ASSIGNMENT~4

21MAT212

MIS ~ 4

Professor~Jithin Sir

Submitted by: Vikhyat Bansal [CB.EN.U4AIE21076]

Creating a graph with a predefined number of nodes and edges.

Code:

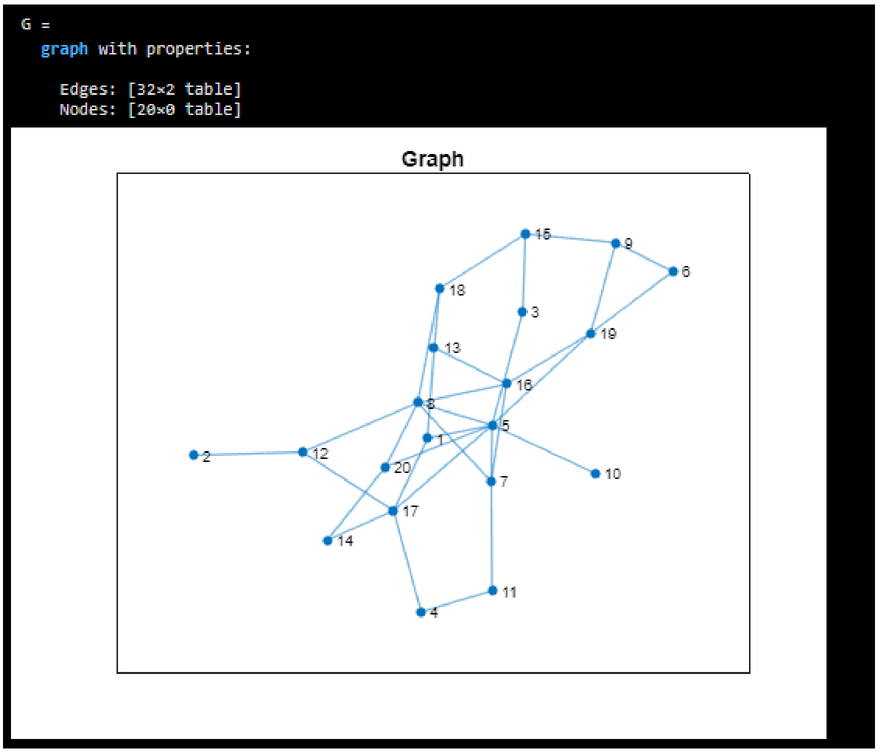
Function

```
function adjMatrix = randomSimpleGraph (n, e)
    adjMatrix = zeros (n);
    while sum (adjMatrix(:)) < 2*e % When the matrix written in row vector
form is less than twice the number of edges
        i = randi (n);
        j = randi (n);
        if i ~= j && adjMatrix(i,j) == 0
            adjMatrix(i,j) = 1;
            adjMatrix(j,i) = 1;
        end
    end
end
```

Input:

```
% for generating a random graph
A = randomSimpleGraph(20,32);
G = graph(A)
plot(G)
title("Graph")
```

Output:



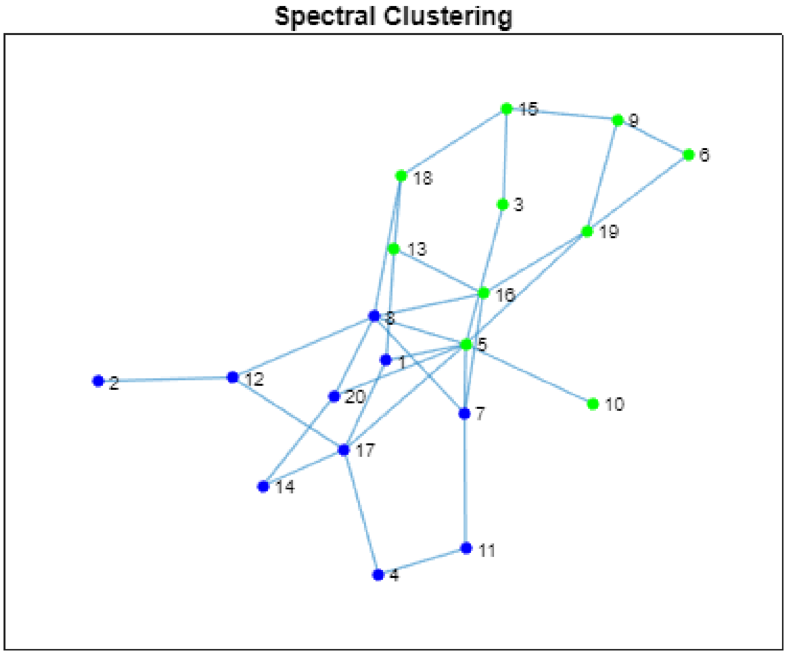
```
G =
  graph with properties:

    Edges: [32×2 table]
    Nodes: [20×0 table]
```

Code for performing Bipartition via spectral method on a graph.

Code:

```
L=laplacian(G); % Getting the laplacian matrix of graph G


[eigvec,eigval]=eigs(L,2,'smallestabs');  %the two smallest absolute eigenvalues and eigenvectors of L


i=eigvec(:,2); % taking the eigenvector corresponding to eigenvalue 2
c1=[];
c2=[];


for j=1:length(i)
    if i(j) > 0
        c1=[c1 j];
    else
        c2=[c2 j];
    end
end

spec_clus=plot(G);
title("Spectral Clustering")
highlight(spec_clus,c1,'NodeColor','g')
highlight(spec_clus,c2,'NodeColor','b')
```

Output:



Code for performing k-means clustering on a graph.

