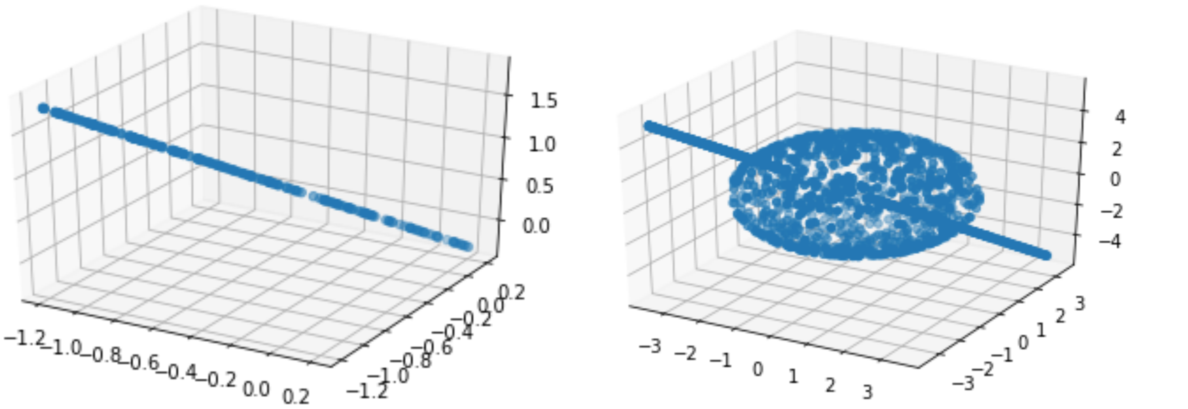
Part 1

I finished reading the paper *A Tutorial on Spectral Clustering*, and understood what spectral clustering trying to do. It said if there exists k totally separate clusters, its graph Laplacian matrix has k zero(or nearly zero) eigenvalues. This means I should take care of eigenvalues which are nearly zero and the corresponding eigenvectors could help me find something. And this paper told me something about adding small perturbations to an ideal Laplacian matrix. In my simulations, objects intersect so I consider there exists small connections between each object.

According to this paper, I tried normalized Laplacian $L_{rw} = I - D^{-1}W$. First on two parallel lines, the result is perfect. The first two eigenvalues are nearly zero(order 10^{-16}) and corresponding eigenvectors are indeed indicating vectors that indicate which cluster should points go.

Then I tried on data of sphere and line (without error). I did K-means on first two smallest eigenvectors and one cluster appeared a linear figure which indicated there existed a line among the data. But when I checked axis of the plot, I found that the linear figure was just a small part of original line. Bases on this fact, could I say the line do exist? And I repeated the experiment, I could only see the line but not the sphere. So I guess we could only detected lower dimensional objects using this method. And there is another question that confused me: my line and sphere only intersected at two points, its similarity matrix should be very close to the ideal one, but I could only classify a small part from the whole data like this:



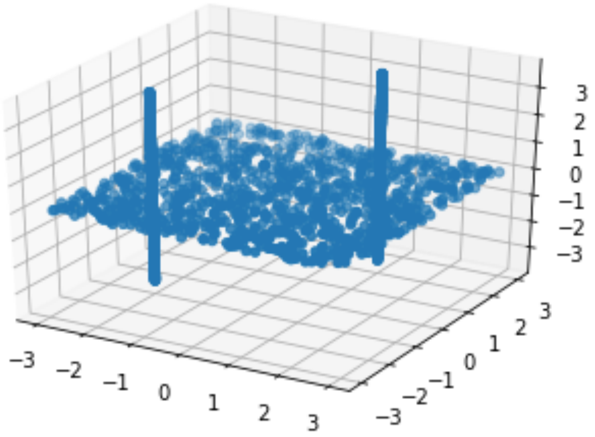
(a) (b)

