Boston University
Department of Electrical and Computer Engineering

ENG EC 500 B1 (Ishwar) Introduction to Learning from Data

Matlab Exercise 4 Solution

Issued: Wed 18 Nov 2015 **Due:** 4pm Tue 1 Dec 2015 PHO440 box + Blackboard
Required reading: Your notes from lectures and additional notes on website on spectral clustering.

- This homework assignment requires some programming background in MATLAB. Please refer to the following link for an introduction (or review) of MATLAB:
`http://www.math.ucsd.edu/~bdriver/21d-s99/matlab-primer.html`
- You will be making two submissions: (1) A paper submission in the box outside PHO440. (2) An electronic submission of all your matlab code to blackboard (in a single zipped file appropriately named as described below).
- **Paper submission:** This must include all plots, figures, tables, numerical values, derivations, explanations (analysis of results and comments), and also printouts of all the matlab .m files that you either created anew or modified. Submit color printouts of figures and plots whenever appropriate. Color printers are available in PHO307 and PHO305. Be sure to annotate figures, plots, and tables appropriately: give them suitable *titles* to describe the content, label the *axes*, indicate *units* for each axis, and use a *legend* to indicate multiple curves in the plots. Please also explain each figure properly in your solution.
- **Blackboard submission:** All the matlab .m files (and only .m files) that you either create anew or modify must be appropriately named and placed into a **single** directory which should be zipped and uploaded into the course website. Your directory must be named as follows: `<yourBUemailID>_matlab4`. For example, if your BU email address is `mary567@bu.edu` you would submit a single directory named: `mary567_matlab4.zip` which contains all the matlab code (and only the code).
- **File naming convention:** Instructions for file names to use are provided for each problem. As a general rule, each file name must begin with your BU email ID, e.g., `mary567_<filename>.m`. The file name will typically contain the problem number and subpart, e.g., for problem 4.1b, the file name would be `mary567_matlab4_1b.m`. Note that the dot . in 4.1 is replaced with an underscore (this is important).

Problem 4.1 ***k*-means vs. Spectral Clustering**
In this problem we will use a circle-shaped dataset and a spiral-shaped dataset. Figure 1 shows examples for circle and spiral shaped datasets with 2 clusters. These synthetic examples can be generated using the provided functions `sample_circle.m` and `sample_spiral.m`.

(a) Use `sample_circle.m` to sample a circle-shaped dataset with $k = 3$ clusters and 500 points for each cluster (denoted as $\mathcal{D}_1$). Similarly, use `sample_spiral.m` to sample a spiral-shaped dataset with $k = 3$ clusters and 500 points for each cluster (denoted as $\mathcal{D}_2$). Use MATLAB's built-in function `kmeans` to cluster $\mathcal{D}_1$ and $\mathcal{D}_2$. **Important:** in any part of this assignment which requires running `kmeans`, before running it, set: `rng(2)`. For both datasets, set `'Replicates'` to 20 and `'Distance'` to `'sqeuclidean'`. We will explore k-means clustering with $k = 2, 3, 4$ to understand how results change when the specified value of k differs from the true value of k. For each choice of k:

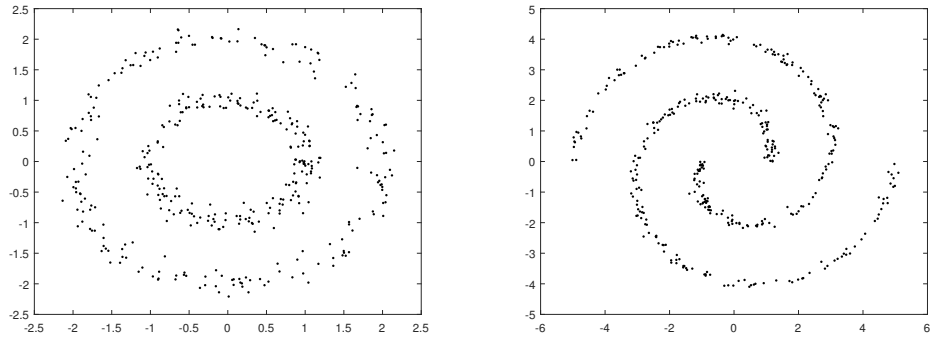

Figure 1: *Example circle-shaped (Left) and spiral-shaped (Right) dataset, each with $k = 2$ clusters and 200 points per cluster.*

