Homework Set 4, CPSC 8420, Fall 2023

Singh, Charanjit

Due 12/10/2023, Sunday, 11:59PM EST

Problem 2

Please use spectral clustering to segment the image 'clemson.jpeg' in the file.

Loading and processing the raw image

```
% Load and preprocess the image
img = imread("clemson.jpeg");

% Downsample the image to fit the matrix in memory
scaleFactor = 0.1;
smallImage = imresize(img, scaleFactor);

% Convert to grayscale
img = rgb2gray(img);
img = imresize(img, scaleFactor);
disp("Downsized Image Size: (Rows X Column)")
```

```
Downsized Image Size: (Rows X Column)
```

```
disp(size(img));
```

```
    96    96
```

Initializing the Adjacency Matrix

```
[rows, cols] = size(img);

% Hyperparameter for the gaussian similarity
sigma = 1;

% Initializing the adjacency matrix
N = rows * cols;
adjMatrix = zeros(N, N);

% Construct the adjacency matrix
for i = 1:N
    for j = 1:N
        % Convert linear indices back to 2D indices
        [row_i, col_i] = ind2sub([rows, cols], i);
        [row_j, col_j] = ind2sub([rows, cols], j);

        xi = img(row_i, col_i);
        xj = img(row_j, col_j);

        % Calculating the Gaussian similarity
        similarity = exp(-sum((xi - xj) .^ 2) / (2 * sigma ^ 2));
        adjMatrix(i, j) = similarity;
    end
end
```

Calculating the Degree Matrix

```
% Construct the degree matrix
```

```
degreeMatrix = diag(sum(adjMatrix));
```

Calculating the Laplacian Matrix

```
laplacianMatrix = degreeMatrix - adjMatrix;
```

Calculate the eigenvectors

```
[eigVectors, eigValues] = eig(laplacianMatrix);
```

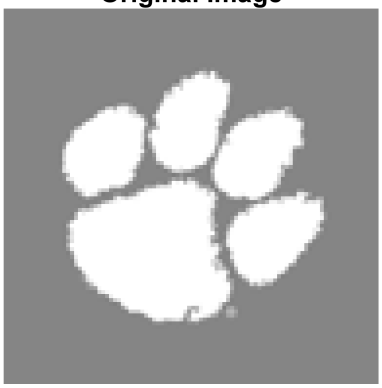
Performing K Means Clustering on the eigenVectors matrix

```
k = 2; % No of Clusters

eigVectors = eigVectors(:, 1:k);

% Normalizing Eigenvectors
rowNorms = sqrt(sum(eigVectors.^2, 2));
normalizedeigVectors = eigVectors ./ (rowNorms + eps);

clusters = kmeans(normalizedeigVectors, k);

% Reshape Clusters
segmentedImage = reshape(clusters, rows, []);
```

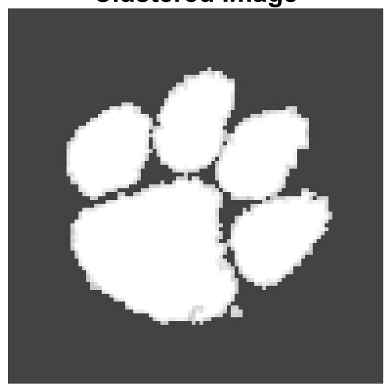
Displaying the Clustered Image

```
% Define a custom colormap for black and white
colormap = [0 0 0; 1 1 1]; % Black and White

% Create the overlay image using the custom colormap
newImage = labeloverlay(img, segmentedImage, 'Colormap', colormap);

% Display the original image and the black-and-white cluster overlay
figure;
subplot(1,2,1), imshow(img), title('Original Image');
subplot(1,2,2), imshow(newImage), title('Clustered Image');
```

Original Image



Clustered Image



Problem 3

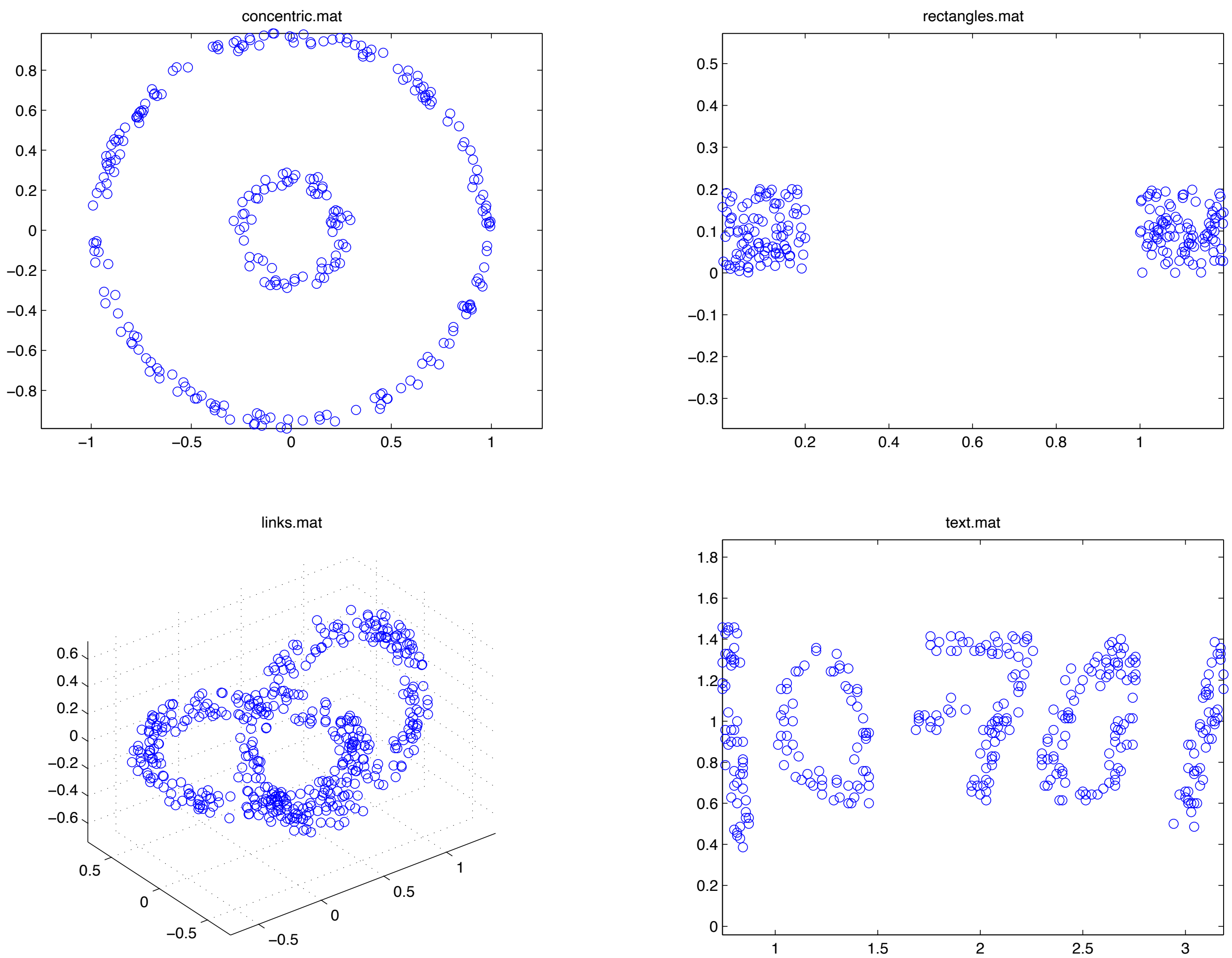
Frequently, the affinity matrix is constructed as:

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

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.

