Mathematics of Machine Learning

Rajen D. Shah r.shah@statslab.cam.ac.uk

1 Introduction

Consider a pair of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with joint distribution P_0, where X is to be thought of as an input or vector of predictors, and Y as an output or response. For instance X may represent a collection of disease risk factors (e.g. BMI, age, genetic indicators etc.) for a subject randomly selected from a population and Y may represent their disease status; or X could represent the number or bedrooms and other facilities in a randomly selected house, and Y could be its price. In the former case we may take $\mathcal{Y} = \{-1, 1\}$, and this setting is known as the (two-class) *classification* setting. The latter case where $Y \in \mathbb{R}$ is an instance of a *regression* setting. We will take $\mathcal{X} = \mathbb{R}^p$ unless otherwise specified. We refer to Y as the output, or response, and X as the input and its components as predictors or variables.

It is of interest to predict the random Y from X; we may attempt to do this via a (measurable) function $h : \mathcal{X} \to \mathcal{Y}$, known in the machine learning literature as a *hypothesis*. To measure the quality of such a prediction we will introduce a *loss* function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}.$$

In the classification setting, loss ℓ given by the *misclassification error* is particularly relevant:

$$\ell(h(x), y) = \begin{cases} 0 & \text{if } h(x) = y, \\ 1 & \text{otherwise.} \end{cases}$$

In this context h is also referred to as a *classifier*. In regression settings, the use of *squared error* $\ell(h(x), y) = (h(x) - y)^2$ is common, and we will take this to be the case unless specified otherwise. We will aim to pick a hypothesis h such that the *risk*

$$R(h) := \int_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \ell(h(x), y) \, dP_0(x, y)$$

is small[1]. For a deterministic h, $R(h) = \mathbb{E}\ell(h(X), Y)$.

Recall that the function h that minimises the risk in a regression setting is $x \mapsto \mathbb{E}(Y \,|\, X = x)$, which we refer to as the *regression function*.

A classifier h_0 that minimises the misclassification risk is known as a *Bayes classifier*, and its risk is called the *Bayes risk*. A key function in the classification context is

$$\eta(x) := \mathbb{P}(Y = 1 \,|\, X = x),$$

which is also known as the regression function here.

[1]Note that this is a different definition from the 'risk' you may have seen in *Principles of Statistics*.

Proposition 1. *A Bayes classifier h_0 is given by*[2]

$$h_0(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2 \\ -1 & \text{otherwise.} \end{cases}$$

In most settings of interest, the joint distribution P_0 of (X, Y), which determines the optimal h, will be unknown. Instead we will suppose we have i.i.d. copies $(X_1, Y_1), \ldots, (X_n, Y_n)$ of the pair (X, Y), known as *training data*. Our task is to use this data to construct a classifier $\hat{h}$ such that $R(\hat{h})$ or $\mathbb{E}R(\hat{h})$ is small.

Important point: $R(\hat{h})$ is a random variable depending on the random training data:

$$R(\hat{h}) = \mathbb{E}(\ell(\hat{h}(X), Y) \,|\, X_1, Y_1, \ldots, X_n, Y_n).$$

A (classical) statistics approach to classification may attempt to model P_0 up to some unknown parameters, estimate these parameters (e.g. by maximum likelihood), and thereby obtain an estimate of the regression function. We will take a different approach and assume that we are given a class $\mathcal{H}$ of hypotheses from which to pick our $\hat{h}$. Possible choices of $\mathcal{H}$ in the context of regression include for instance

- $\mathcal{H} = \{h : h(x) = \mu + x^\top \beta \text{ where } \mu \in \mathbb{R},\ \beta \in \mathbb{R}^p\}$;
- $\mathcal{H} = \left\{h : h(x) = \mu + \sum_{j=1}^d \varphi_j(x)\beta_j \text{ where } \mu \in \mathbb{R},\ \beta \in \mathbb{R}^d\right\}$ for a given set of what are known in this context as *basis functions* $\varphi_1, \ldots, \varphi_d : \mathcal{X} \to \mathbb{R}$;
- $\mathcal{H} = \left\{h : h(x) = \sum_{j=1}^d w_j \varphi_j(x) \text{ where } w \in \mathbb{R}^d,\ \varphi_j \in \mathcal{B}\right\}$ for a given class $\mathcal{B}$ of functions $\varphi : \mathcal{X} \to \mathbb{R}$.