Figure 1.1 (a) shows the first cluster; (b) shows the second cluster

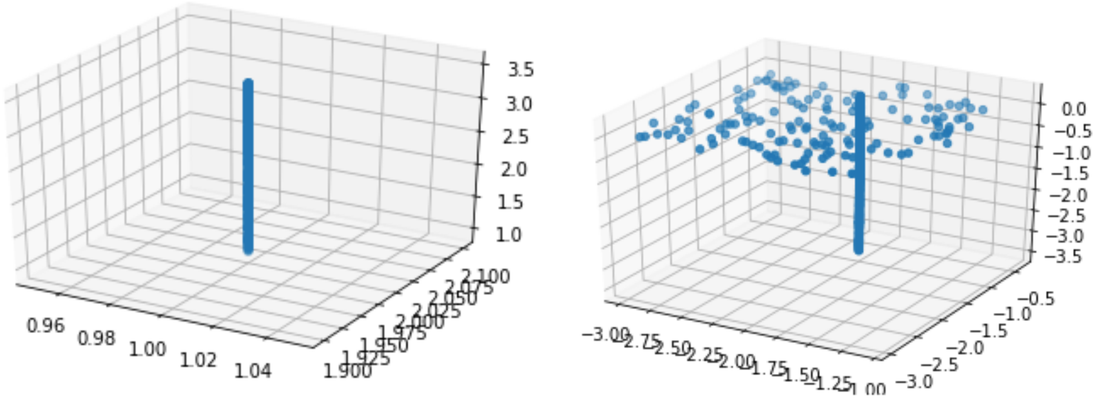
The above plots show that we could detect line using this method but we could not detect sphere or classify two objects.

Part 2

I tried another dataset: two parallel lines and one plane:



My idea is to check eigenvalues of its Laplacian matrix to find how many eigenvectors I should take. In this dataset, first 5 eigenvalues are nearly zero(order 10^{-15}), so I take 5 corresponding eigenvectors. I did K-means assuming there are 5 clusters(for I took 5 vectors). It finally gave me clusters and I plot them out. There are two clusters showing linear figures.



Roughly, I could conclude I detected lines. This result is much better than only consider first 3 eigenpairs and do clustering with only 3 clusters. But this procedure showed something different from data simulation. I simulated 2 lines and 1 plane, and the intersections were just few points. Thus I assumed this situation was much close to ideal situation—only 3 clusters and only the first 3 eigenvalues should be zero. However, the experiment showed 5 eigenvalues and 5 clusters. I read the slides and paper you gave me and I was wondering if the Laplacian matrix really considers the plane as an object or just noise data.

Part 3

From the experiment above, this method could detect objects. Then I tried to add noise to it and test how sensitive this method could be.

Two confusion here:

1. how to add noise. I think of two ways: one is to add noise to each data point; the other is to generate a noise dataset.

2. If I add noise to each data point, the dimension would change. In this case the line is not 2-dim at all. Would the problem change then? For our original problem is to detect different dimensional objects.

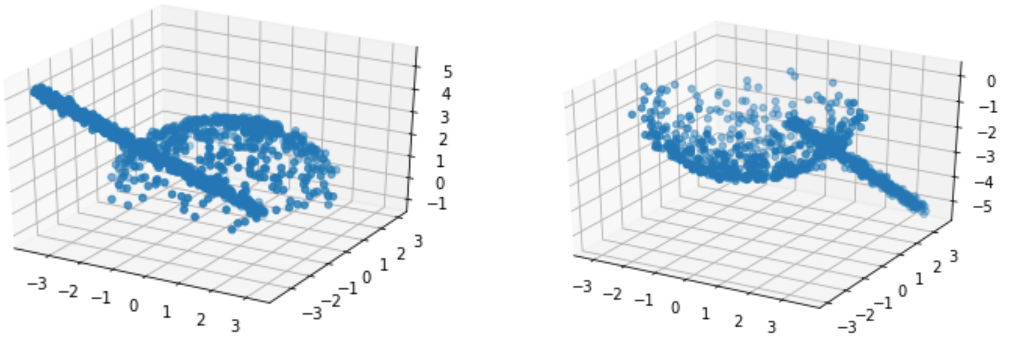
Stage 1 Add Gaussian noise to each data point N(0,σ^2)

1. $\sigma^2 = 0.1$

Only the first smallest eigenvalue is close to 0.

When trying k-Means on the first eigenvector, detect nothing.

When trying k-Means on the first two eigenvectors, detect nothing.

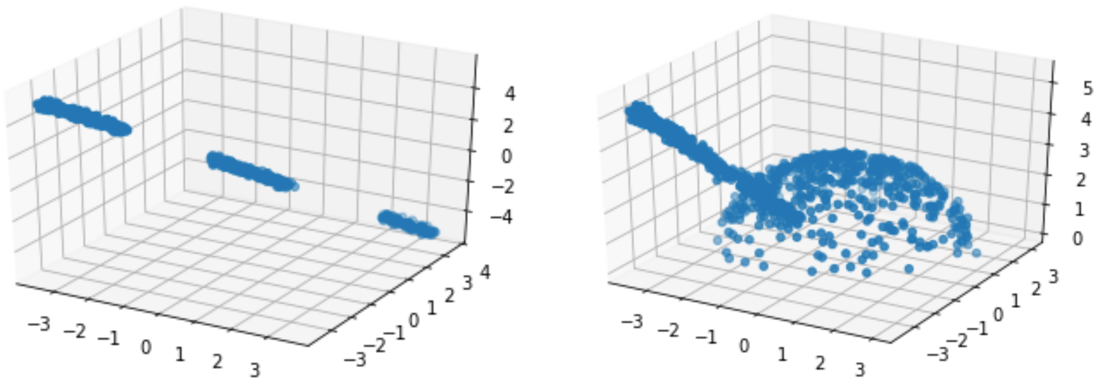


(a) (b)

