K-最近鄰法
K-Nearest Neighbor (KNN)

2018.10.4

先學習「分類」再用來「預測」

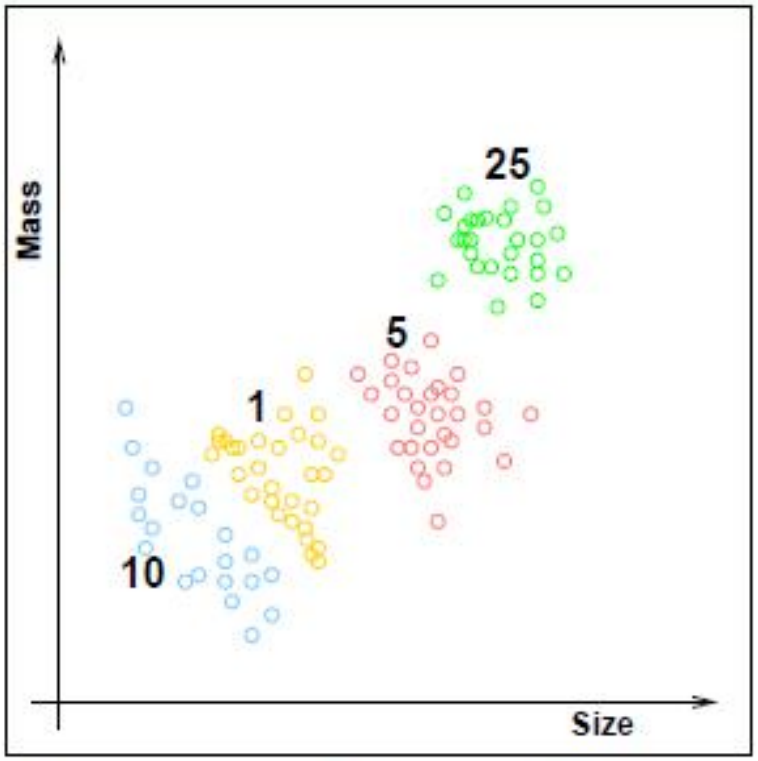
- 所謂的機器學習，是一種人工智慧的程式自行學習的機制
- 而構成學習的基礎元素是「分類」這樣的處理動作。只要能做好分類，既能理解事物，也能做出判斷、採取行動。
- 機器學習可以讓電腦一面處理大量的資料，一面自動學會這樣的分類方式。等到學會分類方式後就可以用來預測未知的資料。

機器學習案例 —「分類」

我是特徵					我是目標
菇種	菌傘形狀	菌傘顏色	分布地帶	氣味	有毒 /無毒
菇菇 A	球狀	棕色	腐木	腥味	0 (標籤)
菇菇 B	圓錐形	淺黃色	草堆	無氣味	1 (標籤)
菇菇 C	圓錐形	白色	樹葉	霉味	1 (標籤)
菇菇 D	鐘形	紫色	腐木	杏仁味	0 (標籤)
菇菇 E	下凹形	黃色	腐木	無氣味	1 (標籤)
菇菇 F	扁平狀	白色	泥土	惡臭味	0 (標籤)

