Homework #3

Sam Fleischer

May 16, 2016

Problem 1

Download the dataset `crescents.mat` from the Class website and load it into Matlab. It contains two-hundred points in two dimensions. If you plot it, you see that the data form two half-moon-like clusters. Clearly, k-means directly applied to this dataset will fail to cluster the data according to these two shapes. Use the graph Laplacian or diffusion maps (followed by k-means) to try to cluster the data as good as possible according to the half-moon shapes. You can use Matlab's k-means function to do the actual clustering once you transformed the data.

Proof. Figures 1 and 2 shows the data and the result after spectral clustering.

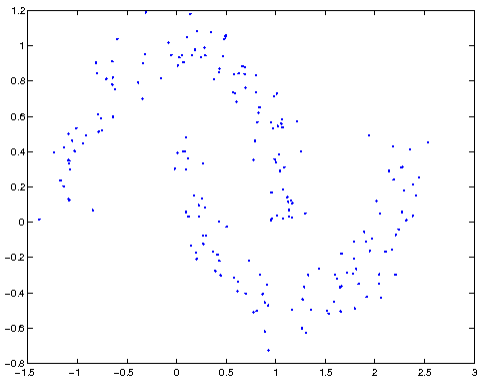

Figure 1: Original Data

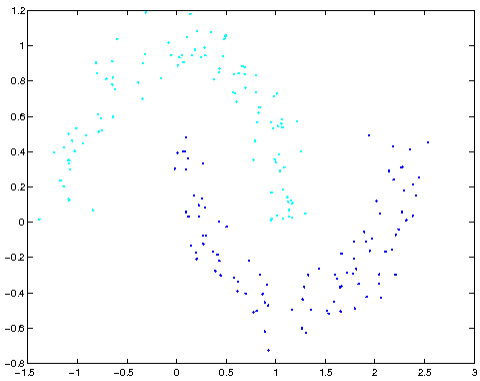

Figure 2: Data after spectral clustering

Here are the steps taken to complete this assignment:

I. Define the weight function w.

$$w_{i,j} = w(p_i, p_j) = \exp\left[-100\left\|p_i - p_j\right\|_2^2\right]$$

II. Define the adjacency matrix A.

$$A_{i,j} = \begin{cases} w_{i,j} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

III. Define the graph-Laplacian $L = D - A$ where D is a diagonal matrix with

$$D_{i,i} = \sum_{j=1}^{n} A_{i,j} \qquad \text{and} \qquad D_{i,j} = 0 \quad \text{for } i \neq j$$

IV. Find the eigenvalues and eigenvectors of L and arrange them in ascending order. These eigenvalues must necessarily all be non-negative.

V. Define v_2 to be the eigenvector corresponding to the second-smallest eigenvalue. If the i^{th} component of v_2 is positive, we associate the i^{th} data point with the first cluster. Otherwise, we associate the i^{th} data point with the second cluster.

□

Problem 2

Download the dataset `genomedata.mat` from the Class website; it contains Single Nucleid Polymorphisms data from the Human Genome Diversity Project. The data consists of an array consisting of 5000 rows, each row has 1043 different strings. The 5000 rows are Single Nucleid Polymorphisms, the columns correspond to 1043 different individuals. The entries are not numerical values (quite annoyingly), but contain the characters 'AA', 'CC', 'GG', 'TT', 'AG', 'AC', 'TC', 'TG', and (even more annoyingly) also '--', the latter represents missing measurements. Your goal is to cluster the data into a small set of clusters. After loading the file into Matlab, you need to convert the characters into numerical values. It is up to you which conversion you use (you can use the file `gen2vec.m` to do the actual conversion, once you have chosen a conversion rule). Since the data are high-dimensional you first need to reduce the dimension before clustering. You should attempt the dimension reduction via PCA as well as via diffusion maps. In both cases you need to decide how many dimensions you want to use. Also, in both cases you may want to use Matlab's k-means function to do the actual clustering after dimension reduction. Note: Your results may differ from mine, because you will likely choose a different conversion rule. I did not get a meaningful clustering via PCA, but did achieve reasonable clustering via diffusion maps.