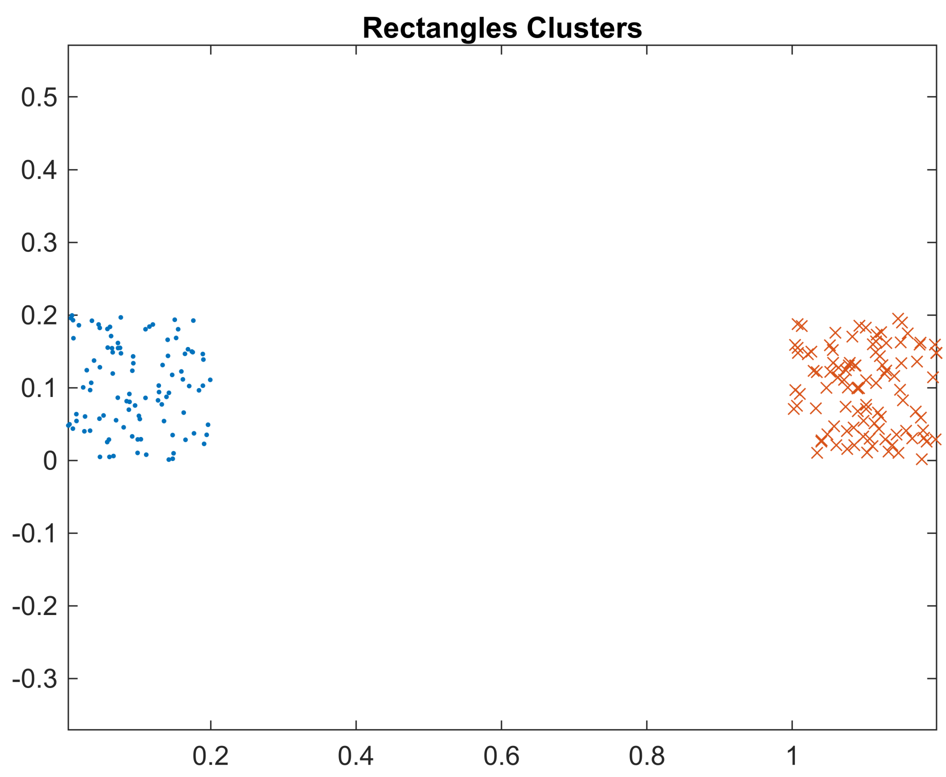
1. Run k-means on the datasets provided in the .zip file. For text.mat, take $k = 6$. For all others use $k = 2$.
2. Implement the above spectral clustering algorithm and run it on the four provided datasets using the same k. Plot your clustering results using $\sigma = .025, .05, .2, .5$. Hints: You 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

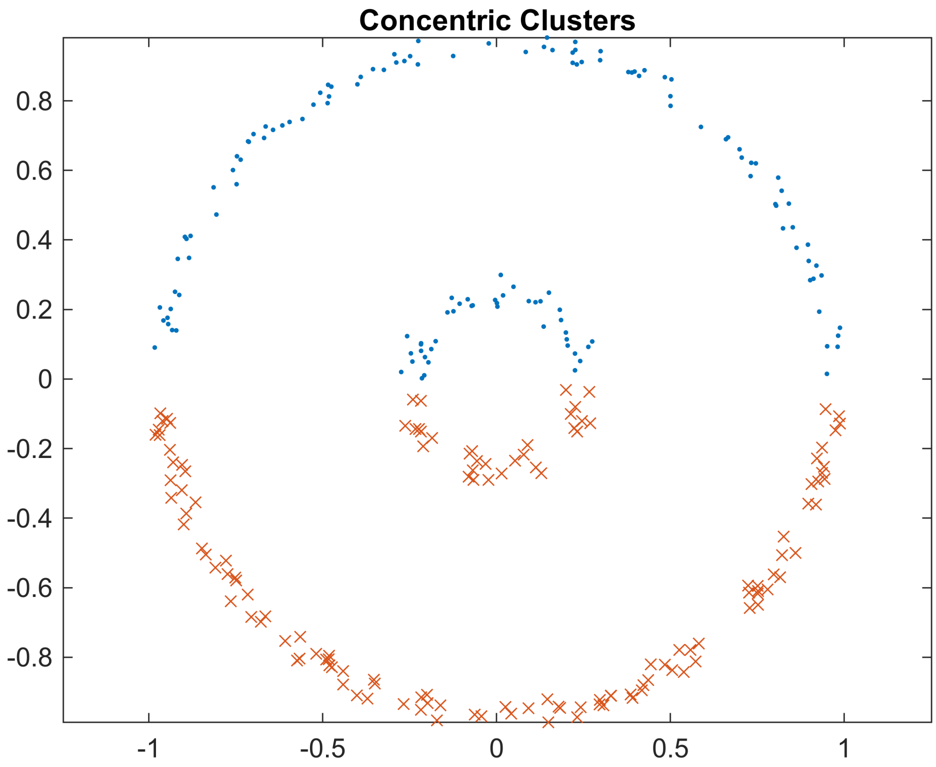
Problem 2

1. K-Means Clustering

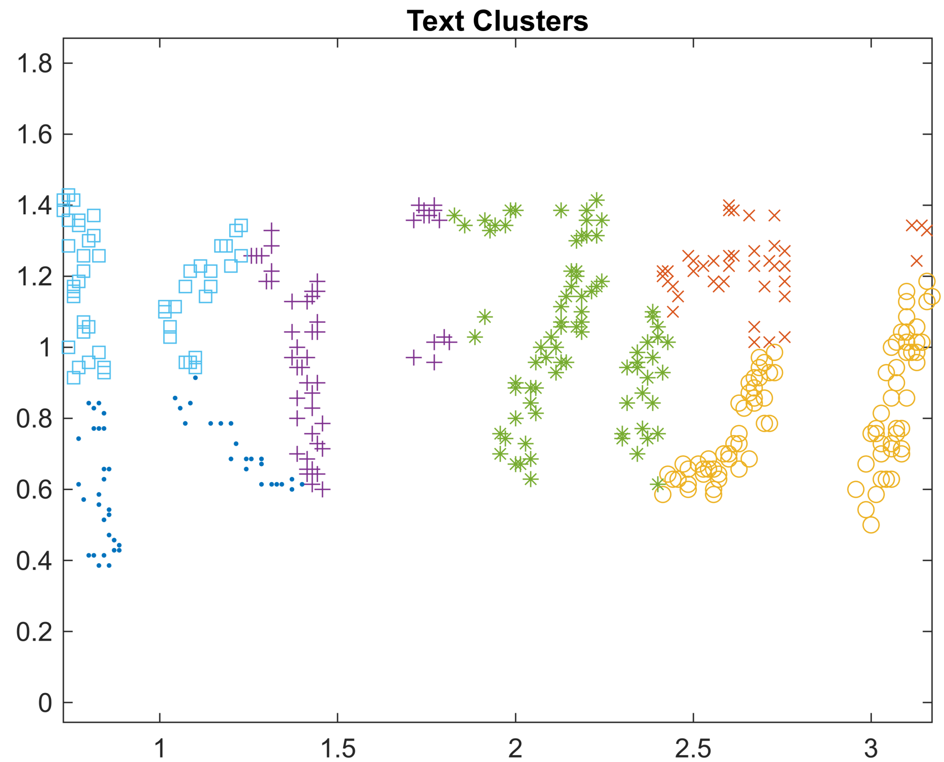
```
% Load the datasets
load("rectangles.mat"); % loads X2
load("concentric.mat"); % loads X1
load("text.mat");       % loads X4
load("links.mat");      % loads X3

% Perform k-means clustering on each dataset
rect_clusterIdx = kmeans(X2, 2);
conc_clusterIdx = kmeans(X1, 2);
text_clusterIdx = kmeans(X4, 6);
links_clusterIdx = kmeans(X3, 2);

plotClusters(X2, rect_clusterIdx);
title('Rectangles Clusters');
```