(i) **Plot** all the data points in $\mathcal{D}_1$ (resp. $\mathcal{D}_2$), and indicate the cluster assignment by coloring the data points in different colors: For $k = 2$, use red and blue; For $k = 3$, use red, blue, and green; For $k = 4$, use red, blue, green, and black. On the same figure, plot the cluster centers. You need to create 6 plots in total. You can use `subplot` to save space.

(ii) **Report** the overall within-cluster sums of points-to-cluster-centroid (Euclidean) ℓ_2 squared distances (**for each cluster**).

(b) We will next **implement** three variants of spectral clustering. Given the $n \times n$ similarity matrix $\mathbf{S}$ and the adjacency matrix $\mathbf{W}$, we use the following three different spectral clustering algorithms:

Un-normalized spectral clustering (SC-1):

- Compute the unnormalized graph Laplacian $\mathbf{L}$.
- Compute the first k eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of $\mathbf{L}$.
- Let $\mathbf{V} \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ as columns.
- Cluster the n rows of $\mathbf{V}$ with the k-means algorithm into k clusters (set: `rng(2)` before running `kmeans` and use default options).

Normalized spectral clustering 1 (SC-2):

- Compute the normalized graph Laplacian $\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}$.
- Compute the first k eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of $\mathbf{L}_{rw}$.
- Let $\mathbf{V} \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ as columns.
- Cluster the n rows of $\mathbf{V}$ with the k-means algorithm into k clusters (set: `rng(2)` before running `kmeans` and use default options).

Normalized spectral clustering 2 (SC-3):

- Compute the normalized graph Laplacian $\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.
- Compute the first k eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of $\mathbf{L}_{sym}$.
- Let $\mathbf{V} \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ as columns. Normalize (by scaling) the rows of $\mathbf{V}$ so that their ℓ_2 norms are 1.
- Cluster the n rows of $\mathbf{V}$ with the k-means algorithm into k clusters (set: `rng(2)` before running `kmeans` and use default options).

Implement the above three spectral clustering algorithms. Apply them to $\mathcal{D}_1$ and $\mathcal{D}_2$ created in part (a). Use the Gaussian similarity $S(i,j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right\}$ to construct $\mathbf{S}$ with $\sigma = 0.2$. We will use the *fully-connected* graph and set $\mathbf{W} = \mathbf{S}$.

(i) **Plot** the eigenvalues of $\mathbf{L}$, $\mathbf{L}_{rw}$, and $\mathbf{L}_{sym}$ for $\mathcal{D}_1$ and $\mathcal{D}_2$ in ascending order. (You need to create 6 plots in total. You can use `subplot` to save space.)

(ii) Set $k = 2, 3, 4$ in your spectral clustering algorithms. For each choice of k, **plot** all the data points in $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) and indicate the cluster assignment from SC-3 using different **colors**: For $k = 2$, use red and blue; For $k = 3$, use red, blue, and green; For $k = 4$, use red, blue, green, and black. You need to create 6 plots in total. You can use `subplot` to save space.

(iii) For $k = 3$, use `plot3` to plot the rows of the $\mathbf{V}$ matrices in SC-1, SC-2, and SC-3. For SC-3, first normalize the rows of $\mathbf{V}$ before plotting them. Generate these plots for both the $\mathcal{D}_1$ and $\mathcal{D}_2$ datasets. Indicate the corresponding cluster assignments using red, blue, and green colors. You need to create 6 plots in total. You can use `subplot` to save space. *Note:* Understand that since $k = 3$, each row of a $\mathbf{V}$ matrix has 3 columns. When you use `plot3` to plot the rows of $\mathbf{V}$, each row is mapped to a point in 3-D space with coordinates given by the entries in the three columns. So these are all 3-D plots.

