Spectral Clustering

0856030, 林正偉, aunbdaunbd@gmail.com

December 26, 2019

1 Introduction

In this project, we are asked to recover the 2D points from the given distance matrix. Next, apply spectral clustering to cluster these points into 2/3/4 classes.

2 Implementation Procedures

2.1 Problem 01

I'm going to talk about how I recover the position matrix.

1. Follow the equation in lecture slides. We get

$$G = -\frac{1}{2} \times (D - 1D_1^T - D_1 1^T) \tag{1}$$

   where D is the squared distance matrix, D_1 is the first column of D, 1 is an N $\times$ 1 vector with N as the number of nodes.

2. Apply eigenvalue decomposition, we get the position matrix X

$$G = Q\Lambda Q^T \implies X = \sqrt{\Lambda} Q^T$$

   where the columns of Q are the eigenvectors of G and $diag(\Lambda)$ contains the eigenvalues of G.

3. Since we want 2D points, we just construct the position matrix by 2 eigenvectors corresponding to the first 2 eigenvalues.

2.2 Problem 02

1. Calculate the weight matrix W, where w_{ij} is defined as $1 - \frac{D_{ij} - min(D)}{max(D) - min(D)}$.

2. Calculate the normalized graph Laplacian

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

3. Use Kmeans to cluster the eigenvector corresponding to the 2^{nd} smallest eigenvalue into 2/3/4 classes. In the extra experiment section, I'll show that we can try different numbers of eigenvectors for Kmeans clustering.

3 Results

The following figures show the good results of spectral clustering. For those bad results, I'll demonstrate them in the discussion section.

3.1 Problem 01

Fig. 1 shows the points we derived from the given distance matrix.

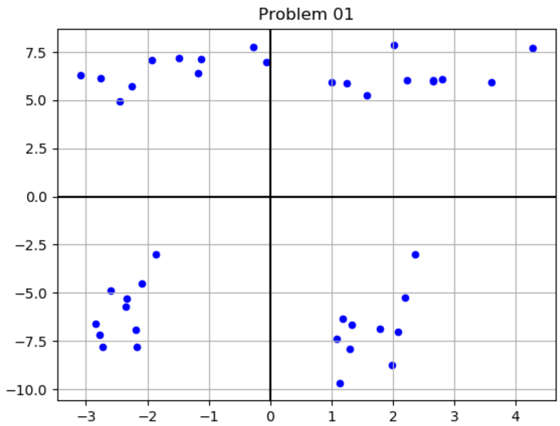
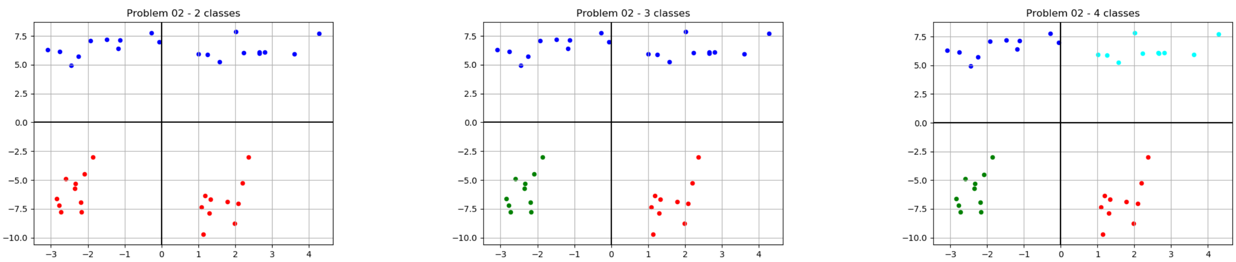


Figure 1: Show the position of the points. The squared distance between them is approximate to the given distance matrix.

3.2 Problem 02

Fig. 2 shows the results that clustering the points into 2/3/4 classes by spectral clustering.



(a) 2 classes. **(b)** 3 classes. **(c)** 4 classes.

Figure 2: The result that clusters the points successfully.

