Machine Learning

COMP2002

Lauren Ansell

Introduction

Today's topics:

- Introduction to machine learning
- Classification methods
- Clustering methods

Session learning outcomes - by the end of today's lecture you will be able to:

- Distinguish between different types of machine learning task.
- Describe different approaches to classification and clustering.

Machine Learning

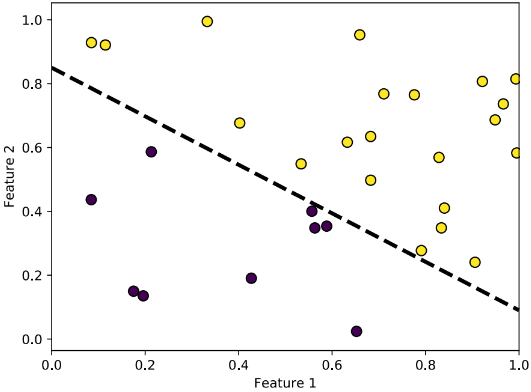
Two varieties of machine learning

- Supervised learning – training models (known target values).
- Unsupervised learning – finding structure in data (no target values).

Machine learning tasks:

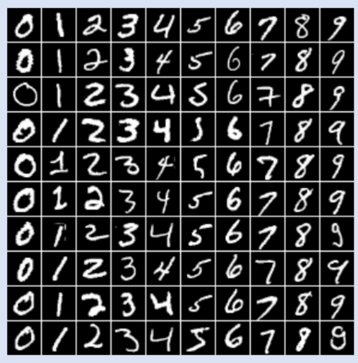
1. Classification - work out which of a set of categories an object belongs to.
2. Clustering - identify groupings occurring within data.
3. Regression - predict a single numerical value.

Classification



Data comprising two linearly separable classes – all members of C1 lie above the hyperplane and all members of C2 lie below it.

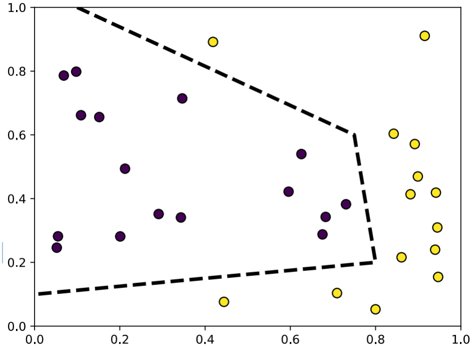
Classification



10 classes – the digits 0-9.
Each observation is a 28x28- pixel image of a digit.
784 features – one for each pixel in the range 0-255 (greyscale).

Challenges With Classification

- Sparse data – e.g., new user problem
- Overlapping classes
- Uneven classes



Data is not often linearly separable – here the hyperplane is **nonlinear**.

Iris Data

Cluster according to

- Sepal width and height
- Petal width and height

The dataset

- Highly cited in the machine learning literature – well-understood
- Labelled data – the ground truth is known
- 150 observations described by four features
- Three classes (50 observations each) – one is linearly separable and the other two are not

The "Curse Of Dimensionality"

Data is often high dimensional – this causes problems for modelling (as well as in other areas of machine learning such as visualizing the data).

Sepal		Petal		Class
length	width	length	width	
5.1	3.5	1.4	0.2	Iris-setosa
5.2	2.7	3.9	1.4	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica

Dimension Reduction

Reduce the dimensionality of the data by discarding redundant data.

- Feature selection – identify redundant features and discard them.
- Feature extraction – find a new set of low-dimensional points that represents the original data well.
- Preserve some characteristics of the original data.

