Graph Spectra

Introduction

The spectral graph clustering algorithm is implemented in the paper "On Spectral Clustering: Analysis and an algorithm" by Andrew Y. Ng, Michael I. Jordan, Yair Weiss. The following figure describes how to implement this algorithm :

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.

How to run

We implemented this assignment in Matlab, so after opening the main.m file, one can edit the 3 variables to test the code :

- K = the number of clustering we want to test
- Sigma = the sigma in Affinity Matrix A
- Filename = the filename of testing data

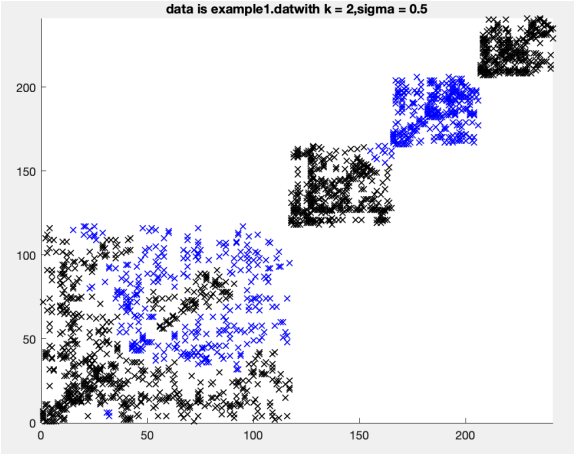
After editing 3 variables, one can just press Run and will get a figure on the result.

Result

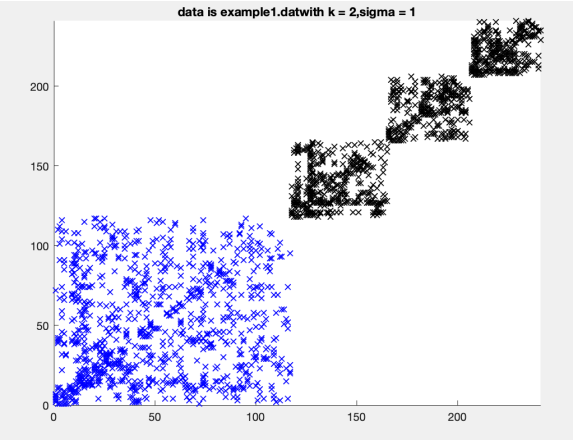
In the result part, two datasets are tested and we will test several different k and sigma and find the best clustering group.