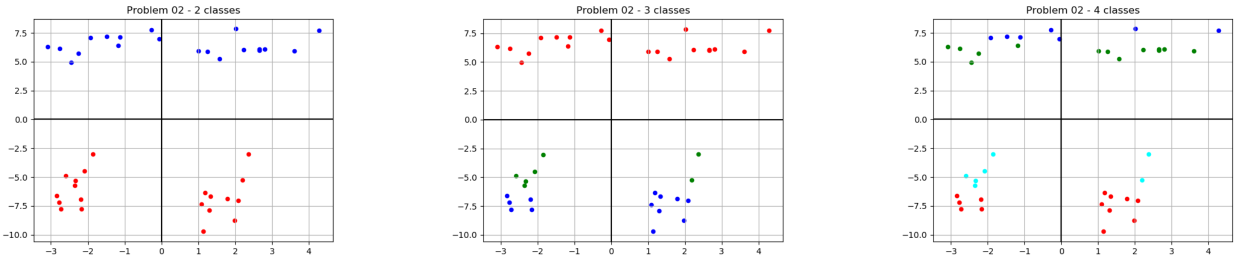
4 Extra experiments

There are 2 following subsections: Kmeans with 1 eigenvector and Kmeans with more than 2 eigenvector.

4.1 Kmeans with 1 eigenvector

The following shows which eigenvector I chosen, and how it affects the result.

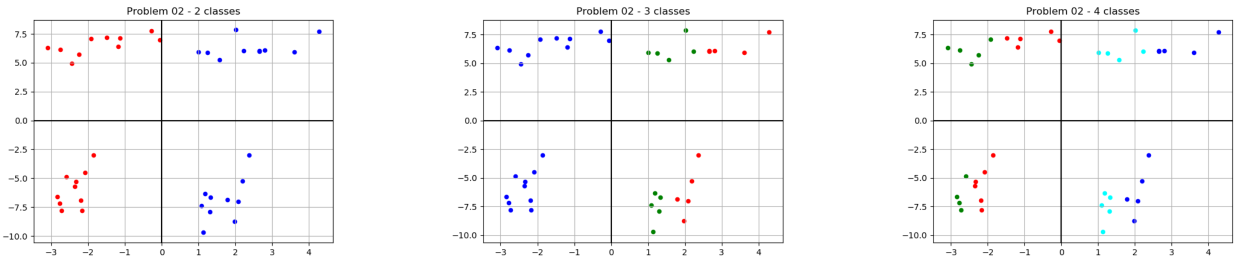
1. 2^{nd} **smallest eigenvector**: It can only cluster the points into 2 classes successfully. Fig. 3 shows the results.



(a) 2 classes. **(b)** 3 classes. **(c)** 4 classes.

Figure 3: Kmeans only with the 2^{nd} smallest eigenvector.

2. 3^{rd} **smallest eigenvector**: It can only cluster the points into 2 classes successfully. Fig. 4 shows the results.

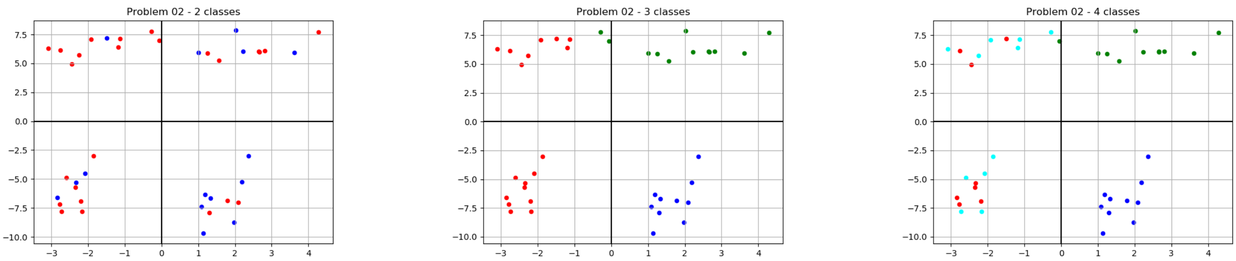


(a) 2 classes. **(b)** 3 classes. **(c)** 4 classes.

Figure 4: Kmeans only with the 3^{rd} smallest eigenvector.

4.2 Kmeans with more than 2 eigenvector

I've shown that we can cluster the points into 2/3/4 classes successfully by 2 eigenvectors. I'll show how many eigenvectors will fail to cluster the points with the 2^{nd} or 3^{rd} smallest eigenvector. Fig. 5 and fig. 6 shows the results.