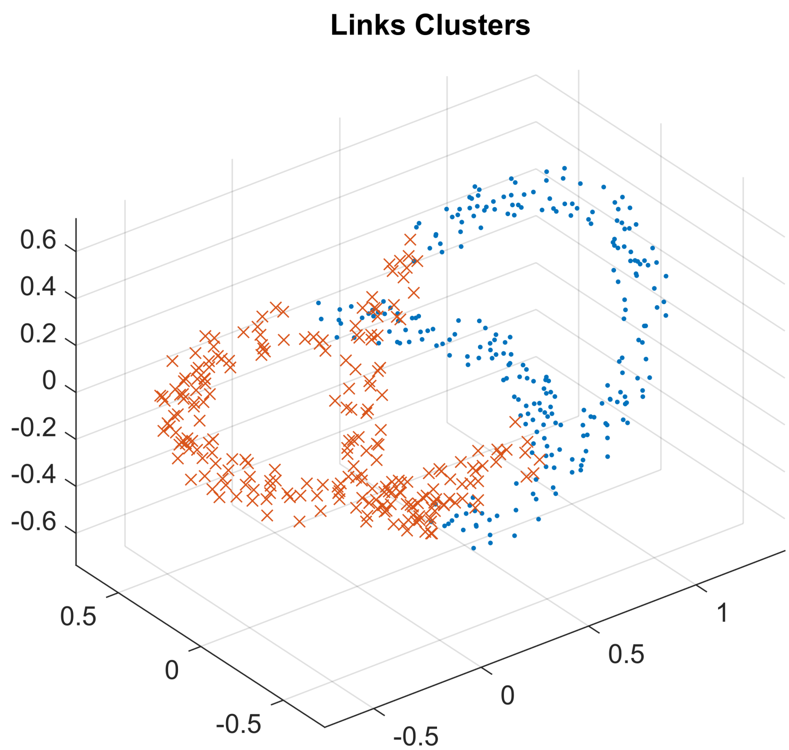

```
plotClusters(X1, conc_clusterIdx);
title('Concentric Clusters');
```




```
plotClusters(X4, text_clusterIdx);
title('Text Clusters');
```



```
plotClusters(X3, links_clusterIdx);
title('Links Clusters');
```