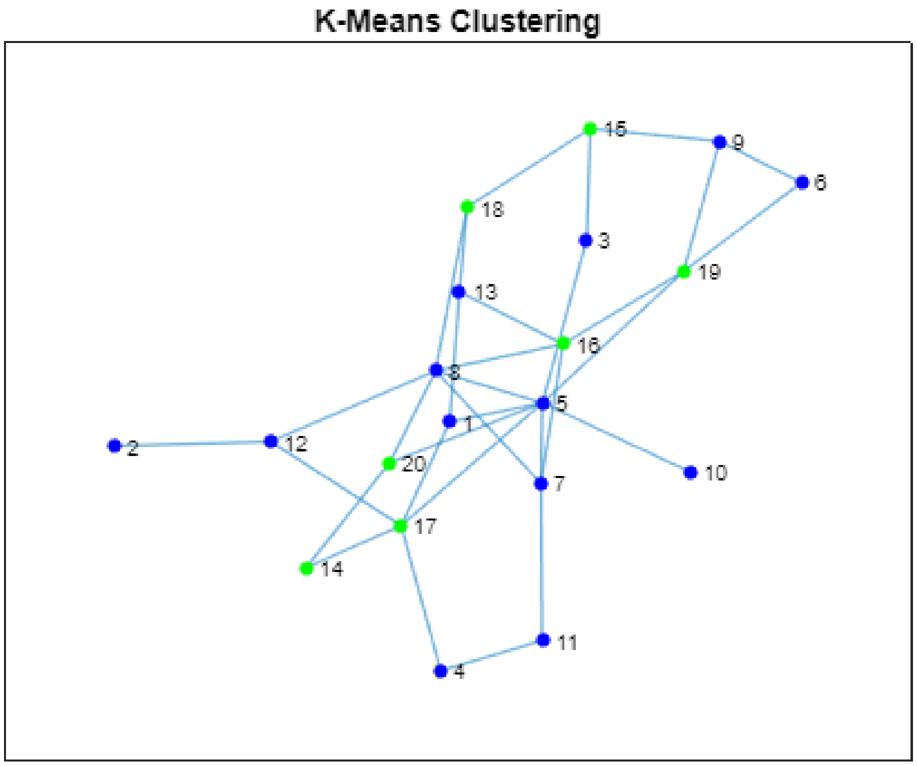
Code: Function

```
function [idx, centroids] = kmeans_using_for_loop(data, k)%function to perform kmeans
% Randomly initialize the centroids
idx = randi(k, size(data, 1), 1);
centroids = zeros(k, size(data, 2));
% Iterate until convergence
while true
    for i = 1:k
        centroids(i, :) = mean(data(idx == i, :));
    end
% Assign data points to the nearest centroid
    prevIdx = idx;
    for i = 1:size(data, 1)
        distances = sum((data(i, :) - centroids).^2, 2);
        [~, idx(i)] = min(distances);
    end
% Check for convergence
    if isequal(prevIdx, idx)
    break;
    end
end
end
```

Input:

```
evalue=eig(L); % The eigenvalues of graph L
idx=kmeans_using_for_loop(evalue,2); %Performing the kmeans on the graph


kc1=[];
kc2=[];


for j=1:length(idx)
    if idx(j)==1  % Splitting the data points to the clusters based on the centroid provided by k-means function
        kc1=[kc1 j];
    else
        kc2=[kc2 j];
    end
end


kmeans_clus=plot(G);
title("K-Means Clustering")
highlight(kmeans_clus,kc1,'NodeColor','g')
highlight(kmeans_clus,kc2,'NodeColor','b')
```

Output:



Difference between spectral clustering and k-means clustering

Spectral Clustering and K-Means Clustering are both unsupervised machine learning algorithms used for clustering, but they differ in how they define clusters.

- K-Means Clustering defines clusters as spherical clusters of points, where each point belongs to the cluster with the nearest mean. Spectral Clustering, on the other hand, defines clusters based on the eigenvectors of the similarity matrix of the data.
- In K-Means, the number of clusters is specified in advance and the algorithm tries to find the best clustering based on that number of clusters.
  In Spectral Clustering, the number of clusters is not specified and the algorithm finds the number of clusters that best fit the data.

- Spectral Clustering is often used when the data is not spherical or when the clusters are not well separated, such as when the data is non-linearly separable. K-Means, on the other hand, is often used when the data is spherical and the clusters are well separated.

In summary, Spectral Clustering is more flexible than K-Means and can be used in more complex scenarios, but it is also more computationally intensive.

Suppose we have a dataset of gene expression levels for different samples, where each sample has thousands of genes. The goal is to cluster the samples based on their gene expression profiles.
In this scenario, the data is likely to be high-dimensional, non-spherical, and the clusters may not be well separated.

Spectral Clustering is well-suited to this type of data because it can capture the underlying structure of the data, even if it is non-linear and high-dimensional. The algorithm can use a similarity matrix to measure the similarity between each pair of data points, and then use the eigenvectors of this matrix to find a low-dimensional representation of the data that preserves the important structure.

On the other hand, K-Means may struggle with this type of data because it assumes that the clusters are spherical and well-separated. It may also be sensitive to the initial choice of centroids and may not be able to capture the complex structure of the data.

Spectral Clustering has been used successfully in bioinformatics for clustering gene expression data, protein-protein interaction networks, and other types of biological data. By using Spectral Clustering, researchers can identify groups of genes or proteins that are co-expressed or functionally related, which can provide insights into important biological processes and pathways.

THANK YOU!!!