School of Electronic Engineering and Computer Science
Queen Mary University of London

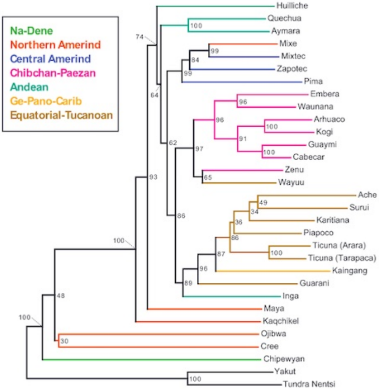
Machine Learning
Unsupervised learning: Structure analysis

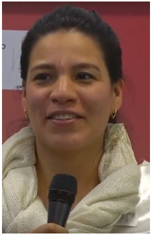
Dr Chao Liu

Credit to Dr Jesus Requena

Nov 2023

English spoken; American understood





"[Languages] reflect how humans perceive and organise the world around them"

Know your population!

Agenda

Unsupervised learning

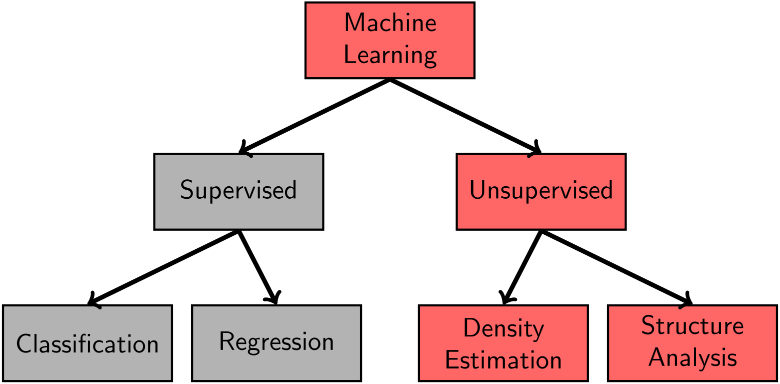
K-means clustering

Density-based clustering

Hierarchical clustering

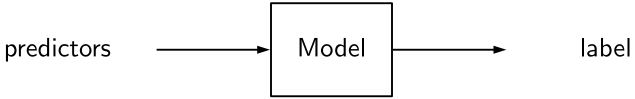
Summary

Machine Learning taxonomy



Supervised learning

In supervised learning, we **designate** one attribute as a **label** and treat the rest as **predictors**. We set out to build a model that estimates the value of the label based on the value of the predictors.



predictors → Model → label

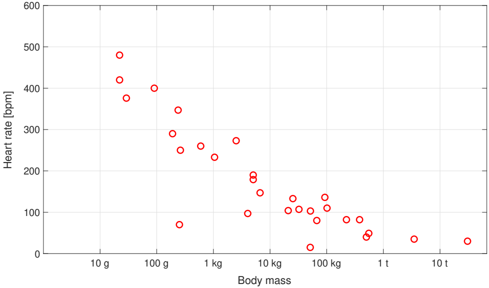
Note that:

- The model mirrors the underlying **structure** of the dataset.
- The notion of quality is defined as a function of the **discrepancy between the true and estimated** values of the label.

The space is empty

Unsupervised learning does not elevate any attribute to the category of label: all the attributes are treated equally. The essence of unsupervised learning is encapsulated in the simple question **where is my data**?

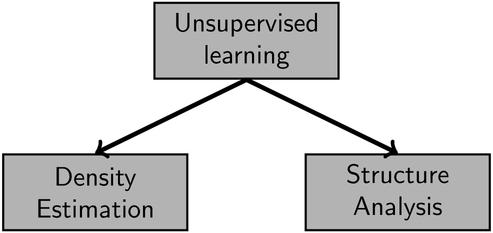
The attribute space is infinite and mostly empty, and the answer to this question will be a **model** that will allow us to identify the regions where we could expect to find samples.