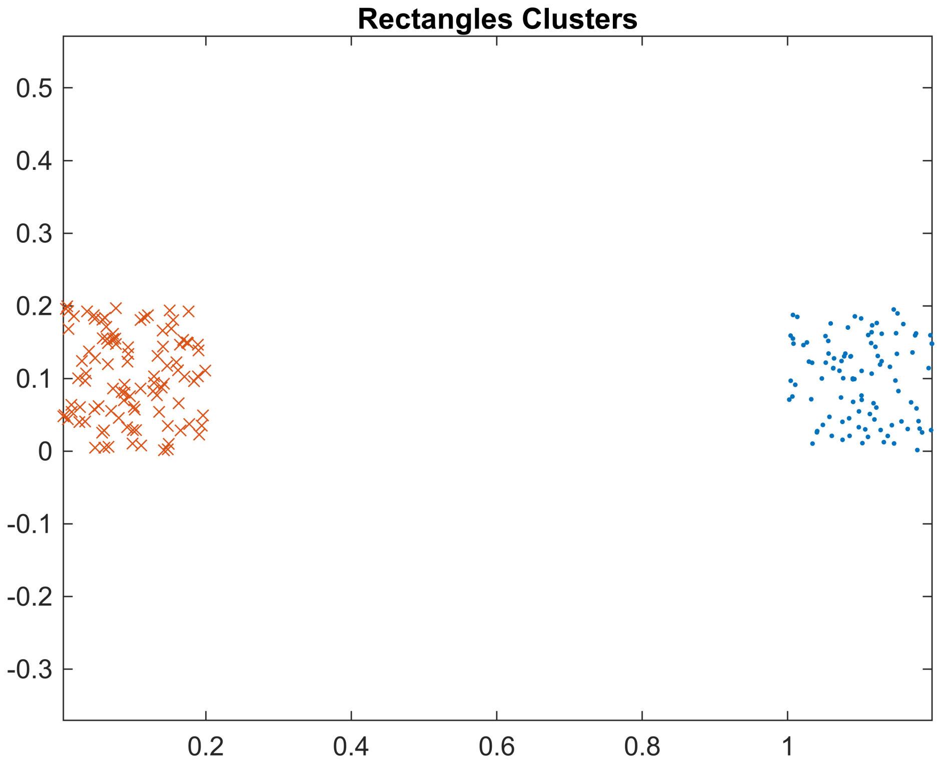


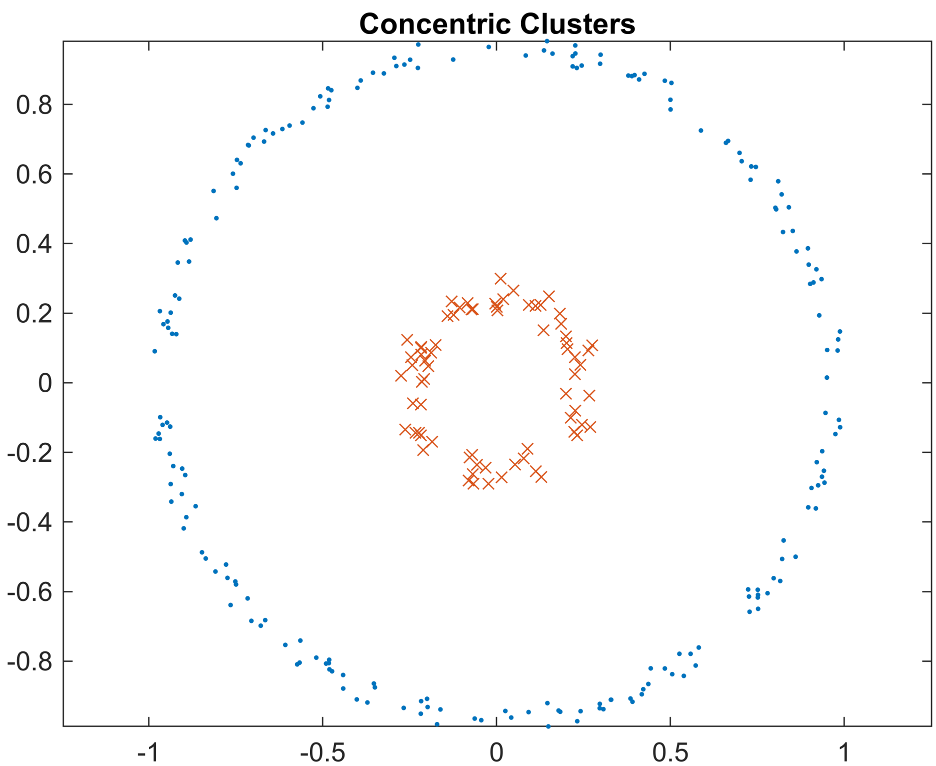
2. Implement matrix spectral clustering

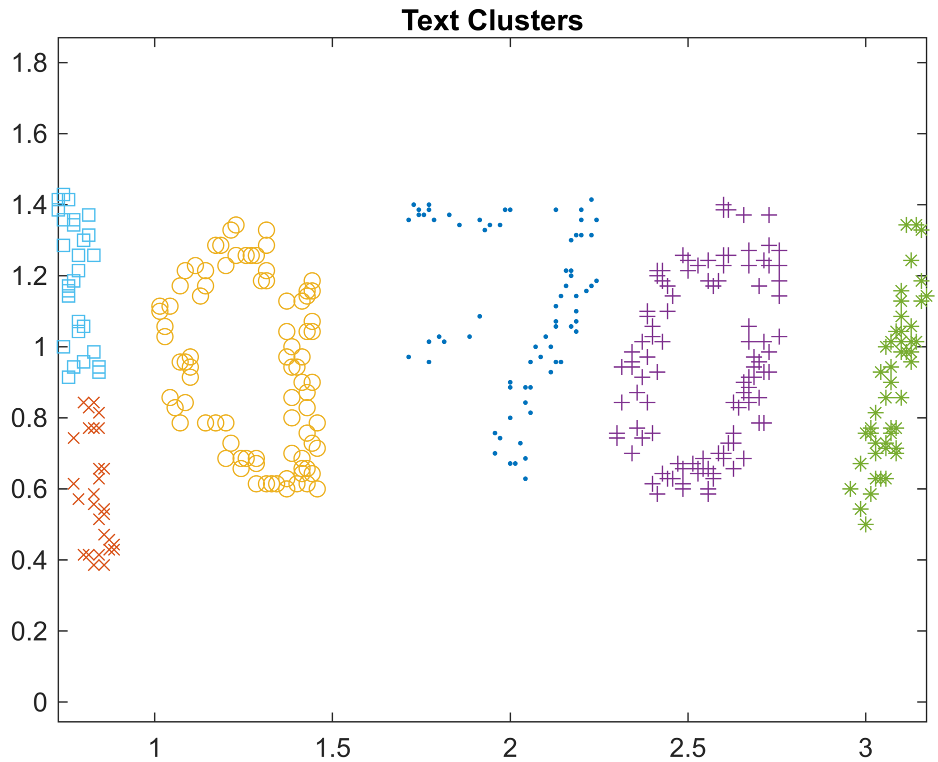
```
sigma = [.025, .05, .2, .5];
for i = sigma
    [rect_cluster, ~] = clustering(X2, 2, i);
    [conc_cluster, ~] = clustering(X1, 2, i);
    [text_cluster, ~] = clustering(X4, 6, i);
    [links_cluster, ~] = clustering(X3, 2, i);

    sprintf('Plots for sigma: %d', i)

    plotClusters(X2, rect_cluster);
    title('Rectangles Clusters');

    plotClusters(X1, conc_cluster);
    title('Concentric Clusters');

    plotClusters(X4, text_cluster);
    title('Text Clusters');

    plotClusters(X3, links_cluster);
    title('Links Clusters');
```

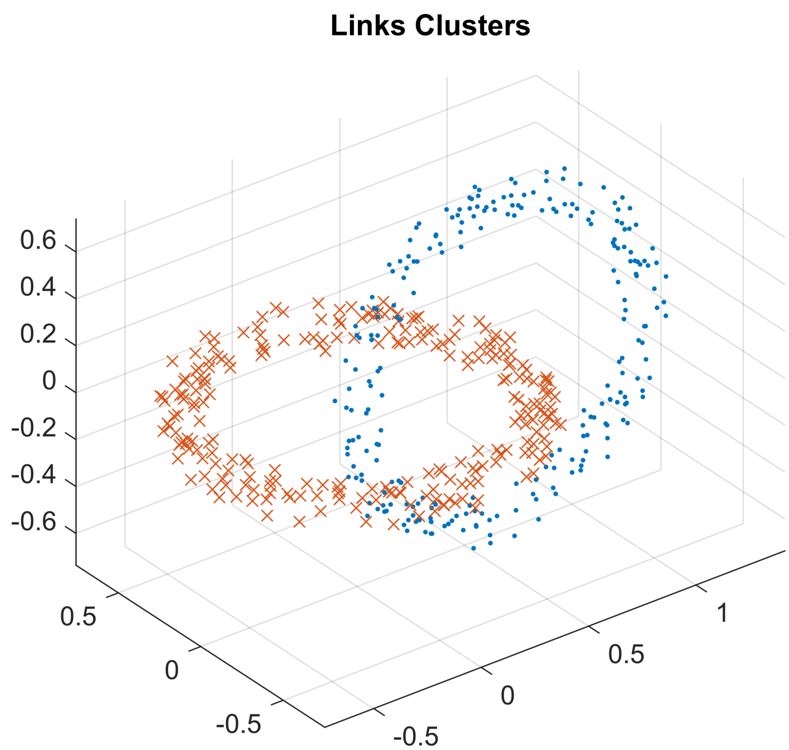
```
end
```

```
ans =
'Plots for sigma: 2.500000e-02'
```

