HW4 - Graph Spectra

Tianxiao Zhao <tzh@kth.se>
Yantian You <yantian@kth.se>

1. Build and run

There are two matlab scripts. ‘baseline.m’ is for graph clustering and ‘gridSearch.m’ is for searching the best sigma. To run our program, just simply replaced data file name in the scripts and then call them in matlab. Value k should be modified corresponding to each graph.

2. Solution

In this homework, we study and implement the spectral graph clustering algorithm in the paper *On Spectral Clustering: Analysis and an algorithm*. The idea is to use the *k* eigenvectors derived from a graph’s Laplacian simultaneously to obtain a *k* way partitioning of the graph. The detailed algorithm is followed below:

Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-\|s_i - s_j\|^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Note that there exists two parameters that may affect the results in the algorithm, namely the number of clusters *k* and the scaling parameter sigma. For the number of clusters *k*, we choose it by first plotting the points in 2D or 3D dimension and then deciding how many clusters there should be according to the distribution. As for the parameter sigma, we simply search over a predefined range of it, and pick the value that gives the smallest distortion after clustering Y’s rows by K-means.

Another issue that happens in our implementation is that the matlab function *eig()* gives double complex solutions when called. This is due to the fact that round-off error during calculations makes the matrix *L* non-symmetric. To deal with this problem, we employ a workaround to convert double precision data to single precision before calculating the

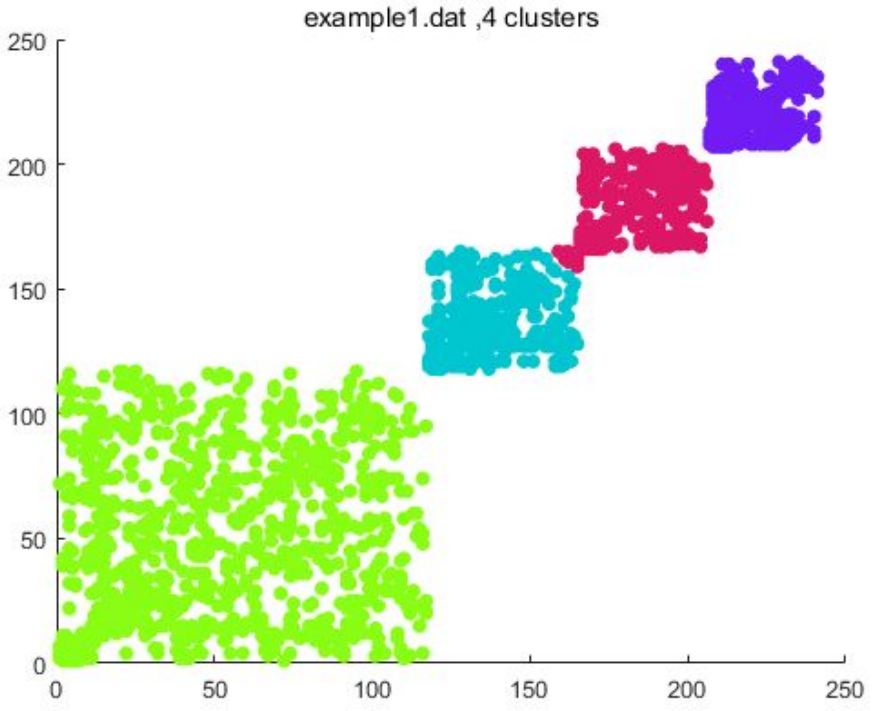
eigenvalues. In this way, matrix L will become symmetric as what we expected and thus we can get real eigenvalues.

