Lab exp : KNN modelling for Alphabet A and B

Table of Contents

Aim: 1
Theory: 1
Part-1 Alphabet K-means 2
Part 2 4
Conclusion: 7

Name: Ventrapragada Sai Shravani

PRN:17070123120

Batch:Entc(2017-21) G-3

Aim:

Implement K-means clustering and Hierarchical clustering

Theory:

K-means is the partitioning method that treats observations in your data as objects having locations and distances from each other. It partitions the bjects into K mutually exclusive clusters are as such that object within each cluster close to each as possible and far from objects in other clusters as possible.

Each cluster is characterised by its centroid and the distances used in clustering often don't represent special distances.

Hierarchical clustering is a way to investigate grouping in your data simultaneously over a variety of scales a distance by creating a cluster tree.

This tree is not a single set of clusters as in K-means but rather a multi-level hierarchy where clusters at one level are joined as clusters at the next higher level. THis allows one to decide what scale or level of clustering is most appropriate in your application.

The function kmeans performs K-Means clustering, using an iterative algorithm that assigns objects to clusters so that the sum of distances from each object to its cluster centroid, over all clusters, is a minimum. Used on Fisher's iris data, it will find the natural groupings among iris specimens, based on their sepal and petal measurements. With K-means clustering, you must specify the number of clusters that you want to create. First, load the data and call kmeans with the desired number of clusters set to 2, and using squared Euclidean distance. To get an idea of how well-separated the resulting clusters are, you can make a silhouette plot. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

```
clc
clear all
close all
```