Example1.dat

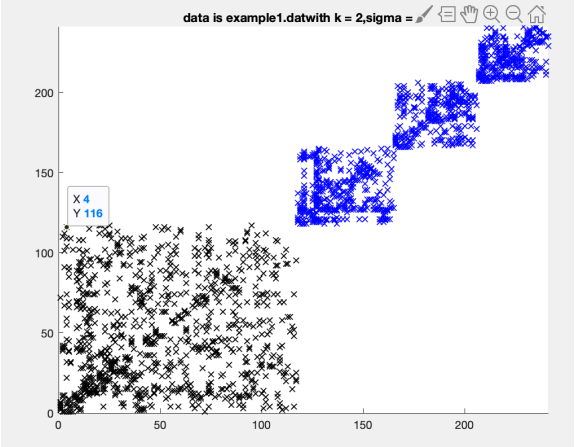
K = 2 and sigma = 0.5



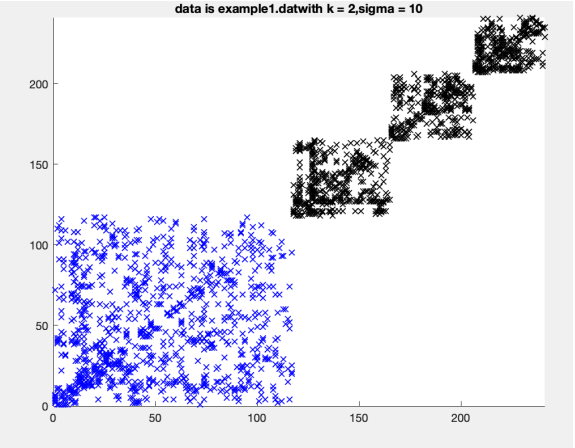
K = 2 and sigma = 1



K = 2 and sigma = 2

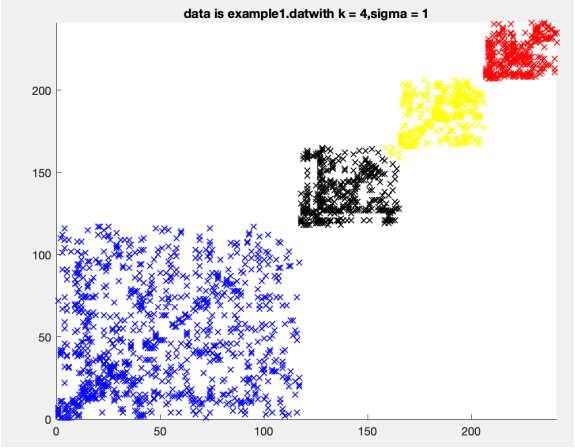


K = 2 and sigma = 10

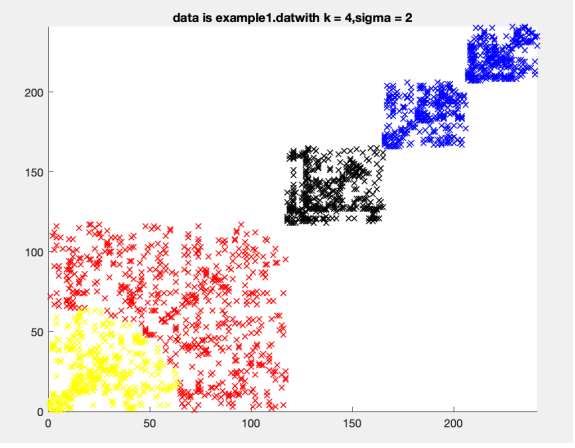


From the previous figures, we know that we should choose sigma larger than 1 so that it can capture the feature of clustering more clearly.