MATLAB functions: `eig, eigs, svd, kmeans`.

(c) Transform the Cartesian coordinates representation of each data point in $\mathcal{D}_1$ into polar coordinates using `cart2pol`. We denote this new dataset as $\mathcal{D}_3$. Now apply the k-means algorithm to $\mathcal{D}_3$ (set: `rng(2)` before running it). Also set `'Replicate'` to 20 and `'Distance'` to `'cityblock'`. For each choice of $k = 2, 3, 4$ in `kmeans`,

(i) **Plot** all the data points in $\mathcal{D}_3$ **with radius along the x-axis and angle along the y-axis**, and indicate the cluster assignment by coloring the data points in different colors: For $k = 2$, use red and blue; For $k = 3$, use red, blue, and green; For $k = 4$, use red, blue, green, and black. On the same figure, plot the cluster centers. You need to create 3 plots in total. You can use `subplot` to save space.

(ii) **Report** the overall within-cluster sums of points-to-cluster-centroid (Euclidean) ℓ_2 squared distances (**for each cluster**).

Hint: when applying k-means to data whose attributes are of different units (say, radius and angle), a standard pre-processing step is to apply a linear transform to each attribute so that the minimum (resp. maximum) of each attribute is 0 (resp. 1). Apply this pre-processing step in part (c).

Solution:

(a) For the circle dataset $\mathcal{D}_1$, the cluster assignments produced by the k-means algorithm, for $k = 2, 3, 4$, are depicted in Fig. 2. For the realization shown in Fig. 2, the WCSS values for clusters are as follows:

- $k = 2 : 2.2138 \times 10^3, 2.2760 \times 10^3$;
- $k = 3 : 0.9097 \times 10^3, 1.0401 \times 10^3, 0.8921 \times 10^3$;
- $k = 4 : 0.4447 \times 10^3, 0.5380 \times 10^3, 0.5478 \times 10^3, 0.569 \times 10^3$

For the spiral dataset $\mathcal{D}_2$, the cluster assignments produced by the k-means algorithm, for $k = 2, 3, 4$, are depicted in Fig. 3. For the realization shown in Fig. 3, the WCSS values for clusters are as follows:

- $k = 2 : 4.3206 \times 10^3, 5.3915 \times 10^3$;

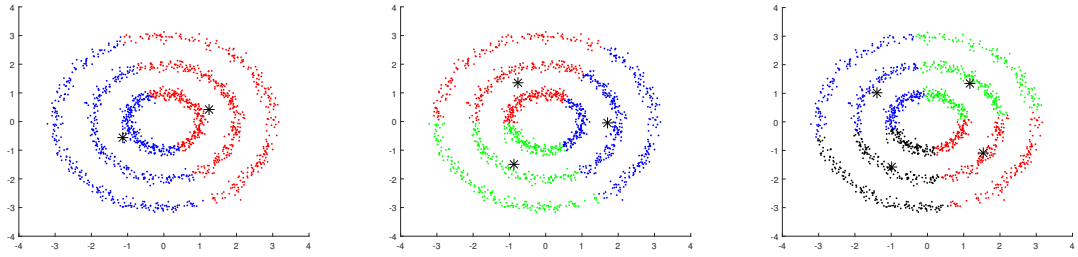

Figure 2: *Cluster assignments for the* **circle dataset** $\mathcal{D}_1$ *produced by the k-means algorithm for* $k = 2$ *(left),* $k = 3$ *(middle), and* $k = 4$ *(right). Different colors indicate different clusters and the black star-points indicate the centroids of each cluster. This figure is best viewed in color.*

- $k = 3 : 1.9785 \times 10^3, 1.8382 \times 10^3, 1.8657 \times 10^3$;
- $k = 4 : 0.9904 \times 10^3, 1.2034 \times 10^3, 0.9751 \times 10^3, 1.2300 \times 10^3$

