Programming Assignment 3: PCA and fastmap

Instructor: Dr. Satish Kumar Thittamaranahalli
Group Member: Yu Hou; Haoteng Tang

In this assignment, we will implement two algorithms, PCA and Fastmap. Both of them is used to reduce the dimensions. First of all, PCA (principle component analysis) is to treat the data as matrix and calculate their means, sigma, and the eigenvetors of this matrix. Only store the first k eigenvetors. The Fastmap is not easy to understand. This method is used to calculate different values' distance, and project them to different axis. In this assignment, we will treat 6000 values with PCA and treat 10 words with fastmap method. In the first section, we will show you the pseudocodes of them. In the second section, we will show you the library in different programming languages. In the third section, we will show you the applications of these two different algorithms.

1 pseudocode of PCA and Fastmap (Yu Hou)

(1) pseudocode

PCA method:

```
    Call miu method to calculate the mean.
    Call sigma method to calculate the sigma.
    Call U matrix to calculate a new matrix which is consist of first k eigenvectors.
    Call z method to times original data (matrix) with U matix.
```

Fastmap method has two ways to implement. We can use two for loop to get k dimension matrix. Also, we can use recursion to get k dimension matrix.

Fastmap method 1:

```
    Determine the A point and B point and distance to point A of every point
    Using $x_i = (d_{ai}^2 + d_{ab}^2 - d_{ib}^2)/2d_{ab}$, i means different point
    Get the x value of different points

    Get the new distance of every points
    Using $D_{new}^2(O_i'O_j') = D_{old}^2(O_i'O_j') - (x_i - x_j)^2$

    Determine the A point and B point again and calculate the distance to point A of every point
    Get the y value of different point.
```

Fastmap method 2 (recursion):

```
    If it is the base case:
        K=k' means we already get the k dimension
    If it is not the base case:
        Determine the A point and B point and distance to point A of every point
        Using $x_i = (d_{ai}^2 + d_{ab}^2 - d_{ib}^2)/2d_{ab}$, i means different point
        Get the $K_i$ dimension of different points
```

Update the new distance of every points Using $D^2_{new}(O^{'}_iO^{'}_j) = D^2_{old}(O^{'}_iO^{'}_j) - (x_i - x_j)^2$
Recursion and k=k+1

Because in the assignment, we just had to represent 10 words in 2D, so we used the method 1 without recursion.

(2) Results
PCA:
Shown as the figure 1, the 6000 point can be plotted in 3D view. In the figure2, they are the same 6000 points plotted in 2D view after dimension reduction.

