Homework Set 4, CPSC 8420, Fall 2023

Singh, Parampreet

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

Problem 1

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

Problem 1

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

```
% Load the image
image = imread('clemson.jpeg');

% Downsample the image
scaleFactor = 0.8; % for example, adjust based on your needs
smallImage = imresize(image, scaleFactor);

[rows, cols, ~] = size(smallImage); % 'image' is the original RGB image
pixels = reshape(smallImage, rows*cols, 3); % Reshape into 2D matrix for clustering
```

Create Adjacency Matrix using K-Nearest Neighbor Graph with Gaussian Similarity

```
sigma = 0.2; % sigma
% Use the knnsearch function to find the k-nearest neighbors
k = 10; % KNN
[neighbors, distances] = knnsearch(double(pixels), double(pixels), 'K', k+1);

% Create the sparse adjacency matrix
rowIdx = repmat((1:size(pixels, 1))', 1, k);
colIdx = neighbors(:, 2:end); % exclude the first column which is the point itself
values = exp(-distances(:, 2:end).^2 / (2*sigma^2));
adjacencyMatrix = sparse(rowIdx, colIdx, values, size(pixels, 1), size(pixels, 1));
```

Create Laplacian Matrix using Adjacency and Degree Matrix

L = D - A

```
% Construct the degree matrix
degreeMatrix = diag(sum(adjacencyMatrix));

laplacianMatrix = degreeMatrix - adjacencyMatrix;
```

Perform Eigen Value Decomposition

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

Sort and select Eigen vectors for clustering

```
k = 2; % No of Clusters

% Find the sorted eigenvalues and their indices
[eigValuesSorted, idx] = sort(diag(eigValues));

% Find the indices of the first two non-zero eigenvalues
```

```
% Assuming the first eigenvalue is zero or near-zero, we skip it
idxNonZero = idx(eigValuesSorted > eps);

% Select the first k non-zero eigenvectors
selectedEigVectors = eigVectors(:, idxNonZero(1:k));
```

Normalize the rows

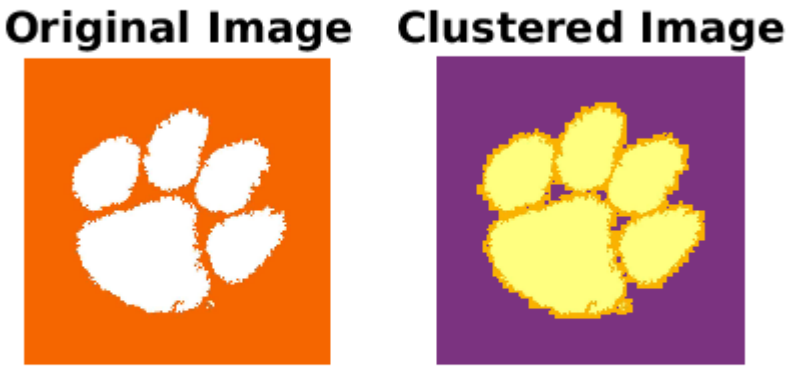
```
% Calculate the norm of each row
rowNorms = sqrt(sum(selectedEigVectors.^2, 2));

% Divide each row by its norm (avoid division by zero)
normalizedEigVectors = selectedEigVectors ./ (rowNorms + eps);
```

Apply the Clustering Algorithm

```
clusters = kmeans(normalizedEigVectors, k);

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

Display the Image

```
newImage = labeloverlay(smallImage, segmentedImage);

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



Problem 2

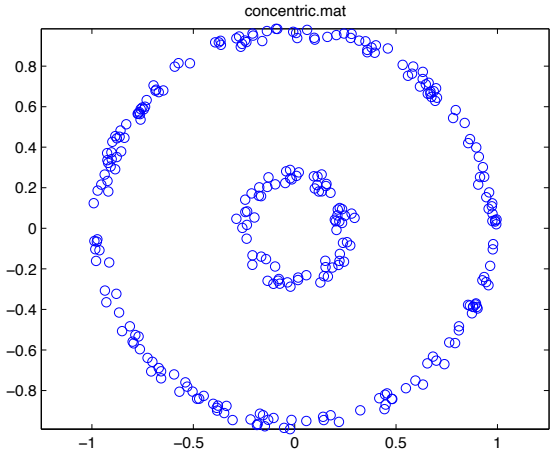
Frequently, the affinity matrix is constructed as:

