K-means clustering

Vishal S
MM14B048

Data Set

The data is visualised using MATLAB. The scatter plot is depicted in Fig. below.

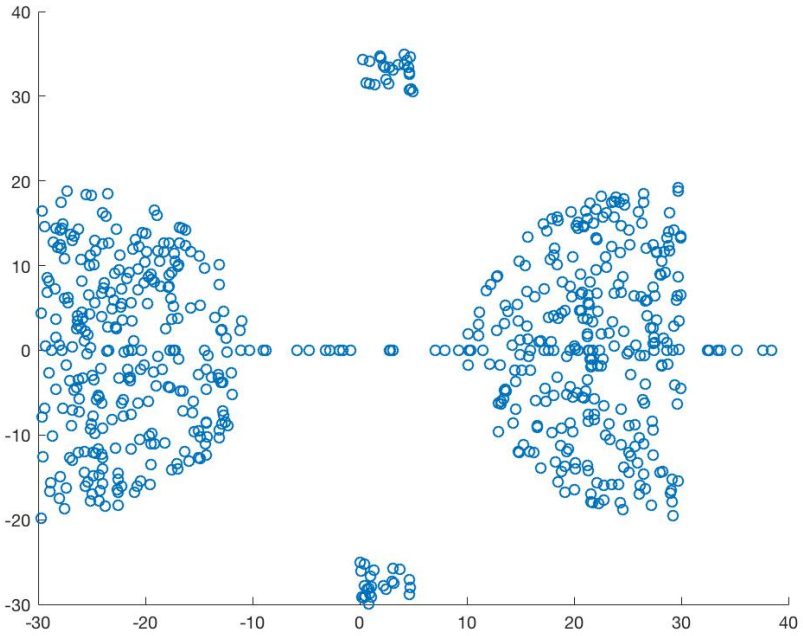


Figure 1: Outlier.txt

Outlier.txt

The data has 4 clusters and some noise points.

Cluster Id	No of Points
Cluster 1	276
Cluster 2	276
Cluster 3	24
Cluster 4	24
Noise	50
Total	650

K-means algorithm

A MATLABcode was written to cluster the data based on K-means algorithm. First the data is read into the code using the code snippet given below.

```
%% Load data
clear all
clf
clc
```

```
hold on
filename = 'outlier.txt';
% read from the file
fileID = fopen(filename,'r');
formatSpec = '%f,%f,%f';
sizeA = [3 Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
```

Once the data is loaded, the K-means algorithm is implemented using the code snippet given below.

```
%% Kmeans algorithm
% initialising the clusters' means
for i = 1:no_of_clusters
    for j = 1:2
    mean_old(j,i) = A(j,initial_mean(i));
    end
end

while ((convergence > convergence_limit)&&(iteration < max_iteration))
    X = ['Iteration = ',num2str(iteration)];
    disp(X);
    % Assigning each point to a cluster
    dist = 0.0;
    for i = 1:max_len
        dist_x = (A(1,i)-mean_old(1,1))^2;
        dist_y = (A(2,i)-mean_old(2,1))^2;
        dist = dist_x + dist_y;
        index = 1;
        for j = 2:no_of_clusters
            dist_x = (A(1,i)-mean_old(1,j))^2;
            dist_y = (A(2,i)-mean_old(2,j))^2;
            if(dist > dist_x + dist_y)
                dist = dist_x + dist_y;
                index = j;
            end
        end
        cluster_id(i,1) = index;
    end

    for i = 1:no_of_clusters
        sum(1,i) = 0.0;
        sum(2,i) = 0.0;
        no_of_points(1,i) = 0.0;
    end
    % Calculating new mean
    for i = 1:max_len
        sum(1,i) = 0.0;
        sum(2,i) = 0.0;
        sum(1,cluster_id(i,1)) = sum(1,cluster_id(i,1)) +  A(1,i);
        sum(2,cluster_id(i,1)) = sum(2,cluster_id(i,1)) +  A(2,i);
        no_of_points(1,cluster_id(i,1)) =no_of_points(1,cluster_id(i,1)) + 1;
    end

    % Calculating the new means
    for i = 1:no_of_clusters
        for j = 1:2
            mean_new(j,i) = sum(j,i)/no_of_points(1,i);
        end
    end

    % Caclulating SSE
    sse_new = 0.0;
    for i = 1:max_len
        for j = 1:2
            sse_new = sse_new + ( A(j,i)-mean_new(j,cluster_id(i)))^2;
```

```
        end
    end
    convergence = abs((sse_new - sse_old)) / sse_new ;
    % update the variables
    iteration = iteration + 1;
    sse_old = sse_new;
    mean_old = mean_new;
end
```