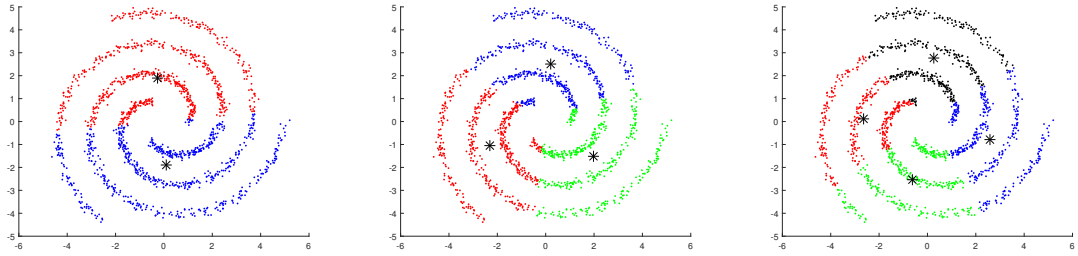

Figure 3: *Cluster assignment for the* **spiral dataset** $\mathcal{D}_2$ *produced by the k-means algorithm for* $k = 2$ *(left),* $k = 3$ *(middle), and* $k = 4$ *(right). Different colors indicate different clusters and the black star-points indicate the centroids of each cluster. This figure is best viewed in color.*

Figures 2 and 3 show that the k-means clustering algorithm fails to capture the cluster structure (using the standard squared Euclidean distance in the original representation space). The WCSS decreases as k increases for both $\mathcal{D}_1$ and $\mathcal{D}_2$.

(b) (i) The eigenvalues for three different graph Laplacians are depicted in Fig. 4 for the circle dataset $\mathcal{D}_1$ and in Fig. 5 for the spiral dataset $\mathcal{D}_2$. One can observe clear transitions of eigenvalues for $\mathbf{L}_{\text{rw}}$ and $\mathbf{L}_{\text{sym}}$.

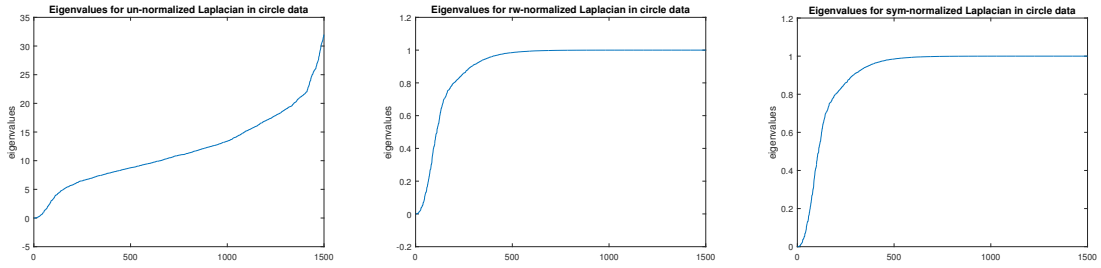

Figure 4: *Eigenvalues for the un-normalized (left), rw-normalized (middle), and sym-normalized (right) graph Laplacians for the* **circle dataset** $\mathcal{D}_1$*. The eigenvalues are sorted in ascending order.*

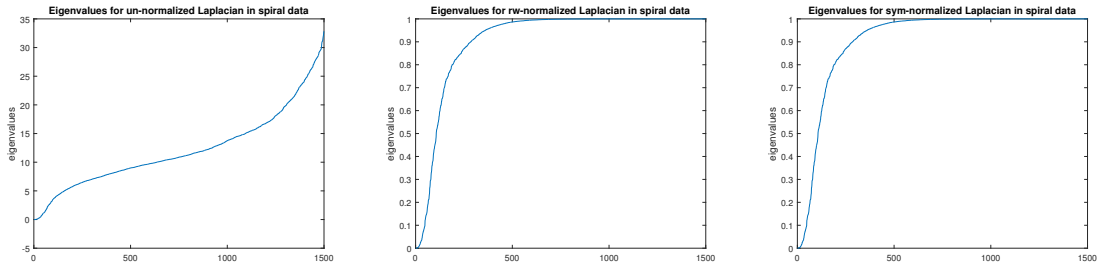

Figure 5: *Eigenvalues for the un-normalized (left), rw-normalized (middle), and sym-normalized (right) graph Laplacians for the* **spiral dataset** $\mathcal{D}_2$*. The eigenvalues are sorted in ascending order.*