In the classification setting, we may consider versions of the above composed with the sgn function e.g. $\mathcal{H} = \{h : h(x) = \operatorname{sgn}(\mu + x^\top \beta) \text{ where } \mu \in \mathbb{R},\ \beta \in \mathbb{R}^p\}$.

Technical note: In this course we will take $\operatorname{sgn}(0) = -1$. (It does not matter much whether we take $\operatorname{sgn}(0) = \pm 1$, but we need to specify a choice in order that the h listed above are well-defined.)

Non-examinable material is enclosed in *stars*.

1.1 Brief review of conditional expectation

For many of the mathematical arguments in this course we will need to manipulate conditional expectations.

[2]When $\eta(x) = 1/2$, we can equally well take $h_0(x) = \pm 1$ and achieve the same misclassification error.

Recall that if $Z \in \mathbb{R}$ and $W = (W_1, \ldots, W_d)^\top \in \mathbb{R}^d$ are random variables with joint probability density function (pdf) $f_{Z,W}$ with respect to measure μ, then the conditional pdf $f_{Z|W}$ of Z given W satisfies

$$f_{Z|W}(z|w) = \begin{cases} f_{Z,W}(z,w)/f_W(w) & \text{if } f_W(w) \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

where f_W is the marginal pdf of W. When one or more of Z and W are discrete, we typically work with probability mass functions.

Suppose $\mathbb{E}|Z| < \infty$. Then the conditional expectation function $\mathbb{E}(Z \,|\, W = w)$ is given by

$$g(w) := \mathbb{E}(Z \,|\, W = w) = \int z f_{Z|W}(z|w)\mu(dz). \tag{1.1}$$

We write $\mathbb{E}(Z \,|\, W)$ for the random variable $g(W)$ (note this is a function of W, not Z).

This is not a fully general definition of conditional expectation (for that see the *Stochastic Financial Models* course) and we will not use it. We will however make frequent use of the following properties of conditional expectation.

(i) **Role of independence:** If Z and W are independent, then $\mathbb{E}(Z \,|\, W) = \mathbb{E}Z$. If additionally for a random variable U, W is independent of (Z, U), then $\mathbb{E}(Z \,|\, U, W) = \mathbb{E}(Z \,|\, U)$.

(ii) **Tower property:** Let $f : \mathbb{R}^d \to \mathbb{R}^m$ be a (measurable) function. Then

$$\mathbb{E}\{\mathbb{E}(Z \,|\, W) \,|\, f(W)\} = \mathbb{E}\{Z \,|\, f(W)\}.$$

In particular, taking $f \equiv c \in \mathbb{R}$ and using (i) gives us that $\mathbb{E}\{\mathbb{E}(Z \,|\, W)\} = \mathbb{E}(Z)$ (as $f(W)$ is a constant it is independent of any random variable).

(iii) **Fixing what is known:** We have

$$\begin{aligned} &\mathbb{E}\{f(W_1, \ldots, W_d) \,|\, W_1 = w_1, \ldots, W_r = w_r\} \\ =&\mathbb{E}\{f(w_1, \ldots, w_r, W_{r+1}, \ldots, W_d) \,|\, W_1 = w_1, \ldots, W_r = w_r\}, \end{aligned}$$

provided the r.h.s. is well-defined. In particular, if $\mathbb{E}Z^2 < \infty$ and $g : \mathbb{R}^d \to \mathbb{R}$ is such that $\mathbb{E}[\{g(W)\}^2] < \infty$, then $\mathbb{E}\{g(W)Z \,|\, W\} = g(W)\mathbb{E}(Z \,|\, W)$, a property sometimes referred to as 'taking out what is known'.

