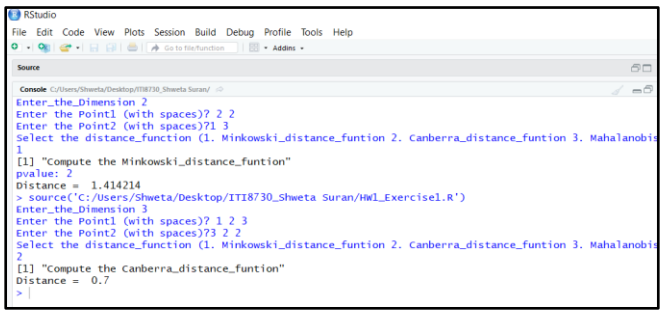
Data Mining – Home Assignment 1

15th October' 2019

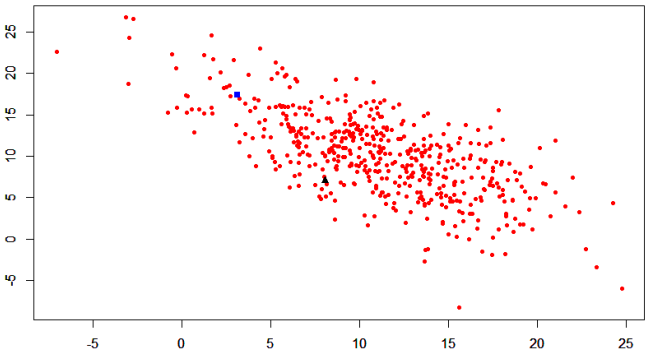
Shweta Suran

- **Exercise 1**

To compute distances using Minkowski distance and Canberra distance, we ask the user to enter the dimension of data points and coordinates. Using for-loops with limit up to dimension, the program calculates ***abs(P1–P2)^p*** and ***abs(P1–P2)/{abs(P1)+abs(P1)}*** for Minkowski and Canberra, resp. Values from each iteration of for-loop are summed to respective distances. For final Minkowski distance is summed distance to the power ***1/p***.



For Mahalanobis distance, we first initialize 500 random points using a covariance matrix (CM) and mean values (from Lab 2). The program then selects two random points and calculates distance using ***sqrt{transpose(P1-P2)*inv(CM)* (P1-P2)}***.

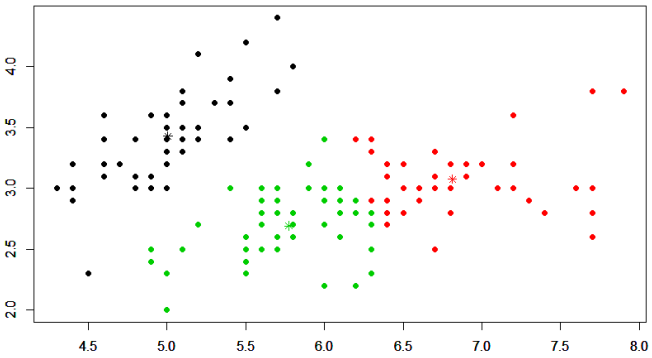


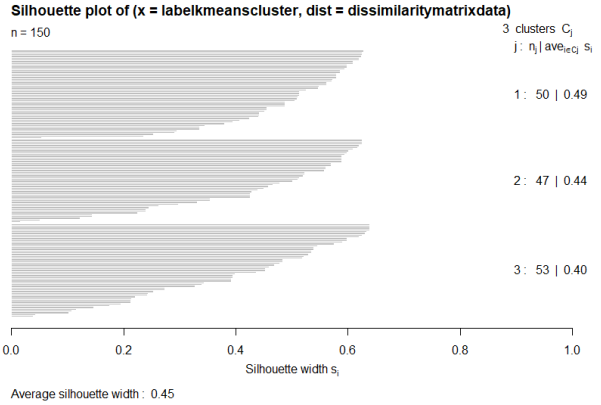
- **Exercise 2**

Input data is from iris-dataset (attributes: sepal length and width). Data has 150 data points with two dimensions segregated into 3 clusters. Input parameters: data points, initial centroids (random), total iterations, number of clusters are passed to k-means and k-medoid functions.

○ *k-means*

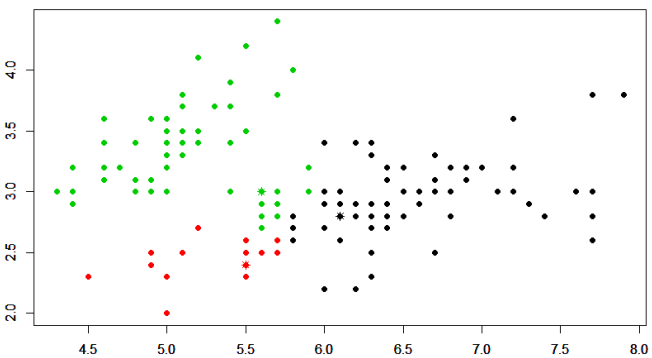
For k-means, the program calculates distance between centroids and all data points using Euclidean distance. Data points closest to any centroid are labeled into the respective cluster. After this, new centroid for each cluster is calculated using mean of data points in that cluster. The same steps are repeated until termination condition (iteration count).