當一筆新資料輸入電腦中，比如特徵具備白色鐘形菌傘、分布在腐木上、杏仁味的香菇，電腦即會判斷這朵香菇有毒或沒毒的機率有多高了。

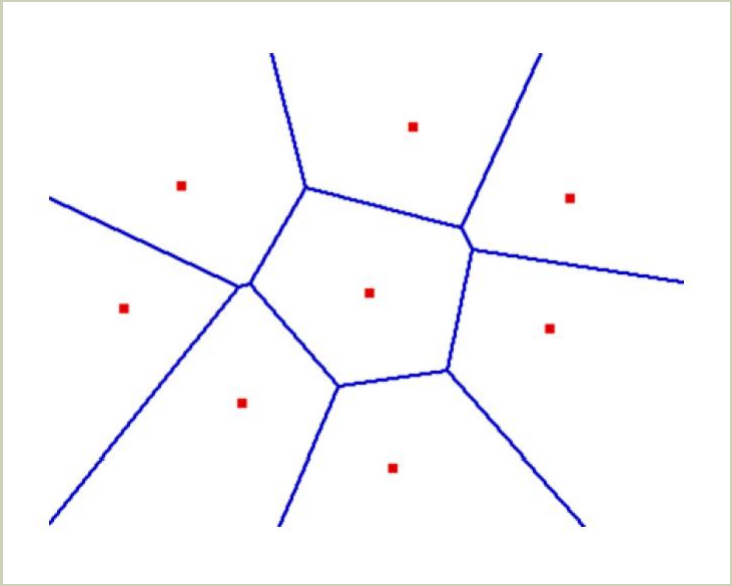
學習將硬幣分類



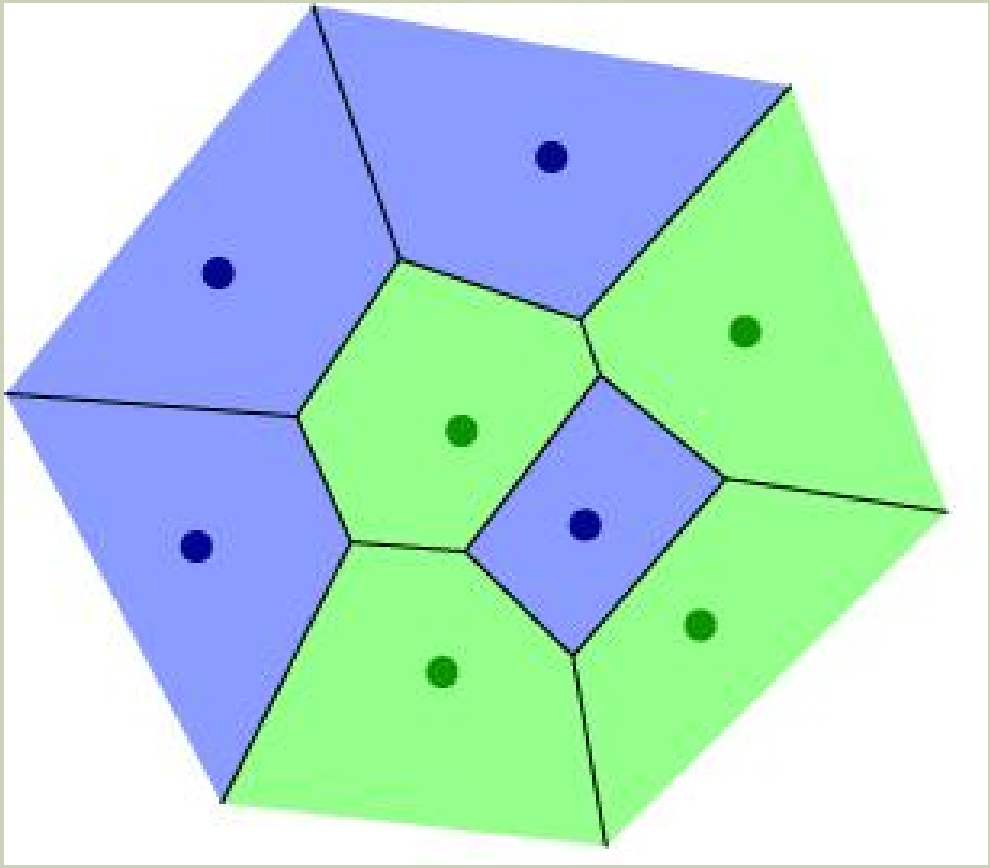
supervised multiclass classification

The Nearest Neighbor Algorithm

- A lazy learning algorithm
- 物以類聚

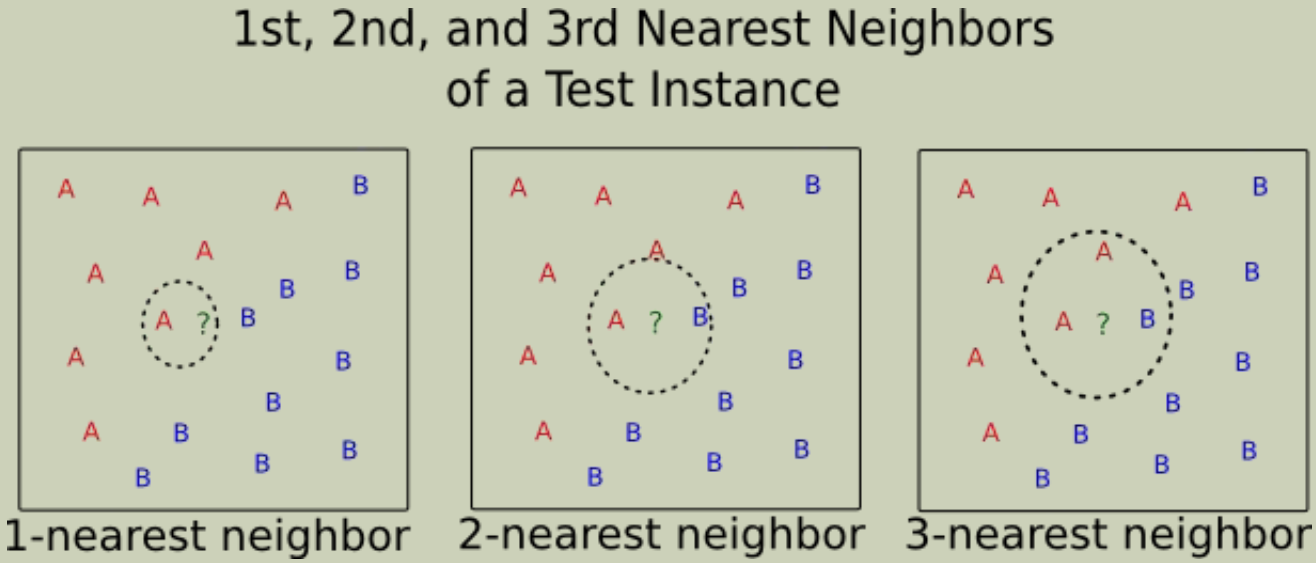


1-Nearest Neighbor Boundary



K-Nearest Neighbor

- 找出K個最近鄰的點，看它們是屬於哪一個類別，多數者獲勝



分類分的好不好? 怎麼評估

預測類別 / 實際類別	Class 1	Class 2
Class 1	TP (true positive)	FN (false negative)
Class 2	FP (false positive)	TN (true negative)

