Homework Set 6, CPSC 8420, Fall 2024

Collins, Matthew

Due 12/05/2024, 11:59PM EST

Problem 1

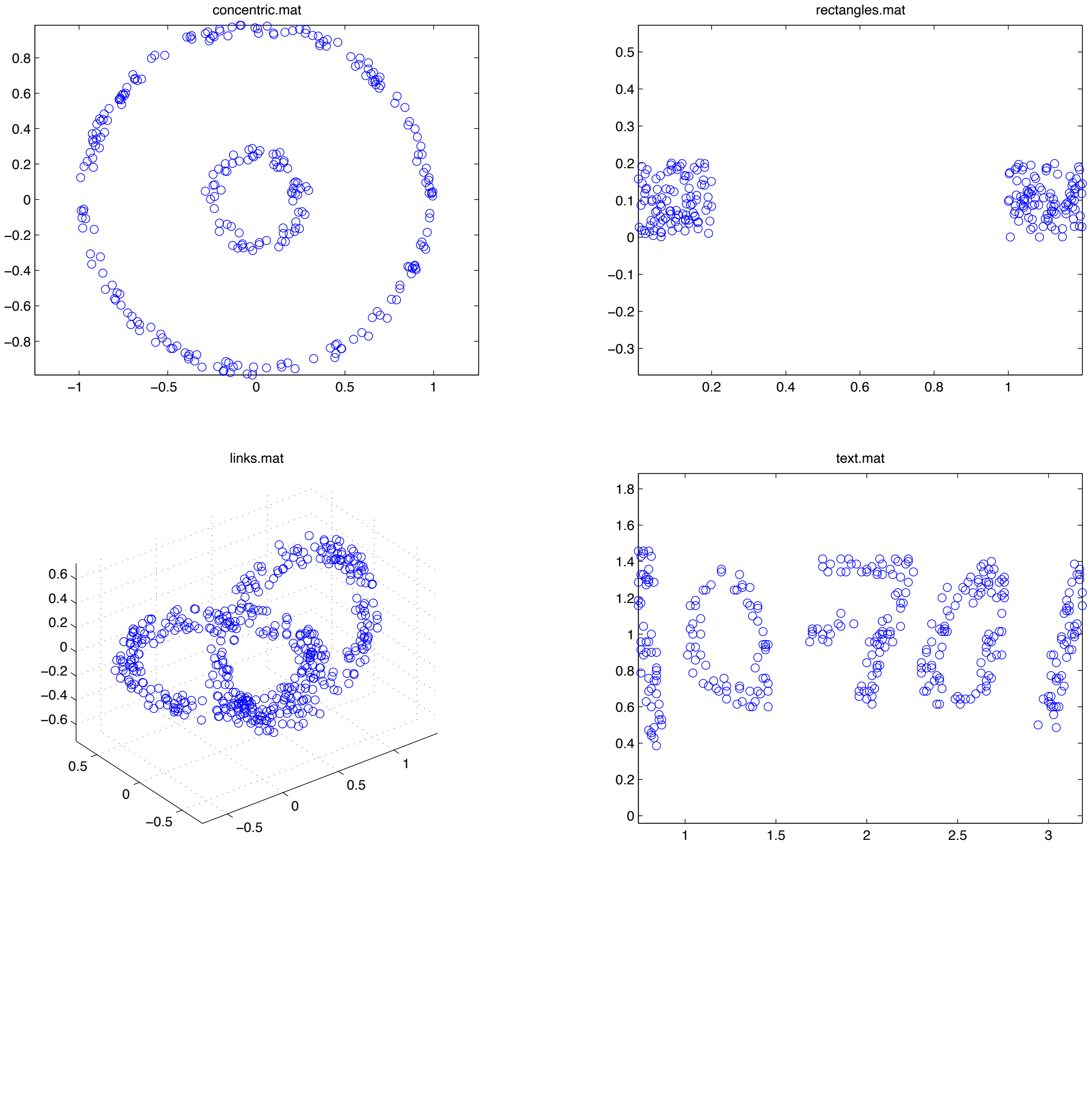
Frequently, the affinity matrix is constructed as:

$$A_{ij} = e^{-d(x_i,x_j)^2/\sigma} \tag{1}$$

where σ is some user-specified parameter. The best that we can hope for in practice is a near block-diagonal affinity matrix. It can be shown in this case, that after projecting to the space spanned by the top k eigenvectors, points which belong to the same block are close to each other in a euclidean sense. The steps are as follows:

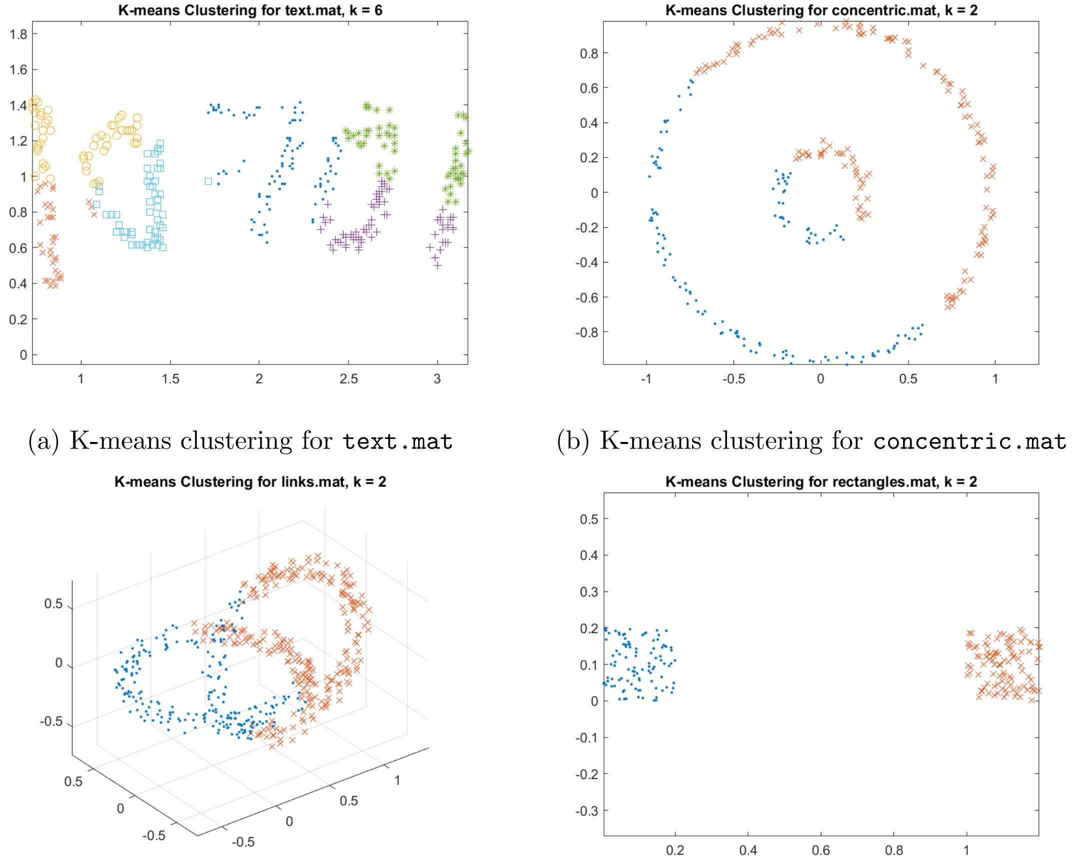
- Construct an affinity matrix A using the above equation.
- Symmetrically 'normalize' the rows and columns of A to get a matrix N such that $N(i,j) = \frac{A(i,j)}{\sqrt{d(i)d(j)}}$, where $d(i) = \sum_k A(i,k)$.
- Construct a matrix Y whose columns are the first k eigenvectors of N.
- Normalize each row of Y such that it is of unit length.
- Cluster the dataset by running k-means on the set of embedded points, where each row of Y is a data-point.