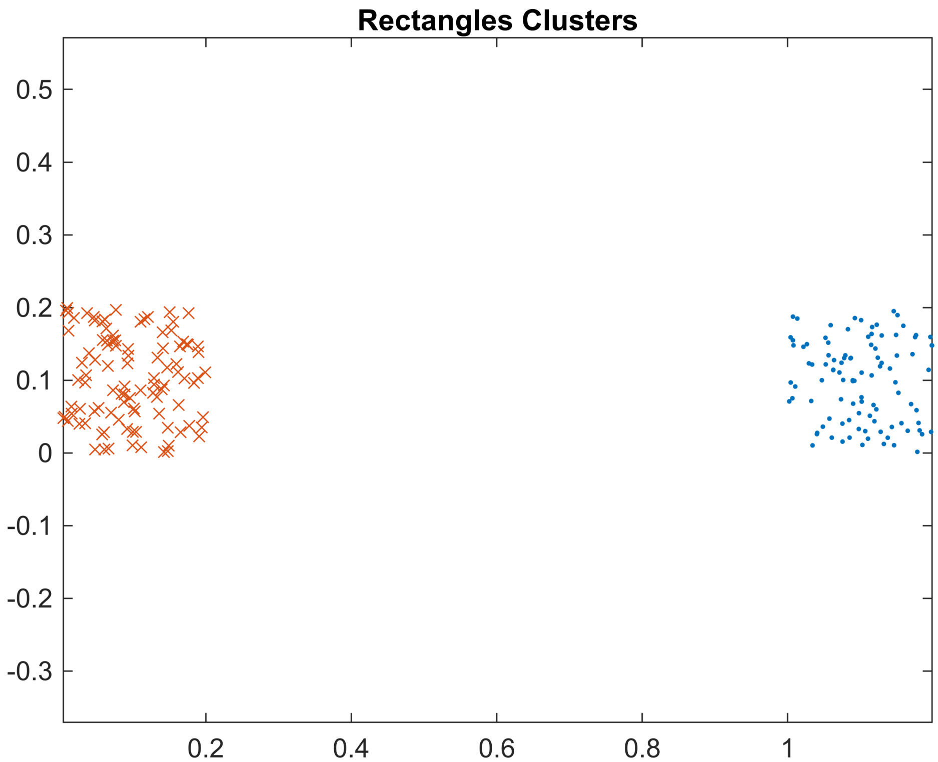


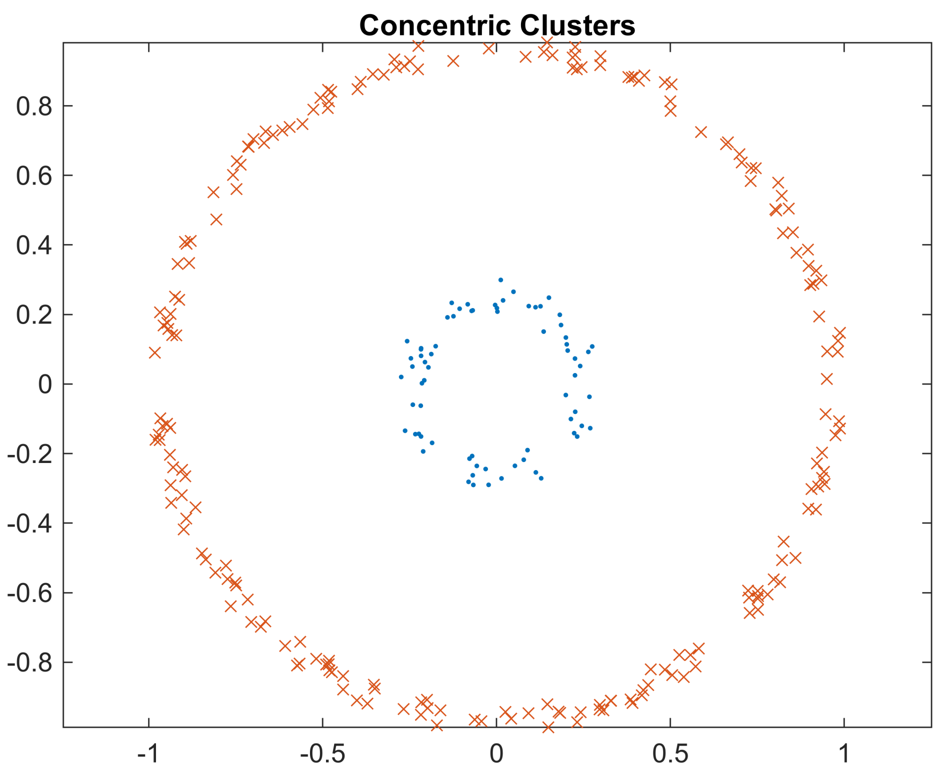


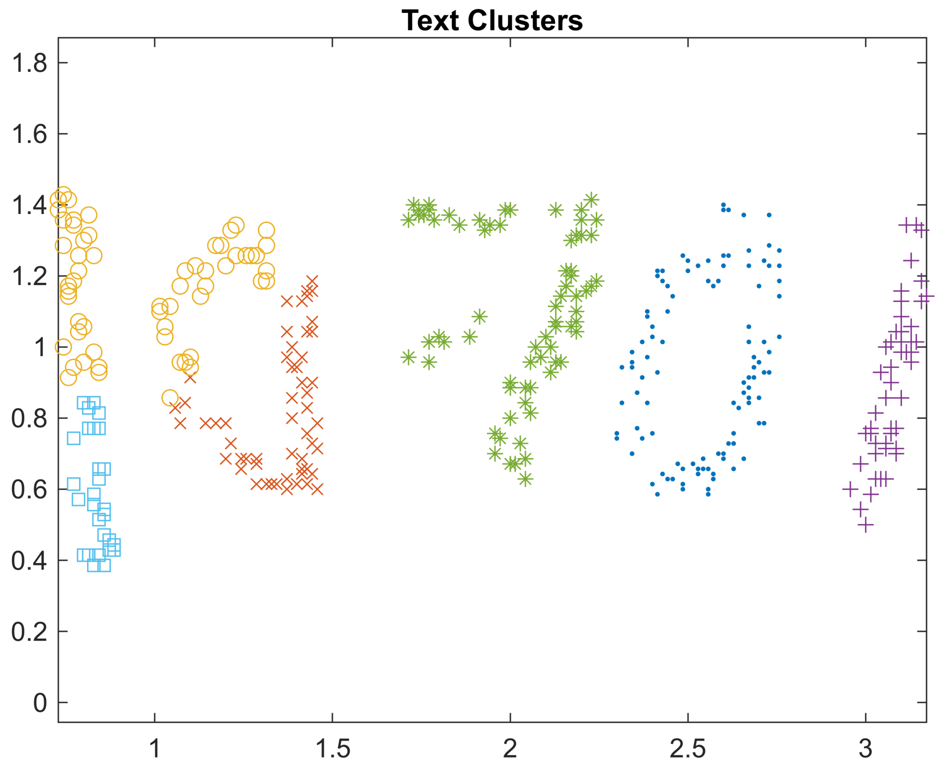


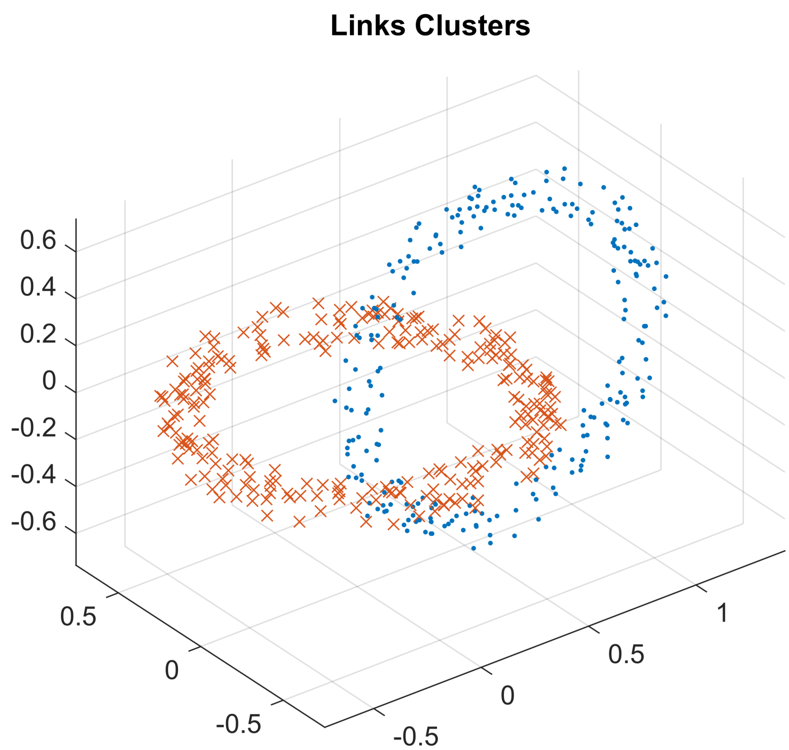



```
ans =
'Plots for sigma: 5.000000e-02'
```