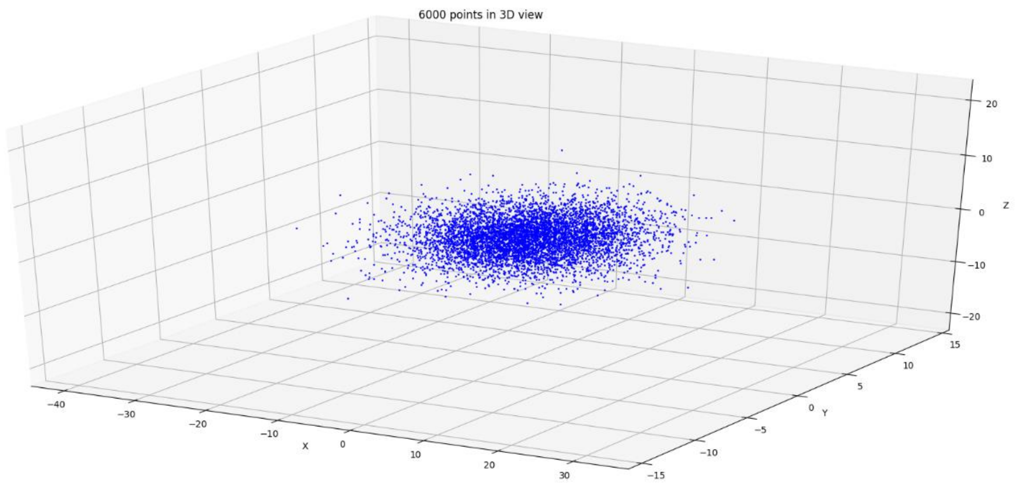

Figure 1 6000 points in 3D view

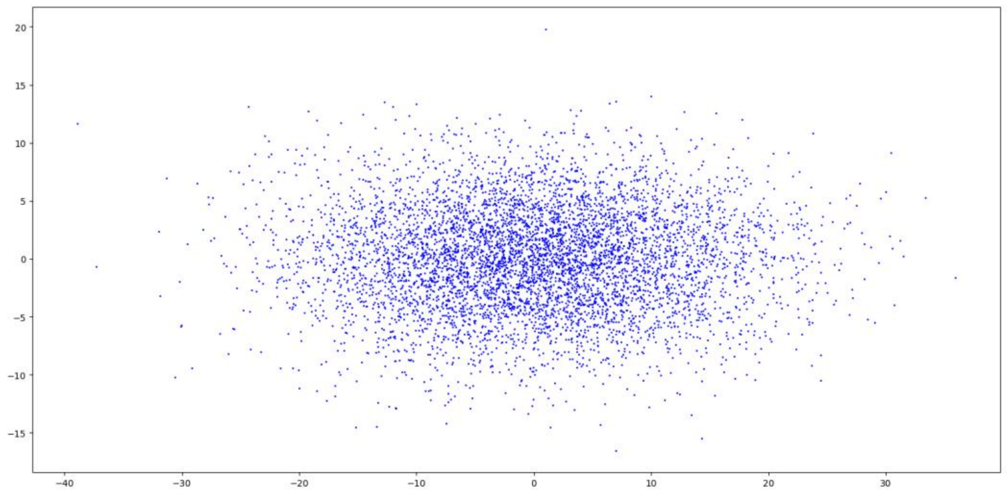

Figure 2 6000 points in 2D view
If we print out the 6000 points in 2D in matrix, it will be as follows:

```
[[ 10.87667009   7.37396173]
 [-12.68609992  -4.24879151]
 [  0.43255106   0.26700852]
 ...,
 [ -2.92254009   2.41914881]
 [ 11.18317124   4.20349275]
 [ 14.2299014    5.64409544]]
```

Fastmap:
If we print out the results, it can be as follows, there will be two different results. In the first iteration, the farthest two points are 3.0 and 10.0, and the distance is 12.0. In the second iteration, there are two cases. One the farthest two points are 5.0 and 7.0, and the distance is 8.0. The other farthest two points are 7.0 and 9.0, the distance is also 8.0. The matrixes of these two cases are shown as follows.

First case:

```
[[ 3.875        6.0625    ]
 [ 3.           7.75      ]
 [ 0.           4.        ]
 [ 1.04166667   1.1875    ]
 [ 2.45833333   0.        ]
 [ 9.5          5.1875    ]
 [ 2.45833333   8.        ]
 [ 1.5          1.5625    ]
 [ 2.45833333   1.        ]
 [12.           4.        ]]
```

Second case:

```
[[ 3.875        2.75      ]
 [ 3.           0.25      ]
 [ 0.           4.        ]
 [ 1.04166667   5.5       ]
 [ 2.45833333   7.        ]
 [ 9.5          4.        ]
 [ 2.45833333   0.        ]
 [ 1.5          7.75      ]
 [ 2.45833333   8.        ]
 [12.           4.        ]]
```

However, according to the requirement of assignment, we should use the one that includes the smallest object ID. If we plot these 10 points in the 2D view, it can be as follows in figure3. Also, we can calculate the distances between each pair of objects, the Damerau–Levenshtein distances between them. Then, we label them on the Fastmap.