Part-1 Alphabet K-means

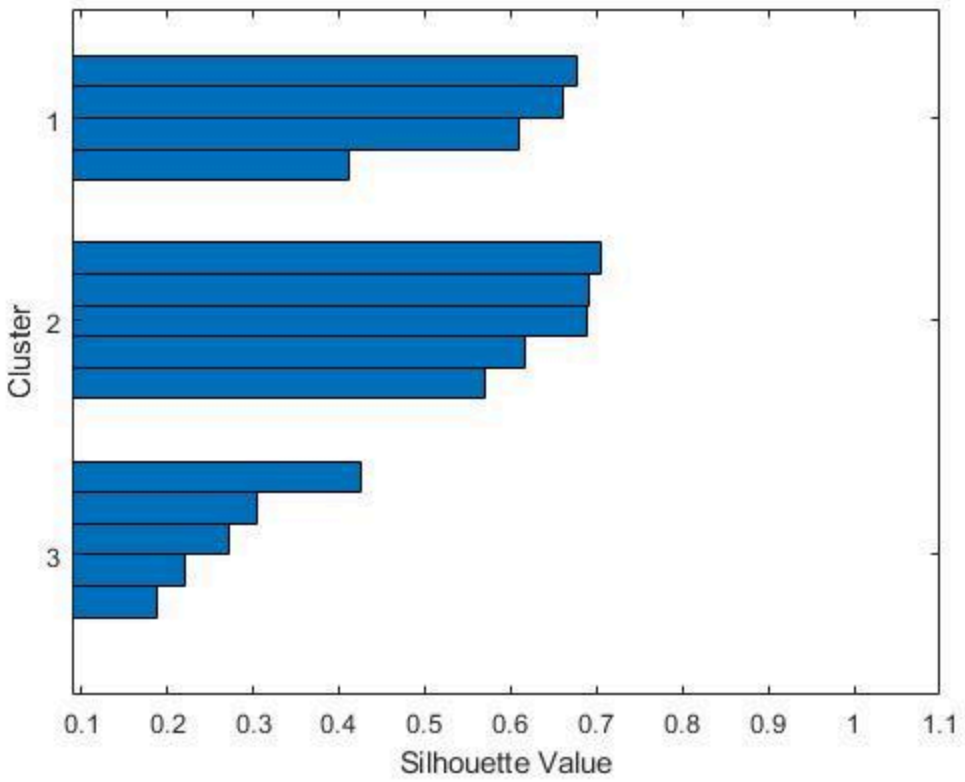
K-means clustering for A,B,C alphabets

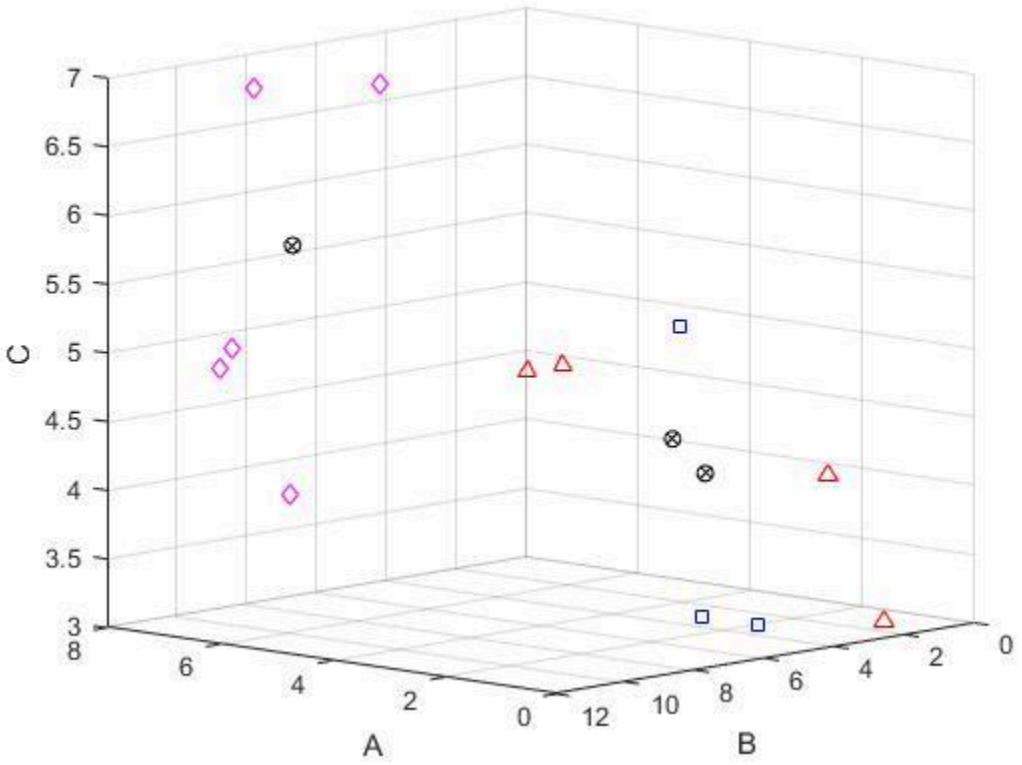
```
num = xlsread('dataset.xlsx');
save ds.mat num
load ds
%which kmeans
%idx= kmeans(num,2)
figure(1);
[cidx2,cmeans2] = kmeans(num,3,'dist','sqeuclidean');
[cidx3,cmeans3,sumd3] =
 kmeans(num,3,'replicates',5,'display','final');
sum(sumd3)
[silh3,h] = silhouette(num,cidx3,'sqeuclidean');
figure(2);
ptsymb = {'bs','r^','md','go','c+'};
for i = 1:3
    clust = find(cidx3==i);
    plot3(num(clust,1),num(clust,2),num(clust,3),ptsymb{i});
    hold on
end
plot3(cmeans3(:,1),cmeans3(:,2),cmeans3(:,3),'ko');
plot3(cmeans3(:,1),cmeans3(:,2),cmeans3(:,3),'kx');
hold off
xlabel('A');
ylabel('B');
zlabel('C');
view(-137,10);
grid on
```

```
Replicate 1, 2 iterations, total sum of distances = 503.1.
Replicate 2, 1 iterations, total sum of distances = 638.8.
Replicate 3, 1 iterations, total sum of distances = 563.79.
Replicate 4, 1 iterations, total sum of distances = 664.889.
Replicate 5, 2 iterations, total sum of distances = 602.171.
Best total sum of distances = 503.1

ans =

  503.1000
```


Cluster
1
2
3
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
Silhouette Value