Unsupervised learning

There are two main approaches to answer the question *where is my data*:

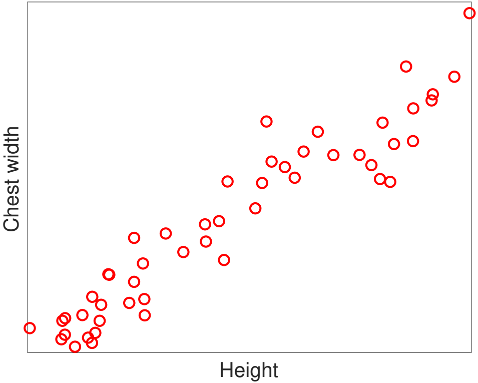
- **Density estimation**: Creates models that allow us to quantify the probability of finding a sample within a region of the attribute space (**probability density**).
- **Structure analysis**: Creates models that identify regions within the attribute space (**cluster analysis**) or directions (**component analysis**) with a high density of samples.



Unsupervised learning: Summarising data

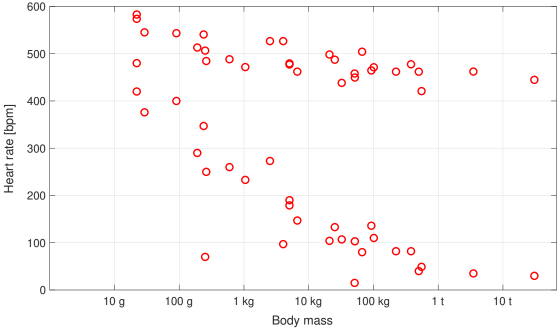
Unsupervised learning can be used to provide summaries of a population in the form of **prototypes** samples.

Look at the dataset below. If you had to produce 3 different t-shirt sizes, which sizes would you choose?

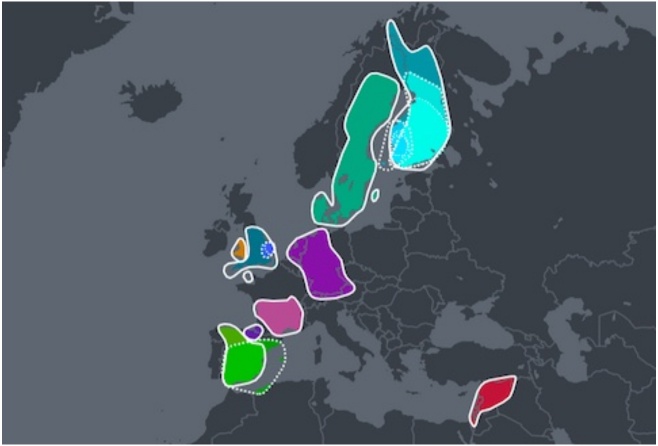


Unsupervised learning: Discovery

Unsupervised learning can also be used to **discover structure**. This can provide advantages (e.g. market segmentation), generate new knowledge (e.g. genetics-based migration studies) or be used to change the way we represent our data (e.g. compression).



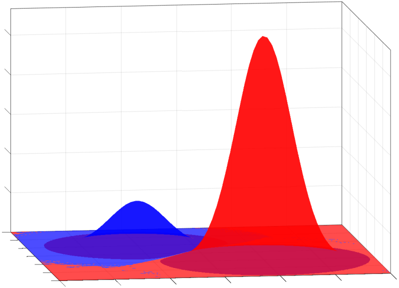
Example: DNA analysis



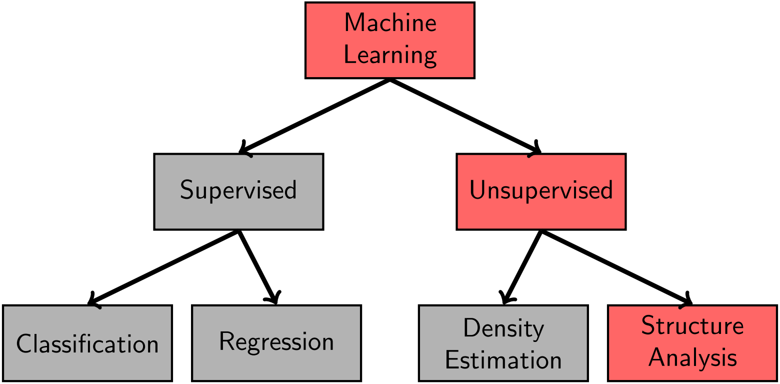
Unsupervised learning: Quantitative applications

Unsupervised learning can be used to build **class densities** that describe the probability of finding a sample from a given class in a region.

A probability density is also useful to identify **anomalies**, i.e. samples that are likely to belong to a different population.



