PAM: introduction

The Partitioning Around Medoids (PAM) algorithm (also known as k-medoids) is a clustering technique aiming at minimizing the sum of dissimilarities between points in a dataset and the nearest medoid, where a medoid is the most centrally located point in a cluster. The main objective function is to minimize the total cost:

$$\sum_{i=1}^{n} \min_{j \in \{1,2,\ldots,k\}} d(x_i, m_j),$$

where $d(x_i, m_j)$ is the dissimilarity between point x_i and medoid m_j, and k is the number of clusters which has to be assessed beforehand. PAM is a better alternative than k-means when dealing with non-Euclidean distances or datasets with noise and outliers, as it is more robust to such irregularities.

'iris' dataset

iris: dataset of 150 observations x 5 variables.

Sepal.Length: continuous variable
Sepal.Width: continuous variable
Petal.Length: continuous variables
Petal.Width: continuous variable
Species.: discrete variables, setosa, virginica, versicolor

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

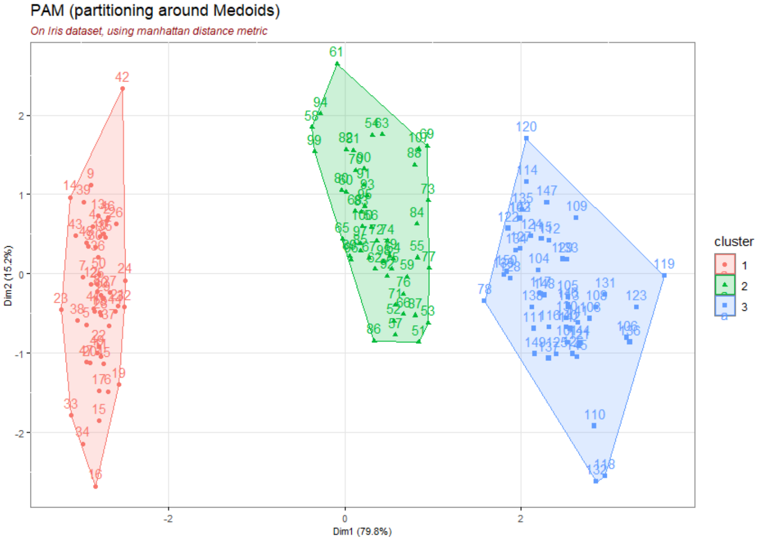
Summary

In order to perform a PAM, we call the function 'pam()' from the package 'cluster', containing three essential arguments: the dataset on which we want to perform clustering, the number of clusters and the distance metric.

```

> # perform PAM
> set.seed(2024)
> pam_result = pam(iris, k = 3, metric = "manhattan", stand = FALSE, medoids =
    NULL, nstart = 50)
>
> # results and interpretation
> pam_result$medoids # medoids
     Sepal.Length Sepal.Width Petal.Length Petal.Width Species
[1,]          5.0         3.4          1.5         0.2       1
[2,]          5.7         2.8          4.5         1.3       2
[3,]          6.5         3.0          5.5         1.8       3
> pam_result$clustering # clustering results
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
 [59] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
    2 2 2 2 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3
[117] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Visualizing PAM clusters



PAM (partitioning around Medoids)

On Iris dataset, using manhattan distance metric

Main observations

- The PAM algorithm successfully grouped the Iris dataset into three clusters, with each cluster roughly corresponding to one of the true species.
- Some overlap is observed between the clusters, particularly between the clusters for versicolor and virginica, indicating difficulty in perfectly separating these two species.
- The ellipses around the clusters show the spread and concentration of data points, with setosa (in red) forming a distinct and compact cluster, while the other two species have more spread-out distributions.
- The comparison of true species labels with PAM clusters reveals that the algorithm captures the overall structure of the dataset but may misclassify some points, especially where species characteristics are similar.

References

https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples/

The R Project for Statistical Computing:
https://www.r-project.org/