3. Results

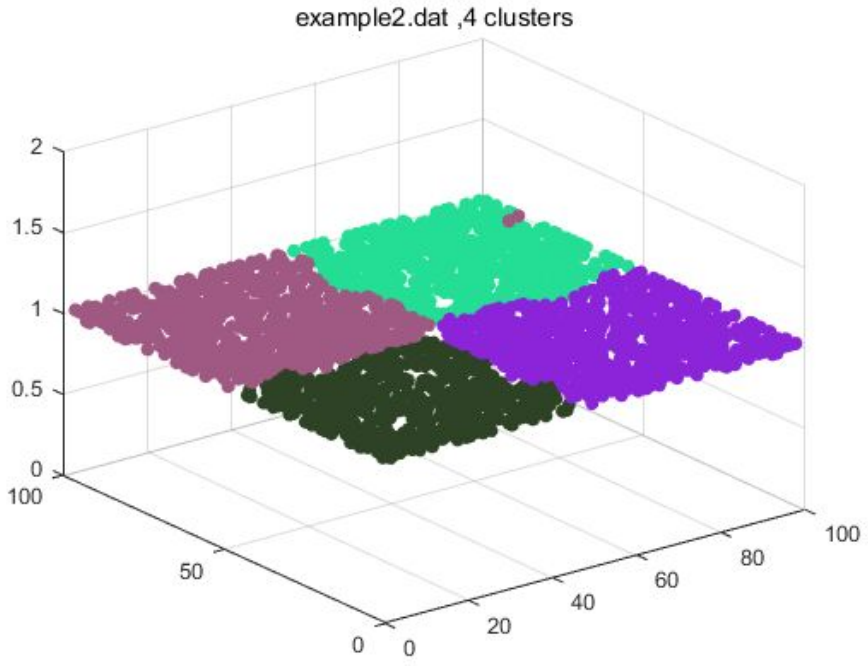
3.1 Performance on test cases

To test out program, we ran our code on different test cases including example1.dat, example2.dat and some other test cases we found on Internet, see below.

