

Programming Assignment 3: PCA and FastMap

Part 1: Group member and contribution

Bin Zhang (5660329599): PCA code, FastMap code, Part2 report

Yihang Chen (6338254416): PCA code, FastMap code, PCA library code, FastMap library code, Part 3 report

Hanzhi Zhang (4395561906): PCA code, FastMap code, and Part 4 report

Part 2: Implementation of PCA and FastMap

PCA :

PCA is a famous and useful algorithm to PCA to reduce the dimensionality of the data points from n dimensionality to k dimensionality. Here we use the sample points to reduce them from 3D to 2D. As for PCA, we split the task into 5 steps:

1. Load data and process it, in this process we normalize it and use the NumPy's ndarray to store it. After being processed, the data's shape is (6000,3)
2. Compute the covariance matrix for data
3. Compute eigenvectors and eigenvalue for data, step 2 and 3 we simply use NumPy's dot function and linalg.eig function. Actually manually calculating covariance matrix, eigenvectors, and eigenvalue would be been a huge challenge.
4. Pick top k principal components, here $k=2$
5. Generate the new dataset

Due to the help of NumPy's function(linalg.eig), it's not so difficult for us to implement it and with the regard to the robustness of our script, we use assert to make sure that the input dimension k is equal or less than raw data's dimension n .

The directions of the first two principal components are:

```
[[ 0.86667137, -0.4962773 ],
```

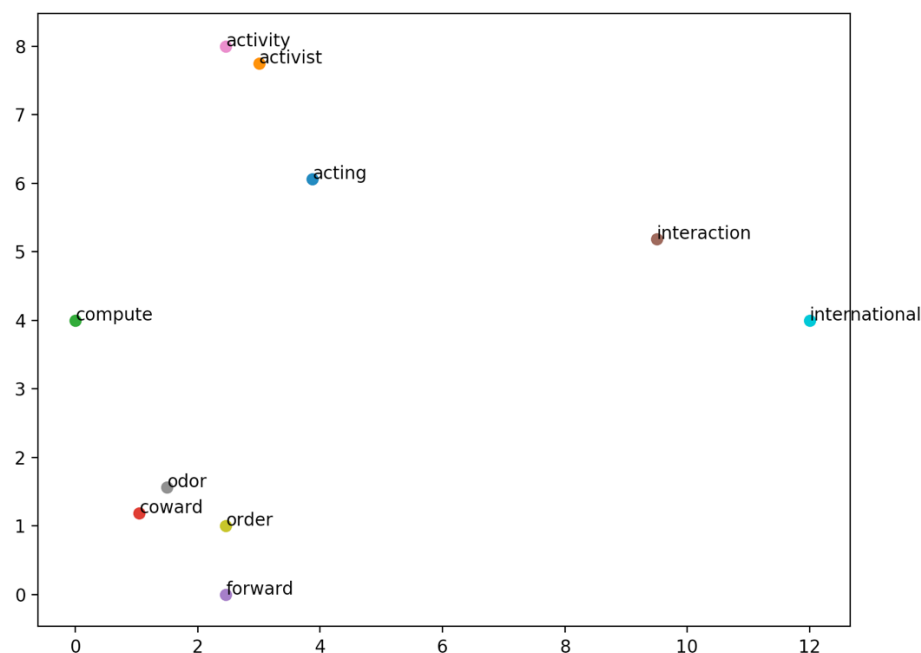
```
[-0.23276482, -0.4924792 ],  
[ 0.44124968,  0.71496368]]
```

each column represents the direction of a component.

FastMap

As its name, FastMap is a fast algorithm to reduce dimensionality and project object to the Euclidean space. Here we use the sample(words) to project them into the Euclidean space. We split the task into 4 steps:

1. Load distance data and store them into the nested list. The reason why we don't use NumPy's ndarray is the problem caused in the step3.
2. Identify the farthest pair of object a and object b.
3. Recursively set the coordinates for object, once we get the object a and object b, we need to use the Cosine law to help us generate the pi for each xi, and then we according to the old distance and new distance, we get the new farthest pair o', b', and recursively get the following projected coordinates. And problem appeared: First we used the NumPy's matrix to store the data, but when we got the new distance, the datatype is float and when we store the new distance into the matrix and try to update it, it automatically cut off the decimal. In order to solve the problem, we turned back to the step 1 and change the data structure----using nested list.
4. Plot the object in a 2d plane, here we only use 2 dimension to represent our object and the result is like following graph:



The projected coordinates for each word are:

```
{'acting': [3.875, 6.0625],
 'activist': [3.0, 7.749999999999999],
 'compute': [0.0, 4.0],
 'coward': [1.0416666666666667, 1.1875],
 'forward': [2.4583333333333335, 0.0],
 'interaction': [9.5, 5.1875],
 'activity': [2.4583333333333335, 8.0],
 'odor': [1.5, 1.5624999999999996],
 'order': [2.4583333333333335, 1.0],
 'international': [12.0, 4.0]}
```

Part 3: Software Familiarization

For the library to implement PCA, we choose PCA method in Sklearn library. It's a very simple method that we only need to set `n_components` (new K dimensions) and fit the data.

Cause when using PCA method, program need to load all data to the memory, it is easy to exceed memory space. Sklearn also provided a Incremental PCA method for the case that the dataset is too large to load to the memory. "The IPCA method builds a low-rank approximation for the input data. It is still dependent on the input data features, but changing the batch size allows for control of memory usage."