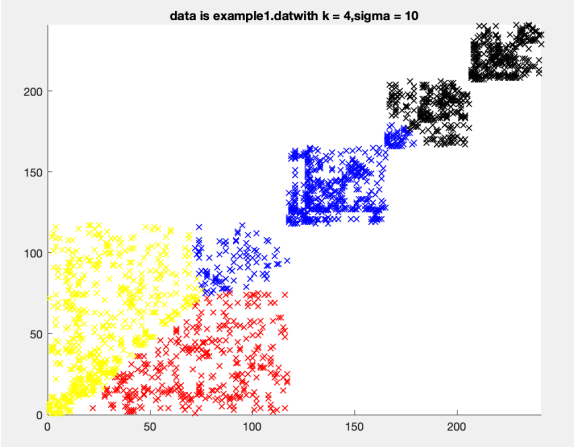
K = 4 and sigma = 1



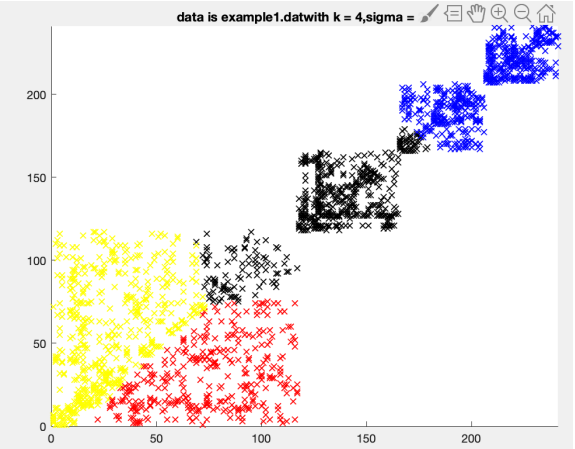
K = 4 and sigma = 2



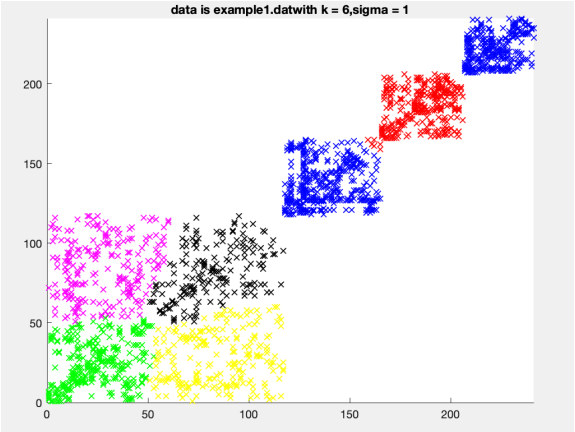
K = 4 and sigma = 10



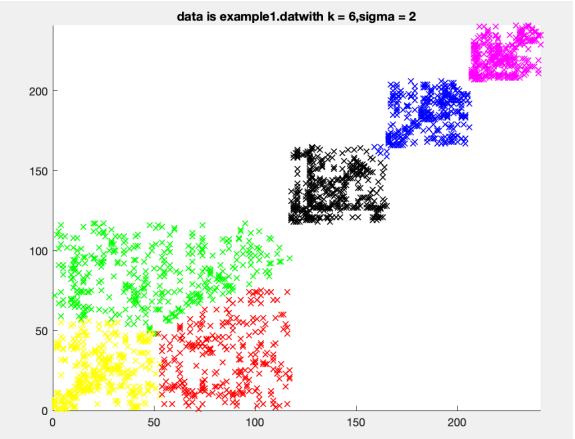
K = 4 and sigma = 20



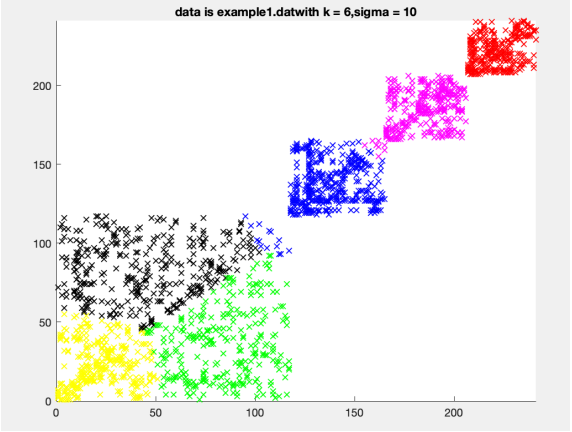
K = 6 and sigma = 1



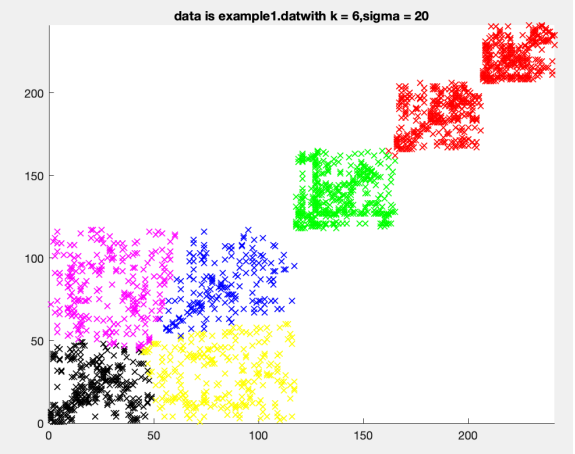
K = 6 and sigma = 2



K = 6 and sigma = 10