1. Run k-means on the datasets provided in the .zip file. For text.mat, take $k = 6$. For all others use $k = 2$.
2. Implement the above spectral clustering algorithm and run it on the four provided datasets using the same k. Plot your clustering results using $\sigma = .025, .05, .2, .5$. Hints: You may find the MATLAB functions pdist and eig to be helpful. A function plotClusters.m has been provided to help visualize clustering results.
3. Plot the first 10 eigenvalues for the rectangles.mat and text.mat datasets when $\sigma = .05$. What do you notice?
4. How do k-means and spectral clustering compare?


concentric.mat
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−0.5
0
0.5
1
rectangles.mat
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0.2
0.4
0.6
0.8
1
links.mat
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
0.5
0
−0.5
−0.5
0
0.5
1
text.mat
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
1.5
2
2.5
3

P1.1, Answer:

The clustering algorithm groups the data into k clusters, where k=6 for text.mat and k=2 for the other datasets.



(a) K-means clustering for `text.mat`

(b) K-means clustering for `concentric.mat`

(c) K-means clustering for `links.mat`

(d) K-means clustering for `rectangles.mat`

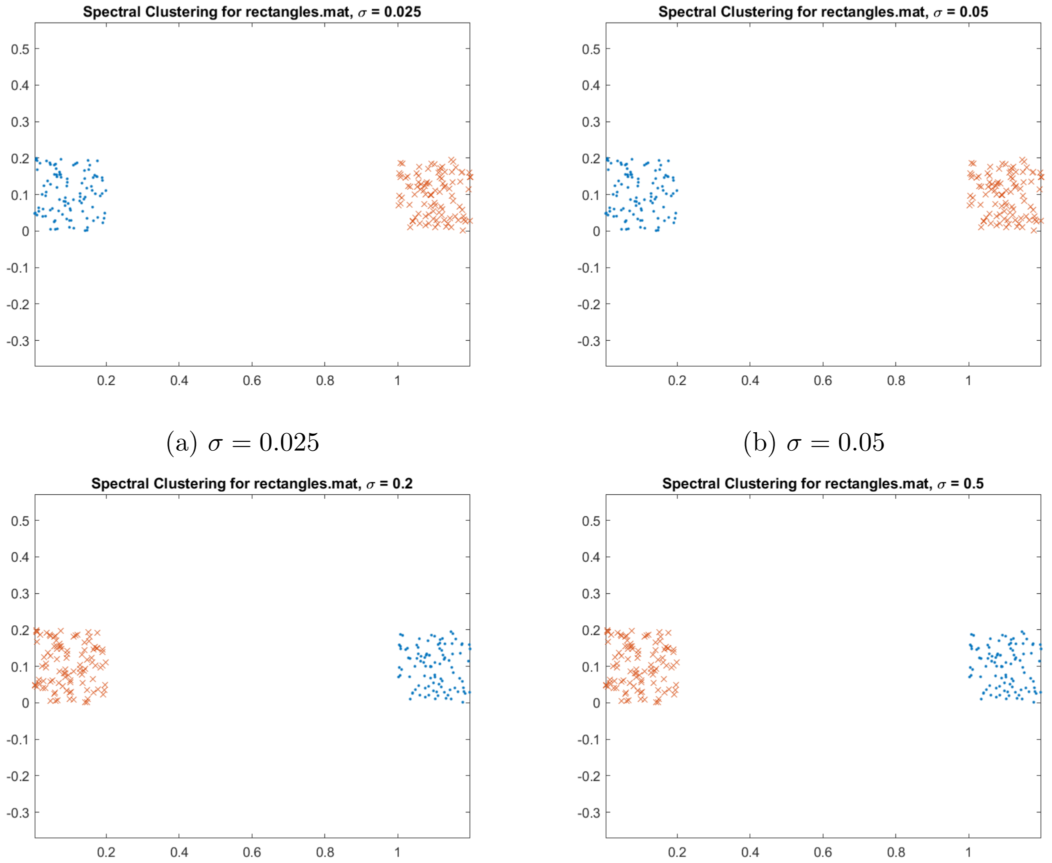
Figure 1: K-means clustering results for the provided datasets.