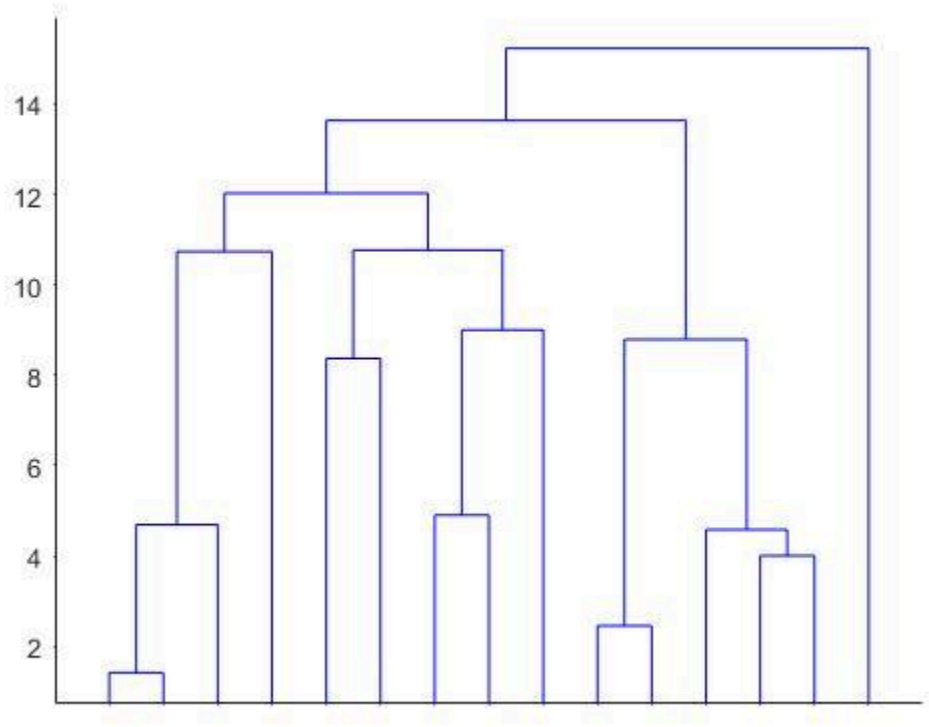

C
7
6.5
6
5.5
5
4.5
4
3.5
3
A
8
6
4
2
0
B
12
10
8
6
4
2
0

Alphabet characterisation Using Hierarchical Clustering

```
eucD = pdist(num,'euclidean');
clustTreeEuc = linkage(eucD,'average');
cophenet(clustTreeEuc,eucD)
figure(3);
[h,nodes] = dendrogram(clustTreeEuc,0);
h_gca = gca;
h_gca.TickDir = 'out';
h_gca.TickLength = [.002 0];
h_gca.XTickLabel = [];
```

```
ans =

    0.8162
```



Part 2

K-means Clustering for fisher iris

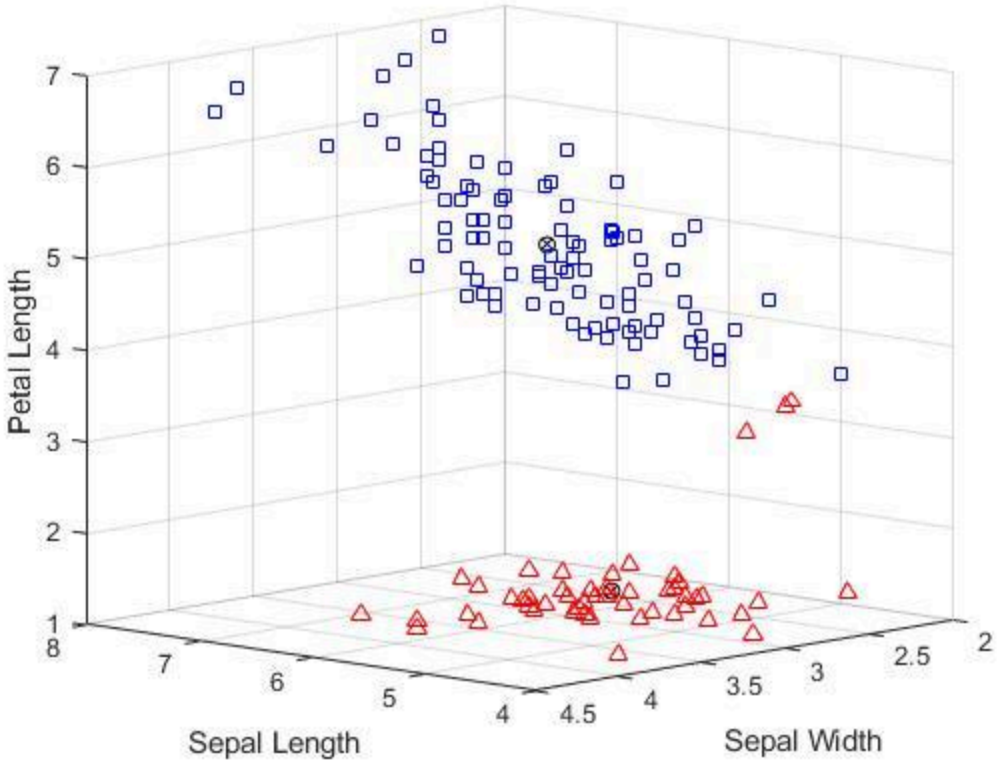
```
load fisheriris
figure(4);
[cidx2,cmeans2] = kmeans(meas,2,'dist','sqeuclidean');
[silh2,h] = silhouette(meas,cidx2,'sqeuclidean');
```