(ii) For the circle dataset $\mathcal{D}_1$, the cluster assignment produced by the SC-3 algorithm for $k = 2, 3, 4$ are depicted in Fig. 6. When $k = 3$, the SC-3 algorithm can correctly identify the true clusters. We also note that when we set $k = 2$, the SC-3 algorithm merges two clusters into one cluster (the red cluster in Fig. 6 left). On the other hand, when we set $k = 4$, SC-3 splits one cluster into two clusters (the green and black clusters in Fig. 6 right).

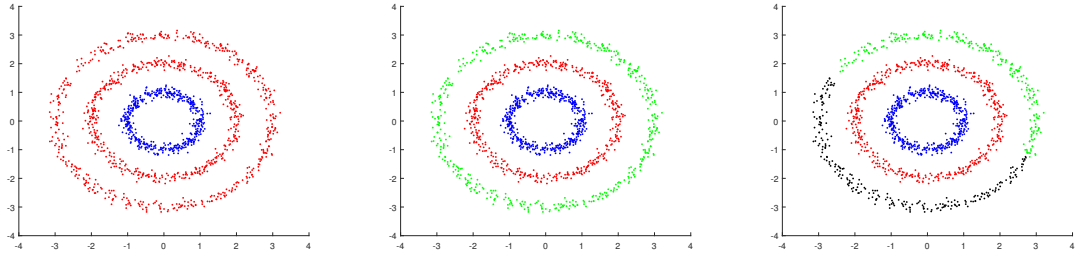

Figure 6: *Cluster assignment for the* **circle dataset** $\mathcal{D}_1$ *using SC-3 for* $k = 2$ *(left),* $k = 3$ *(middle), and* $k = 4$ *(right). Different colors indicate different clusters.*

For the spiral dataset $\mathcal{D}_2$, the cluster assignments produced by the SC-3 algorithm with $k = 2, 3, 4$ are depicted in Fig. 7. When $k = 3$, the SC-3 algorithm can correctly identify the true clusters. We observe the same clustering patterns as in the circle dataset $\mathcal{D}_3$. Specifically, when we set $k = 2$, the SC-3 algorithm merges two clusters into one cluster (the blue cluster in Fig. 7 left). On the other hand, when we set $k = 4$, SC-3 splits one cluster into two clusters (the green and black clusters in Fig. 7(right)).

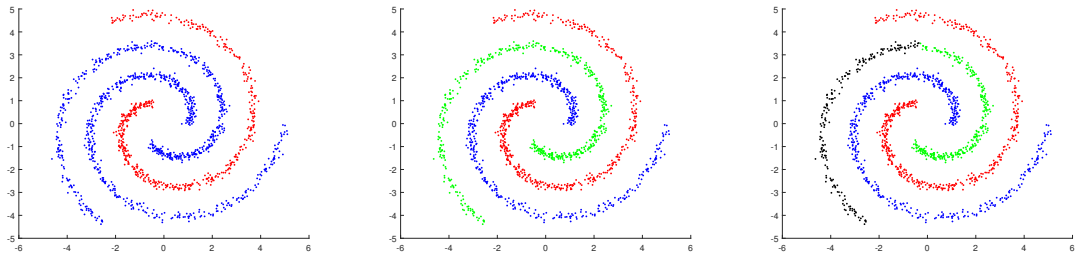

Figure 7: *Cluster assignments for the* **spiral dataset** $\mathcal{D}_2$ *using SC-3 for* $k = 2$ *(left),* $k = 3$ *(middle), and* $k = 4$ *(right). Different colors indicate different clusters.*

Overall, we observe that the spectral clustering algorithm can capture the correct cluster structure in both $\mathcal{D}_1$ and $\mathcal{D}_2$ if the value of k equals the true number of clusters. If we are incorrect about the number of clusters k, the SC algorithm can still capture the cluster structure to a certain degree.