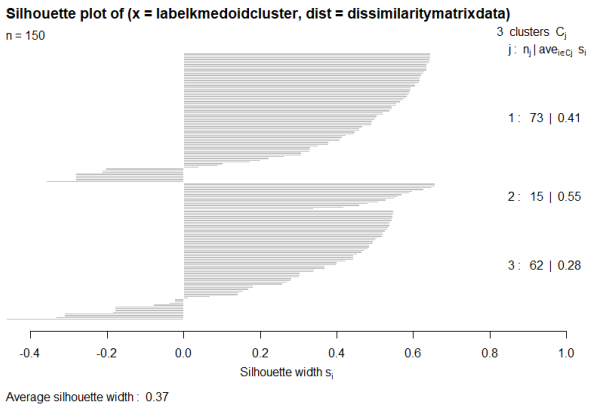


○ *k-medoids*

For k-medoid, the program calculates the distance between centroids and all data points using Manhattan distance. Data points closest to any centroid are labeled into the respective cluster. After this, new centroid for each cluster is initialized by selecting the data points closest to the previous centroids. The same steps are repeated until termination condition (iteration count). The least cluster cost through all iterations and its respective centroids are returned as final centroids.



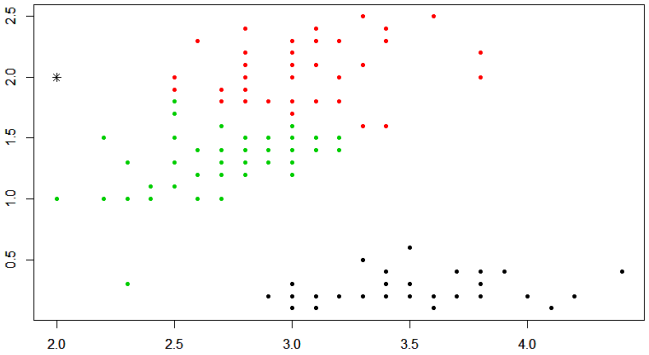
To evaluate clusters, Silhouette is plotted using ***silhouette*** R-Studio function. Inter-Intra distance ratios are calculated using ***cls.scatt.data*** R-Studio function.



- **Exercise 3**

Input data is from iris-dataset (attributes: petal length, petal width, and label). Data [DP_{ix}, DP_{iy}] has 150 data points segregated into 3 clusters.

User is asked to enter the value of 'k' (nearest neighbor count) for knn. Then user is asked to enter the x-y coordinates for the data point [Pnt_x, Pnt_y].



The program them calculates:

Euclidean: ***[{(DP_{ix}-Pnt_x)^2} + {(DP_{iy}-Pnt_y)^2}]^0.5***

Manhattan: ***abs(DP_{ix}-Pnt_x) + abs(DP_{iy}-Pnt_y)***

Chebyshev: ***max(abs(DP_{ix}-Pnt_x), abs(DP_{iy}-Pnt_y))***

Canberra: ***abs(DP_{ix}-Pnt_x)/{abs(DP_{ix})+abs(Pnt_x)} + abs(DP_{iy}-Pnt_y)/{abs(DP_{iy})+abs(Pnt_y)}***

… distances between the user's point and all input data points. 'k' points with least distances (for each distance function) are selected. Among the 'k' closest points, the label with highest number of occurrences is the resulting cluster for the respective distance function.

```
Console C:/Users/Shweta/Desktop/ITI8730_Shweta Suran/
> source('C:/Users/Shweta/Desktop/ITI8730_Shweta Suran/HW1_Exercise3.R')

Favourable XrangeAxis 2 - 4.4
Favourable YrangeAxis 0.1 - 2.5
Put value-p (0 to exit):
p? 20
enter x-coordinate: 2
enter y-coordinate: 2

KNN for eucli 2
KNN for manh 2
KNN for cheb 3
KNN for can 2
```

Results: *User's point [2,2] is clustered as Cluster 2 for Euclidean, Cluster 2 for Manhattan, Cluster 3 for Chebyshev and Cluster 2 for Canberra* [in reference to the above figure].

- **Exercise 4**

For this problem, the same iris-dataset (attributes: sepal length and petal width) is used for Fisher's exact test. Attributes are paired into groups and fisher.test R-Studio function is called to calculate the p-values (the fisher.test computes the p-value by summing the probabilities for all values with probabilities less than or equal to that of the observed value). The finest 20 pairs are chosen and the results for the same are displayed in the console.

```
Element petal.length & petal.width
        Fisher's Exact Test for Count Data with simulated p-value (based on 2000 replicates)

data:  tabledata
p-value = 0.0004998
alternative hypothesis: two.sided


Element petal.width & sepal.length
        Fisher's Exact Test for Count Data with simulated p-value (based on 2000 replicates)

data:  tabledata
p-value = 0.0004998
alternative hypothesis: two.sided


Element petal.width & sepal.width
        Fisher's Exact Test for Count Data with simulated p-value (based on 2000 replicates)

data:  tabledata
p-value = 0.0004998
alternative hypothesis: two.sided


Element petal.width & petal.length
        Fisher's Exact Test for Count Data with simulated p-value (based on 2000 replicates)

data:  tabledata
p-value = 0.0004998
alternative hypothesis: two.sided
```

- **References**

- R-Documentation (https://www.rdocumentation.org)
- Dimensionality (https://en.wikipedia.org/wiki/Curse_of_dimensionality)
- Tutorials Point (https://www.tutorialspoint.com/)
- Euclidean Distance (https://stackoverflow.com/questions/45780199/function-to-calculate-euclidean-distance-in-r)
- Tutorial Gateway (https://www.tutorialgateway.org)
- Fisher's test (https://www.youtube.com/watch?v=POiHEJqmiC0)
- kNN Algorithm | R Programming (https://www.youtube.com/watch?v=IDCWX6vCLFA)
- R Programming Tutorial - Learn the Basics of Statistical Computing (https://www.youtube.com/watch?v=_V8eKsto3Ug)
- k-means (http://rpubs.com/Nitika/kmeans_Iris)