Part2

1. PCA
2. How to run the code which implement PCA with using sklearn:

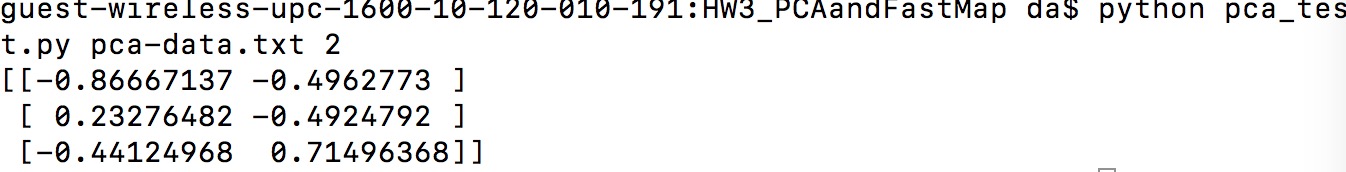
python pca\_test.py pca-data.txt 2

1. Explanation:

We learned how to implement the PCA with using sklearn.

And then output the components.

Here is the snapshot of result.



1. Difference:

I think the main difference is that we generate the components with using Eigenvalue Decomposition, and the sklearn use the SVD Decomposition.

And this is the main reason that the result is not same.

Another difference is that sklearn used combine strategy to do the SVD Decomposition:

the solver is selected by a default policy based on X.shape and n\_components:

if the input data is larger than 500x500 and the number of components to extract is lower than 80% of the smallest dimension of the data, then the more efficient ‘randomized’ method is enabled. Otherwise the exact full SVD is computed and optionally truncated afterwards.

1. FastMap
2. To be honest, from online, I can search too much library which implemented the FastMap, however I found a lot of codes in github and blogs. Here we found a blogger who had implemented the FastMap algorithm in python as well (http://gromgull.net/blog/2009/08/fastmap - in - python/ )
3. Difference:  
   The result of each run is not same, sometimes is good and sometimes is bad.

This is because that it chooses the first object randomly. And in our code, the result is same and stable.

In his code, he used two dimensional array to represent distances. But in our code, we used a graph which nodes represent ID and edges represent distance.