Figure 3.1.1 (a) shows the cluster from k-Means of the first eigenvector. (b) shows the cluster from k-Means of the first two eigenvectors.

2. $\sigma^2 = 0.08$

Similar result to $\sigma^2 = 0.1$. But I do see something in one cluster and I could not explained that.



(a) (b)

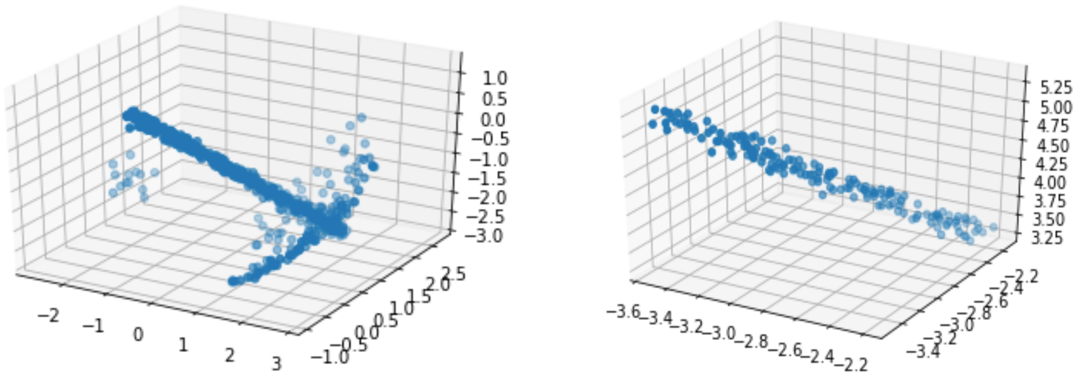
Figure 3.1.2 (a) shows the cluster from k-Means of the first eigenvector and I could find 3 separate linear figures. Approximately, they seem to be parts of the original line. (b) shows the cluster from k-Means of the first two eigenvectors.

3. $\sigma^2 = 0.06$

Only the first eigenvalue is close to 0 (order 10^{-16}).

When trying k-Means on the first eigenvector, detect a line approximately. To be specific, a line and some noise from sphere.

When trying k-Means on the first two eigenvectors, detect a line approximately.



(a) (b)

Figure 3.1.3 (a) shows the cluster from k-Means of the first eigenvector. (b) shows the cluster from k-Means of the first two eigenvectors.

4. $\sigma^2 = 0.05$

First and second smallest eigenvalues are close to 0.

When trying k-Means on the first two eigenvectors, detect a linear figure.

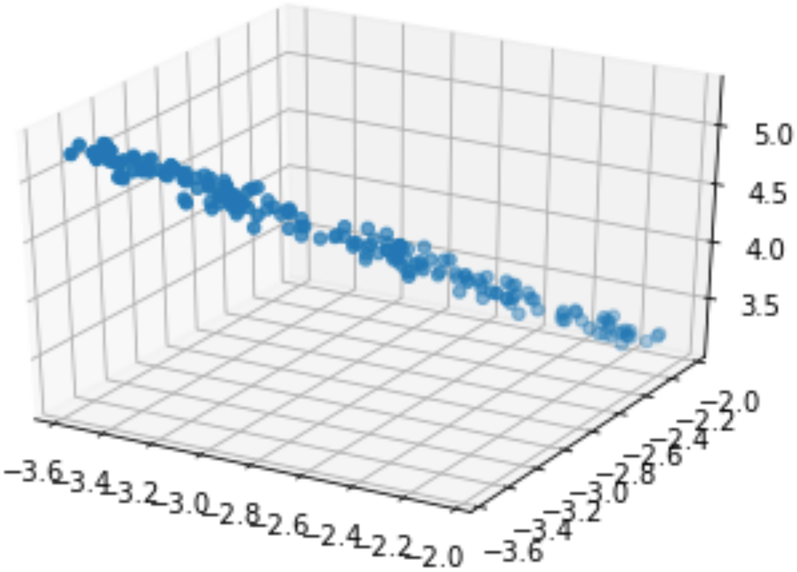


Figure 3.1.4 This plot shows linear figure in one cluster

5. $\sigma^2 = 0.03$

Similar to $\sigma^2 = 0.05$.