Proof. First, define the following map from $\{\texttt{AA},\texttt{CC},\texttt{GG},\texttt{TT},\texttt{AG},\texttt{AC},\texttt{TG},\texttt{TC},\texttt{--}\}$ to the unit cirtcle (and 0) in $\mathbb{C}$:

$$\begin{aligned}
\texttt{AA} &\mapsto 1 \\
\texttt{TT} &\mapsto -1 \\
\texttt{CC} &\mapsto i \\
\texttt{GG} &\mapsto -i \\
\texttt{AC} &\mapsto \frac{1+i}{\sqrt{2}} \\
\texttt{AG} &\mapsto \frac{1-i}{\sqrt{2}} \\
\texttt{TC} &\mapsto \frac{-1+i}{\sqrt{2}} \\
\texttt{TG} &\mapsto \frac{-1-i}{\sqrt{2}} \\
\texttt{--} &\mapsto 0
\end{aligned}$$

We choose this map since `A` never pairs with `T` and `C` never pairs with `G`. Transform the data given in genomedata.mat in to complex vectors using this map. We get a 5000×1043 matrix of complex numbers, which represents 1043 data points in 5000 dimensions. Next we define the weight function w

$$w_{i,j} = w(p_i, p_j) = \exp\left[-\frac{\|p_i - p_j\|_2^2}{1700}\right],$$

and the adjacency matrix W

$$W_{i,j} = \begin{cases} w_{i,j} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}$$

Then define the matrix $M = D^{-1}W$ where D is a diagonal matrix with

$$D_{i,i} = \sum_{j=1}^{n} A_{i,j} \qquad \text{and} \qquad D_{i,j} = 0 \quad \text{for } i \neq j.$$

Next, denote M_S as the normalized matrix M:

$$M_S = D^{\frac{1}{2}} M D^{-\frac{1}{2}}.$$

Finally, perform a spectral decomposition of M_S and done it as $M_S = V\Lambda V^T$ where Λ is the diagonal matrix containing the eigenvalues of M_S and the columns of V contain the eigenvectors of M_S with respect to their corresponding values in Λ. Noting that M_S always has an eigenvalue with norm 1, we then use MATLAB to find the next four largest (in norm) eigenvalues of M_S. They are:

$$\lambda_2 \approx 0.1024963 \qquad \lambda_3 \approx 0.0721643 \qquad \lambda_4 \approx 0.0522477 \qquad \lambda_5 \approx 0.0192484$$

Note how quickly the eigenvalues decay in Figure (3) (this is a log-log plot). Only the first few are greater than 10^{-2}. We then construct the following diffusion map U:

$$U = \begin{bmatrix} \lambda_1^t\phi_1(1) & \lambda_1^t\phi_1(2) & \dots & \lambda_1^t\phi_1(1043) \\ \lambda_2^t\phi_2(1) & \lambda_2^t\phi_2(2) & \dots & \lambda_2^t\phi_2(1043) \\ \lambda_3^t\phi_3(1) & \lambda_3^t\phi_3(2) & \dots & \lambda_3^t\phi_3(1043) \\ \lambda_4^t\phi_4(1) & \lambda_4^t\phi_4(2) & \dots & \lambda_4^t\phi_4(1043) \end{bmatrix}$$

In order to visualize the relationship of these columns, we compare their magnitudes and arguments in Figure (4). □

