K-MEANS

Authors: G. H. Zerwes, *Universidade Federal do Espirito Santo, Vitória, Brazil*

Research Supervisor: L. Homri, *Arts et Métiers ParisTech, Metz, France*

1 Introduction

The K-means algorithm is a technique for unsupervised machine learning that aims to solve clustering problems, that is, identify groups and assign each data point to one of the groups. The basic working principle of this algorithm consists of randomly placing k cluster centers in the dataset. After that, each data point is assigned to its closest cluster center. The position of the cluster center is then updated to be in the center of all of its points. This procedure continues to iterate until the centers move only a small amount per iteration.

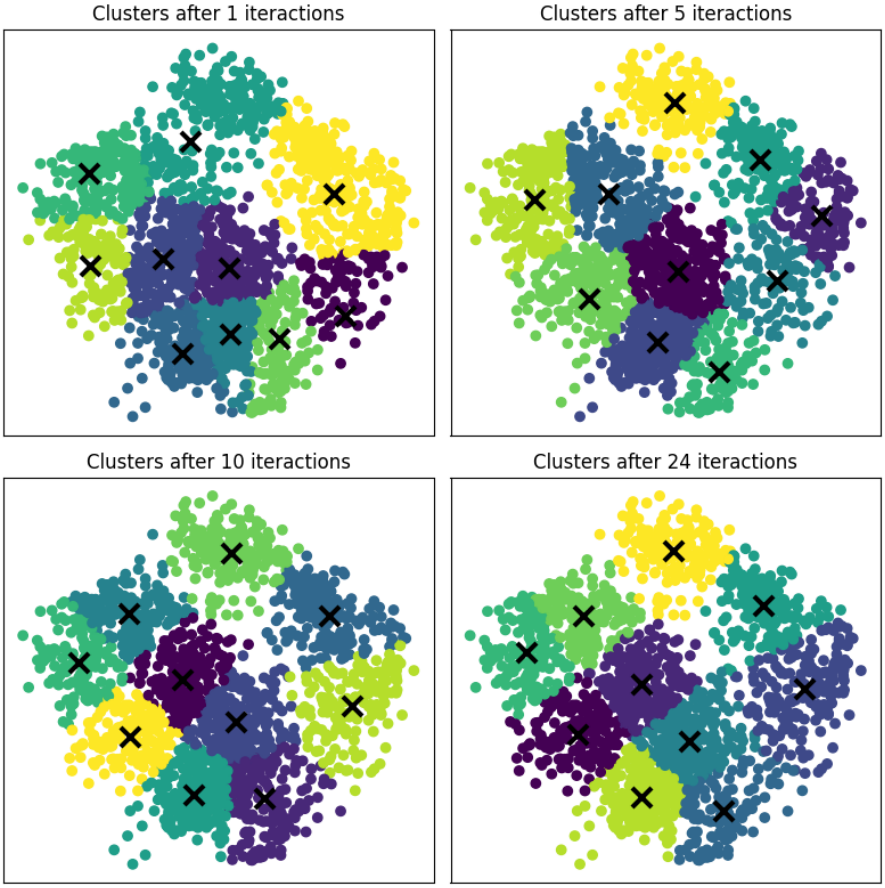

Figure 1: Postion of cluster centers during the execution

An example of this algorithm can be seen in figure 1. The first figure shows the initial position of the cluster centers, with white crosses, and the data points that are assigned to them. After five iterations it can be seen that the cluster center position is re-evaluated, which gives the second plot. The same re-assessment continues to go on until the convergence is achieved after 24 iterations.

2 Pseudo-code

To use this algorithm on your data, the pseudo-code below shows a procedure that should be followed.

```
#Step 1: Gather the data. The X data is the independent variable and should be already
    pre-treated.

X = [x_value_1, x_value_2, ...]
```

```
#Step 2: Fit the model.
model = KMeans.fit(X)

#Step 3: Make new predictions.
y_prediction = model.predict(X)
```

3 Other recommendations

There are some problems with this algorithm that arise from some assumptions taken to build it.

At first, it is important to note that this algorithm is sensitive to outliers. That is because it aims to minimize the distance to the closest center squared, so outliers will pull this value up.

A similar problem arises when the input data has a high number of features to be considered. Because the distance metric is not normalized when the data has many features, the values of the distance metrics tend to be overestimated, which gives bad results. In this case, it is recommended to perform a dimensionality reduction technique beforehand.

Another point to be considered is that this algorithm makes the assumption that clusters are regularly shaped, which may not be true in some cases. If this is the case for a said dataset, it is recommended that another algorithm is used.

The choice of the number of clusters can be a difficult one, and an improper choice leads to unwanted results. One alternative for this is to perform what is known as silhouette analysis, which allows us to graphically observe if the clusters have roughly the same size, or not.

At last, it is noted that this algorithm will always arrive at a solution, given enough iterations, however, this may not be the optimal minima. To avoid this problem, it is recommended to perform a greater number of initializations in random positions and choose the one that yields the best results.

References

Bishop, C. M. (2006). *Pattern Recognition And Machine Learning*. Number 758. Springer.