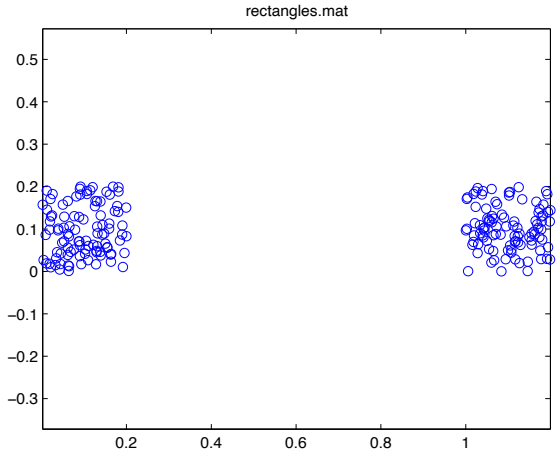
$$A_{ij} = e^{-d(x_i,x_j)^2/\sigma} \tag{1}$$

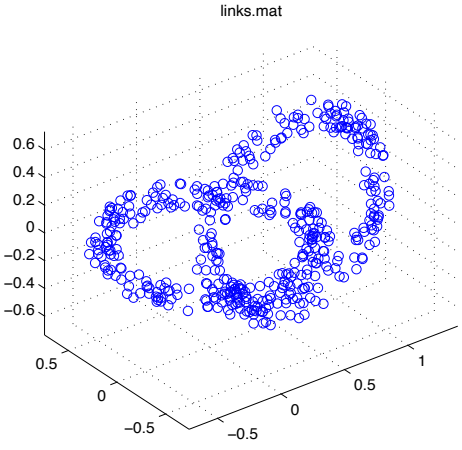
where σ is some user-specified parameter. The best that we can hope for in practice is a near block-diagonal affinity matrix. It can be shown in this case, that after projecting to the space spanned by the top k eigenvectors, points which belong to the same block are close to each other in a euclidean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

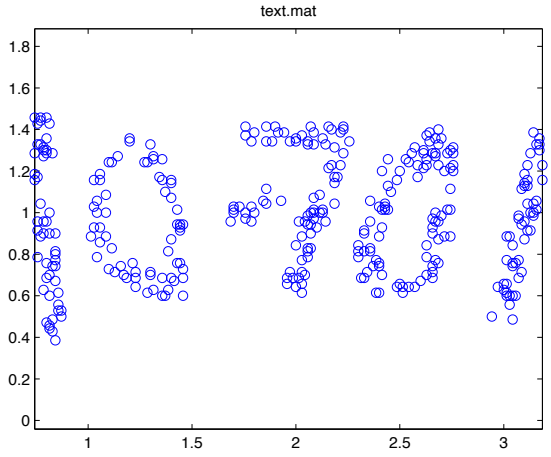
- Construct an affinity matrix A using the above equation.
- Symmetrically 'normalize' the rows and columns of A to get a matrix N such that $N(i,j) = \frac{A(i,j)}{\sqrt{d(i)d(j)}}$, where $d(i) = \sum_k A(i,k)$.
- Construct a matrix Y whose columns are the first k eigenvectors of N.
- Normalize each 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