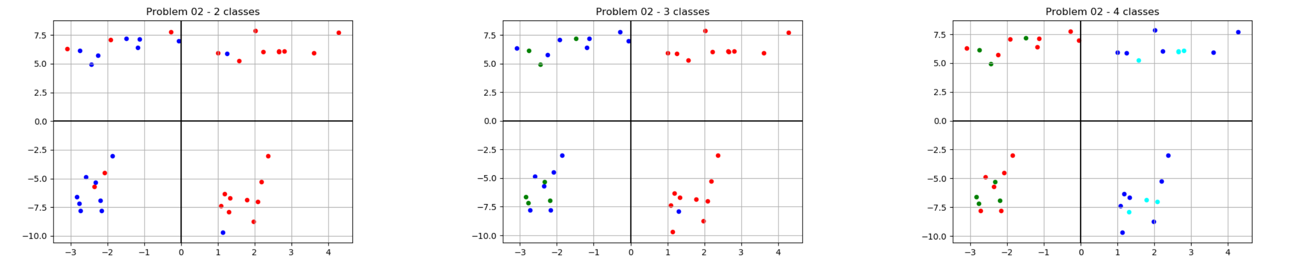


(a) 2^{nd} - 7^{th} smallest eigenvectors will fail to cluster the points into 2 classes.

(b) 2^{nd} - 3^{rd} smallest eigenvectors will fail to cluster the points into 3 classes.

(c) 2^{nd} - 4^{th} smallest eigenvectors will fail to cluster the points into 4 classes.

Figure 5: Show how many small eigenvectors starting from the 2^{nd} smallest eigenvector will fail to cluster the points.



(a) 3^{rd} - 9^{th} smallest eigenvectors will fail to cluster the points into 2 classes.

(b) 3^{rd} - 4^{th} smallest eigenvectors will fail to cluster the points into 3 classes.

(c) 3^{rd} - 4^{th} smallest eigenvectors will fail to cluster the points into 4 classes.

Figure 6: Show how many small eigenvectors starting from the 3^{rd} smallest eigenvector will fail to cluster the points.

1. **Starting from the 2^{nd} smallest eigenvector**
   (a) 2 classes: The 2^{nd} to 7^{th} smallest eigenvectors will fail to cluster them.
   (b) 3 classes: The 2^{nd} to 3^{rd} smallest eigenvectors will fail to cluster them.
   (c) 4 classes: The 2^{nd} to 4^{th} smallest eigenvectors will fail to cluster them.
2. **Starting from the 3^{rd} smallest eigenvector**
   (a) 2 classes: The 3^{rd} to 9^{th} smallest eigenvectors will fail to cluster them.
   (b) 3 classes: The 3^{rd} to 4^{th} smallest eigenvectors will fail to cluster them.
   (c) 4 classes: The 3^{rd} to 4^{th} smallest eigenvectors will fail to cluster them.

5 Discussion

5.1 The S.P.D property of G

In eq 1, G must be S.P.D if the distance matrix satisfies the triangular inequality property. Furthermore, the distance matrix is calculated from a set of 2D points. We may recover the 2D points by the largest 2 eigenvectors. But, in fact, we can only get the approximate 2D points and G is not S.P.D (see fig 7). The distance we get from the generated 2D points just approximate the distance matrix, they do not match. The reason is the floating point error. The position of the origin 2D points are stored as floating point. The floating points error makes G indefinite. Also, this makes us impossible to recover the origin 2D points. However, the largest 2 eigenvectors are enough to approximate the origin 2D points. In fig 7, we can see that the largest 2 eigenvalues are much larger than other eigenvalues This gives us a good approximation.

```
[-2.06801608e-04 -7.12314856e-05 -4.92605226e-05 -3.74994123e-05
 -3.23943807e-05 -2.91311761e-05 -2.56013839e-05 -2.42165130e-05
 -2.01124026e-05 -1.94168284e-05 -1.60698865e-05 -1.43358102e-05
 -1.26243652e-05 -9.75309583e-06 -8.11494093e-06 -6.00217274e-06
 -2.48584511e-06 -8.32415057e-07 -6.14505328e-08 -0.00000000e+00
  2.11255666e-06  5.00416329e-06  5.91411750e-06  8.35513697e-06
  1.00161672e-05  1.22301419e-05  1.30302055e-05  1.48966382e-05
  1.83031937e-05  2.04830411e-05  2.33519992e-05  2.69683908e-05
  3.14676795e-05  3.32847328e-05  3.70416990e-05  4.80710212e-05
  7.02819531e-05  1.90457805e-04  1.94900855e+02  4.13207667e+03]
```

Figure 7: The eigenvalues of G in eq 1

5.2 Number of chosen eigenvector

In my observation, clustering N dimensional points with N eigenvector gets a better results. I don't know how to prove my observation mathematically. I think that less dimension provides less information, and too many dimensions suffer from the curse of dimensionality.