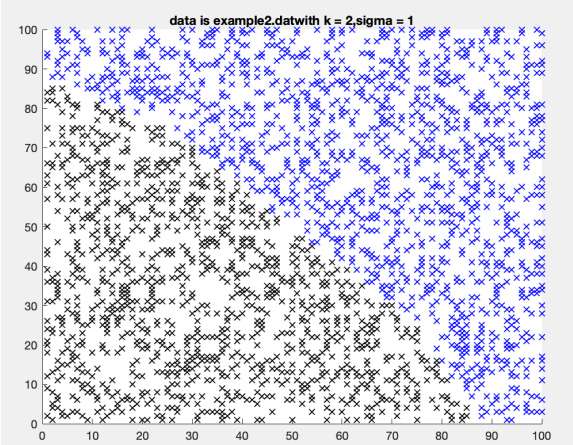
K = 6 and sigma = 20



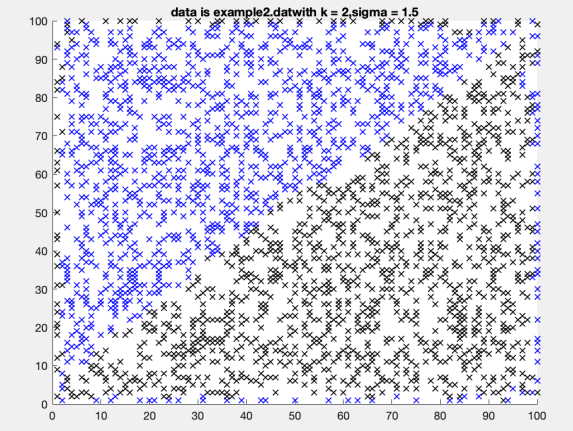
After trying for different K and sigma, we found that when K = 4 and sigma = 2 the algorithm gives a reasonable result in example1.dat.

Example2.dat

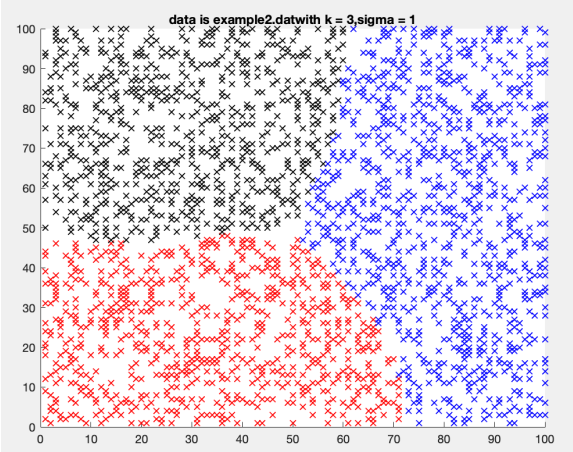
K = 2 sigma = 1



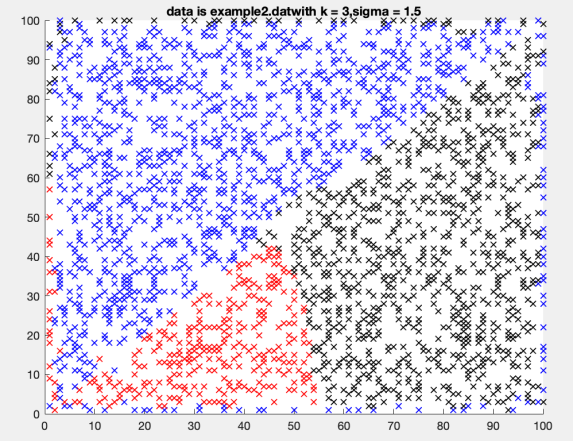
K = 2 and sigma = 1.5 with eig(...,'SA')