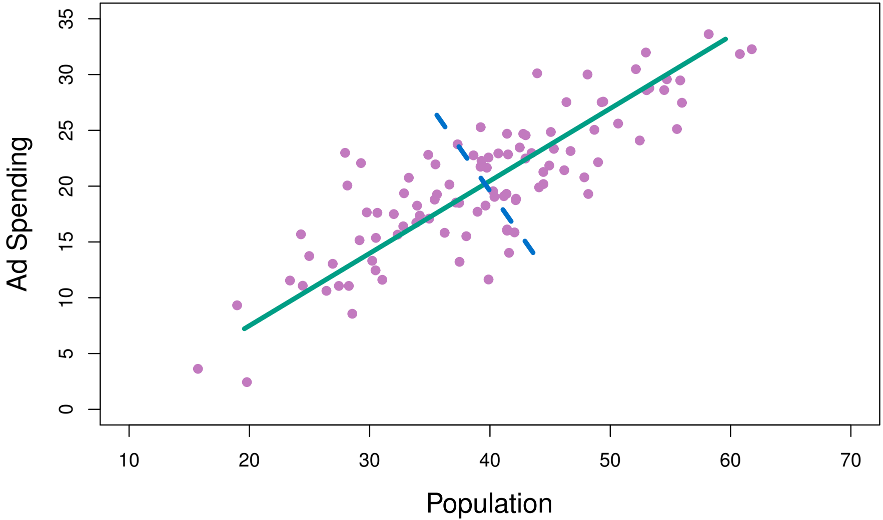
Principal Component Analysis

PCA is a technique to reduce the dimension of an $n \times p$ data matrix X.

The first principal component is in the direction of the data along which there is the greatest amount of variation.

The second principal component is a linear combination of the variables that is uncorrelated with the principal component subject to the constraint that it has the largest variance.

Example



Eigenvectors and Eigenvalues

It is eigenvectors and eigenvalues who are behind all the magic of principal components.

Eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

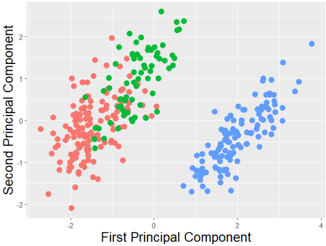
By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

Penguin Data – Principal Component Analysis Projection

We construct a low-dimensional representation of the data by projecting onto the principal components.

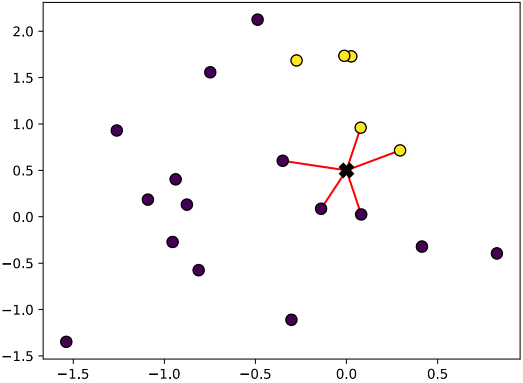
PCA in practice:

- Commonly used for data visualization – retain eigenvectors corresponding to the two or three biggest eigenvalues.
- Use your favourite data science package to compute it easily.



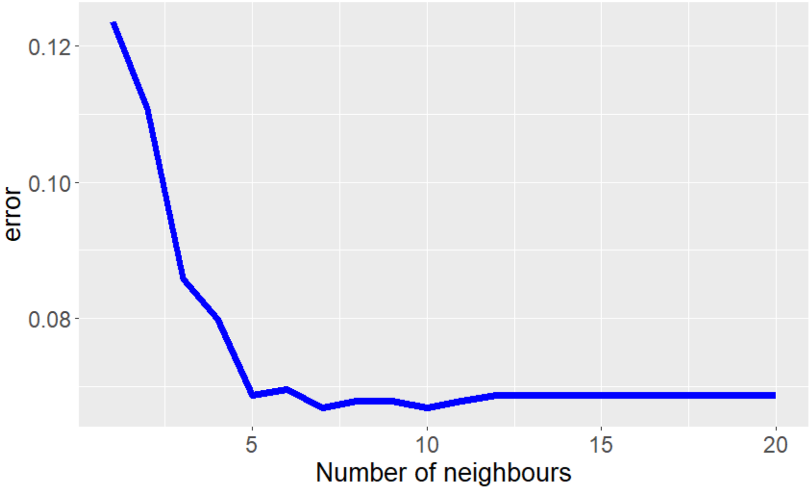
Classification – k-nearest neighbours (kNN)

- Given a set of points with known classes and a new point of unknown class.
- Find the *k* nearest known points to the unknown point.
- Assign the class held by the majority of the k points to be the class of the unknown point.
- The clusters do not overlap.