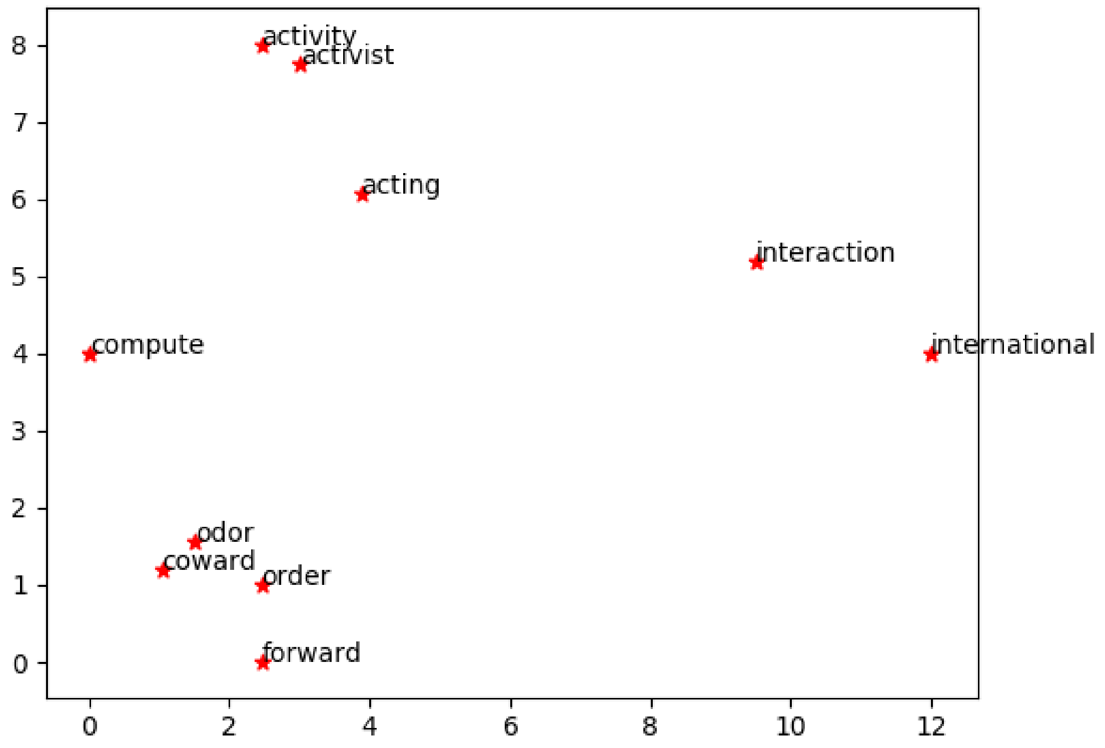

Figure3 10 words using Fastmap

(3) data structure
PCA
This algorithm is easy to store the data. We create a matrix to store all 6000 3D points, matrix [6000] [3]. This is easy to calculate the mean, sigma, eigenvector using matrix. We also store the mean, sigma, and eigenvector in different matrix. After computation, we will get a matrix [6000] [2] to store all 2D points after reduction.

Fastmap
There are two ways to store data in fastmap method.
First method is using a 2D array. Shown as the table 1.

Table 1 store the distance between each two record.

	01	02	03	……	09	10
01	0.0	4.0	7.0	……	6.0	10.0
02	4.0	0.0	7.0	……	8.0	11.0
03	7.0	7.0	0.0		6.0	12.0
……	……	……	……	……	……	……
09	6.0	8.0	6.0	…….	0.0	11.0
10	10.0	11.0	12.0	…….	11.0	0.0

In this 2D array, for example, the intersection of the “01” column and the “02” row is 4.0 means the distance of point “01” and “02” is 4.0.

The second method is using a dictionary in Python. The dictionary consists of several key-value pairs, so the key can represent the first record and the value can be also a key-value pair. The key of it is the second record, and the value can be the distance of the first record and the second record. Shown as follows.