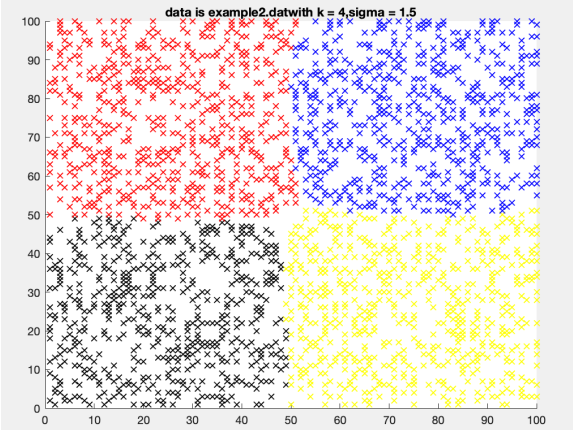
K = 3 sigma = 1



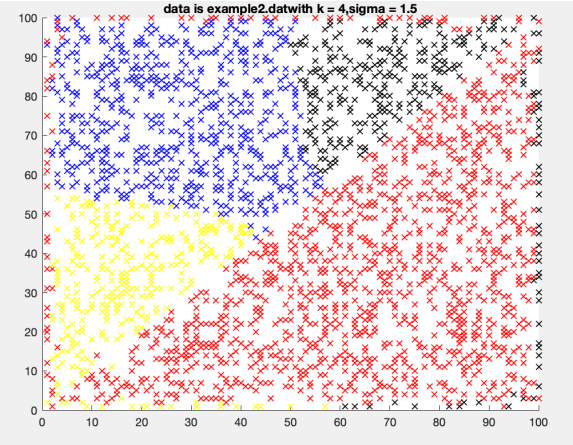
K = 3 sigma = 1 with eig(...,'SA')



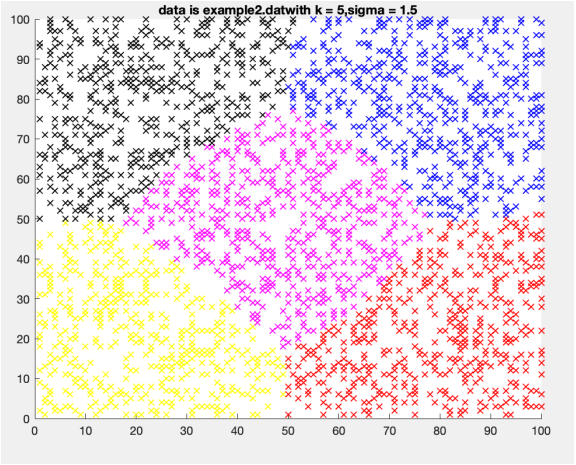
K = 4 sigma =1.5



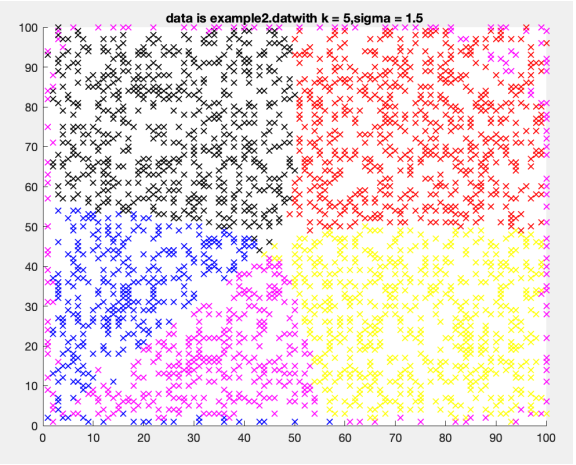
K = 4 sigma =1.5 with eig(...,'SA')



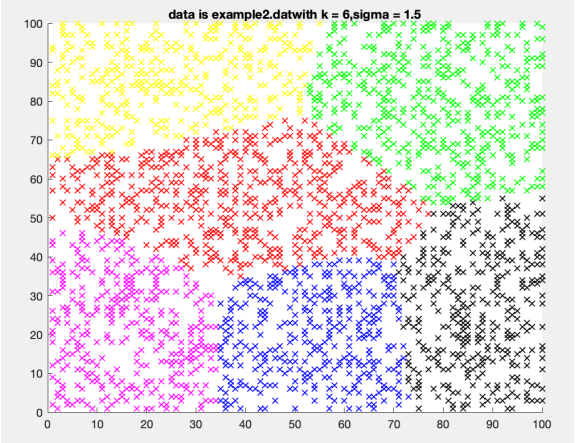
K = 5 sigma = 1.5



K = 5 sigma = 1.5 with eig(...,'SA')



K = 6 sigma = 1.5



K = 6 sigma = 1.5 with eig(...,'SA')