6. $\sigma^2 = 0.01$

Similar to $\sigma^2 = 0.03$ and the line is clearer.

I think variance smaller than 0.01 will have better performance and we could definitely detect a line. Thus roughly saying, when variance is larger than 0.06, the figure is not clear and variance smaller than that will show clear figure and detection.

Stage 2

Denoising procedure.

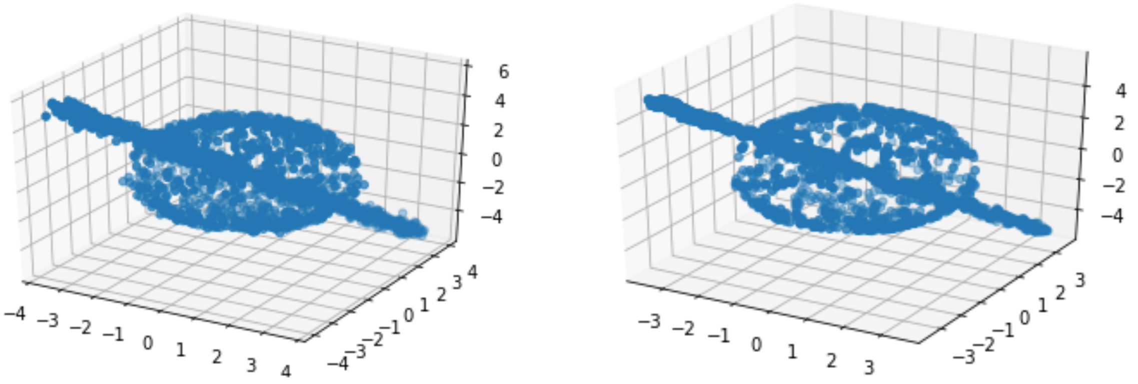
Idea: 1. Let $\widetilde{x_i}$ =mean of k nearest neighbor of x_i;

2. remove any point have a large distance to their k-th nearest neighbor.

I simulated original data with few combos of variances for line and sphere. Let ${\sigma_1}^2$ denote variance for the line and ${\sigma_2}^2$ denote variance for the sphere.

1. ${\sigma_1}^2 = 0.2, {\sigma_2}^2 = 0.1$, and set k=8

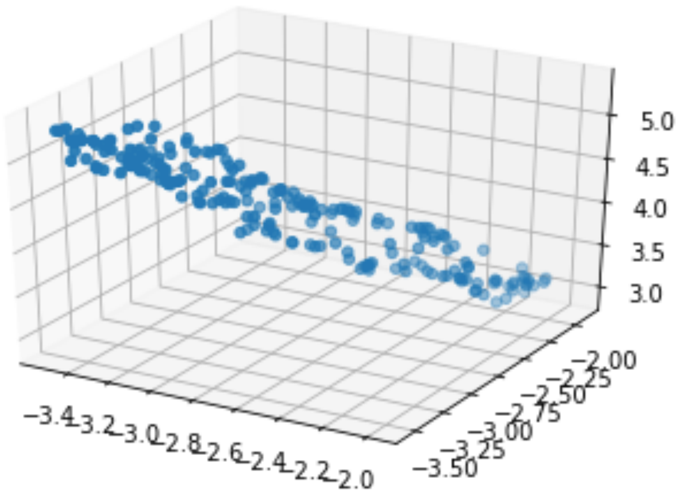
Plot of data became clearer like below:



(a) (b)

Figure 3.2.1 (a) is the original data and (b) is the denoising data. Obviously, the new data becomes denser and each object is clearer.

Then I obtained Laplacian matrix of denoised data and its eigenpairs. The first 5 eigenvalues were close to 0. I did k-Means assuming there were 5 clusters. In one cluster, I observed linear figure:



This result was better than original data which did not detect anything. But this figure was not that strong to support the detection.

2. $\sigma_1{}^2 = 0.3, \sigma_2{}^2 = 0.1,$

Similarly, plot of data became a little bit better but not very ideal, for the variance was relatively big.

Thus I was thinking if this denoising procedure was not strong enough. The paper ***Path-Based Spectral Clustering: Guarantees, Robustness to Outliers, and Fast Algorithms*** mentioned some ways like defining a new distance and another denoising procedure. And I was considering other noise that I could add to the data. I am still working on these issues.