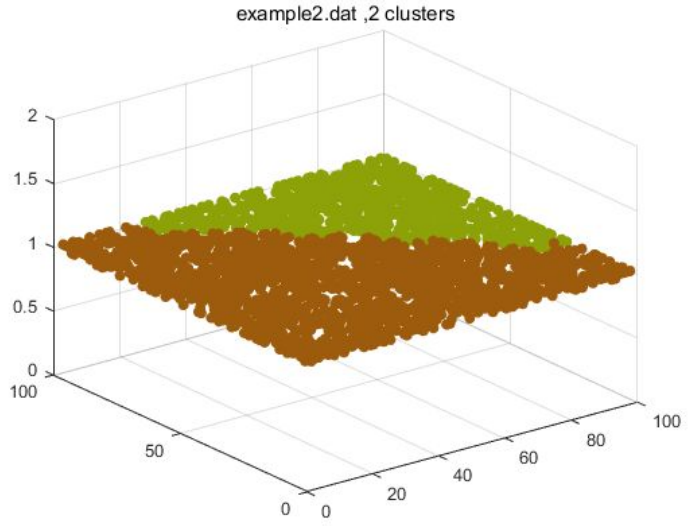
1. example1.dat
   K = 4, sigma = 1.5



2. example2.dat
   K = 4, sigma = 1.5

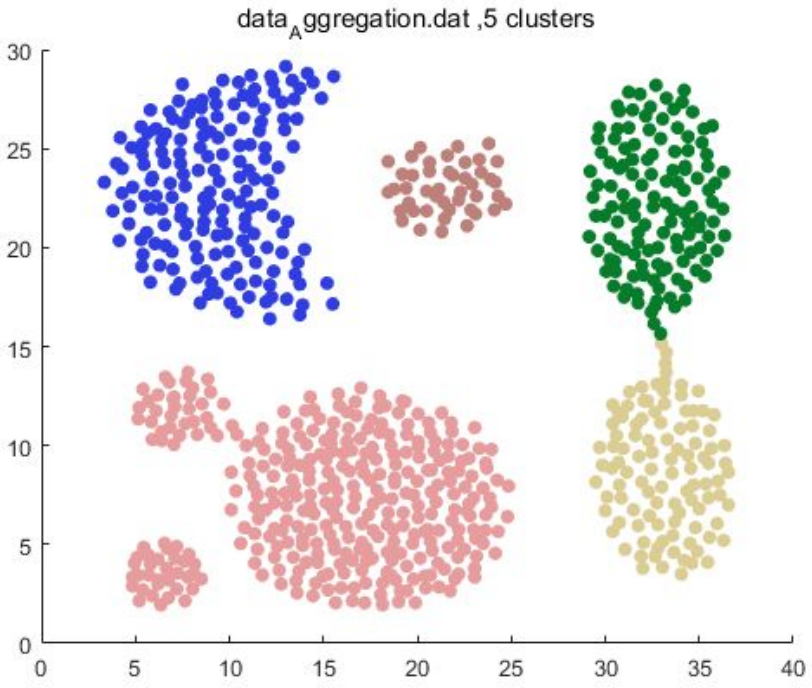


K = 2, sigma = 1



3. data_Aggregation.dat

K = 5, sigma = 1.5

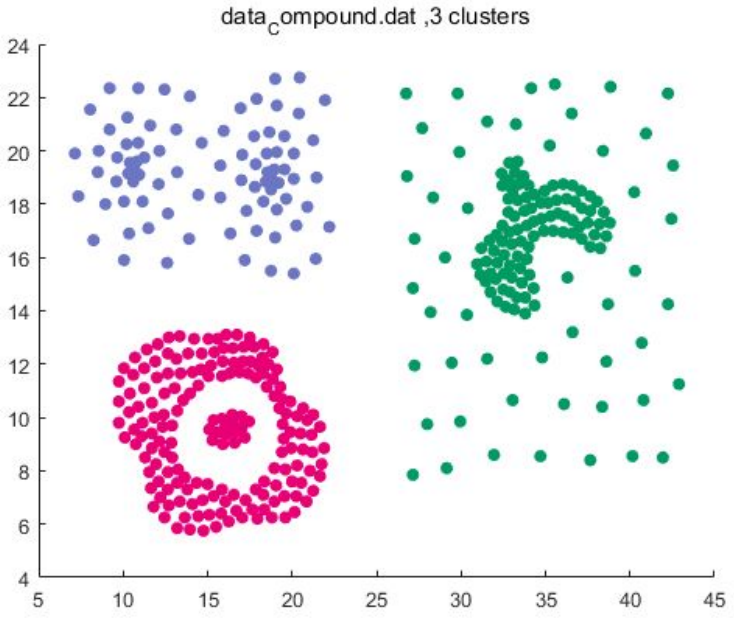


4. data_Bridge.dat

K = 2, sigma = 1.5

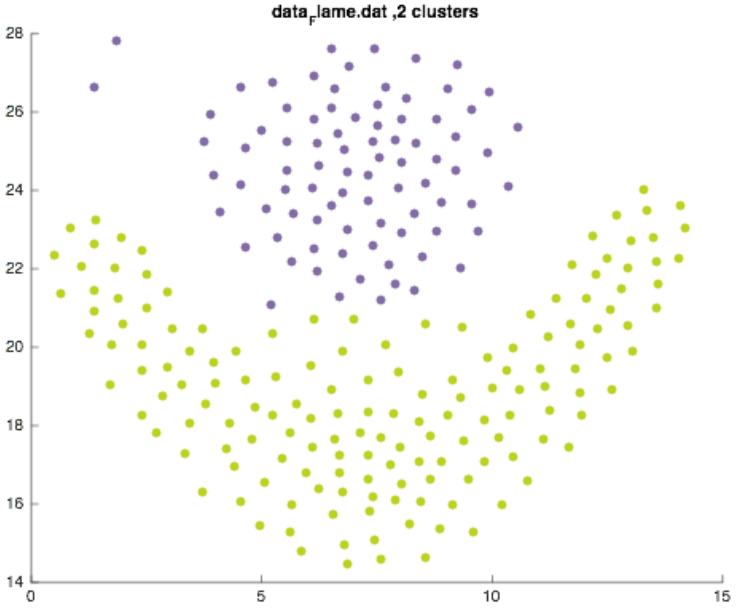
5. data_Compound.dat

K = 3, sigma = 1.5



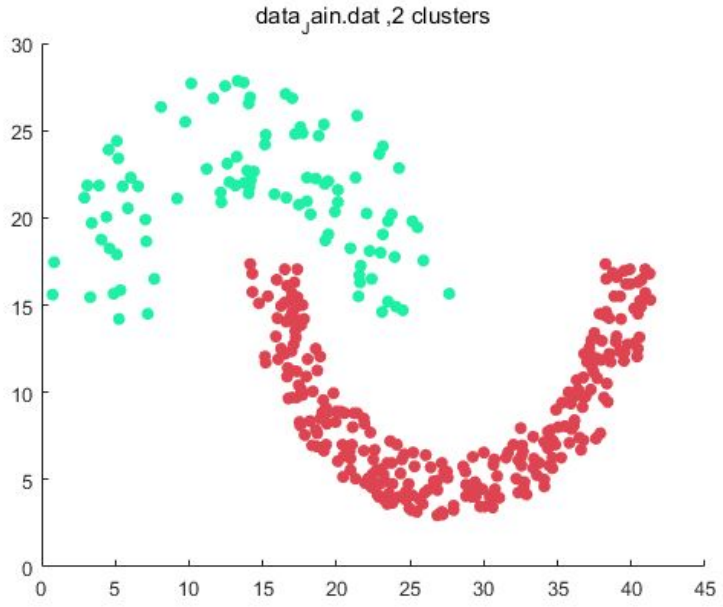
6. data_Flame.dat

K = 2, sigma = 1.5



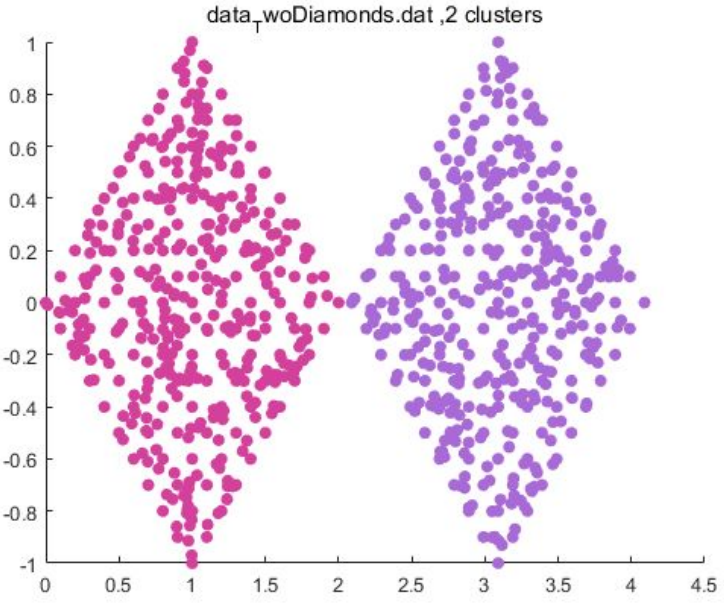
7. data_Jain.dat

K = 2, sigma = 1.5



8. data_TwoDiamonds.dat

K = 2, sigma = 1.5



3.2 Finding best sigma

We did some experiment on how to find the best sigma for K-mean clustering. By running the gridSearch.m script we achieved the best sigma value with smallest sums of point-to-centroid distances for example1 and example2.

example1.dat (K=4) :

sigma_optimal1 = 1.5

example2.dat (K=2) :

sigma_optimal2 = 1

4. Conclusion

From the results we could found out that the K-mean algorithm could cluster spectral graph appropriately. However, the number of clusters should be carefully chosen for each graph. For example, if we choose K=4 and Sigma=0.5 for example2 data set, the graph can not be clearly separated even though we can achieve minimum sum of point-to-centroid distances.