(iii) We visualize the row vectors of the $\mathbf{V}$ matrices in SC-1, SC-2 and SC-3 for both the circle and spiral datasets in Figs. 8 and 9. Recall that each row vector of $\mathbf{V}$ is a representation of the corresponding data point. We observe that a clear cluster structure emerges in this new representation space. This explains the key intuition behind why the k-means algorithm that is used in the final step of a spectral clustering algorithm can succeed in the new representation space when it cannot in the original space.

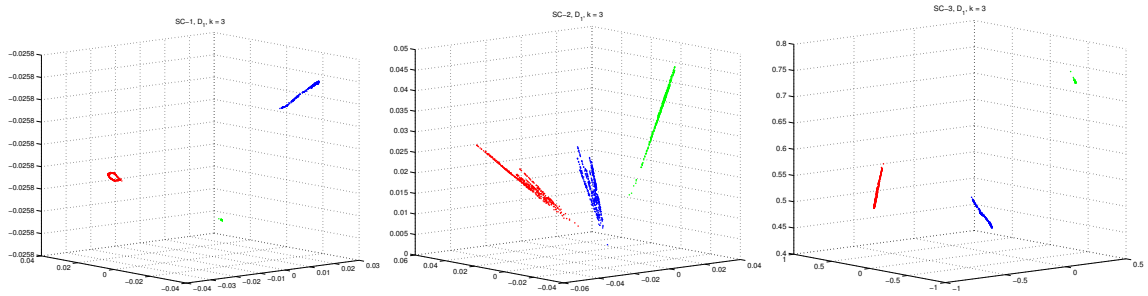

Figure 8: *3D plots of the row-vectors of the* $\mathbf{V}$ *matrices produced by SC-1 (left), SC-2 (middle), and SC-3 (right) for the circle dataset* $\mathcal{D}_1$ *with* $k = 3$*. Different colors indicate different clusters. This figure is best viewed in color.*

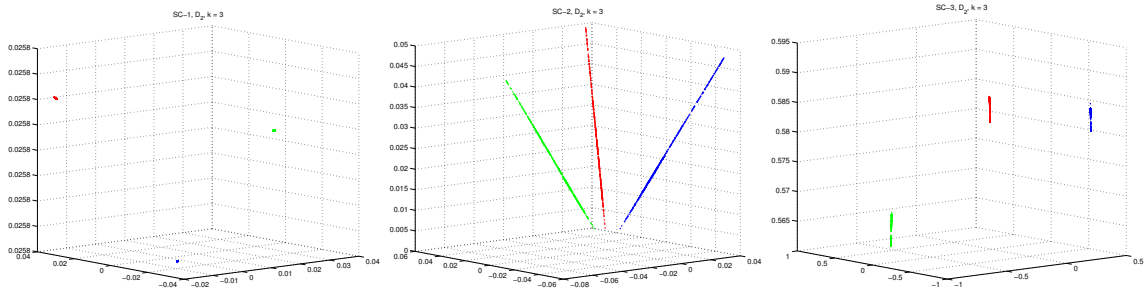

Figure 9: *3D plots of the row-vectors of the* $\mathbf{V}$ *matrices produced by SC-1 (left), SC-2 (middle), and SC-3 (right) for the spiral dataset* $\mathcal{D}_2$ *with* $k = 3$*. Different colors indicate different clusters. This figure is best viewed in color.*

(c) For the circle dataset represented in the polar coordinate system: $\mathcal{D}_3$, the cluster assignments produced by the k-means algorithm for $k = 2, 3, 4$ are depicted in Fig. 10. We observe that in this representation space, if we specify the correct number of clusters, i.e., set $k = 3$, then the k-means algorithm can identify the correct cluster structure. For the realization shown in Fig. 3, the WCSS values (for the 'cityblock' distance) are as follows:

- $k = 2 : 258, 274$;
- $k = 3 : 130, 141, 135$;
- $k = 4 : 85, 44, 73, 135$

We note that this distance is after attribute-wise re-normalization. The maximum value of each attribute is 1 and the minimum is 0.

