TDT4173: Task 1

Jim Totland

September 2021

K-means clustering

The K-means algorithm works by clustering the input data in K clusters by minimizing

$$\sum_{i=1}^{n} \|x^{(i)} - \mu_{c^{(i)}}\|, \tag{1}$$

where $x^{(i)}$ is the i'th data point and $\mu_{c^{(i)}}$ is the cluster centroid. K-means clustering is an unsupervised learning method, meaning that it does not use a response variable associated with the input (features). It is thus suited for problems where there is no appropriate response variable, but where we want to investigate whether there are groups (clusters) in the data.

The algorithm assumes that the feature space can be divided into K clusters and that the clusters can be found by minimizing (1). This implicitly assumes that the clusters are isotropic, or uniform in all direction[1], which is part of the method's inductive bias.

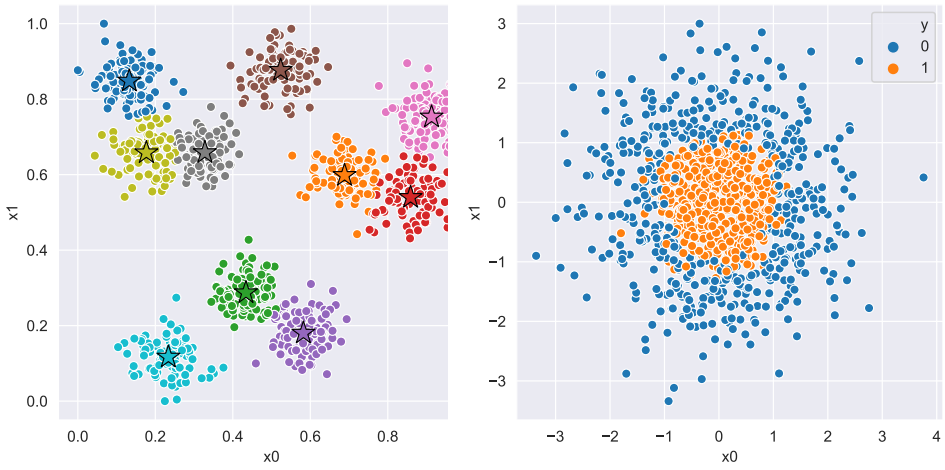
In the second dataset one can see, visually, that the scales of the second and first features differ by a factor of $\sim$10 (in order for there to be approximately uniform clusters). Because of the isotropic assumption, the K-means algorithm struggles to find good clusters (with respect to distortion and silhouette score). I remedied this by scaling the features with a min-max transformation, so that they are between 0 and 1. To produce a consistent clustering, I also introduced the variable `n_runs`, which signals how many times the algorithm should be run (with random initialization of clusters). The clustering which gives the highest silhouette score is chosen as the final clustering. With `n_runs = 50`, the clustering shown in figure 1a is consistently produced.

Logistic regression

Let

$$h_\theta(x) = \frac{1}{1 - e^{\theta^T x}}.$$

[1] https://antoinebrl.github.io/blog/kmeans/



(a) (b)

Figure 1: (a) The clustering produced on the second dataset. (b) The second dataset for logistic regression.

Logistic regression assumes the following relationship between the input, x, and the response, y:

$$\begin{aligned} P(y = 1|x; \theta) &= h_\theta(x), \\ P(y = 0|x; \theta) &= 1 - h_\theta(x), \end{aligned} \tag{2}$$

where θ is referred to as the weights. Logistic regression is therefore suited to problems where the response is binary.

Logistic regression assumes that the input data can be separated with a hyperplane (because the hypothesis function is a function of the linear combination of features and bias). This constitutes its inductive bias. For the second dataset, shown in figure 1b, there is no hyperplane separating the data (in that coordinate system). Thus, performing logistic regression on x_0 and x_1 yields subpar results.

This can be remedied by observing that the data seems to be linearly separable along the the distance from the origin, i.e. $r = \sqrt{x_0^2 + x_1^2}$. Training the logistic regression model on simply r, or r^2, yields much better results (90.6% and 90.2% accuracy on training and test data, respectively, with 1000 epochs).