ASSIGNMENT-3

21MAT212

MIS - 4

Professor-Jithin Sir

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



1) Create a data set containing 3 different groups of data points and develop the MATLAB code to perform K-means clustering.

Code:

% Define the number of data points in each group

N = 75;

% Define the centers of each group

center1 = [-4 -2];

center2 = [2 -3];

center3 = [0 1];

% Generate the data points for each group

group1 = center1 + randn(N,2);

group2 = center2 + randn(N,2);

group3 = center3 + randn(N,2);

% Concatenate the data points into a single matrix

X = [group1; group2; group3];

% Define the number of clusters (groups)

K = 3;

% Initialize the cluster centroids randomly

cent\_arr = randperm(N\*3,K);

centroids = X(cent\_arr,:);

% Perform K-means clustering

max\_iter = 50;

for iter = 1:max\_iter

% Assign each data point to the nearest centroid

distances = zeros(N\*3,K);

for k = 1:K

for i = 1:N\*3

distances(i,k) = norm(X(i,:) - centroids(k,:));

end

end

[min\_dist,labels] = min(distances,[],2);

% Update the cluster centroids

for k = 1:K

centroids(k,:) = mean(X(labels==k,:),1);

end

end

% Plot the results

colors = ['r','g','b'];

figure;

hold on;

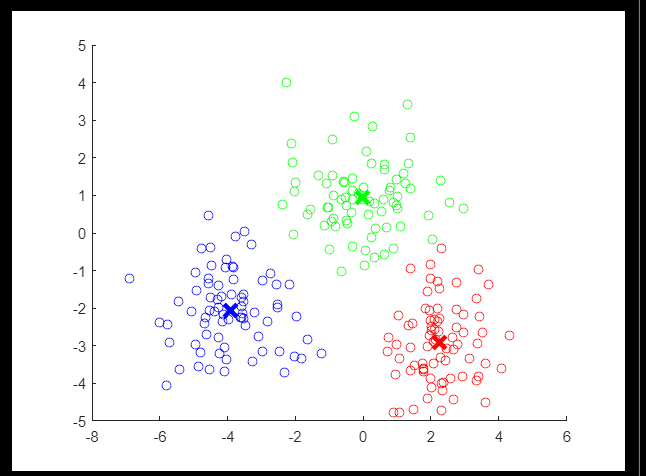
for k = 1:K

plot(X(labels==k,1),X(labels==k,2),[colors(k) 'o']);

plot(centroids(k,1),centroids(k,2),[colors(k) 'x'],'MarkerSize',12,'LineWidth',3);

end

hold off;

Output:

2.) Download a data set from an open source and develop MATLAB code to perform the K-means clustering. Use appropriate methodology to choose the best value for K.

Code:

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

% Update the centroids

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

function idealK = chooseIdealK(data, maxK)%function to get an ideal value

% of k using elbow method

% Initialize sum of squared distances array

sse = zeros(maxK, 1);

% Perform k-means clustering for different k values

for k = 1:maxK

[idx, centroids] = kmeans\_using\_for\_loop(data, k);

% Calculate sum of squared distances for each cluster

for i = 1:k

sse(k) = sse(k) + clustersum(data,idx,centroids,i);

end

end

% Plot the sum of squared distances

figure;

plot(1:maxK, sse, 'o-');

xlabel('Number of Clusters (k)');

ylabel('Sum of Squared Distances');

title('Elbow Method');

% Taking ideal k value from the graph made by elbow point

prompt = 'Enter the ideal k value: ';

idealK = input(prompt);

end

function sum = clustersum(data,idx,centroids,k)%function to calculate varianace

% in the clusters using

% sum of square distances

sum=0;

for i = 1:k

for j=1:length(data(idx == i, :))

sum=sum+norm(data(idx == i, :)-centroids(i,:))^2;

end

end

end

Input: Dataset

clc;clear all;

load fisheriris % Loading the dataset from matlab inbuilt

X = meas(:,3:4);

figure;

plot(X(:,1),X(:,2),'k\*','MarkerSize',5);

title 'Fisher''s Iris Data';

xlabel 'Petal Lengths (cm)';

ylabel 'Petal Widths (cm)';

%getting the ideal value of k using elbow method

k=chooseIdealK(X,10);

%performing kmeans with the ideal k

[idx, centroids] = kmeans\_using\_for\_loop(X, k);

% Plot the results

figure;

gscatter(X(:, 1), X(:, 2),idx,'bgmkr');

hold on;

scatter(centroids(:, 1), centroids(:, 2),'kx');

G = max(idx) ;

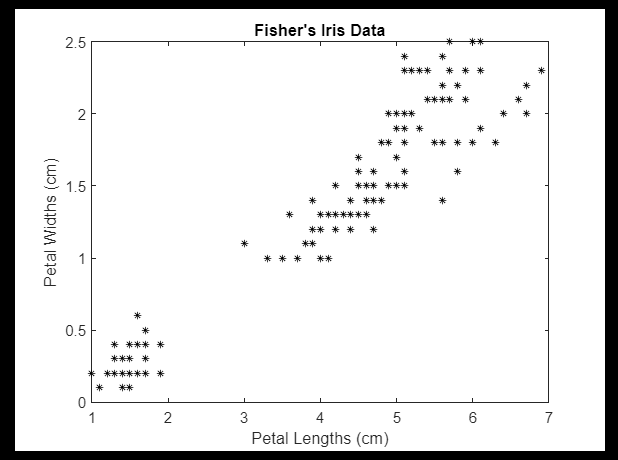
str = repmat("Cluster ",G,1) ;

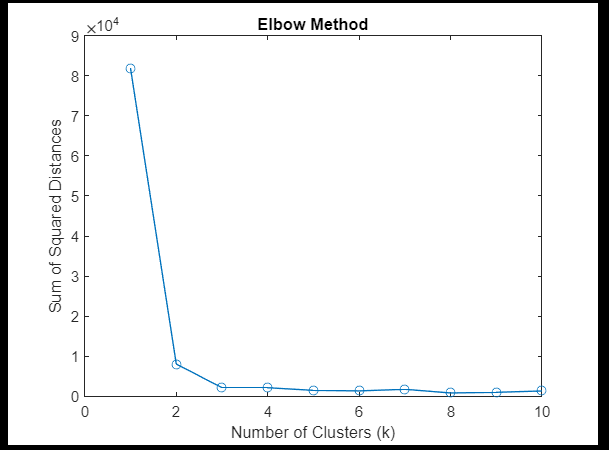
str = strcat(str,string((1:G)')) ;

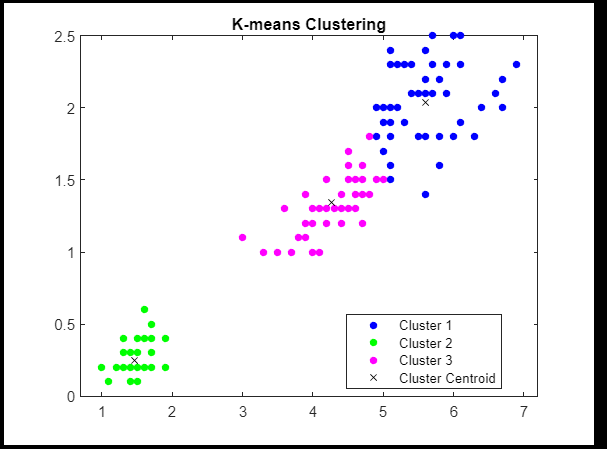
lgd = legend([str ; "Cluster Centroids"]) ;

title('K-means Clustering');

Output:







THANK YOU !!

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