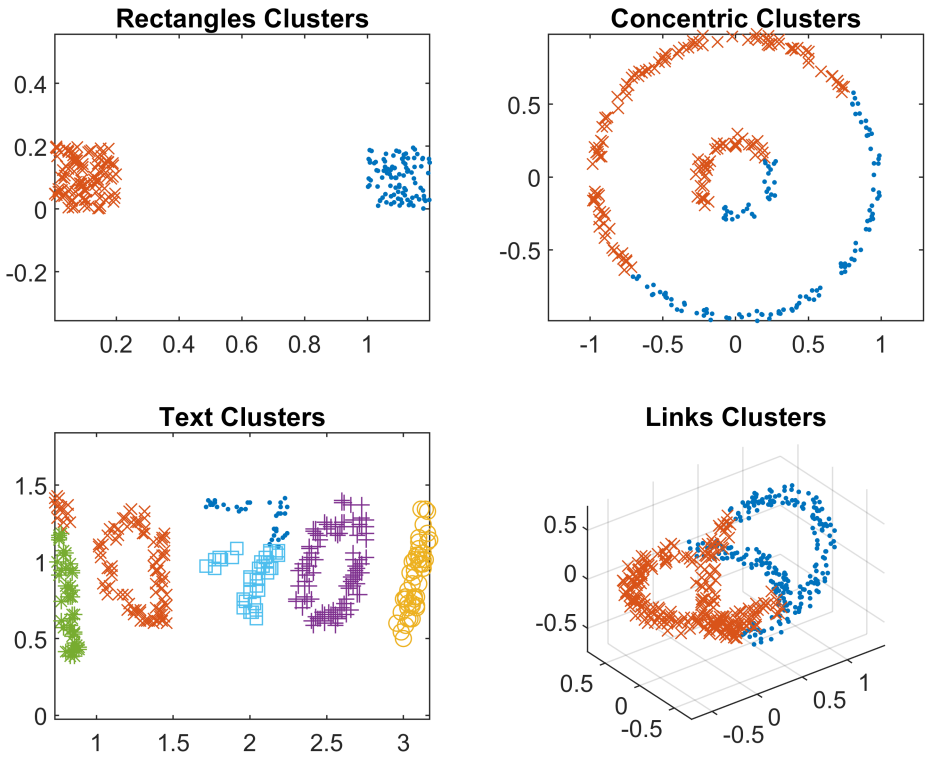

links.mat
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
0.5
0
−0.5
−0.5
0
0.5
1


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. **Implementing K-Means Clustering**

```
% Note - I removed the new figure initialization [figure;] in plotClusters.m in order
% to use subplots in my code.

% 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);

% Create a 2x2 grid of subplots
figure;

% Plot and title for rectangles
subplot(2,2,1);
plotClusters(X2, rect_clusterIdx);
title('Rectangles Clusters');
% Plot and title for concentric
subplot(2,2,2);
plotClusters(X1, conc_clusterIdx);
title('Concentric Clusters');
% Plot and title for text
subplot(2,2,3);
plotClusters(X4, text_clusterIdx);
title('Text Clusters');
% Plot and title for links
subplot(2,2,4);
plotClusters(X3, links_clusterIdx);
title('Links Clusters');
```