For the implementation for the Fastmap, we only found a vector version library. That is to say, the library is designed for dimensionality reduction. The input data is original data vector. The data need to be measured into high dimensional vector. It is different with the data in the assignment, which only contains the distance between any two objects. It only takes the advantage of lower time cost.

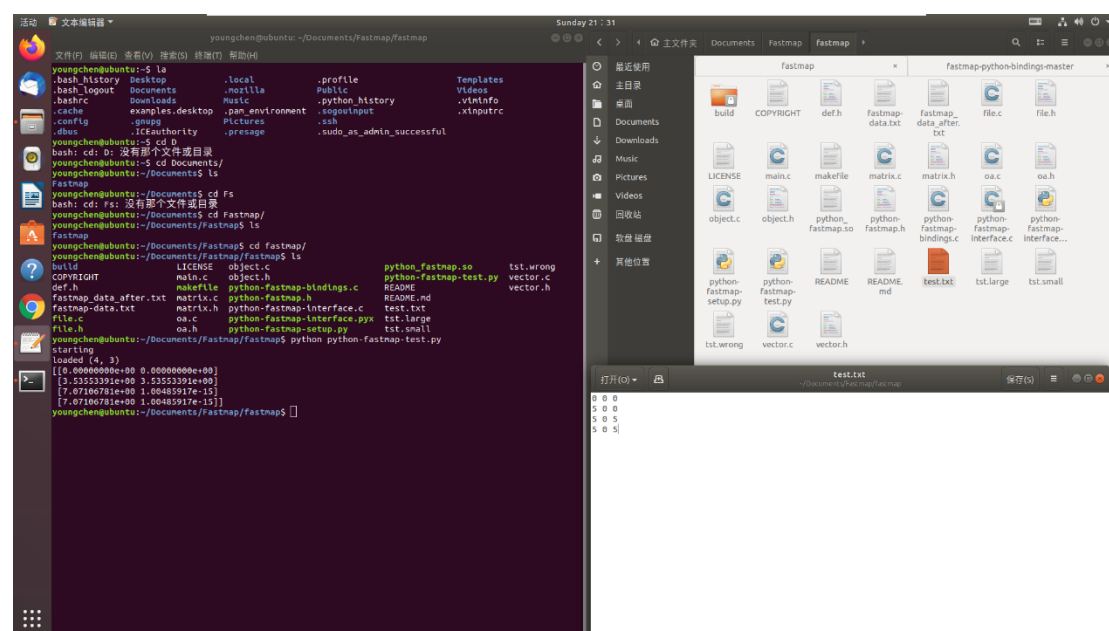
The core Fastmap code is written by Christos Faloutsos in C.

<http://www.cs.cmu.edu/~christos/software.html>

There is a project on the GitHub which build a python bindings for the Fastmap library by Christos Faloutsos.

<https://github.com/definitelyuncertain/fastmap-python-bindings>

We implemented this project on the Ubuntu 18.04.4LTS and Python 2.7.17. The library and the python binding are fully functional. The only pity is that the data need to be in vector form.



```
youngchen@ubuntu:~$ la
.bash_history Desktop .local .profile Templates
.bash_logout Documents .mozilla Public Videos
.bashrc Downloads Music .python_history .viminfo
.cache examples.desktop .pan_environment .ssinputrc .xinputrc
.config .gnupg Pictures .ssh
.dbus .ICEauthority .presage .sudo_as_admin_successful

youngchen@ubuntu:~$ cd 0
bash: cd: 0: 没有那个文件或目录

youngchen@ubuntu:~$ cd Documents/
youngchen@ubuntu:~/Documents$ ls
Fastmap

youngchen@ubuntu:~/Documents$ cd Fastmap/
youngchen@ubuntu:~/Documents/Fastmap$ ls
Fastmap

youngchen@ubuntu:~/Documents/Fastmap$ cd fastmap/
youngchen@ubuntu:~/Documents/Fastmap/fastmap$ ls
build COPYRIGHT def.h fastmap-data.txt file.c file.h
def.h fastmap-data-after.txt fastmap-data.txt fastmap.h
LICENSE main.c makefile matrix.c matrix.h oa.c oa.h
fastmap-data.txt matrix.h python-fastmap-Interface.c test.txt
file.c oa.h python-fastmap-Interface.py test.large
file.h python-fastmap-Interface.py test.small

youngchen@ubuntu:~/Documents/Fastmap/fastmap$ python python-fastmap-test.py
starting
loaded (4, 3)
[[0.00000000e+00 0.00000000e+00]
 [3.53533391e+00 3.53533391e+00]
 [7.07106781e+00 1.00485917e-15]
 [7.07106781e+00 1.00485917e-15]]
youngchen@ubuntu:~/Documents/Fastmap/fastmap$
```

Part 4: Applications

PCA (Principle Component Analysis)

PCA is a statistical procedure that reduce the datasets in high dimension to the lower dimension with less information loss. PCA can be applied on the data visualization and image processing.

In terms of visualization, many high-dimensional datasets are hard to analysis and visualization. However, PCA can solve it. An application of PCA to detection and visualization of computer network attack is a good example.

Image processing can be divided into face recognition and image compression. For high-dimensional datasets, a small eigenvalue means this dimensional dataset has less relationship with the total data. Therefore, in image processing, a small eigenvalue represents noise dataset for noise reduction and redundancy for image compression.

FastMap

FastMap is also a measure for dimensionality reduction. Some high-dimension dataset cannot be represented by Euclidean space, which means PCA cannot be applied for them. In this case, FastMap can be helpful. DNA structure is a good example. FastMap can have a good performance on both gene expression and high-density genotype data.