(iv) **Best least squares predictor:** With the conditions in (iii) above, we have

$$\mathbb{E}\{Z - g(W)\}^2 = \mathbb{E}\{Z - \mathbb{E}(Z \,|\, W)\}^2 + \mathbb{E}\{\mathbb{E}(Z \,|\, W) - g(W)\}^2. \tag{1.2}$$

Indeed, using the tower property,

$$\begin{aligned} \mathbb{E}\{Z - g(W)\}^2 &= \mathbb{E}\{Z - \mathbb{E}(Z \,|\, W) + \mathbb{E}(Z \,|\, W) - g(W)\}^2 \\ &= \mathbb{E}\{Z - \mathbb{E}(Z \,|\, W)\}^2 + \mathbb{E}\{\mathbb{E}(Z \,|\, W) - g(W)\}^2 \\ &\quad + 2\mathbb{E}\,\mathbb{E}[\{Z - \mathbb{E}(Z \,|\, W)\}\{\mathbb{E}(Z \,|\, W) - g(W)\} \,|\, W], \end{aligned}$$

but by 'taking what is known', half the final term is

$$\mathbb{E}[\{\mathbb{E}(Z \mid W) - g(W)\} \underbrace{\mathbb{E}\{Z - \mathbb{E}(Z \mid W) \mid W\}}_{=0}] = 0.$$

Property (iv) verifies that the $h : \mathcal{X} \to \mathbb{R}$ minimising $R(h)$ under squared loss is $h_0(x) = \mathbb{E}(Y \mid X = x)$.

Probabilistic results can be 'applied conditionally', for example:

Conditional Jensen. Recall that $f : \mathbb{R} \to \mathbb{R}$ is a convex function if

$$tf(x) + (1-t)f(y) \geq f\big(tx + (1-t)y\big) \quad \text{for all } x, y \in \mathbb{R} \text{ and } t \in (0,1).$$

The conditional version of *Jensen's inequality* states that if $f : \mathbb{R} \to \mathbb{R}$ is convex and random variable Z has $\mathbb{E}|f(Z)| < \infty$, then

$$\mathbb{E}\big(f(Z) \mid W\big) \geq f\big(\mathbb{E}(Z \mid W)\big).$$

1.2 Bayes risk

Proof of Proposition 1. We have $R(h) = \mathbb{E}\mathbb{1}_{\{Y \neq h(X)\}} = \mathbb{E}\mathbb{E}[\mathbb{1}_{\{Y \neq h(X)\}} \mid X]$.

Now

$$\begin{aligned}\mathbb{E}[\mathbb{1}_{\{Y \neq h(X)\}} \mid X] &= \mathbb{E}[\mathbb{1}_{\{Y=1\}}\mathbb{1}_{\{h(X)=-1\}} + \mathbb{1}_{\{Y=-1\}}\mathbb{1}_{\{h(X)=1\}} \mid X] \\ &= \mathbb{1}_{\{h(X)=-1\}}\eta(X) + \mathbb{1}_{\{h(X)=1\}}(1-\eta(X)).\end{aligned}$$

When $\eta(X) > 1 - \eta(x)$ and so $\eta(X) > 1/2$, this is minimised by taking $h(X) = 1$, and similarly when $\eta(X) < 1/2$, this is minimised by taking $h(X) = -1$. If $\eta(X) = 1/2$, then the above is constant so any $h(X)$ minimises this. □

1.3 Empirical risk minimisation

Empirical risk minimisation replaces the expectation over the unknown P_0 in the definition of the risk with the empirical distribution, and seeks to minimise the resulting objective over $h \in \mathcal{H}$:

$$\hat{R}(h) := \frac{1}{n}\sum_{i=1}^{n} \ell(h(X_i), Y_i), \qquad \hat{h} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \hat{R}(h).$$

$\hat{R}(h)$ is the *empirical risk* or *training error* of h and $\hat{h}$ is the *empirical risk minimiser* (ERM).