Finally, the results are plotted.

```
%% Plot results
k=max(cluster_id);
Colors=hsv(k);
Legends = {};
for i=0:k
    A_i=A(:,cluster_id==i);
    if i~=0
        Style = 'x';
        MarkerSize = 8;
        Color = Colors(i,:);
        Legends{end+1} = ['Cluster #' num2str(i)];
    end
    if ~isempty(A_i)
        %scatter3(A_i(1,:),A_i(2,:),A_i(3,:),Style,'MarkerSize',MarkerSize,'
    MarkerFaceColor',Color);
        scatter(A_i(1,:),A_i(2,:),Style,'MarkerFaceColor',Color);
    end
    hold on;
end
Style = 'o';
MarkerSize = 10;
Color = [0 0 0];
Legends{end+1} = 'Centroids';
scatter(mean_new(1,:),mean_new(2,:),Style,'MarkerFaceColor',Color);

hold off;
grid on;
legend(Legends);
legend('Location', 'NorthEastOutside');
```

K-means Implementation

Outlier.txt

No of clusters = 2

The above code was executed with the number of clusters equal to two and the following output was obtained. The centroids and the SSE obtained for each case is given below.

```
SSE =138047.9529
Centroids
      Centroid #1=[18.9961 0.78908]
      Centroid #2=[-21.0431
    -0.27064]
```

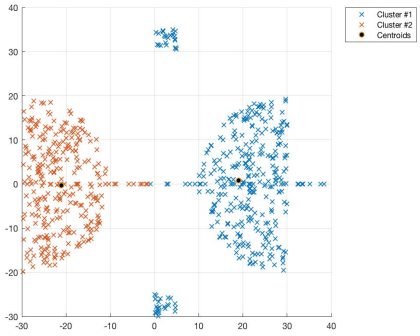
Listing 1: Case 1

```
SSE =137470.4669
Centroids
      Centroid #1=[20.122 2.4939]
      Centroid #2=[-19.5722 -2.0021]
```

Listing 2: Case 2

No of clusters = 4

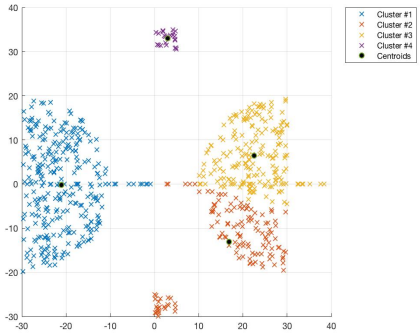
The above code was executed with the number of clusters equal to four and the following output was obtained. The centroids and the SSE obtained for each case is given below.



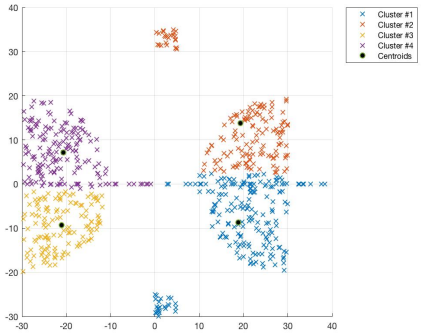
(a) Case 1

(b) Case 2

Figure 2: K-means with K=2



(a) Case 1



(b) Case 2

Figure 3: K-means with K=4

```
SSE =71740.5202
Centroids
      Centroid #1=[-21.0418
    -0.25116]
      Centroid #2=[16.8663 -13.1015]
      Centroid #3=[22.5359 6.4484]
      Centroid #4=[3.0439 32.9744]
```

Listing 3: Case 1

```
SSE =74377.92
Centroids
      Centroid #1=[18.8815 -8.6685]
      Centroid #2=[19.2873 13.8065]
      Centroid #3=[-21.094 -9.3189]
      Centroid #4=[-20.6517 7.1059]
```

Listing 4: Case 2

Observation

We observe that, the bigger clusters are getting classified as two different clusters. The noise points are also getting clustered.