How Many Neighbours?

Training Error

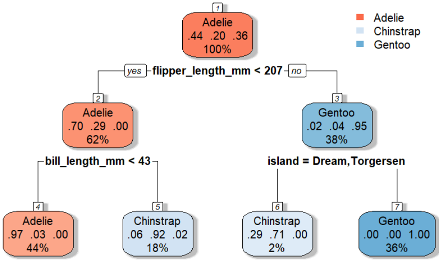


Decision Trees

Decision trees can be applied to both regression and classification problems.

Use classification trees to predict the class label of an observation.

We are interested in both the class prediction for an observation and the proportions of the data which fall into each class.



Creating Trees

To create a tree we continuously split the data.

We use the classification error rate to make the splits.

The classification error rate is the fraction of the training observations in a region that do not belong to the most common class within that region:

$$E = 1 - \max_k(\hat{p}_{mk})$$

How Good Is The Tree?

Gini index:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{1}$$

entropy:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \tag{2}$$

These two approaches are more sensitive to node purity.

We use any of the three methods when pruning the tree.

Pros And Cons of Decision Trees

- Trees are very easy to explain.
- Some people believe they mirror human decision making closely
- Trees can be displayed graphically.
- Trees can easily handle qualitative predictors without the need to create dummy variables.
- Trees do not have the same level of predictive accuracy as other methods.
- Trees can be very non-robust.

Random Forests

A set of decision trees working together – ensemble learning

Each tree is constructed from a random set of features – each tree is different

Unlike decision trees, each time a split is considered only a random sample of the predictors is considered.

We build a number of different decision trees.

At each split the algorithm is allowed to use only one of the *m* predictors.

$$m \approx \sqrt{p}. \tag{3}$$

This reduces the probability of a strong predictor being used in all of the trees, as on average only

$$\frac{p - m}{p} \tag{4}$$

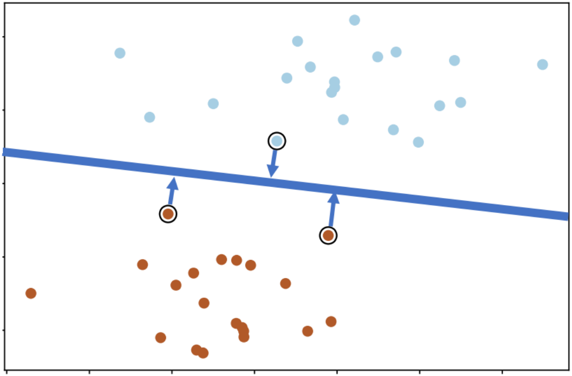
of the splits will not even consider the strong predictor.

This results in the average of the trees having less variability and therefore more reliable.

Support Vector Machines (SVM)

Generalisation of simple and intuitive classifer called *maximal margin classifer*.

A popular approach for classification due to its flexibility and good performance in comparison to other classifiers.



Fitting SVM

First, we decide what shape the class boundary should have.

We can choose from:

- linear - $K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$
- polynomial - $K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$
- radial - $K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$
- sigmoid - $K(x_i, x_{i'}) = \tanh(\alpha \sum_{j=1}^{p} (x_{ij} x_{i'j}) + \beta)$

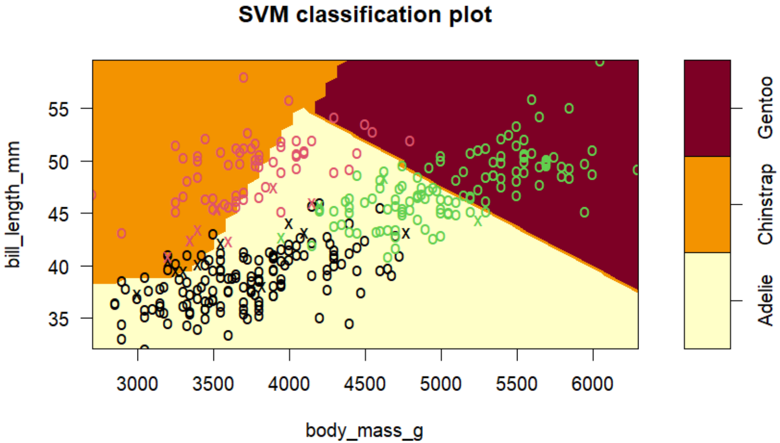
This is the so-called kernel of the SVM classifier.