P1.2, Answer:

The `rectangles.mat` dataset exhibits two clearly separated clusters, and spectral clustering is largely insensitive to the value of σ. Across all tested σ values, the clusters are correctly identified, reflecting the dataset's inherent simplicity and large separation between clusters. Even with a large σ, where global similarities dominate, the clustering remains accurate due to the strong geometric distinction between the rectangles.

The `links.mat` dataset has two intertwined clusters, making it more sensitive to changes in σ. For smaller σ values (0.025 and 0.05), the clustering is highly accurate, as the algorithm emphasizes local neighborhood relationships. As σ increases (0.2 and 0.5), global similarities blur the cluster boundaries, resulting in some misclassification near the overlap of the "links."

The `concentric.mat` dataset shows the importance of σ in clustering performance. For smaller



(a) $\sigma = 0.025$

(b) $\sigma = 0.05$

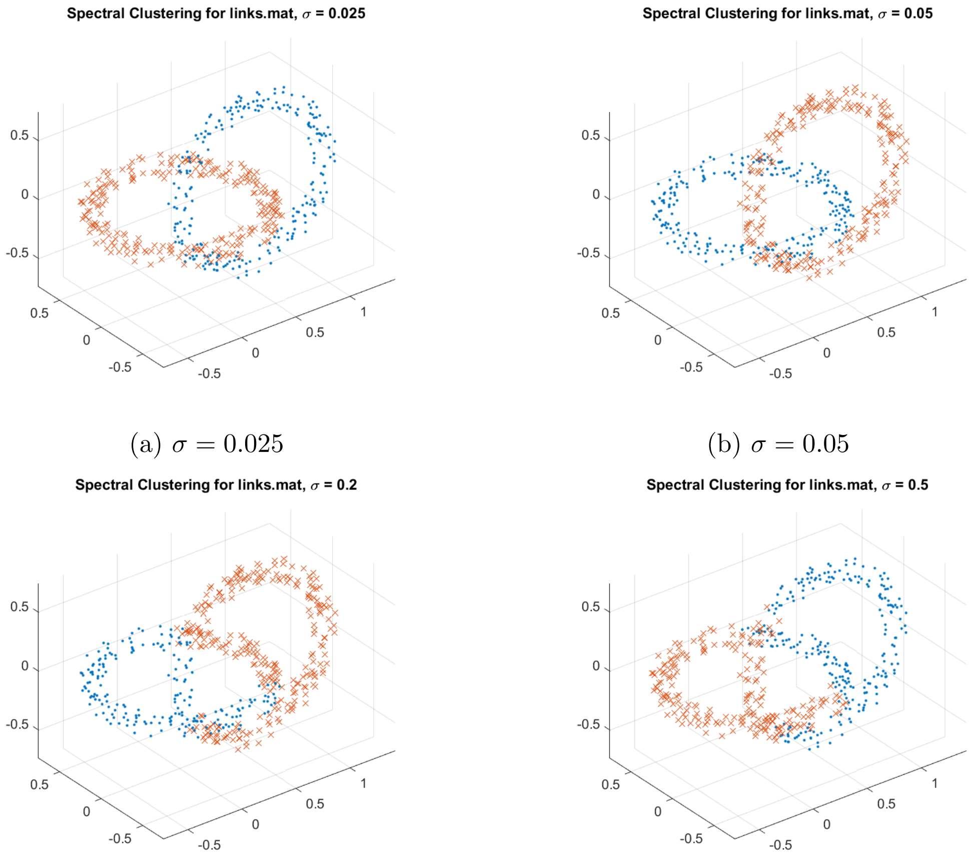
(c) $\sigma = 0.2$

(d) $\sigma = 0.5$

Figure 2: Spectral clustering results for `rectangles.mat` with varying σ.

σ values (0.025 and 0.05), the clustering is highly accurate, as local similarities dominate, allowing the algorithm to distinguish between the two concentric rings. As σ increases (0.2 and 0.5), the clusters become less distinct, and points from different rings begin to influence each other, leading to errors.