- Confusion matrix (混亂矩陣、混淆矩陣)
- 正確率 = (TP+TN) / (TP+TN+FP+FN)
- 錯誤率 = 1-正確率
- 敏感度 (sensitivity) = TP / (TP+FN)
- 準確度 (specificity) = TN/ (TN+FP)
- 精確率 (precision) = TP / (TP+FP)
- 回想率 (recall) = TP / (TP+FN)

KNN in R

- **library**(class)
- knn(train, test, cl, k = 1, l = 0, prob = FALSE, use.all = TRUE)

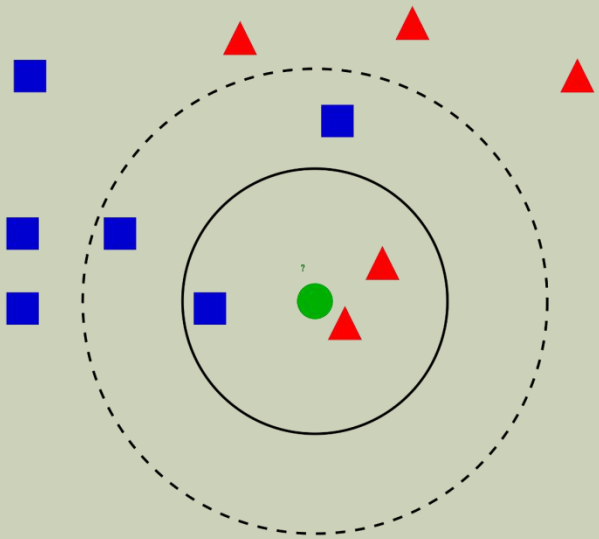
- x <- sample(c(1:150),120)
- train <- iris[x,1:4]
- test <- iris[-x,1:4]
- train.cl <- iris[x,5]
- test.cl <- iris[-x,5]
- iris.knn <- knn(train, test, train.cl, k = 1)