{1.0: {2.0: 4.0, 3.0: 7.0, 4.0: 6.0, 5.0: 7.0, 6.0: 7.0, 7.0: 4.0, 8.0: 6.0, 9.0: 6.0, 10.0: 10.0},
2.0: {1.0: 4.0, 3.0: 7.0, 4.0: 7.0, 5.0: 8.0, 6.0: 9.0, 7.0: 2.0, 8.0: 8.0, 9.0: 8.0, 10.0: 11.0},
3.0: {1.0: 7.0, 2.0: 7.0, 4.0: 5.0, 5.0: 6.0, 6.0: 10.0, 7.0: 6.0, 8.0: 6.0, 9.0: 6.0, 10.0: 12.0},
……,
10.0: {1.0: 10.0, 2.0: 11.0, 3.0: 12.0, 4.0: 12.0, 5.0: 11.0, 6.0: 4.0, 7.0: 11.0, 8.0: 12.0, 9.0: 11.0}}

For example, in the dictionary the keys are 1.0, 2.0, 3.0, ……, 10.0. the values of key 1.0 are also a dictionary. In this dictionary, the keys are 2.0, 3.0, ……. And the value of the key 2.0 (the values is the 4.0) is the distance of 1.0 and 2.0.

The data structure of storing the result is a matrix, matrix [10] [2]. As shown in the result section above.

(4) challenges, code-level optimizations

The biggest challenges in the assignment is to understand the procedure of Fastmap, especially to understand these two formulas, $x_i = (d_{ai}^2 + d_{ab}^2 - d_{ib}^2)/2d_{ab}$ and $D_{new}^2(O_i^{'}O_j^{'}) = D_{old}^2(O_i^{'}O_j^{'}) - (x_i - x_j)^2$. The key is to understand these two pictures. First of all, we should treat the distance as springs. Find the most remote two points and compute all points to the line. This value can be the first axis, for example X axis. Using the second formula and the second figure to compute the new distance of each different points. Do this again to compute the second axis value of different points, for example Y axis. If we want to project the points in higher dimensions, we can do this procedure more times. Another challenges of this method is to find the base case if we want to implement this method using recursion.1 Next step, we can improve our data using recursion.

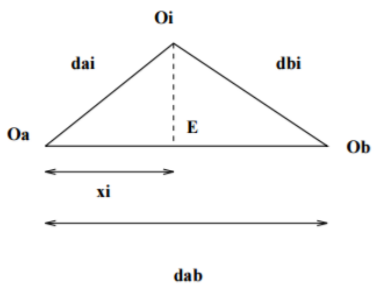

Figure5 projection on the line OaOb

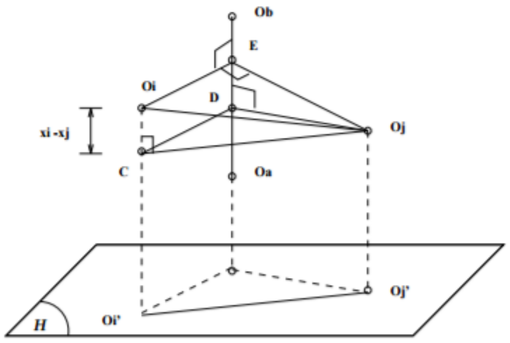

Figure6 Projection on a higher dimension

In this assignment, we can search the farthest distance between two points through all data. However, we improved my code using a heuristic way to find the first point randomly, and find the farthest distance in several iterations like the professor talked in the class. Because when we have many data, it is better to use a heuristic way.

2 Software Familiarization (Yu Hou, Haoteng Tang)

(1) Python for principle component analysis(PCA)

In Python programming, we have a library called sklearm. In the library, we can find some statements to implement this algorithm. Here is the main statements.

```
>>> import numpy as np
>>> from sklearn.decomposition import PCA
>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])          //data set need to reduce
>>> pca = PCA(n_components=2)
>>> pca.fit(X)
PCA(copy=True, iterated_power='auto', n_components=2, random_state=None, svd_solver='auto',
tol=0.0, whiten=False)
```

In this statement, number of components to keep. if n components are not set all components are kept, so n_components == min(n_samples, n_features).