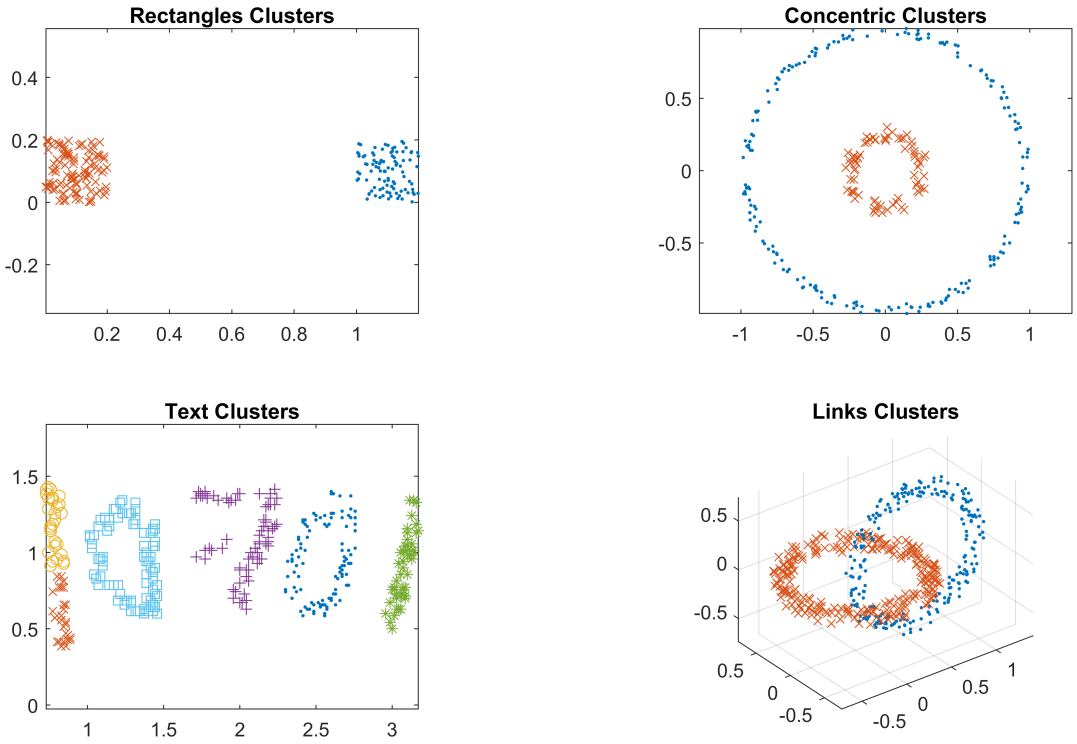

```
% Adjust the layout
sgtitle;
```

2. Implement affinity matrix spectral clustering

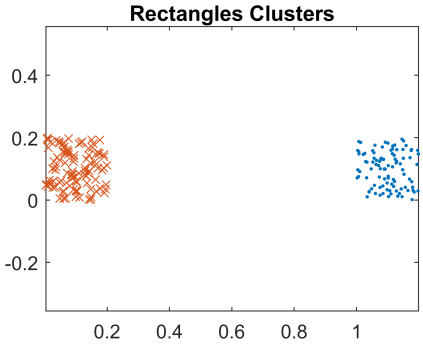
```
sigma = [.025, .05, .2, .5];
for i = sigma
    [rect_clusterIdx, ~] = affinity_matrix_clustering(X2, 2, i);
    [conc_clusterIdx, ~] = affinity_matrix_clustering(X1, 2, i);
    [text_clusterIdx, ~] = affinity_matrix_clustering(X4, 6, i);
    [links_clusterIdx, ~] = affinity_matrix_clustering(X3, 2, i);

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

    % Create a new figure window
    figure;

    % First subplot for rectangles
    subplot(2,2,1)
    plotClusters(X2, rect_clusterIdx);
    title('Rectangles Clusters');

    % Second subplot for concentric
    subplot(2,2,2)
    plotClusters(X1, conc_clusterIdx);
```

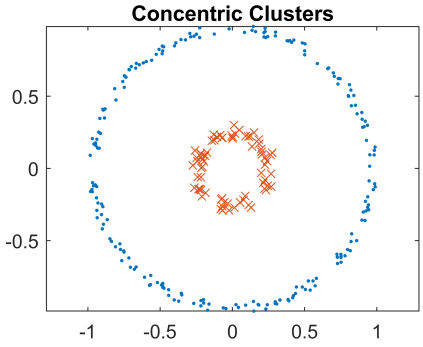
```
    title('Concentric Clusters');

    % Third subplot for text
    subplot(2,2,3)
    plotClusters(X4, text_clusterIdx);
    title('Text Clusters');

    % Fourth subplot for links
    subplot(2,2,4)
    plotClusters(X3, links_clusterIdx);
    title('Links Clusters');

    % Optionally, adjust the layout if the plots are overlapping
    set(gcf, 'Position', get(0, 'Screensize')); % For fullscreen, if needed
end
```

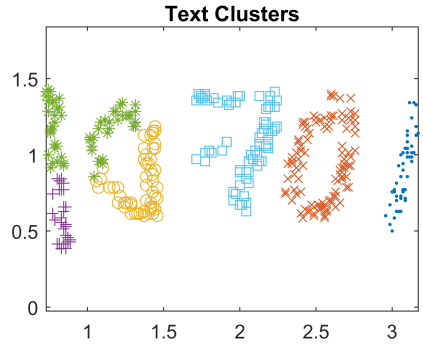
```
ans =
'Plots for sigma: 2.500000e-02'
```

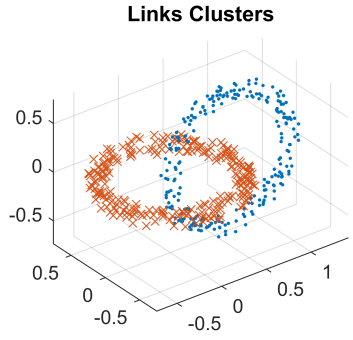



```
ans =
'Plots for sigma: 5.000000e-02'
```