難道KNN就這麼簡單無聊嗎？

- 好K讓你上天堂，K要多少最合適，怎麼樣挑一個好K。
- 怎麼算出資料的***距離***，例如座標距離、顏色相近程度、字詞重疊程度...等，這個會跟你的資料還有分類方法有關
- 每次都要掃描全部的資料算出距離，才能知道最近的k筆資料是什麼，需要相對高的記憶體跟計算時間。

K值選取

- 如果K選大了的話，可能求出來的k最近鄰集合可能包含了太多隸屬於其它類別的樣本點，最極端的就是k取訓練集的大小，此時無論輸入實例是什麼，都只是簡單的預測它屬於在訓練實例中最多的類，忽略了訓練實例中大量有用資訊。
- 如果K選小了的話，結果對噪音樣本點很敏感。
- 那麼到底如何選取K值，其實完全靠經驗或者**交叉驗證**



歐氏距離(EUCLIDEAN DISTANCE)

- 歐氏距離是最易於理解的一種距離計算方法，源自歐氏空間中兩點間的距離公式。

- 二維平面空間距離：$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- N維空間距離：$d_{12} = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2}$

曼哈頓距離(MANHATTAN DISTANCE)

- 想像你在曼哈頓要從一個十字路口開車到另外一個十字路口，駕駛距離是兩點間的直線距離嗎？顯然不是，除非你能穿越大樓。實際駕駛距離就是這個「曼哈頓距離」。而這也是曼哈頓距離名稱的來源， 曼哈頓距離也稱為**城市街區距離(City Block distance)**。

- 二維平面空間距離：$d_{12} = |x_1 - x_2| + |y_1 - y_2|$

- **N**維空間距離：$d_{12} = \sum_{k=1}^{n} |x_{1k} - x_{2k}|$

切比雪夫距離（CHEBYSHEV DISTANCE）

- 二個點之間的距離定義為其各座標數值差的最大值。若將西洋棋棋盤放在二維直角座標系中，格子的邊長定義為1，座標的x軸及y軸和棋盤方格平行，原點恰落在某一格的中心點，則國王從一個位置走到其他位置需要的步數恰為二個位置的切比雪夫距離，因此切比雪夫距離也稱為棋盤距離

- 二維平面空間距離：$d_{12} = \max(|x_1 - x_2|, |y_1 - y_2|)$

- N維空間距離：$d_{12} = \max_i(|x_{1i} - x_{2i}|)$

閔可夫斯基距離(MINKOWSKI DISTANCE)

- 其中p是一個變參數。
- 當p=1時，就是曼哈頓距離
- 當p=2時，就是歐氏距離
- 當p→∞時，就是切比雪夫距離

- N維空間距離：$$d_{12} = \sqrt[p]{\sum_{k=1}^{n} |x_{1k} - x_{2k}|^p}$$

關於距離，好像有一個問題

- 假設有三間房子，猜猜誰跟誰的房價比較接近
- A房子：200坪，距離捷運站100公尺
- B房子：30坪，距離捷運站90公尺
- C房子：199坪，距離捷運站400公尺

標準化歐氏距離 (STANDARDIZED EUCLIDEAN DISTANCE)