Machine Learning taxonomy



Agenda

Unsupervised learning

K-means clustering

Density-based clustering

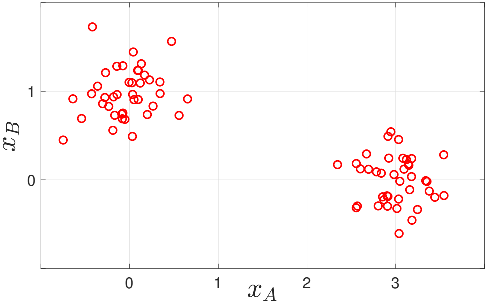
Hierarchical clustering

Summary

Cluster analysis

Clustering is a family of unsupervised learning algorithms that describe the structure of a dataset as groups, or clusters, of **similar samples**.

A notion of **similarity** is therefore needed in order for us to partition a dataset into clusters.

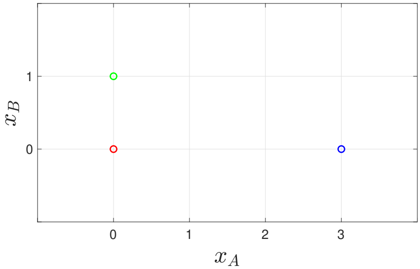


Similarity as proximity

Clusters can be defined as groups of samples that are **close** to one another. In this case, we use proximity as our notion of similarity.

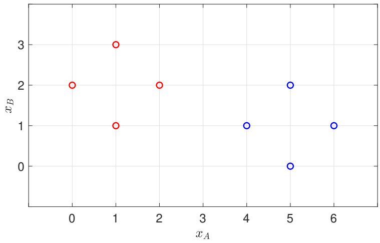
Mathematically, there are different ways of defining a distance. Given two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ consisting of P attributes, $x_{i,1}$, ..., $x_{1,P}$ and $x_{j,1}$, ..., $x_{j,P}$, the **squared distance** $d_{i,j}$ is defined as

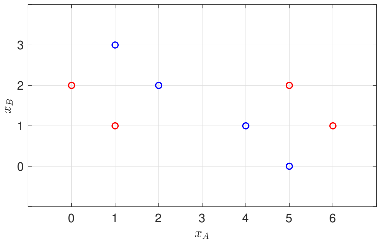
$$d_{i,j} = (x_{i,1} - x_{j,1})^2 + \cdots + (x_{i,P} - x_{j,P})^2$$



A proximity-based quality metric

Using distance as our notion of similarity, samples within the same cluster should be close to one another and samples from different clusters should be far apart.





Our next step will be to create a **quality metric** for a clustering arrangement based on the notion of distance between samples.

A proximity-based quality metric

Assume we have two clusters C_0 and C_1. The **intra-cluster sample scatter** $I(C_0)$ and $I(C_1)$ is the sum of the square distances between samples in the same cluster:

$$I(C_0) = \frac{1}{2} \sum_{x_i, x_j \text{ in } C_0} d_{i,j}, \qquad I(C_1) = \frac{1}{2} \sum_{x_i, x_j \text{ in } C_1} d_{i,j}$$

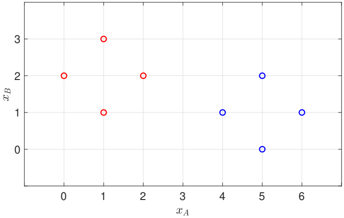
and the **inter-cluster sample scatter** $O(C_0, C_1)$ is defined as the sum of the distances between samples in different clusters:

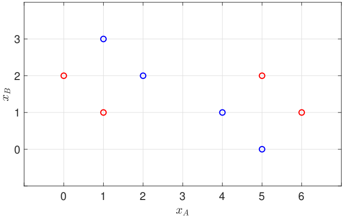
$$O(C_0, C_1) = \sum_{x_i \text{ in } C_0, x_j \text{ in } C_1} d_{i,j}$$

The **best** clustering arrangement has the **lowest intra-cluster** sample scatter and **highest inter-cluster** sample scatter. We can show that reducing the intra-cluster scatter increases the inter-cluster scatter!

A proximity-based quality metric

The intra-cluster and inter-cluster sample scatters allow us to compare and rank different clustering arrangements.