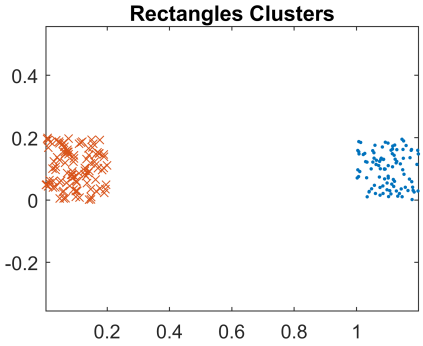


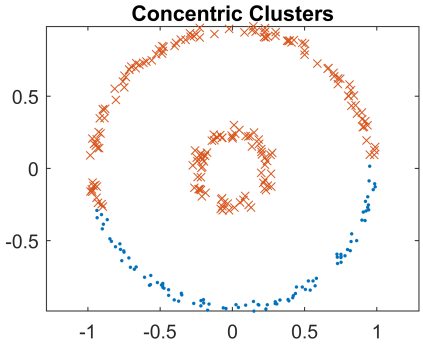


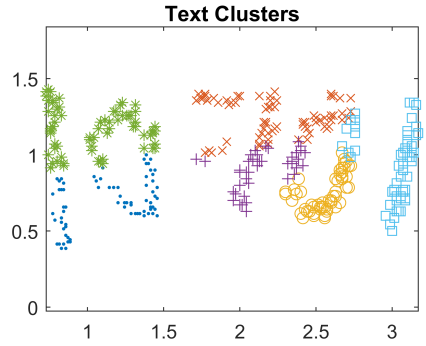


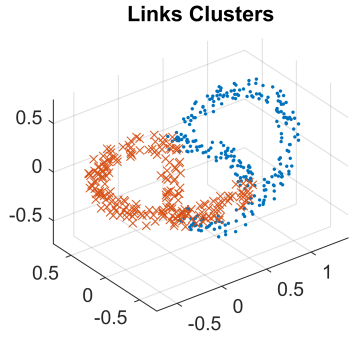



```
ans =
'Plots for sigma: 2.000000e-01'
```

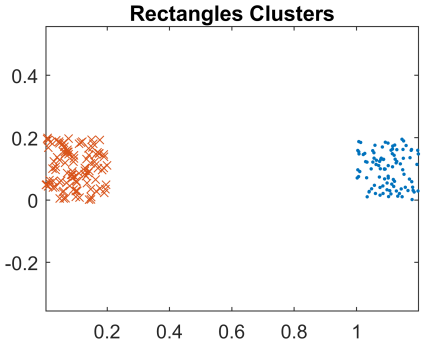


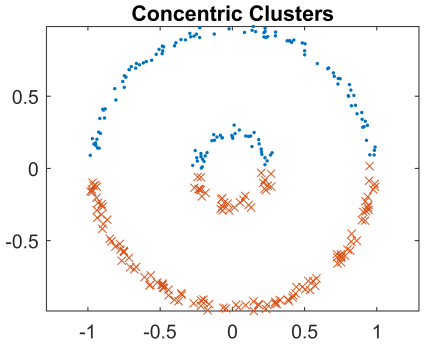


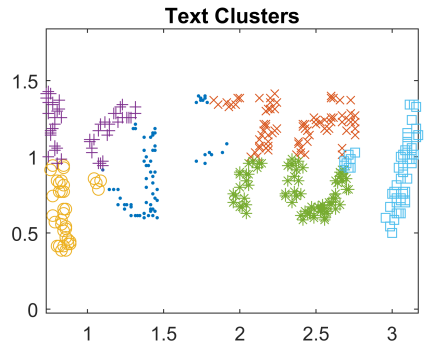


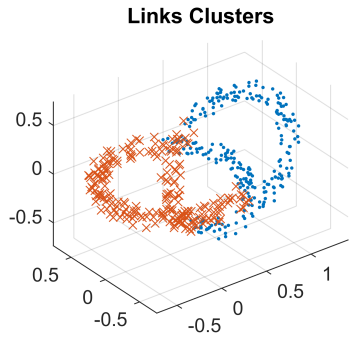



```
ans =
'Plots for sigma: 5.000000e-01'
```