Example 1. Consider the regression setting with $\mathcal{Y} = \mathbb{R}$, squared error loss and $\mathcal{H} = \{x \mapsto \mu + x^\top\beta \text{ for } \mu \in \mathbb{R},\ \beta \in \mathbb{R}^p\}$. Then empirical risk minimisation is equivalent to ordinary least squares, i.e. we have

$$\hat{h}(x) = \hat{\mu} + \hat{\beta}^\top x \quad \text{where } (\hat{\mu}, \hat{\beta}) \in \operatorname*{arg\,min}_{(\mu,\beta) \in \mathbb{R}\times\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \mu - X_i^\top\beta)^2.$$

We can consider applying this more generally where

$$\mathcal{H} = \left\{ x \mapsto \sum_{j=1}^{d} \varphi_j(x)\beta_j \text{ where } \beta \in \mathbb{R}^d \right\}$$

and $\varphi_j : \mathbb{R}^p \to \mathbb{R}$ for $j = 1, \ldots, d$. For instance in the case where $p = 1$, we could have $\varphi_j(x) = x^{j-1}$. Then forming matrix $\Phi \in \mathbb{R}^{n \times d}$ with entries $\Phi_{ij} = \varphi_j(X_i)$ assumed to be of full column rank, and writing $\varphi(x) = (\varphi_1(x), \ldots, \varphi_d(x))$, we have that the ERM $\hat{h} : x \mapsto \hat{\beta}^\top \varphi(x)$ where

$$\hat{\beta} = (\Phi^\top \Phi)^{-1} \Phi^\top Y_{1:n} \tag{1.3}$$

and $Y_{1:n} := (Y_1, \ldots, Y_n)^\top$. △

A good choice for the class $\mathcal{H}$ will result in a low *generalisation error* $R(\hat{h})$. This is a measure of how well we can expect the ERM $\hat{h}$ to predict a new data point $(X, Y) \sim P_0$ given only knowledge of X. To understand the competing factors that drive this sort of quantity, it is helpful to consider the case of squared error loss where, as we shall see, this may be related to a sum of (squared) bias and variance terms.

1.4 Bias–variance tradeoff

Let us consider $\hat{h} = \hat{h}_D$ trained on data $D = (X_i, Y_i)_{i=1}^n$ formed of iid copies of an independent random pair (X, Y). We first consider its expected performance in terms of squared error at X. To this end, it is helpful to introduce

$$\bar{h} : x \mapsto \mathbb{E}(\hat{h}_D(x)),$$

i.e. the average over the training data of $\hat{h}_D$, and the related function

$$\widetilde{h}_{X_{1:n}} : x \mapsto \mathbb{E}(\hat{h}_D(x) \,|\, X_{1:n}).$$

Recall property (iv) of conditional expectations, that for random variables $Z, W \in \mathbb{R} \times \mathcal{W}$ and $f : \mathcal{W} \to \mathbb{R}$, we have

$$\mathbb{E}\{Z - f(W)\}^2 = \mathbb{E}\{Z - \mathbb{E}(Z \,|\, W)\}^2 + \mathbb{E}\{\mathbb{E}(Z \,|\, W) - f(W)\}^2.$$

Using, this we have

$$\begin{aligned}
&\mathbb{E}[\{Y - \hat{h}_D(X)\}^2 \,|\, X] \\
&= \mathbb{E}[\{Y - \underbrace{\mathbb{E}(Y \,|\, X, D)}_{=\mathbb{E}(Y \,|\, X)}\}^2 \,|\, X] + \mathbb{E}[\{\mathbb{E}(Y \,|\, X) - \hat{h}_D(X)\}^2 \,|\, X] \\
&= \mathrm{Var}(Y \,|\, X) + \mathbb{E}[\{\hat{h}_D(X) - \underbrace{\mathbb{E}(\hat{h}_D(X) \,|\, X)}_{=\bar{h}(X)}\}^2 \,|\, X] + \mathbb{E}[\{\mathbb{E}(Y \,|\, X) - \bar{h}(X)\}^2 \,|\, X].
\end{aligned} \tag{1.4}$$

Here, we have used the fact that

$$\mathbb{E}(\hat{h}_D(X) \,|\, X = x) = \mathbb{E}(\hat{h}_D(x) \,|\, X = x) = \bar{h}(x).$$