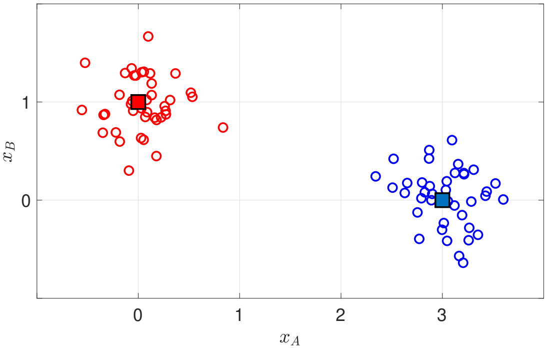




The question is, can we create an algorithm capable of automatically identifying the **best clustering** arrangement? This is an **optimisation** question.

K-means clustering: Prototypes

A simple way to describe a cluster is by using **cluster prototypes**, such as the centre of a cluster.



K-means clustering: Intra-cluster sample scatter

Given a cluster C_0 consisting of N_0 samples, its centre (or mean) $\boldsymbol{\mu}_0$ can be calculated as:

$$\boldsymbol{\mu}_0 = \frac{1}{N_0} \sum_{\boldsymbol{x}_i \text{ in } C_0} x_i$$

Interestingly, the intra-cluster sample scatter can be calculated using the **distance between each sample and the cluster prototype** d_i:

$$I(C_0) = N_0 \sum_{\boldsymbol{x}_i \text{ in } C_0} d_i$$

Therefore, our notion of clustering quality can be expressed as follows: in a good clustering arrangement, samples are **close to their prototype**.

K-means clustering

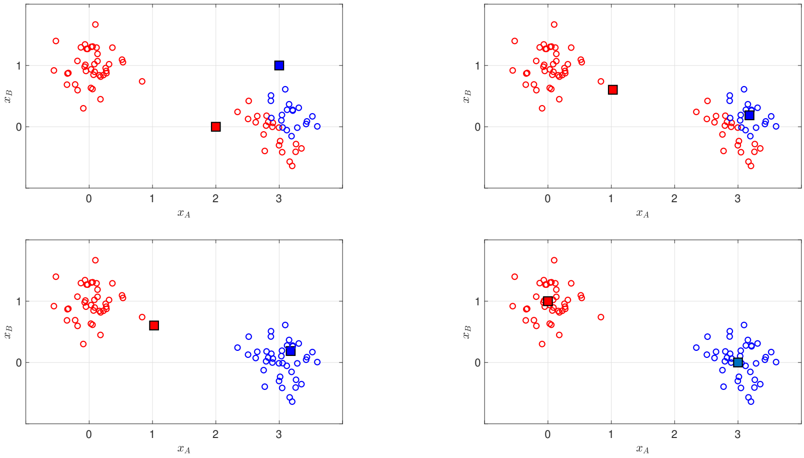
K-means partitions a dataset into K clusters represented by their **mean** and proceeds iteratively as follows:

- Prototypes are obtained as the centre (or mean) of each cluster.
- Samples are re-assigned to the cluster with the closest prototype.

As the K-means algorithm proceeds, we will see samples been reassigned to different clusters until at some point we reach a stable solution, where no sample is reassigned.

The final solution is a **local optimum**, not necessarily the global one.

K-means clustering



How many clusters?

K-means requires that we specify the number of clusters K that we want to partition our dataset into.

A natural question is, what's the right number of clusters? The answer to this question depends on the application:

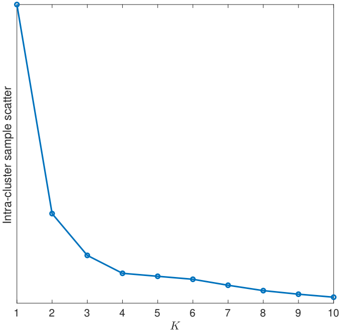
- In some cases we are given the number of clusters (e.g. t-shirt sizes).
- In a discovery scenario we want to find the underlying structure and the number of clusters is unknown.

Validation strategies can suggest a suitable value for the hyperparameter K. Choosing the value of K producing the lowest $I(C_0)$ would not work however, as $I(C_0)$ always decreases as the number of clusters increase.

The elbow method

Assume the true number of clusters is K_T. For $K > K_T$, we should expect the increase in quality to be slower than for $K < K_T$, as we will be splitting true clusters.

The true number of clusters can be identified by observing the value of K beyond which the improvement slows down.


Intra-cluster sample scatter
K