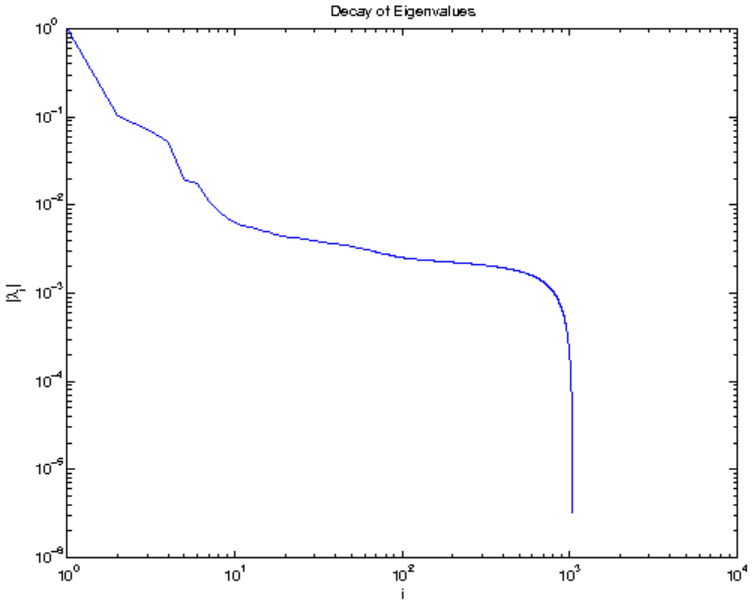

Figure 3: Decay of Eigenvalues

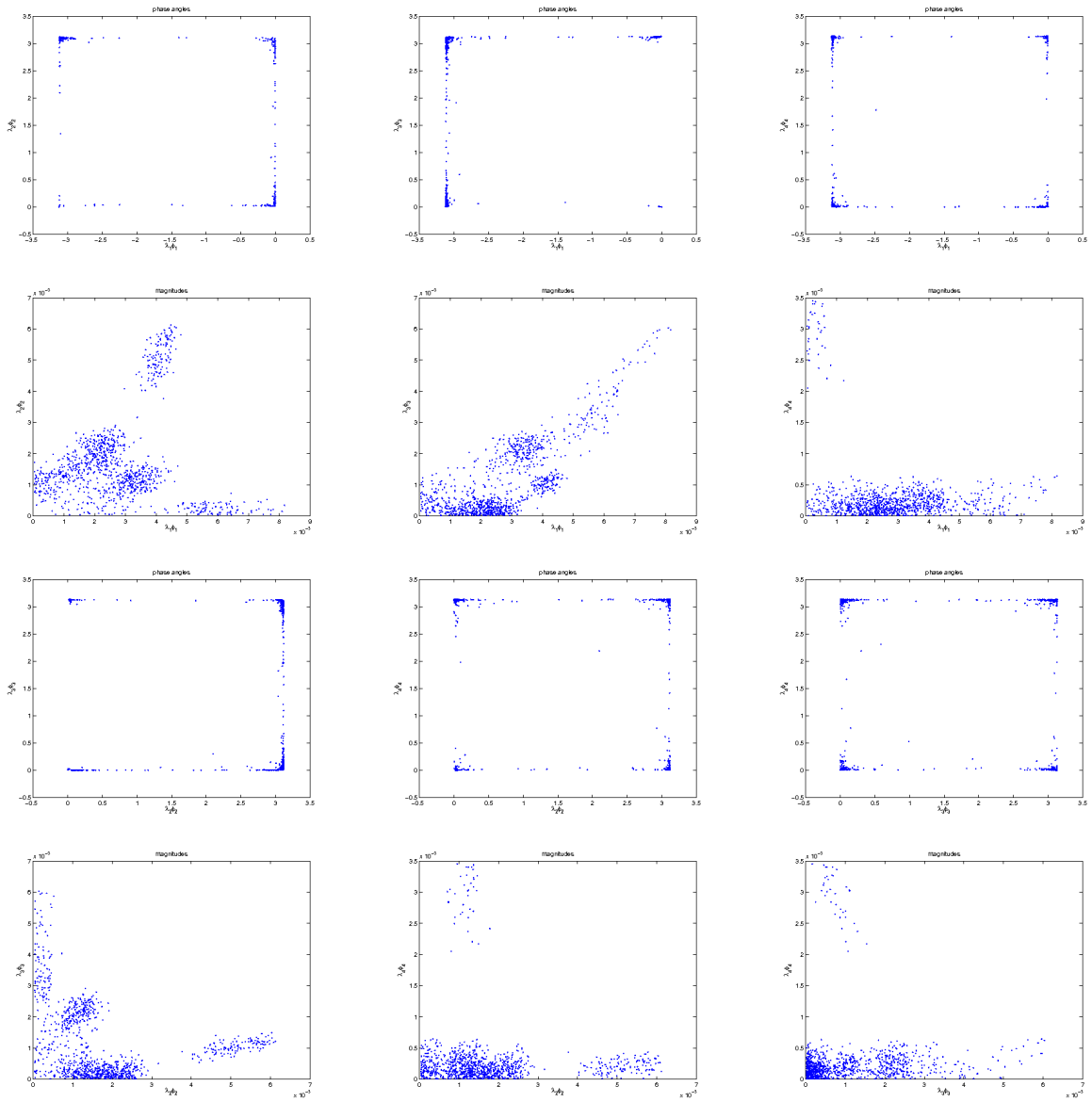

Figure 4: Comparisons of arguments and magnitudes of the columns in U. The top six figures compare column 1 with columns 2, 3, and 4. The bottom left four figures compare column 2 with columns 3 and 4, and the bottom right two figures compare column 3 and column 4.