The `text.mat` dataset demonstrates the importance of σ in separating six distinct clusters that correspond to different groups. At smaller σ=0.025 the clustering is fairly well definied. Slight overlap is visible between cluster 1 and cluster 2, but the clusters are otherwise well separated. At σ=0.05, the clustering is still pretty well-defined, but the algorithm begins to identify unintened clusters. Increasing the total number of cluster from the correct number of five to seven. As σ increases (0.2 and 0.5), the clustering becomes less accurate, with more clusters beginning to merge or become noisy. The sensitivity of this dataset highlights how spectral clustering relies on the proper choice of σ to emphasize local relationships while avoiding excessive smoothing.


Spectral Clustering for links.mat, σ = 0.025

Spectral Clustering for links.mat, σ = 0.05

Spectral Clustering for links.mat, σ = 0.2

Spectral Clustering for links.mat, σ = 0.5

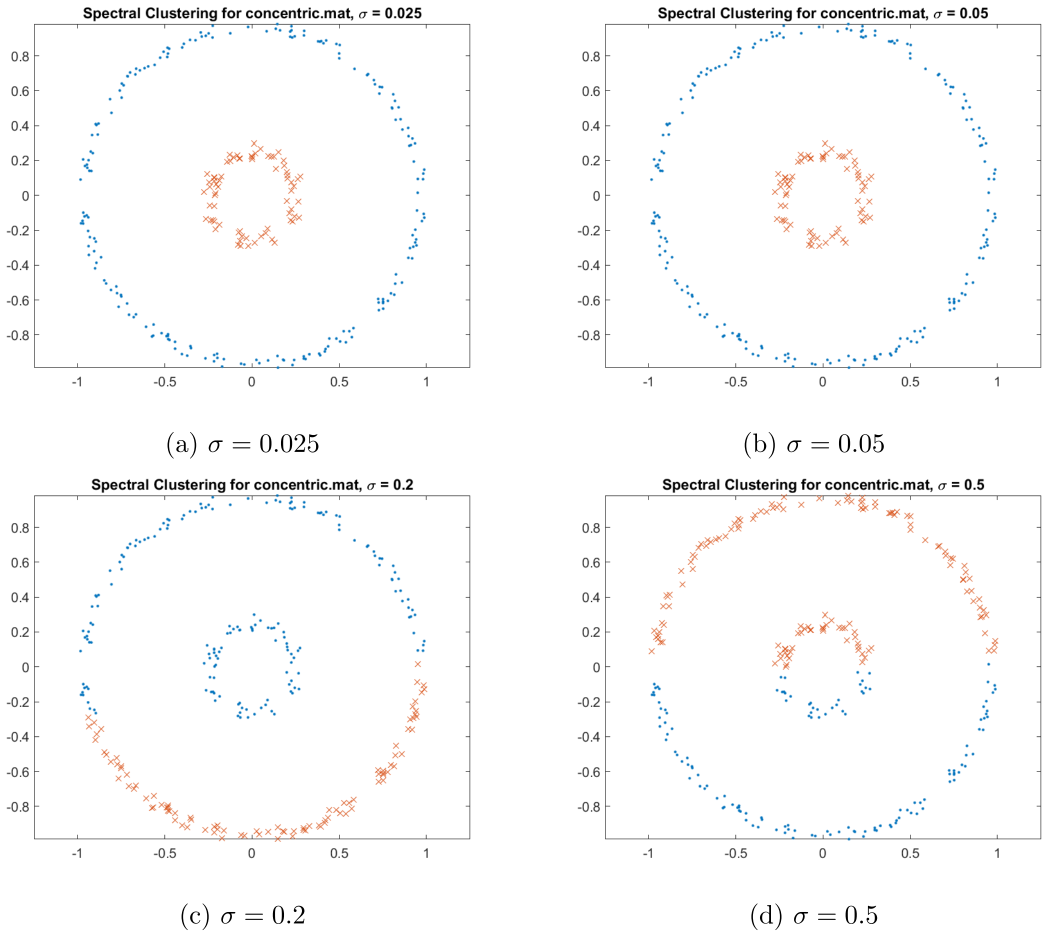
(a) $\sigma = 0.025$

(b) $\sigma = 0.05$

(c) $\sigma = 0.2$

(d) $\sigma = 0.5$

Figure 3: Spectral clustering results for `links.mat` with varying σ.



