Seminar Week 4: Clustering

Please note there is **more** here than you may possibly handle in one hour so please do not worry. This is for you to work with in your own time.

Q&A session

I will be answering any question you may have on the lectures / seminars so far. Please do not hesitate to ask me to clarify anything you are unsure about.

Clustering with the k-means algorithm

As you will remember from the lecture, the k-means algorithm is a clustering technique. Given unlabelled data, it tries to find meaningful groupings (clusters) of the data. In this exercise, you will generate data (in 1D as well as in 2D) and experiment with the algorithm using various values of k (remember that in the absence of information, we do not know what the best value of k is). **Matlab users**: use function `kmeans`; **Python users**: use function `KMeans` in `sklearn.cluster`.

1. Start by generating 100 random values from a unimodal distribution (i.e., a distribution with just one peak – in other words, we will assume all data are from the same class). You learned how to do this last week! Apply k-means with k taking values from 5 to 1. Plot the final clustering to see what has happened. Which value of k is best? Does that make sense? **Matlab users**: You will find this page useful, particular the section entitled "Partition Data into Two Clusters". **Python users**: See this excellent tutorial.

2. Generate 200 random values from a bimodal distribution (i.e., a distribution with two peaks – in other words, we now assume the data come from two classes that are not exactly overlapping). The simplest way to do this is simply to generate N (with N between 1 and 200) data points from a first unimodal distribution and then generate (200-N) data points from another unimodal distribution (with different mean and standard deviations from the first – how different is for you to decide). Again, apply k-means with k taking values from 5 to 1. Plot the final clustering to see what happened. Which value of k is best? Does that make sense?

3. Finally, do the same with 1000 two-dimensional random data points from a distribution of your choice (e.g., bivariate Gaussian distribution). Here, you could generate many more classes (or consider that a single class is made up of multiple blobs). Again, apply k-means for various values of k. Plot the final clustering to see what happened. See whether the results make sense given how you generated the data.

4. **In your own time and if you are interested,** consider coding your own version of the algorithm, ensuring it generalises to data of any dimension.