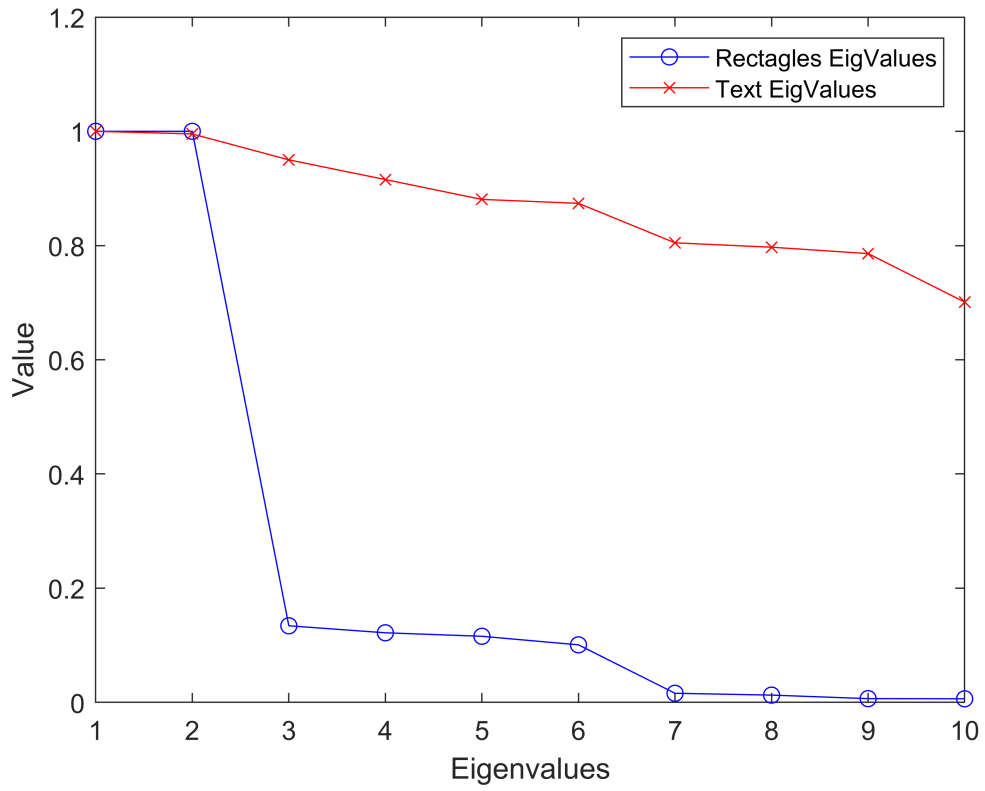






3. Plot first 10 Eigenvalues for rectangles.mat and text.mat

```
[~, rect_Eigs] = affinity_matrix_clustering(X2, 10, .05);
[~, text_Eigs] = affinity_matrix_clustering(X4, 10, .05);

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



ANS -

"Rectangles EigValues" shows a significant drop between the first and second eigenvalue, suggesting a strong principal component. This steep decline indicates that the first eigenvalue captures a substantial amount of the variance within the "Rectangles" dataset.

"Text EigValues" decreases more gradually, indicating that the variance in the "Text" dataset is more evenly spread across the components. There isn't a single eigenvalue that dominates, which could suggest a more complex underlying structure.

4. How do k-means and spectral clustering compare?

K-means clustering is great for simple, round clusters of similar size but falls short with intricate shapes or clusters of varying sizes. It separates rectangular clusters well but doesn't quite capture the layered pattern of concentric circles due to its reliance on straight-line distances.

Spectral clustering, on the other hand, excels in these complex scenarios. By considering how all data points relate to one another, it effectively reveals the circular layers and the subtle groupings in the text and link data.

In essence, while K-means imposes a straightforward but sometimes overly simplistic structure, spectral clustering adapts to the data's natural patterns, making it a better choice for intricate cluster shapes.