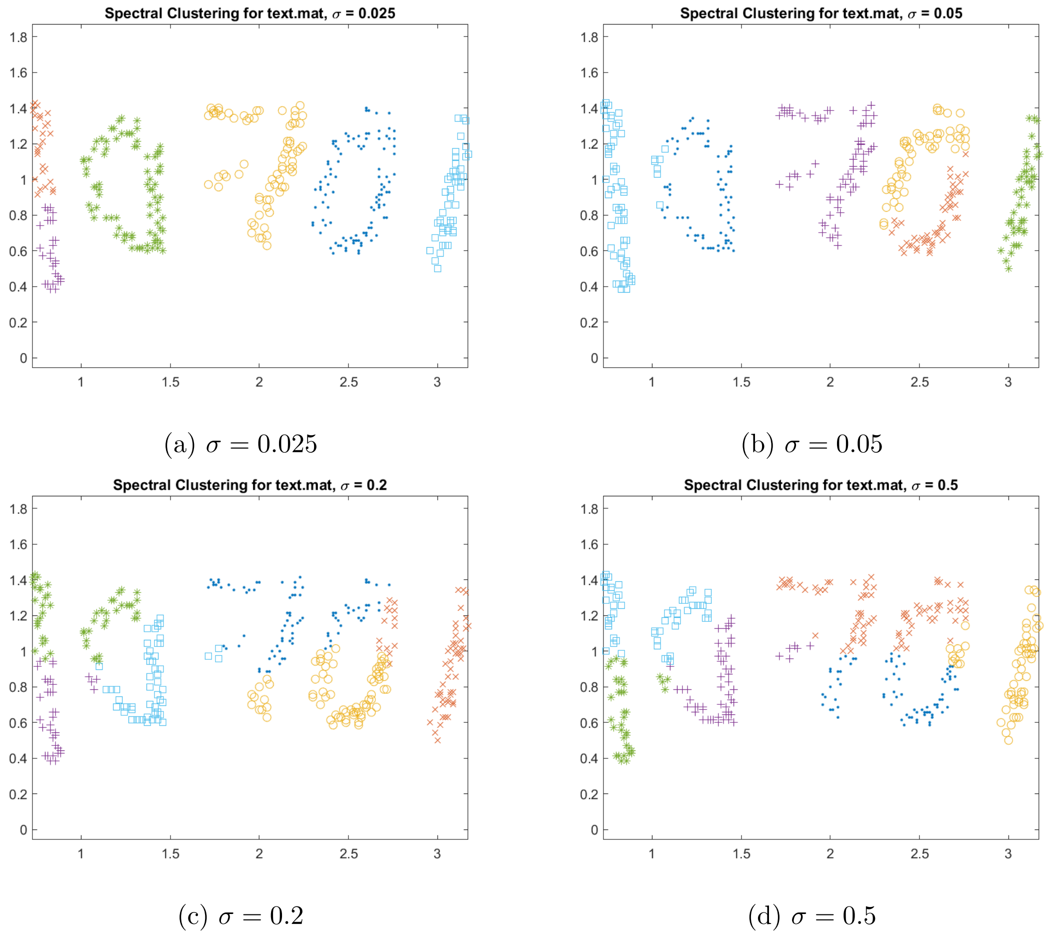
(a) $\sigma = 0.025$

(b) $\sigma = 0.05$

(c) $\sigma = 0.2$

(d) $\sigma = 0.5$

Figure 4: Spectral clustering results for `concentric.mat` with varying σ.