Thus, taking expectations:

$$\mathbb{E}R(\hat{h}_D) = \underbrace{\mathbb{E}\{\mathbb{E}(Y \,|\, X) - \bar{h}(X)\}^2}_{\text{squared bias}} + \underbrace{\mathbb{E}\mathrm{Var}(\hat{h}_D(X) \,|\, X)}_{\text{variance of } \hat{h}} + \underbrace{\mathbb{E}\mathrm{Var}(Y \,|\, X)}_{\text{irreducible variance}} . \tag{1.5}$$

If $\hat{h}$ were an ERM over class $\mathcal{H}$, we would expect a rich class of hypotheses to result in a smaller squared bias term. However, the variance would likely increase as empirical risk minimisation may fit to the realised $Y_1, \ldots, Y_n$ closely and so $\hat{h}_D$ would be very sensitive to the training data D.

To see this tradeoff more clearly, it is instructive to consider a related decomposition to (1.4) involving $\widetilde{h}$: we have

$$\mathbb{E}[\{Y - \hat{h}_D(X)\}^2 \,|\, X = x] = \mathbb{E}\{\mathbb{E}(Y \,|\, X = x) - \widetilde{h}_{X_{1:n}}(x)\}^2 + \mathbb{E}\{\hat{h}_D(x) - \widetilde{h}_{X_{1:n}}(x)\}^2 + \mathrm{Var}(Y \,|\, X = x).$$

We examine the middle term in more detail, and consider the special case where $\hat{h}_D$ is the ERM of Example 1 given by (1.3), that is $\hat{h}_D(x) = \varphi(x)^\top (\Phi^\top \Phi)^{-1} \Phi^\top Y_{1:n}$ with $\varphi(x) \in \mathbb{R}^d$. To facilitate our analysis, let us assume that $\mathrm{Var}(Y \,|\, X = x) =: \sigma^2$ is constant in x. Then we have

$$\begin{aligned}
&\mathbb{E}[\{\hat{h}_D(x) - \widetilde{h}_{X_{1:n}}(x)\}^2 \,|\, X_{1:n}] \\
&= \mathbb{E}[\{\varphi(x)^\top (\Phi^\top \Phi)^{-1} \Phi^\top (Y_{1:n} - \mathbb{E}(Y_{1:n} \,|\, X_{1:n}))\}^2 \,|\, X_{1:n}] \\
&= \varphi(x)^\top (\Phi^\top \Phi)^{-1} \Phi^\top \mathbb{E}[\{Y_{1:n} - \mathbb{E}(Y_{1:n} \,|\, X_{1:n})\}\{Y_{1:n} - \mathbb{E}(Y_{1:n} \,|\, X_{1:n})\}^\top \,|\, X_{1:n}] \Phi (\Phi^\top \Phi)^{-1} \varphi(x).
\end{aligned}$$

Note that by property (i) of conditional expectations, $\mathbb{E}(Y_j \,|\, X_{1:n}) = \mathbb{E}(Y_j \,|\, X_j)$ and also,

$$\begin{aligned}
\mathbb{E}[\{Y_j - \mathbb{E}(Y_j \,|\, X_j)\}\{Y_k - \mathbb{E}(Y_k \,|\, X_k)\} \,|\, X_{1:n}] &= \mathbb{E}[\{Y_j - \mathbb{E}(Y_j \,|\, X_j)\}\{Y_k - \mathbb{E}(Y_k \,|\, X_k)\} \,|\, X_j, X_k] \\
&= \mathbb{E}(Y_j Y_k \,|\, X_j, X_k) - \mathbb{E}(Y_j \,|\, X_j)\mathbb{E}(Y_k \,|\, X_k),
\end{aligned}$$