Second, we decide how wide the so-called margin should be.

- The margin defines the area in the direct neighbourhood of class boundary.
- Only the data points inside the margin area will be taken into account when forming the class boundary.
- These data points are called support vectors.
- The width of the margin is specified indirectly via the so-called tuning parameter.

In the last step, the SVM algorithm determines the class boundary by finding the line or curve, depending on step 1, that separates the support vectors in an optimal way.

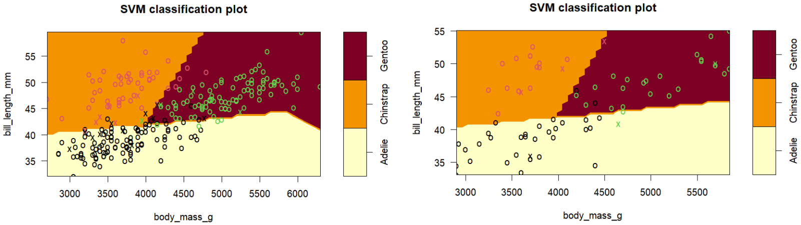
Penguin Data - SVM



No classes are correctly classified

More difficult for the two overlapping classes – try a **different kernel**

Penguin Data - SVM Training/Testing Results



(a) Training accuracy: 1.000

(b) Test accuracy: 1.000

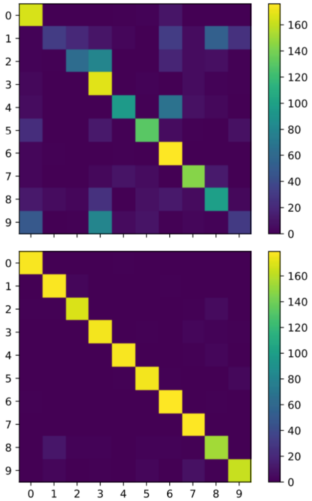
The model performs well on the training data.

Fortunately it also performs well on the testing data so we can say that the model generalises.

Is It Working? How Good Is The Classifier?

Confusion matrices

- Compare predictions against targets
- High value (yellow) in diagonal entries indicates high accuracy
- Two variants of kNN classifiers – k = 100 (top) and k = 50 (bottom)
- So the bottom example has higher accuracy than the top row



Other Metrics For Evaluating Classification

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Recall (Sensitivity)