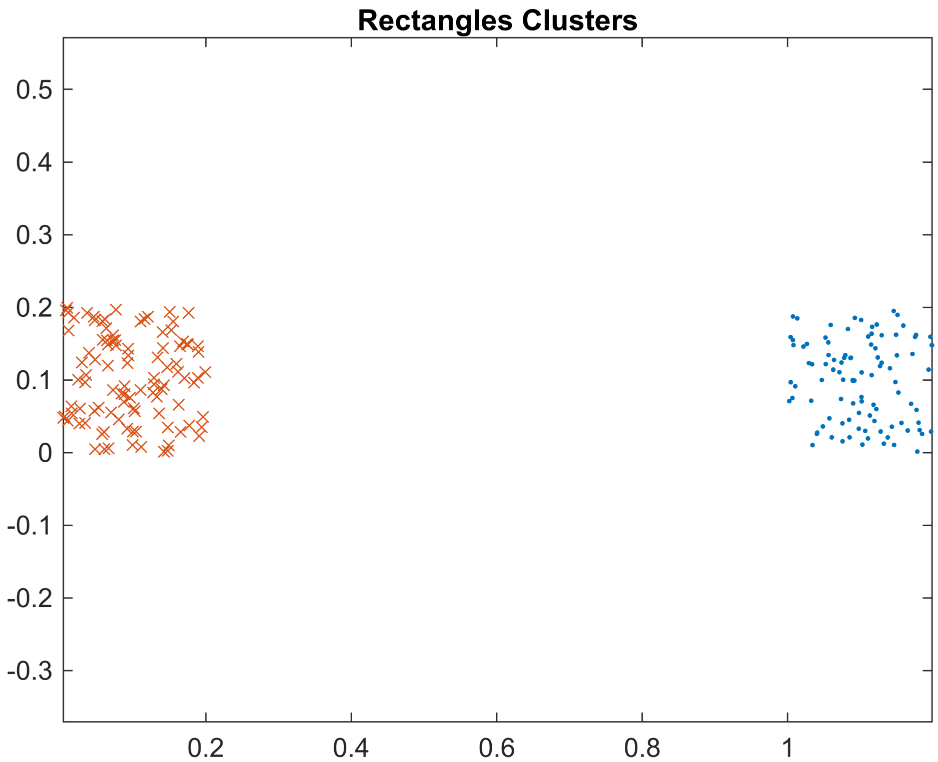

Rectangles Clusters
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1

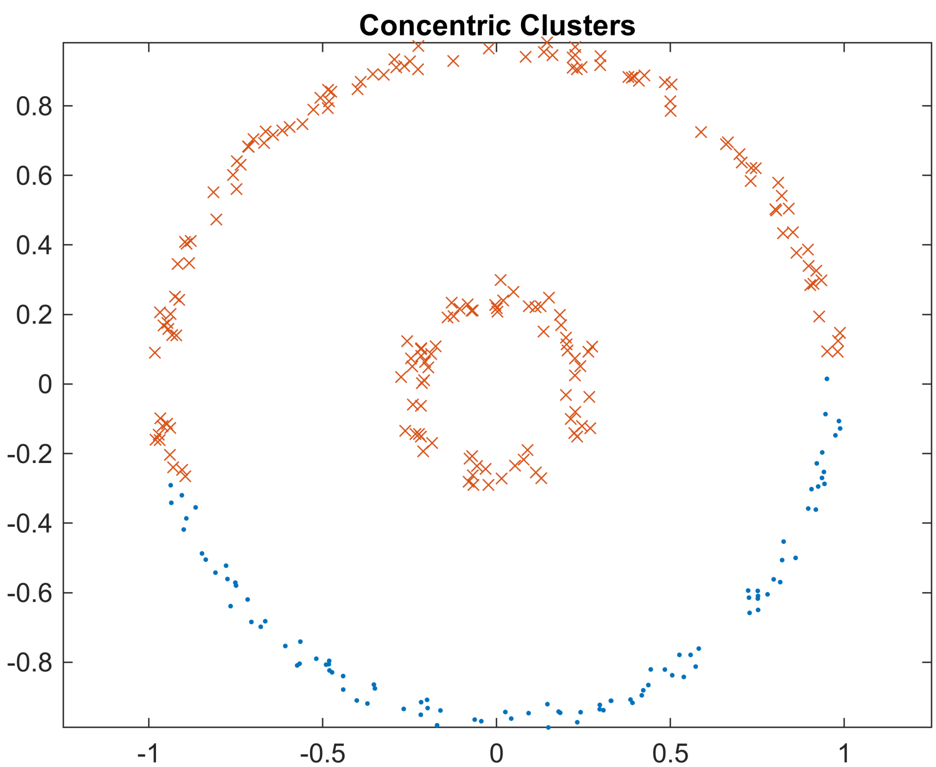

Concentric Clusters
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1

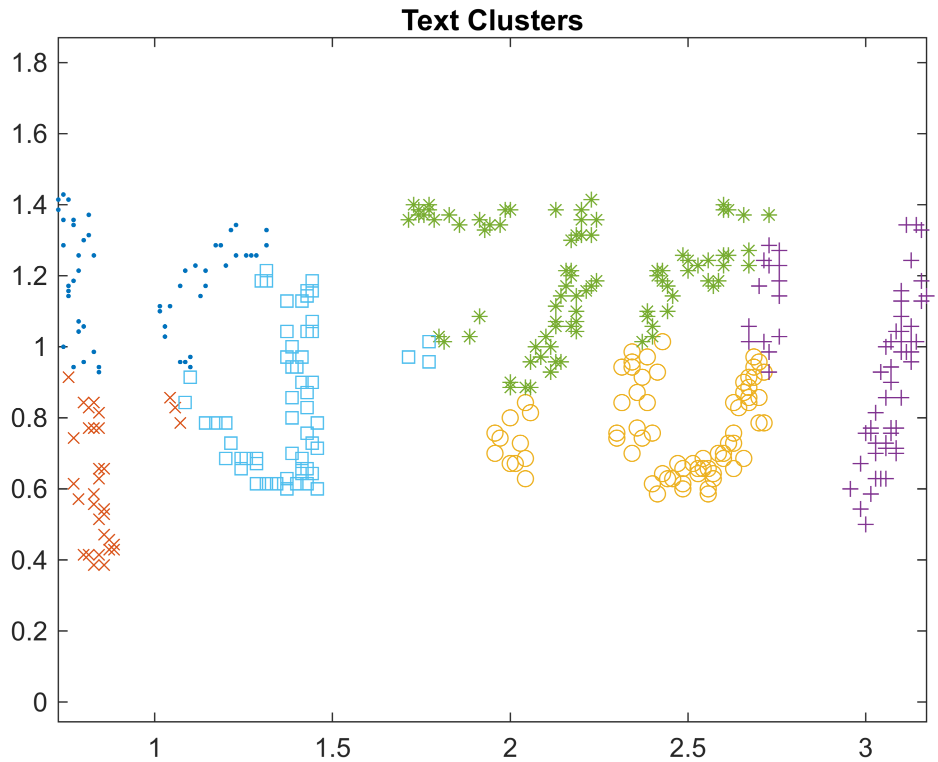


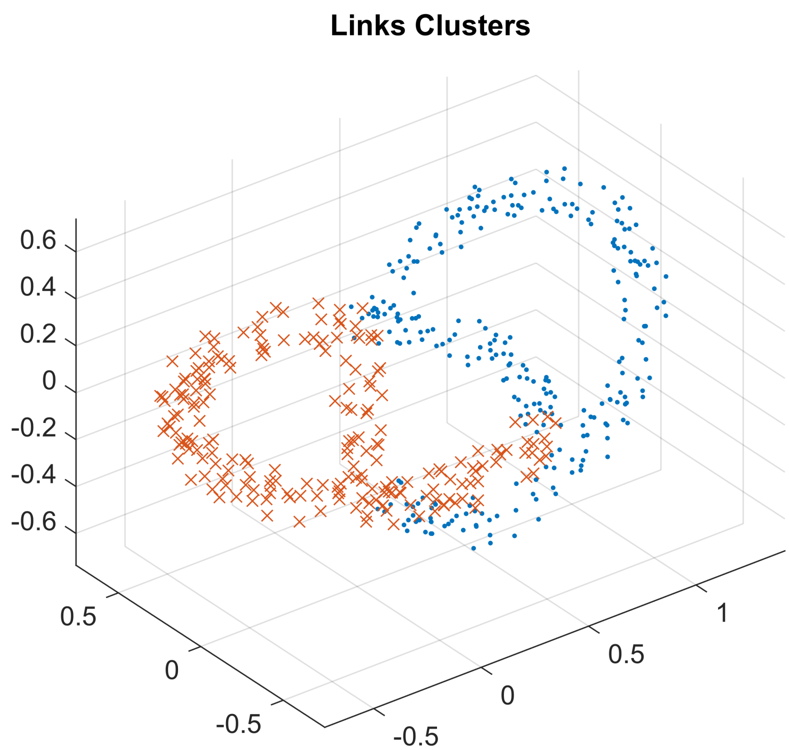



```
ans =
'Plots for sigma: 2.000000e-01'
```