- 標準化歐氏距離是針對簡單歐氏距離的缺點而作的一種改進方案。標準歐氏距離的思路：既然數據各維份量的分佈不一樣，先將各個份量都「標準化」到均值、變異數相等。假設樣本集X的均值(mean)為m，標準差(standard deviation)為s，那麼X的「標準化變量」表示為：

$$X^* = \frac{X - m}{s}$$

- 而且標準化變量的數學期望為0，變異數(標準差)為1
- N維空間距離：

$$d_{12} = \sqrt{\sum_{k=1}^{n} \left(\frac{x_{1k} - x_{2k}}{s_k} \right)^2}$$

馬氏距離(MAHALANOBIS DISTANCE)

- 與歐氏距離不同的是它考慮到各種特性之間的聯繫（例如：一條關於身高的信息會帶來一條關於體重的信息，因為兩者是有關聯的）
- S為共變異數矩陣
  - 二維平面空間距離： $D(X_i,X_j)=\sqrt{(X_i-X_j)^T S^{-1}(X_i-X_j)}$
  - N維空間距離： $D(X_i,X_j)=\sqrt{(X_i-X_j)^T (X_i-X_j)}$
- 如果共變異數矩陣為單位矩陣，馬氏距離就簡化為歐氏距離；如果共變異數矩陣為對角陣，其也可稱為正規化的歐氏距離。

夾角餘弦(COSINE)

- 幾何中夾角餘弦可用來衡量兩個向量方向的差異，機器學習中借用這一概念來衡量樣本向量之間的差異。
- 在二維空間中向量**A(x1,y1)**與向量**B(x2,y2)**的夾角餘弦公式：

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}}$$

- **N**維空間夾角餘弦：

$$\cos(\theta) = \frac{\sum_{k=1}^{n} x_{1k}\, x_{2k}}{\sqrt{\sum_{k=1}^{n} x_{1k}^2}\sqrt{\sum_{k=1}^{n} x_{2k}^2}}$$

關於距離，再想一個問題

- 如果資料當中有類別變數，例如：學歷、婚姻狀況，該怎麼辦？

簡單配對係數
SIMPLE MATCHING COEFFICIENT (SMC)

$$\mathrm{SMC} = \frac{\text{number of matching attributes}}{\text{number of attributes}}$$

$$= \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

where:

M_{11} is the total number of attributes where *A* and *B* both have a value of 1.

M_{01} is the total number of attributes where the attribute of *A* is 0 and the attribute of *B* is 1.

M_{10} is the total number of attributes where the attribute of *A* is 1 and the attribute of *B* is 0.

M_{00} is the total number of attributes where *A* and *B* both have a value of 0.

The **simple matching distance (SMD)**, which measures dissimilarity between sample sets, is given by $1 - \mathrm{SMC}$.

傑卡德相似係數
(JACCARD SIMILARITY COEFFICIENT)

- 兩個集合A和B的交集元素在A，B的並集中所佔的比例，稱為兩個集合的傑卡德相似係數，用符號J(A,B)表示。

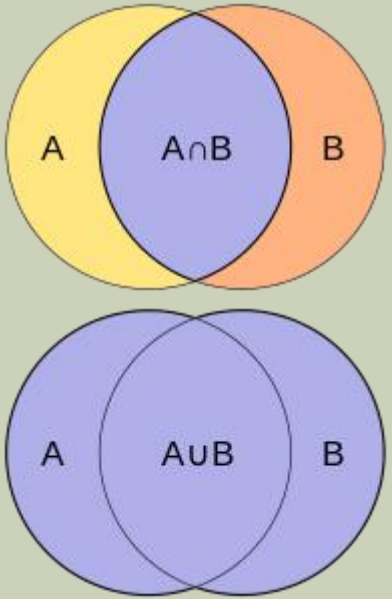
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

- 傑卡德距離

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

- 與SMC很相似

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}.$$



KNN的進階版本

- 為了解決資料量大時的計算複雜度，而產生另外3種演算法
- **K-D tree**
- **Ball tree**
- **Cover tree**
- **Package ‘FNN’**