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Precision

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

f_1 score

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

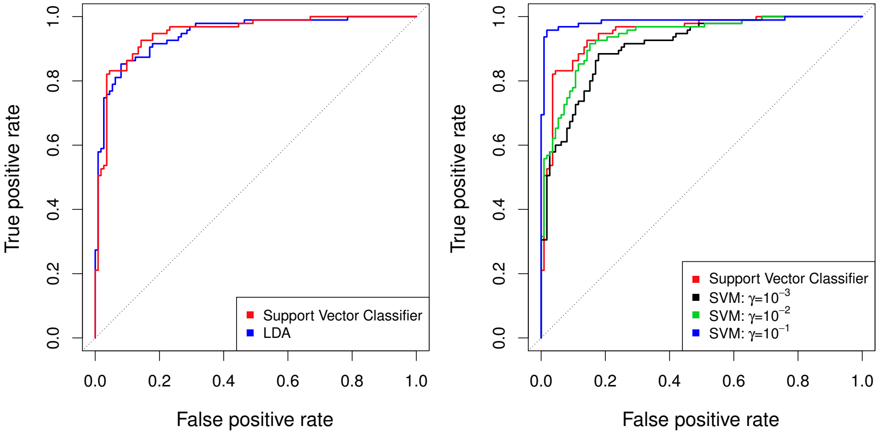
ROC Curve

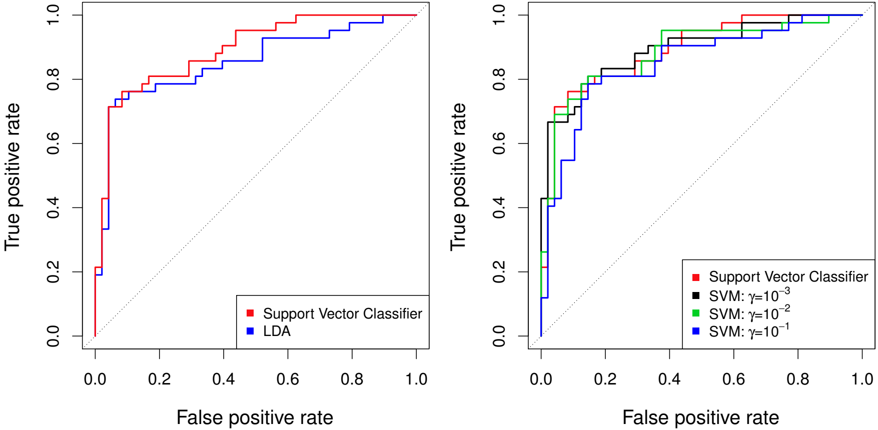
ROC stands for Receiver Operating Characteristics.

The ROC curve displays two types of errors simultaneously.

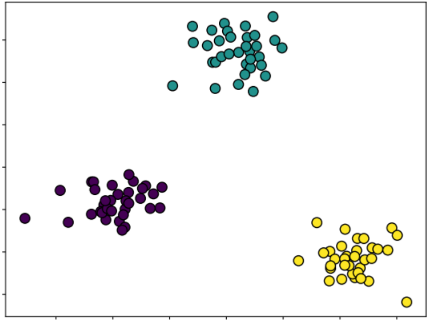
The overall performance of the classifier is given by the area under the curve.

We often plot the ROC for both the training and the test data.


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
🟥 Support Vector Classifier
🟦 LDA
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
🟥 Support Vector Classifier
⬛ SVM: $\gamma=10^{-3}$
🟩 SVM: $\gamma=10^{-2}$
🟦 SVM: $\gamma=10^{-1}$


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
🟥 Support Vector Classifier
🟦 LDA
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
🟥 Support Vector Classifier
■ SVM: $\gamma=10^{-3}$
🟩 SVM: $\gamma=10^{-2}$
🟦 SVM: $\gamma=10^{-1}$

Clustering



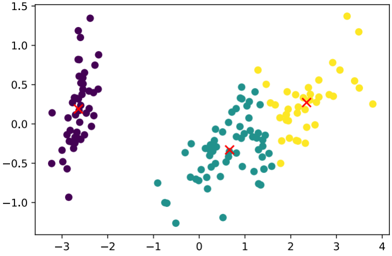
Various approaches – generally in terms of a distance.

An example of such a distance is Euclidean distance – given two vectors a and b

$$d(\vec{a},\vec{b}) = d(\vec{b},\vec{a}) = \sqrt{\sum_{d=1}^{D}(a_i - b_i)^2} \tag{9}$$

k-means Clustering

```
Require: $X$ – a set of $N$ points to be clustered
Require: $k$ – the number of clusters required
1 Select a set of $k$ cluster centres $c_j$ where $j = 1, \ldots, k$
2 while not converged do
3     Allocate each member of $X$ to their nearest $c_j$
4     Replace each $c_j$ with the new mean of cluster $j$
5 end while
```



How k-means Works

We let $C_1, ..., C_k$ denote sets containing the indices of the observations in each cluster.