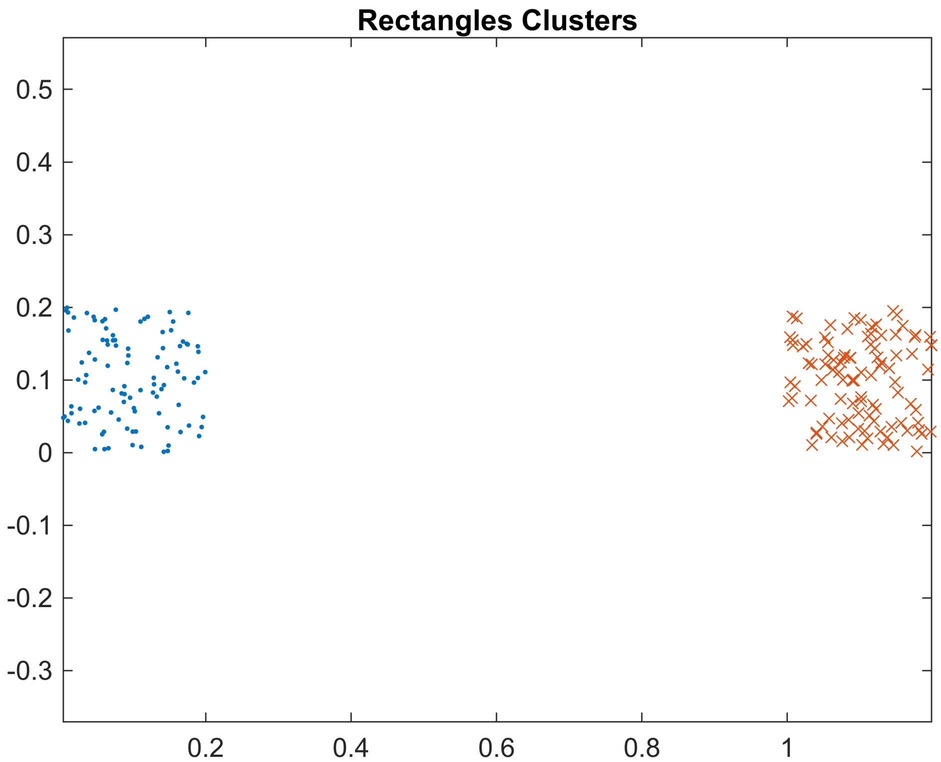

Rectangles Clusters
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1

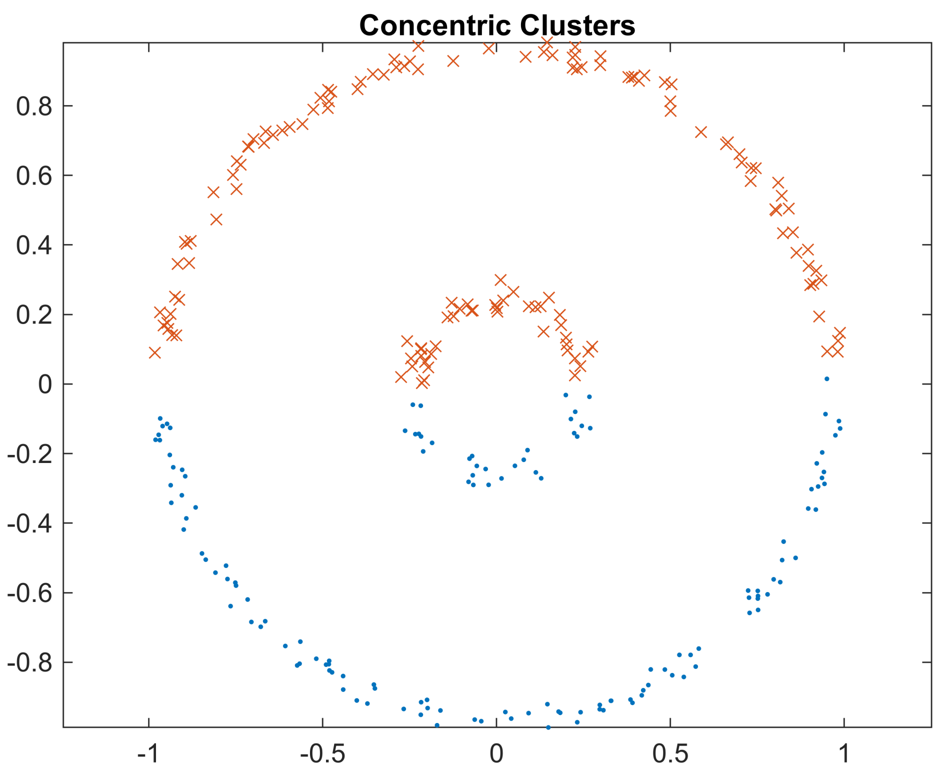

Concentric Clusters
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1

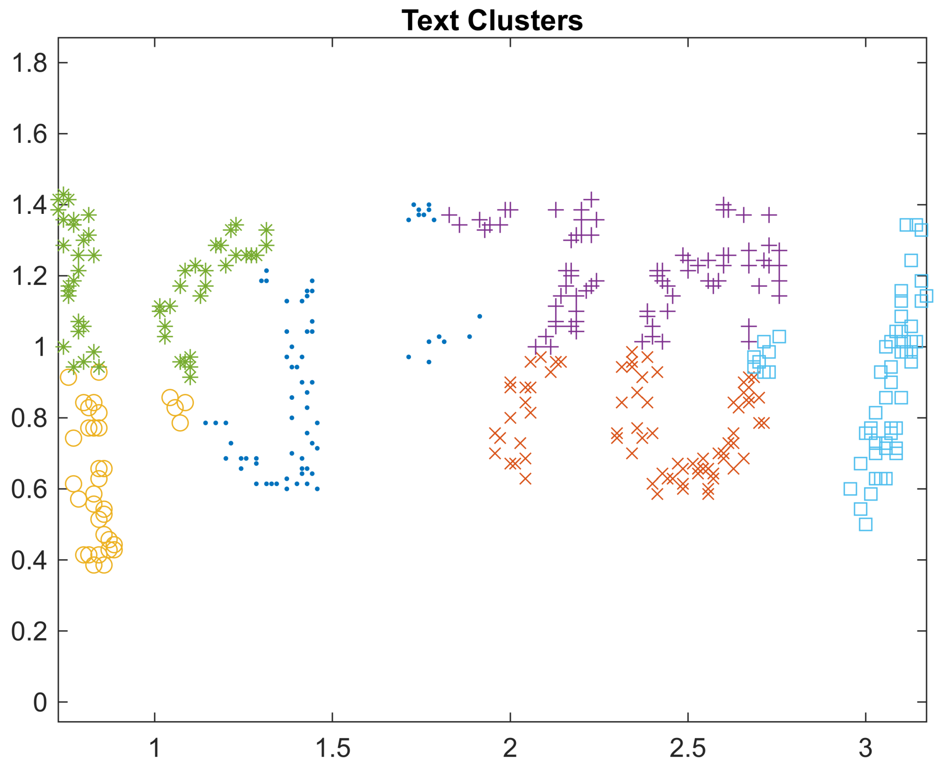


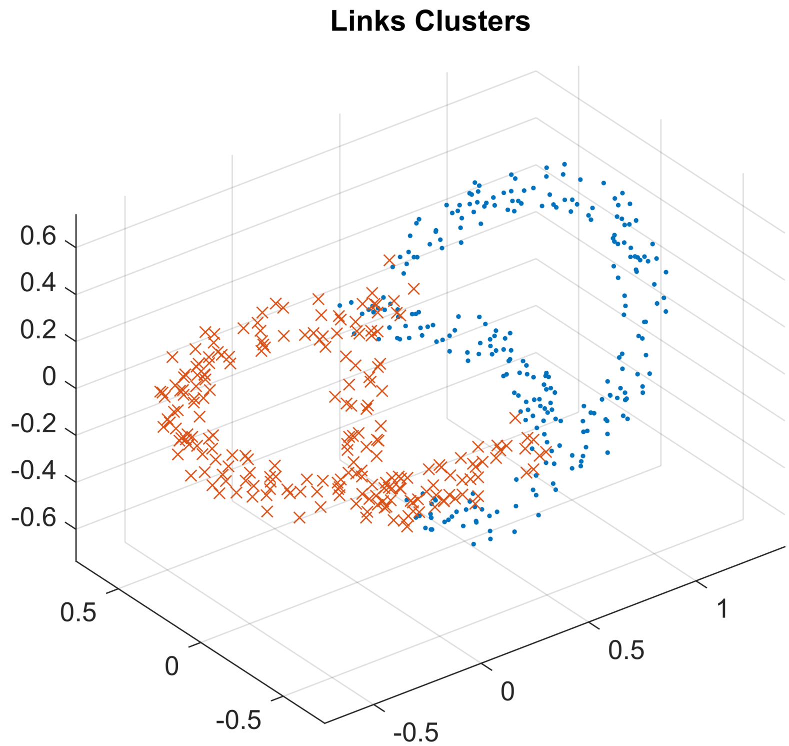



```
ans =
'Plots for sigma: 5.000000e-01'
```


Rectangles Clusters
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

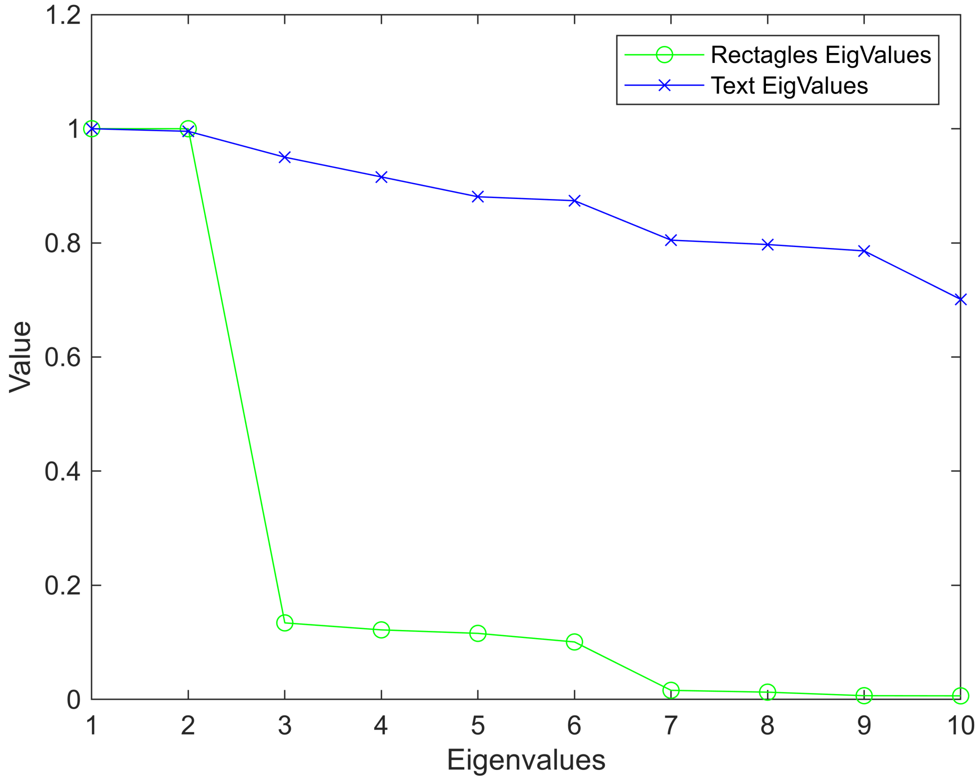

Concentric Clusters
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1





3. Plot first 10 Eigenvalues for rectangles.mat and text.mat

```
[~, rectangleEigValues] = clustering(X2, 10, .05);
[~, textEigValues] = clustering(X4, 10, .05);

figure;
plot(1:10, diag(rectangleEigValues), 'g-o');
hold on;
plot(1:10, diag(textEigValues), 'b-x');
xlabel("Eigenvalues")
ylabel("Value")
legend('Rectagles EigValues', 'Text EigValues');
```



ANS

1. Rapid Decline in Rectangles Eigenvalues: The eigenvalues for the "Rectangles" dataset drop quickly after the first eigenvalue. This suggests that the first eigenvalue is significantly larger than the others, indicating that there may be one dominant dimension or feature in this dataset.
2. Gradual Decline in Text Eigenvalues: The "Text" eigenvalues show a more gradual decline. This implies a more distributed variance across the dimensions or features of the "Text" dataset.
