

Classification with	→ K/NN	1
	→ trees	2
	→ boosting (AdaBoost)	4
• support vector machines (svm)		3

K-nearest-neighbours

• for regression $\hat{y}_0 = \frac{1}{K} \sum_{i: x_i \in N_K(x_0)} y_i$

• for classification $\hat{g}_0 = \max_c \left(\sum_{i: x_i \in N_K(x_0)} \frac{1}{K} (g_i = c) \right)$

basically each point is classified in the class that is most frequent in its K neighbours

Again, K is the tuning parameter, related to the complexity of the model

- larger K → larger bias, smaller variance
- lower K → smaller bias, larger variance

We can compute K by cross-validation

Classification trees

Let start with a simple example, $p=1$, $K=2 \rightarrow \{0,1\}$

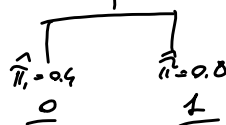
Instead of estimating $\hat{y} = \frac{1}{n} \sum_{i \in R_n} y_i$, we estimate $\Pr[G=1 | X=x]$

$$\hat{\pi}_1(x) = \frac{1}{K} \sum_{i \in R_n} I(y_i = 1) \quad I = \begin{cases} 1 & \text{if } y_i = 1 \\ 0 & \text{if } y_i = 0 \end{cases}$$

$\mathbb{1}, \underline{\mathbb{1}}, \mathbb{I}, \underline{\mathbb{I}}$ indicator function $\begin{cases} 1 & \text{if the condition is met} \\ 0 & \text{if not} \end{cases}$

Then we allocate the new observation (x_0) to category 1 if $\hat{\pi}_1 > 0.5$, 0 if $\hat{\pi}_1 \leq 0.5$

→ estimating the probability of belonging to a class based on its frequency within the region defined by the tree $x_1 < a$



To split the space (grow the tree) we need a loss-function. The most immediate is using the deviance $-2 \log \text{lik}$

↳ since we have binary outcome, we use the binomial log-likelihood

$$D = -2 \sum_{i=1}^n \left\{ y_i \log \hat{\pi}_i + (1-y_i) \log (1-\hat{\pi}_i) \right\}$$

that becomes, when we consider the split into K regions

$$D = -2 \sum_{k=1}^K n_k \left(\hat{\pi}_k \log \hat{\pi}_k + (1-\hat{\pi}_k) \log \hat{\pi}_k \right)$$

We can rewrite this formula as

$$D = 2n \sum_{k=1}^K \frac{n_k}{n} \underbrace{Q(\hat{\pi}_k)}_{\substack{\text{entropy} \\ \text{size of the leaf } k}} \quad \text{in this example } K=2$$

$$D = 2n \sum_{k=1}^K \frac{n_k}{n} Q(\hat{\pi}_k)$$

$$K=2 \quad Q(\pi) = -\pi_1 \log \pi_1 + (1-\pi_1) \log(1-\pi_1)$$

in general

$$Q(\hat{\pi}) = -\sum_k \hat{\pi}_k \log \hat{\pi}_k \quad \leftarrow \text{entropy}$$

↑
Alternatives:

- Gini index
- missclassification error

Gini index:

$$Q(\hat{\pi}) = \sum_k \hat{\pi}_k (1 - \hat{\pi}_k)$$

missclassification error:

$$Q(\hat{\pi}) = \sum_k \frac{1}{n_k} \sum_{i \in R_k} \mathbb{1}(\hat{y}_i \neq y_i)$$

same
missclassification
error



entropy
Gini index higher



entropy
Gini index lower

Support vector machines

Let us first consider the separable case. There is an hyperplane (affine set) (or a line in $\mathbb{R}^2$) that can completely separate the classes. Among all the possible hyperplane, we want to choose that that maximise the distance (m) from the closest representative of each class.

In general, an hyperplane in $\mathbb{R}^p$ is defined as

$$\beta_0 + \beta^T x = 0 \quad x \in \mathbb{R}^p$$

and identified by β_0 and β^T , $\beta_0 \in \mathbb{R}$, $\beta \in \mathbb{R}^p$

Properties:

- for each point x of the hyperplane $\beta^T x = -\beta_0$
- if x' and x'' are two points of the hyperplane, $\beta^T (x' - x'') = 0$
- therefore β is orthogonal to the hyperplane
- the signed distance from a point $x \in \mathbb{R}^p$ to the hyperplane is

$$\beta^T (x - x_0) = \frac{1}{\|\beta\|} (\beta^T x + \beta_0)$$

without loss of generality

where x_0 is projection of x to the hyperplane

With $k=2$ $y \in \{-1, 1\}$ and hyperplane $\beta_0 + \beta^T x = 0$, assuming wlog $\|\beta\|=1$

$$y (\beta_0 + \beta^T x) > 0$$

correctly classified

$$y (\beta_0 + \beta^T x) < 0$$

incorrectly classified

$$G(x) = \text{sign}[x^T \beta + \beta_0]$$

distance from the hyperplane

$$\max_{\beta_0, \beta} M \quad \text{subject to} \quad \begin{cases} \|\beta\| = 1 \\ y_i (\beta_0 + \beta^T x_i) \geq M \end{cases} \quad \begin{matrix} (4.46) \\ i=1, \dots, n \end{matrix}$$

where $2M$ is called the margin (2 because we are in a 2 class problem)

We can rewrite the problem as

$$\max_{\beta_0, \beta} M \quad \text{subject to} \quad \frac{1}{\|\beta\|} y_i (\beta_0 + \beta^T x_i) \geq M \quad i=1, \dots, n$$

$$\text{or } y_i (\beta_0 + \beta^T x_i) \geq M \|\beta\| \quad (4.47)$$

since it is a constant, we can set $\|\beta\| = \frac{1}{M}$

then the optimization problem becomes

$$\min_{\beta_0, \beta} \frac{1}{2} \|\beta\|^2 \quad \text{subject to} \quad y_i (\beta_0 + \beta^T x_i) \geq 1 \quad (4.48)$$

$$\min_{\beta_0, \beta} \|\beta\| \quad \text{subject to} \quad y_i (\beta_0 + \beta^T x_i) \geq 1 \quad (12.4)$$

quadratic optimization with 2 linear constraint $\rightarrow$ SVM

In the case of non-perfect separability, the constraint

$$y_i (\beta_0 + \beta^T x_i) \geq 1$$

becomes

$$y_i (\beta_0 + \beta^T x_i) \geq 1 (1 - \xi_i) \quad i=1, \dots, n$$

$\underbrace{1 - \xi_i \text{ and not } \geq}_{\text{mathematical convenience}}$
 $\uparrow$ slack variables
 (how far a misclassified observation is from the hyperplane)

using the same procedure as before,

$$\min_{\beta_0, \beta} \frac{1}{2} \|\beta\|^2 + \gamma \sum_{i=1}^n \xi_i \quad \text{subject to} \quad \begin{cases} \xi_i \geq 0 & \forall i \\ y_i (\beta_0 + \beta^T x_i) \geq 1 - \xi_i \end{cases}$$

where $\gamma > 0$ is a regularization parameter (it controls how much I want to penalise the distance of misclassified observation from the hyperplane)

subject to $\sum_{i=1}^n \xi_i < t$