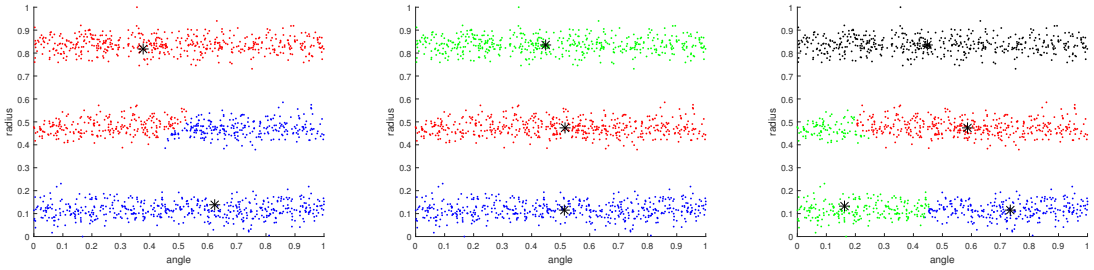

Figure 10: *Cluster assignments for the* **circle dataset** *represented in the* **polar coordinate system** *produced by the k-means clustering algorithm with $k = 2$ (left), $k = 3$ (middle), and $k = 4$ (right). Different colors indicate different clusters and the black star-points indicate the centroids of each cluster.*

Problem 4.2 Spectral Clustering on Airbnb data

In this problem we will use a newly released real-world dataset obtained from

`http://insideairbnb.com/get-the-data.html`

It consists of $n = 2558$ houses listed on the Airbnb website in the Boston area in Oct. 2015. We have pre-processed the data into a MATLAB file "`BostonListing.mat`". For each listing, we will keep its **latitude**, **longitude**, and **neighborhood**. The "neighborhood" attribute indicates the region of each house listing such as "allston", "brighton", etc. We will use **latitude** and **longitude** as a 2-D feature vector for each listing and treat **neighborhood** as the ground-truth cluster label. Our goal is to cluster the listings into clusters that can reflect the neighborhood structure based on latitude and longitude. Figure 11 is a visualization of this dataset on Google Map.

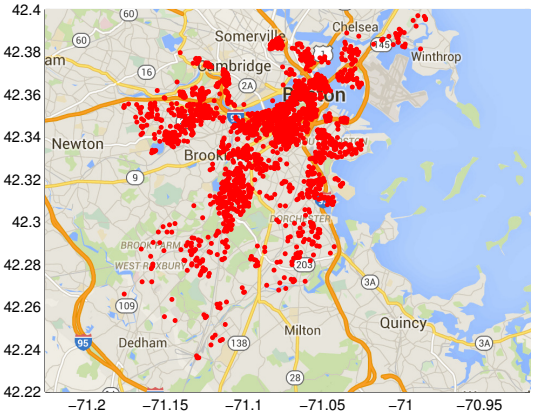

Figure 11: *House listings on Airbnb websites in the Boston area in Oct. 2015. Each point indicates a house listing. The dataset is obtained from Inside Airbnb.*

(a) We will use the Gaussian similarity distance defined in problem 4.1 and construct a fully-connected graph. We set $\sigma = 0.01$ in this case and use the symmetrically normalized graph Laplacian (SC-3

defined in problem 4.1) for spectral clustering. For $k = 1, 2, \ldots, 25$, **calculate** the "purity" metric of the obtained cluster by treating the "neighborhood" label as the ground truth.[1] **Plot** the purity metric (y-axis) as a function of k (x-axis).

