k-Means Clustering

k-means clustering is a popular unsupervised learning algorithm utilised to group unlabelled data into k clusters of similar samples. The algorithm begins with the initialisation of k centroids that we eventually hope to situate at the geometric centre of each cluster. Next, the euclidean distance between each sample and the centroids is measured, with samples then being assigned to the nearest centroid. Following this, each centroid is relocated to the mean position of the points that are assigned to it. These steps are repeated iteratively until the centroids converge on the centres of their respective clusters.

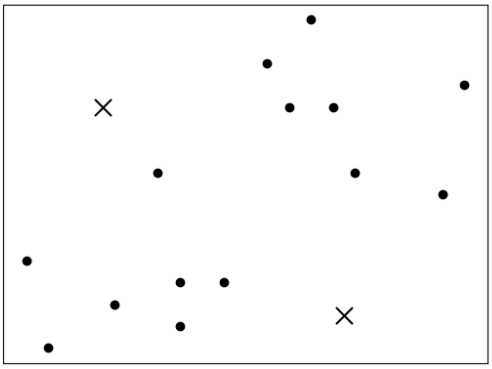

Figure 1: Here we have a very small two-dimensional dataset. The first step in the algorithm is to randomly initialise the points known as the cluster centroids (black crosses). We choose to initialise two centroids and proceed to measure the euclidean distance between each data sample and the centroids.

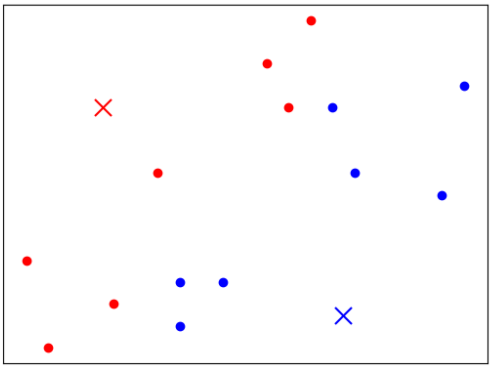

Figure 2: Once the distances between the samples and the cluster centroids have been measured, each sample is assigned to the nearest centroid.

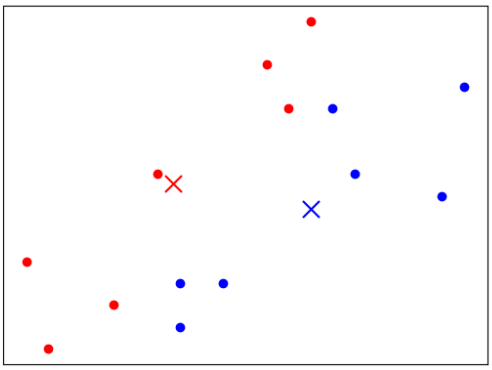

Figure 3: The next step is to move each centroid to the mean position of the samples that have been assigned to it.

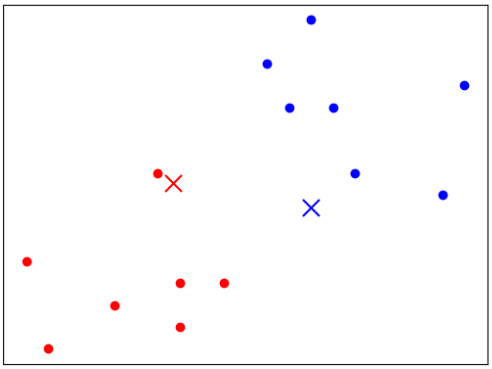

Figure 4: The distance between each sample and the centroids is measured again, with each sample being reassigned to the nearest centroid.

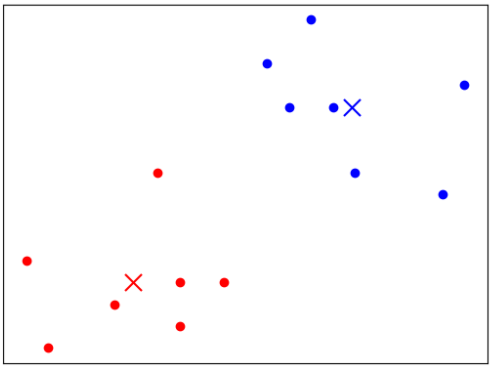

Figure 5: The centroids are moved to the mean positions of their newly assigned samples. Further iterations will not alter the assignments of the samples nor the positions of the centroids, so the algorithm stops here and the clustering has been completed.

The k-Means Algorithm

1. **Input**: Number of clusters, k.
2. **Initialise**: Randomly choose initial centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k$ (position vectors).
3. **Repeat until convergence**:
   (a) Set c^i as the index of the cluster centroid closest to the sample x^i. For example, if the sample x^9 is closest to the centroid $\boldsymbol{\mu}_3$, then $c^9 = 3$.
   Mathematically, we set $c^i := \arg\min_j ||\boldsymbol{x}^i - \boldsymbol{\mu}_j||^2$, where $\boldsymbol{x^i}$ is the position vector of the sample x^i. $\arg\min_j ||\boldsymbol{x}^i - \boldsymbol{\mu}_j||$ returns the j that minimises $||\boldsymbol{x}^i - \boldsymbol{\mu}_j||^2$.
   (b) Relocate $\boldsymbol{\mu}_j$ to the mean position of the samples assigned to cluster j.
   Mathematically, For $j = 1, 2, ..., k$:

$$\text{update } \boldsymbol{\mu}_j := \frac{1}{|c^i|} \sum_{x^i \in c^i} \boldsymbol{x}^i$$

   We take the sum of the position vectors $\boldsymbol{x}^i$ of the samples x^i that have been assigned to the cluster with index c^i, and then divide by the number of samples assigned to this cluster, $|c^i|$.

For m samples, the k-means optimisation objective is

$$\min_{\substack{c^1, c^2, ..., c^m \\ \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k}} J(c^1, c^2, ..., c^m; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k) \tag{1}$$

where

$$J(c^1, c^2, ..., c^m; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k) = \frac{1}{m} \sum_{i=1}^{m} ||\boldsymbol{x}^i - \boldsymbol{\mu}_{c^i}||^2 \tag{2}$$

When we minimise the cost function we are ensuring that the sum of the distances between $\boldsymbol{\mu_{c^i}}$ and all of the samples assigned to the cluster with index c^i is minimised. For example, $\boldsymbol{\mu}_2$ is placed at the location where the sum of the distances between $\boldsymbol{\mu}_2$ and all the samples assigned to the cluster with index $c^i = 2$ are minimised. The aim of the k-means algorithm is to find the optimal $\boldsymbol{\mu_j}$ and c^i to minimise the cost function J. Step 3(a) minimises J with respect to $(c^1, c^2, ..., c^m)$, whilst step 3(b) minimises J with respect to $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k)$.

With random utilisation of $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k$, the centroids can converge on different solutions depending on where they were initially placed, i.e., they can converge on different local optima. To overcome this issue, the k-means algorithm should be run a large number of times and the initialisation that produced the smallest cost function chosen.

References

Ng, A. *Machine Learning*. Offered by Stanford University on Coursera.

Shalev-Shwartz, S., Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, USA.