(2) MATLAB for principle component analysis(PCA)

In order to do the PCA of a set of data X which is showed in an m*n dimension matrix by using MATLAB library function, we can code as below:

```
[COEFF,SCORE,Latent]=princomp(X);
```

COEFF is a p-by-p matrix, each column containing coefficients for one principal component. The columns are in order of decreasing component variance. For data matrix in assignment, the COEFF matrix is:

```
[0.866671367084373   -0.496277304498303   -0.0508879014698239
 -0.232764816475979  -0.492479204508382   0.838620756562641
 0.441249681798195   0.714963684509004    0.542333521134645];
```

which is the same as the result based on our own code.

SCORE is the principal component scores that is the representation of X in the principal component space. Rows of SCORE correspond to observations, columns to components.

Latent, a vector containing the eigenvalues of the covariance matrix of X.

(3) Python for fastmap

It is very hard to find a library using python to implement fastmap. However, teaching assistance suggested us to search a code somebody else wrote on the website. We found one on Github, https://github.com/mahmoudimus/pyfastmap. [2]The author is “mahmoudimus”. We used his code to run our data and get some results as follows.

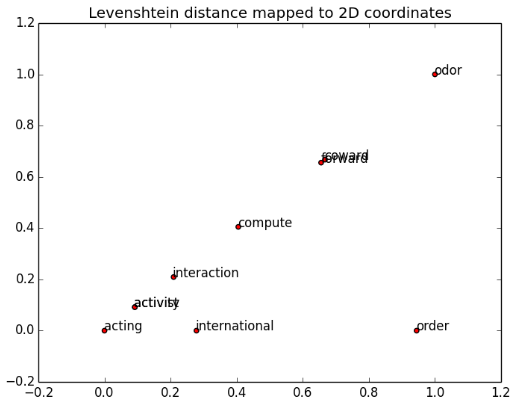

Figure 7

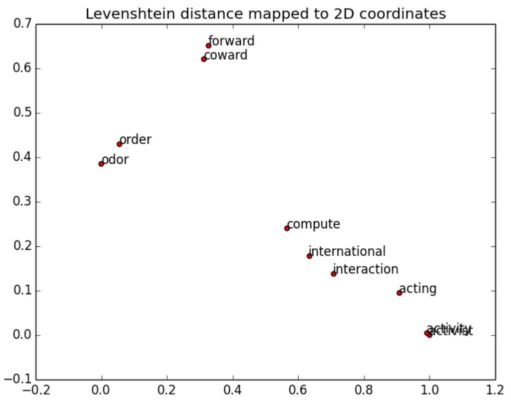

Figure 8

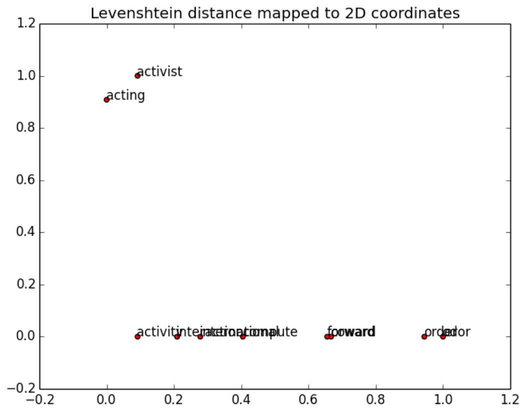

Figure 9

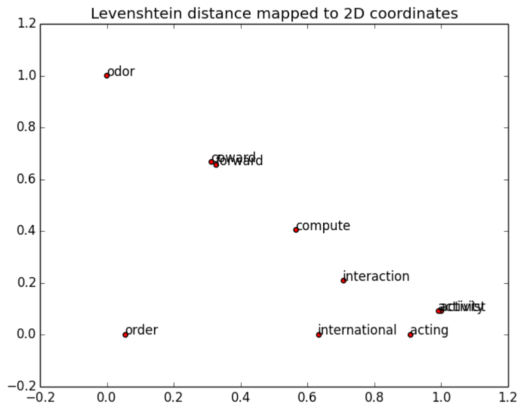

Figure 10

However, we found that the results cannot make sense. In these four pictures above, we can find that there were great possibilities that some words were in a line Because the author find the farthest distance between two points randomly, but there were some mistakes in his code. Sometimes, the code will choose the same two words as the two farthest distance words. In the author's code, he wrote a test code to generate 10 5-D points randomly distributed between [0-10]. Here are twice running his test code.

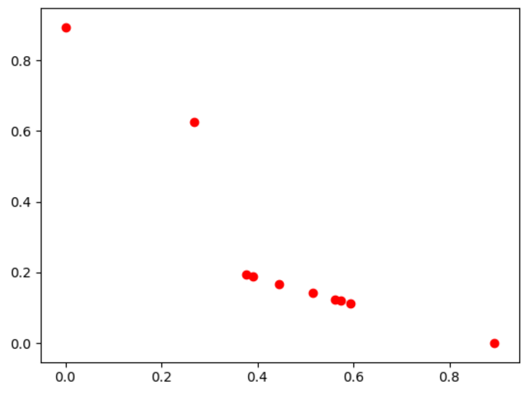

Figure 11

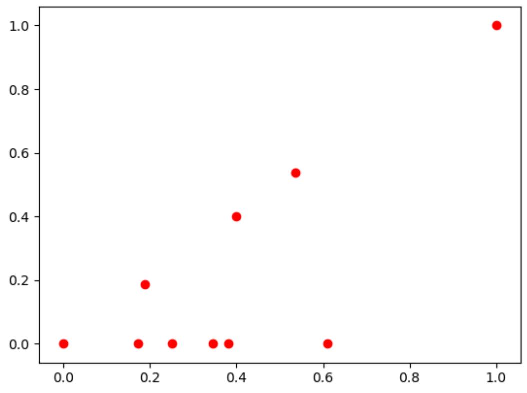

Figure 12