(b) Use the `plot_google_map.m` to **plot** all the data points on the map and indicate the cluster assignment with $k = 5$ using different colors.

Solution:

(a) Figure 12 summarizes how the purity metric changes as a function of k. We observe that as we increase k, the performance increases in terms of the purity. This indicates that the spectral algorithm can partially capture the neighborhood structure.

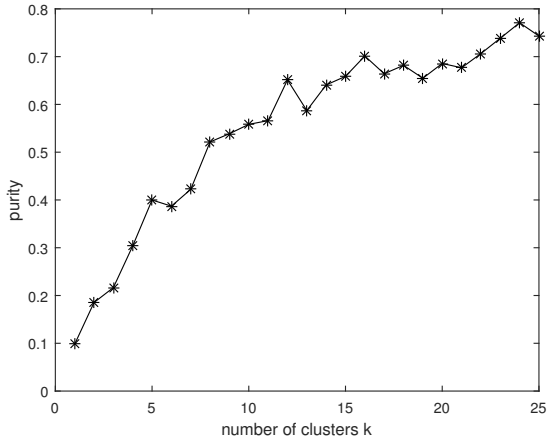

Figure 12: *Purity of the spectral clustering on Airbnb data for* $1 \leq k \leq 25$. *The Gaussian similarity function and a fully connected graph is used for the spectral clustering. The graph Laplacian is symmetrically normalized. Overall, there are* 25 *ground truth classes in the dataset. The purity increases as the number of clusters* k *increases.*

(b) Figure 13 depicts the cluster assignment produced by spectral clustering (SC-3 as denoted in problem 4.1) for $k = 5$. The clusters are indicated using 5 different colors. The results illustrate some high-level neighborhood structure in the Boston area: the red cluster in Fig. 13 represents the Allston/Brighton neighborhood; the blue cluster corresponds to the South Boston/Dorchester region; the black cluster corresponds to the Jamaica Plain/Roslindale/Roxbury region.

[1] The purity metric of a cluster assignment with k clusters is defined as follows. Let $n_{i,j}$ be the number of objects in cluster i that belong to class j, where $i = 1, \ldots, k$ and $j = 1, \ldots, m$. Here m is the number of ground-truth classes. Let $n_i = \max_{j=1,\ldots,m} n_{i,j}$. Purity is then defined as $\sum_{i=1}^{k} n_i / n$.

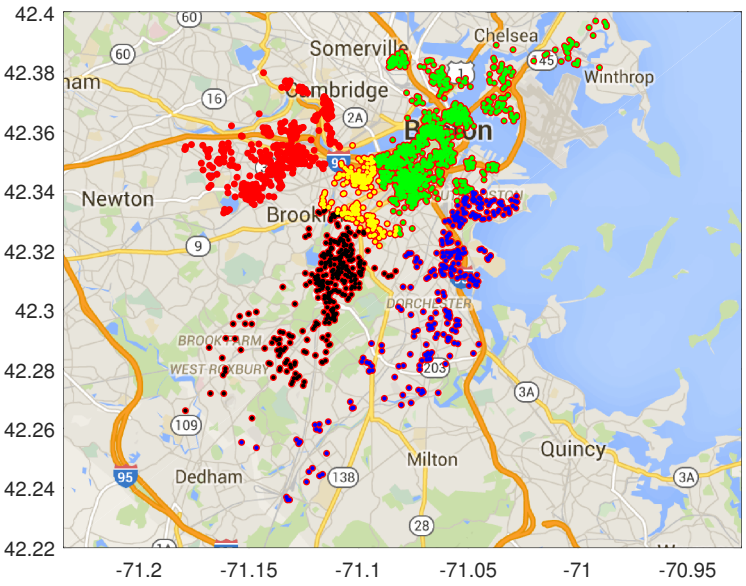

Figure 13: *Cluster assignments for the Airbnb dataset produced by the SC-3 spectral clustering algorithm with $k = 5$. Different colors indicate different clusters.*