3. Difference in Dataset Structures: The different shapes of the curves suggest that the inherent structures of the "Rectangles" and "Text" datasets are quite different. The "Rectangles" dataset may have a simpler or more coherent structure that can be captured by fewer dimensions, whereas the "Text" dataset seems to require more dimensions to capture its variance.

Ques 4) How do K means and Spectral Clustering compare?

Ans) K-means and spectral clustering are fundamentally different clustering techniques. K-means, a distance-based clustering method, assumes that clusters are spherical and is best suited for well-separated data. It's efficient for large datasets but can struggle with non-convex clusters and is sensitive to initial centroid placement.

Spectral clustering, on the other hand, relies on graph theory, using the eigenvectors of a similarity matrix to reduce dimensionality before clustering. This allows it to detect complex cluster structures that K-means cannot. However, spectral clustering can be computationally intensive, especially for large datasets, due to the eigenvalue decomposition required.

While K-means excels in computational efficiency and ease of implementation, spectral clustering is more robust in handling data with intricate geometries. The choice between them depends on the specific characteristics of the dataset and the computational resources available.

```
function [clusters, eigen_Values] = clustering(dataset, k, s)

    A = exp(-pdist2(double(dataset), double(dataset), "squaredeuclidean")/s);

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

    [eigVectors, eigen_Values] = eigs(N, k, 'largestabs');
    Y = eigVectors;

    Y = Y ./ sqrt(sum(Y.^2, 2));

    clusters = kmeans(Y, k);
end
```