(a) $\sigma = 0.025$

(b) $\sigma = 0.05$

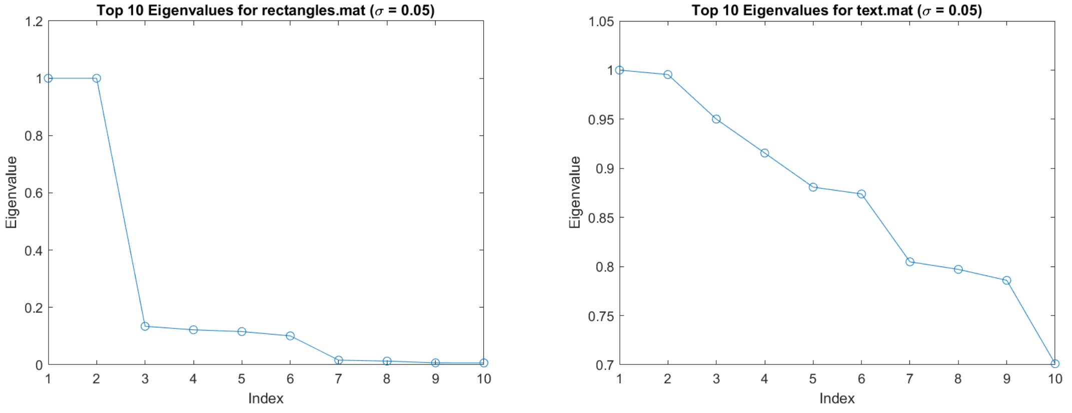
(c) $\sigma = 0.2$

(d) $\sigma = 0.5$

Figure 5: Spectral clustering results for `text.mat` with varying σ.

P1.3, Answer:

The plots of the top 10 eigenvalues for `rectangles.mat` and `text.mat` in Figure 6 reveal distinct differences in their clustering structure when $\sigma = 0.05$. For `rectangles.mat`, there is a sharp spectral gap between the second and third eigenvalues. The first two eigenvalues are significantly larger, indicating the presence of two dominant clusters in the dataset. This aligns with the expectation, as `rectangles.mat` contains two well-separated rectangular clusters. The remaining eigenvalues are close to zero, suggesting that most of the data's variance is captured by the first two eigenvectors, which correspond to the two clusters. In contrast, the eigenvalues for `text.mat` show a gradual decrease, with no distinct spectral gap. This indicates that the dataset has a more complex structure and does not have well-separated clusters. Instead, the clusters may overlap, requiring more eigenvectors to adequately represent the data. The gradual decline in eigenvalues suggests that the clustering quality might depend heavily on the choice of parameters such as σ or k.



(a) `rectangles.mat` (b) `text.mat`

Figure 6: First 10 eigenvalues for `rectangles.mat` and `text.mat` with $\sigma = 0.05$.

P1.4, Answer:

The comparison between k-means and spectral clustering contain advantages and limitations for each method based on the dataset's characteristics. k-means assumes that clusters are convex and roughly spherical in Euclidean space, which makes it effective for datasets like `rectangles.mat`, where clusters are well-separated and aligned with Euclidean distance. However, k-means struggles with datasets like `concentric.mat` and `links.mat`, where clusters are non-convex or intertwined.

Spectral clustering, on the other hand, does not rely on such assumptions and performs well with non-linear or complex cluster shapes. By leveraging the graph-based affinity matrix, spectral clustering captures relationships between points more effectively, making it suitable for datasets like `concentric.mat` and `links.mat`. On the `rectangles.mat` dataset, both k-means and spectral clustering perform well because the clusters are linearly separable and geometrically simple. For `concentric.mat`, k-means fails because the clusters are circular and cannot be separated using straight lines, while spectral clustering identifies the circular clusters accurately.

Similarly, on the `links.mat` dataset, k-means struggles due to the intertwined structure of the clusters, whereas spectral clustering handles the dataset effectively. On the `text.mat` dataset, k-means provides reasonable clustering but may struggle if clusters overlap or are not spherical. Spectral clustering, in contrast, offers better separation, especially for complex or overlapping clusters.

Regarding parameter sensitivity, k-means requires the number of clusters (k) to be specified explicitly and is sensitive to the initialization of cluster centroids, which can lead to different results across runs. Spectral clustering relies on the choice of the similarity function and the parameter σ. The eigenvalue spectrum in spectral clustering provides insights into the optimal number of clusters, which can be an advantage in some contexts.

