    df = load_data(fpath, flist)

    ###### Get Statistics from dataset ############
    # statistics(df)

    ###### Recommendations based on Collaborative Filtering (Matrix Factorization) #######
    # collaborative_filtering(df)

    ###### Recommendations based on Content-based Method (Cosine Similarity) ############
    content_recommendation(df, k=20)
```

The results and more information of the two recommendation methods can be found in Dataset Description.pdf on Blackboard.