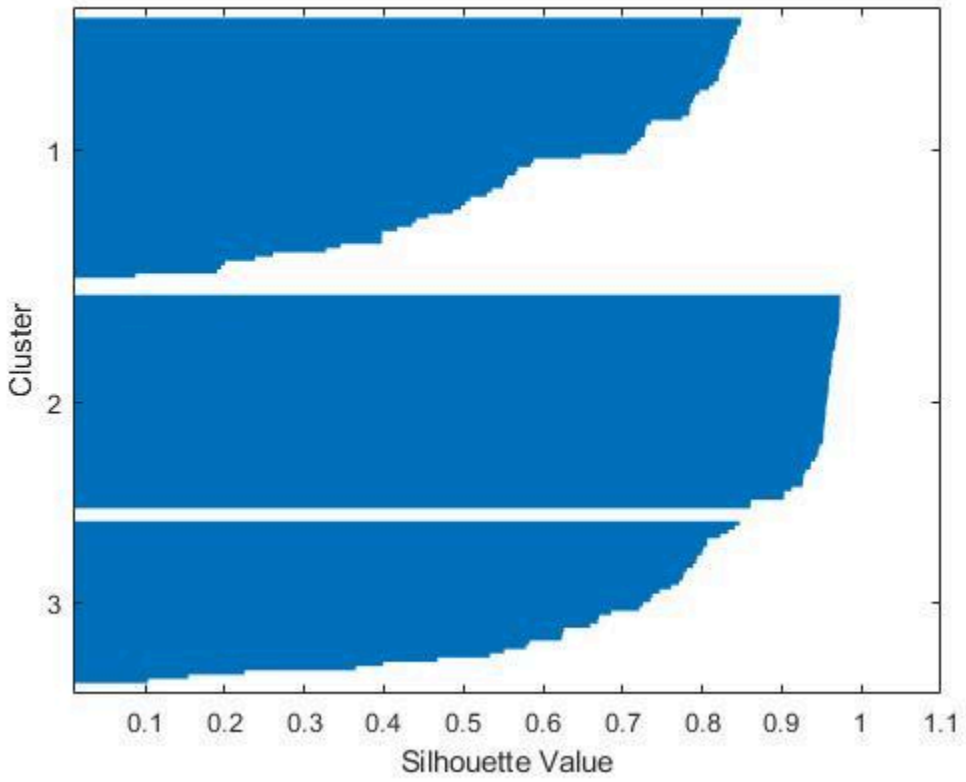
```
ptsymb = {'bs','r^','md','go','c+'};
for i = 1:2
    clust = find(cidx2==i);
    plot3(meas(clust,1),meas(clust,2),meas(clust,3),ptsymb{i});
    hold on
end
plot3(cmeans2(:,1),cmeans2(:,2),cmeans2(:,3),'ko');
plot3(cmeans2(:,1),cmeans2(:,2),cmeans2(:,3),'kx');
hold off
xlabel('Sepal Length');
ylabel('Sepal Width');
zlabel('Petal Length');
view(-137,10);
grid on
[cidx3,cmeans3] = kmeans(meas,3,'Display','iter');
[cidx3,cmeans3,sumd3] =
 kmeans(meas,3,'replicates',5,'display','final');
sum(sumd3)
figure(5);
[silh3,h] = silhouette(meas,cidx3,'sqeuclidean');
```

```
  iter  phase      num          sum
     1      1      150      96.6413
     2      1       10      84.8977
     3      1        7      79.5415
     4      1        3      78.8514
Best total sum of distances = 78.8514
Replicate 1, 6 iterations, total sum of distances = 78.8514.
Replicate 2, 9 iterations, total sum of distances = 78.8557.
Replicate 3, 10 iterations, total sum of distances = 78.8557.
Replicate 4, 7 iterations, total sum of distances = 78.8557.
Replicate 5, 4 iterations, total sum of distances = 78.8514.
Best total sum of distances = 78.8514

ans =

   78.8514
```