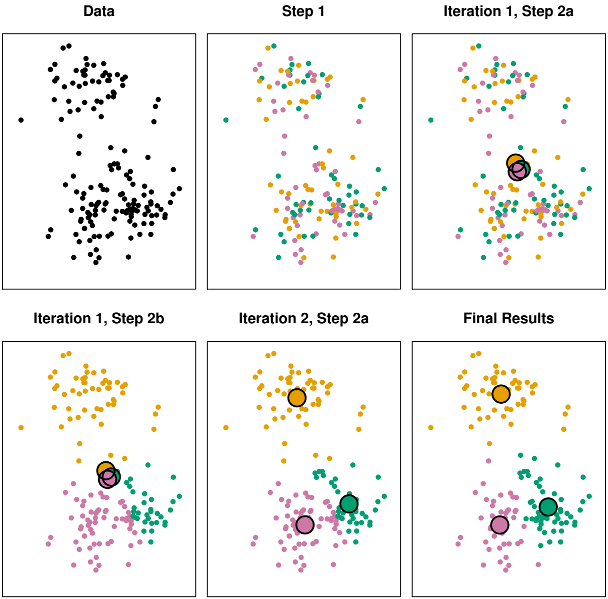
These need to satisfy the following two properties:

$$C_1 \cup C_2 \cup ... \cup C_k = \{1, ..., n\} \tag{10}$$

$$C_k \cap C_{k'} = \emptyset \text{ for all } k \neq k' \tag{11}$$

The aim is to minimise within-cluster variation

$$\underset{C_1,..,C_K}{\text{minimise}} \left\{ \sum_{k=1}^{K} W(C_k) \right\} \tag{12}$$



Data

Step 1

Iteration 1, Step 2a

Iteration 1, Step 2b

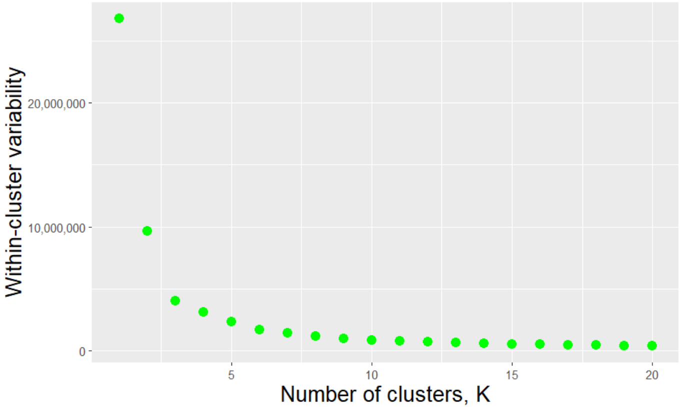
Iteration 2, Step 2a

Final Results

How Many Clusters?

Within-clusters Variability

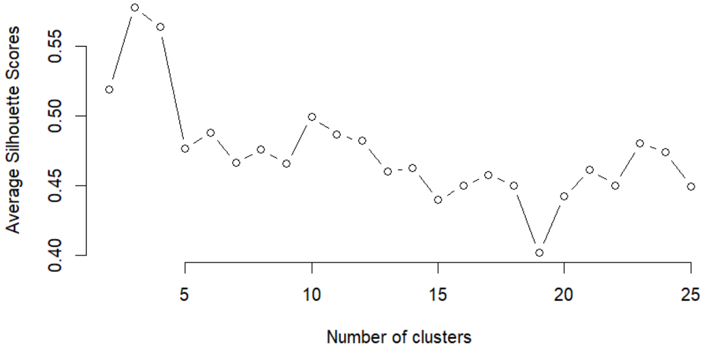
- Minimise the variation within the cluster.
- The more broad the clusters, the higher the within-cluster variation.



How Many Clusters?

Silhouette score

- Maximise the nearest-cluster distance and minimize the intra-cluster distance
- Score lies between 1 and -1
- If there are overlapping clusters a score close to 0 is likely