In the figure11, the first iteration, the code picked point 2 and 1, the same as the second iteration. In the figure12, the first iteration picked point 9 and 3, the same as the second iteration. Here are the results printed.

Picked 2, 1 at K = 2
Picked 1, 2 at K = 1

Picked 9, 3 at K = 2
Picked 9, 3 at K = 1

So, this library should be improved.

3 Application (Haoteng Tang)

Both PCA and fast-map are a kind of method to reduce the dimension. It will be complicated for our algorithm to be executed and be a lot of time and space cost if the attributes (dimensions) of the data is very high when we use machine learning algorithms such as neural network to process the data. For instance, it the dimension is very high, each level of the neural network should be implanted many perceptron which is very complex. However, by using this kind of dimension reduction, we can use a few representative dimensions to replace previous whole dimensions. This is a logical and wise preprocess before we deal with our data into following steps.
For example, in construction management. We want to find the most appropriate architectural designer. We have a big dataset with many attributes. Here are some of these attributes.

1Gender
2Age
3Worktime in this company （how many years）
4Salary
5Education background
6Designer style
7Position in the company
8Marriage
9Kids (number)
10Work time (how many hours a week)
11Work delay (how many days)
12Extra work time (how many hours a week)
13Design mistakes
14Typical design mistakes, account the number
15Design omissions
16Typical design omission, account the number
17Design conflicts
18Typical design confilicts, account the number
19Type of work (new-build，rebuild，extension)
20Discipline of work (Steel, concrete， wood，combination)
21Complexity of the work
22Profitable of the work
23Service fee of the work
24Priority of the work

There are too much attributes, we need to reduce the dimension of the dataset. So we can use PCA and fastmap and other methods.

Reference

[1] Christos Faloutsos, King-Ip Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. 1995: 163-174
[2] Fastmap Github: https://github.com/mahmoudimus/pyfastmap.