A Appendix: MATLAB Code

A.1 Homework 6.1

```
clear all
close all
clc

datasets = {'text.mat', 'concentric.mat', 'links.mat', 'rectangles.mat'};
data_vars = {'X4', 'X1', 'X3', 'X2'}; % Variable names in the .mat files

for i = 1:length(datasets)
    load(datasets{i});
    var_name = data_vars{i};
    if exist(var_name, 'var')
        data = eval(var_name);
    else
        error(['Variable␣', var_name, '␣not␣found␣in␣dataset␣', datasets{i}]);
    end


    if strcmp(datasets{i}, 'text.mat')
        k = 6;
    else
        k = 2;
    end

    [idx, ~] = kmeans(data, k);

    plotClusters(data, idx);
    title(['K-means␣Clustering␣for␣', datasets{i}, ',␣k␣=␣', num2str(k)]);
    filename = sprintf('k_means_clustering_%s.png', datasets{i});
    saveas(gcf, filename);
end
```

A.2 Homework 6.2

```
clear all
close all
clc

datasets = {'text.mat', 'concentric.mat', 'links.mat', 'rectangles.mat'};
data_vars = {'X4', 'X1', 'X3', 'X2'};
dataset_names = {'text', 'concentric', 'links', 'rectangles'};
sigma_values = [0.025, 0.05, 0.2, 0.5];
```

```
for i = 1:length(datasets)
    load(datasets{i});
    var_name = data_vars{i};
    if exist(var_name, 'var')
        data = eval(var_name);
    else
        error(['Variable ', var_name, ' not found in dataset ', datasets{i}]);
    end

    if strcmp(datasets{i}, 'text.mat')
        k = 6;
    else
        k = 2;
    end

    for j = 1:length(sigma_values)
        sigma = sigma_values(j);
        idx = spectralClustering(data, k, sigma);

        plotClusters(data, idx);
        title(['Spectral Clustering for ', datasets{i}, ', \sigma = ', num2str(

        filename = sprintf('%s_sigma_%.3f.png', dataset_names{i}, sigma);
        saveas(gcf, filename);
    end
end

function [idx] = spectralClustering(data, k, sigma)
    n = size(data, 1);
    A = zeros(n, n);
    for i = 1:n
        for j = 1:n
            A(i, j) = exp(-norm(data(i, :) - data(j, :))^2 / sigma);
        end
    end

    D = diag(1 ./ sqrt(sum(A, 2)));
    N = D * A * D;

    [V, ~] = eigs(N, k);
    Y = V ./ vecnorm(V, 2, 2);
    [idx, ~] = kmeans(Y, k);
end
```

A.3 Homework 6.3

```
clear all
close all
clc

datasets = {'rectangles.mat', 'text.mat'};
data_vars = {'X2', 'X4'};
sigma = 0.05;

for i = 1:length(datasets)
    load(datasets{i});
    var_name = data_vars{i};
    if exist(var_name, 'var')
        data = eval(var_name);
    else
        error(['Variable␣', var_name, '␣not␣found␣in␣dataset␣', datasets{i}]);
    end

    n = size(data, 1);
    A = zeros(n, n);
    for p = 1:n
        for q = 1:n
            A(p, q) = exp(-norm(data(p, :) - data(q, :))^2 / sigma);
        end
    end

    D = diag(1 ./ sqrt(sum(A, 2)));
    N = D * A * D;

    [V, E] = eig(N);
    eigenvalues = diag(E);
    eigenvalues = sort(eigenvalues, 'descend');

    figure;
    plot(1:10, eigenvalues(1:10), '-o');
    title(['Top␣10␣Eigenvalues␣for␣', datasets{i}, '␣(\sigma␣=␣0.05)']);
    xlabel('Index');
    ylabel('Eigenvalue');
    filename = sprintf('first_10_eigenvalues_%s.png', datasets{i});
    saveas(gcf, filename);
end
```