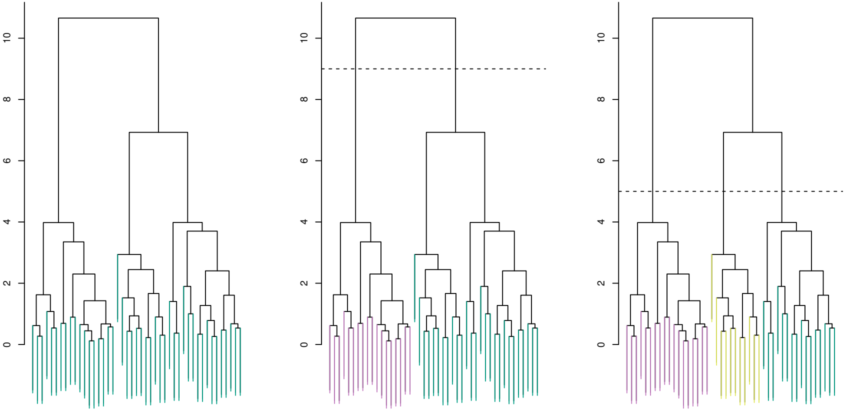
Hierarchical Clustering

Unlike K-means clustering, we do not specify the number of clusters at the start.

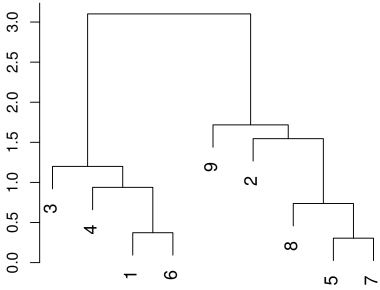
We can also produce tree-based representations of the observations called a dendrogram.

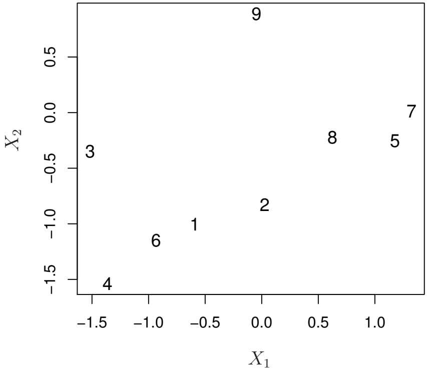
The most common type of hierarchical clustering is *bottom-up* or *agglomerative* clustering.

Denodrograms



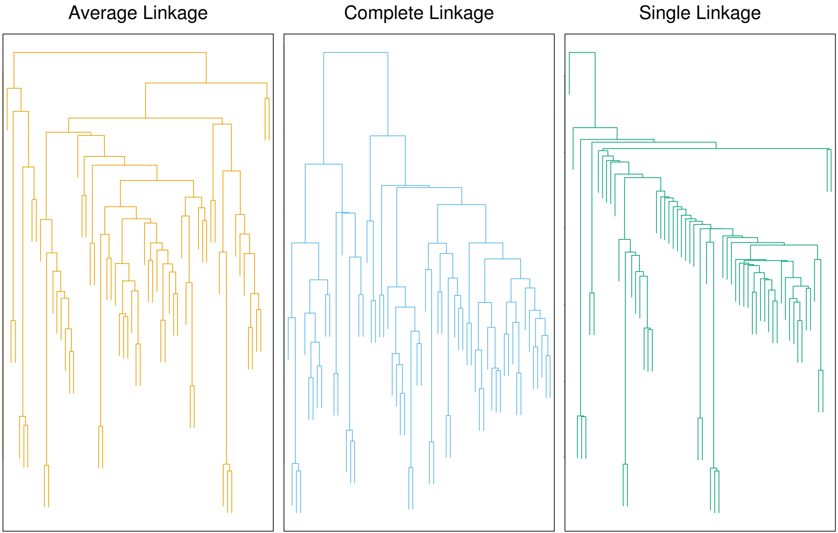
Interpreting Denodrograms





Methods Of Linkage

- Complete - Maximal intercluster dissimilarity
- Average - Mean intercluster dissimilarity
- Single - Minimal intercluster dissimilarity
- Centroid - dissimilarity between the centroid for cluster A and the centroid of cluster B



Average Linkage

Complete Linkage

Single Linkage

Summary

Machine Learning

- Learning from data
- Various types of learning task – classification, clustering and regression
- Classification – which class does an observation belong to?
- Clustering – identify naturally occurring groupings in the data
- Training/testing – confusion matrices (classification) and silhouette scores (clustering)

Lab Session

We will apply different Machine Learning algorithms to datasets.

Algorithms we will be applying:

- decision trees
- PCA
- KNN