Petal Length
7
6
5
4
3
2
1
8
7
6
5
4
Sepal Length
4.5
4
3.5
3
2.5
2
Sepal Width

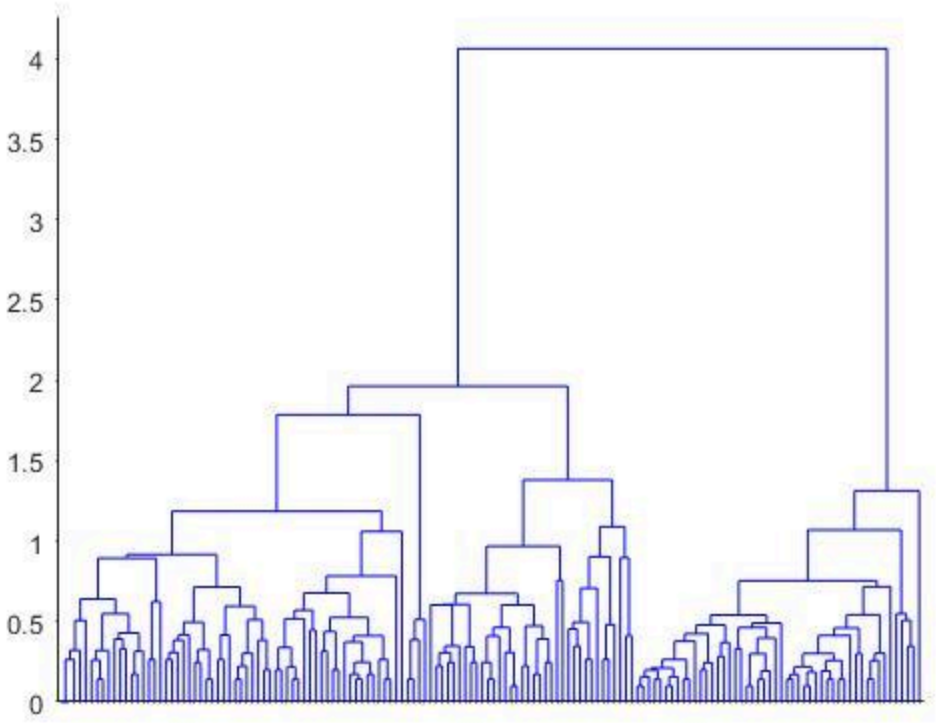

Cluster
1
2
3
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
Silhouette Value

Iris Data Using Hierarchical Clustering

```
eucD = pdist(meas,'euclidean');
clustTreeEuc = linkage(eucD,'average');
cophenet(clustTreeEuc,eucD)
figure(6);
[h,nodes] = dendrogram(clustTreeEuc,0);
h_gca = gca;
h_gca.TickDir = 'out';
h_gca.TickLength = [.002 0];
h_gca.XTickLabel = [];
```

```
ans =

    0.8770
```



Conclusion:

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). We extracted features for Alphabet A and alphabet B and also for fisher iris data the dendrograms were plotted respectively and KNN function was observed.

Published with MATLAB® R2020a