using the tower property in the final line. Now if $j \neq k$,

$$\begin{aligned}
\mathbb{E}(Y_j Y_k \,|\, X_j, X_k) &= \mathbb{E}\{\mathbb{E}(Y_j Y_k \,|\, Y_j, X_j, X_k) \,|\, X_j, X_k\} \qquad \text{(tower property)} \\
&= \mathbb{E}\{Y_j \mathbb{E}(Y_k \,|\, Y_j, X_j, X_k) \,|\, X_j, X_k\} \qquad \text{(taking out what is known)} \\
&= \mathbb{E}\{Y_j \mathbb{E}(Y_k \,|\, X_k) \,|\, X_j, X_k\} \qquad \text{(property (i))} \\
&= \mathbb{E}(Y_j \,|\, X_j)\mathbb{E}(Y_k \,|\, X_k) \qquad \text{(taking out what is known and (i))}.
\end{aligned}$$

Thus $\mathbb{E}[\{Y_{1:n} - \mathbb{E}(Y_{1:n} \,|\, X_{1:n})\}\{Y_{1:n} - \mathbb{E}(Y_{1:n} \,|\, X_{1:n})\}^\top \,|\, X_{1:n}] = \sigma^2 I$, and so

$$\mathbb{E}[\{\hat{h}_D(x) - \widetilde{h}_{X_{1:n}}(x)\}^2 \,|\, X_{1:n}] = \sigma^2 \varphi(x)^\top (\Phi^\top \Phi)^{-1} \varphi(x).$$

Consider now averaging this over the training points $x = X_1, \ldots, X_n$. Noting that $\varphi(X_i)$ is the ith row of Φ, we may compute, using the 'trace trick' (and that trace is invariant to cyclic permutations),

$$\frac{1}{n}\sum_{i=1}^{n}\sigma^2 \mathrm{tr}\{\varphi(X_i)^\top(\Phi^\top\Phi)^{-1}\varphi(X_i)\} = \frac{\sigma^2}{n}\mathrm{tr}\Bigg(\underbrace{\sum_{i=1}^{n}\varphi(X_i)\varphi(X_i)^\top}_{=\Phi^\top\Phi}(\Phi^\top\Phi)^{-1}\Bigg)$$
$$= \frac{\sigma^2 d}{n}.$$

Thus the variance term increases linearly with d, while the squared bias should decrease when adding further basis functions φ_j.

At least two questions may arise at this stage: how should we choose the number of basis functions in practice in order to obtain a small expected risk? And, particularly in multivariate settings, what are sensible ways of choosing the basis functions themselves? We turn to the first of these questions next.

1.5 Cross-validation

The question of selecting the appropriate number of basis functions in a linear regression may be seen as a special case of the following problem: given a number of competing machine learning methods, select from these (ideally) the best one i.e. one that trades off bias and variance most favourably. In the case of linear regression, each regression using a given set of basis functions may be thought of as one of the competing methods.

Now let $\hat{h}^{(1)}, \ldots, \hat{h}^{(m)}$ be a collection of machine learning methods: for instance $\hat{h}^{(j)}$ could correspond to performing linear regression using basis functions $\varphi_1, \ldots, \varphi_j$. Each $\hat{h}^{(j)}$ takes as its argument i.i.d. training data $(X_i, Y_i)_{i=1}^n =: D \in (\mathcal{X} \times \mathcal{Y})^n$ and outputs a hypothesis, so $\hat{h}^{(j)}_D : \mathcal{X} \to \mathbb{R}$. Given a loss function ℓ with associated risk R, we may ideally want to pick a $\hat{h}^{(j)}$ such that the risk

$$R(\hat{h}^{(j)}_D) = \mathbb{E}\{\ell(\hat{h}^{(j)}_D(X), Y) \,|\, D\} \tag{1.6}$$

is minimised. Here $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ is independent of D and has the same distribution as (X_1, Y_1). This $\hat{h}^{(j)}$ is such that conditional on the original training data, it minimises the expected loss on a new observation drawn from the same distribution as the training data.

A less ambitious goal is to find a j to minimise the expected risk

$$\mathbb{E}R(\hat{h}^{(j)}_D) = \mathbb{E}[\mathbb{E}\{\ell(\hat{h}^{(j)}_D(X), Y) \,|\, D\}] \tag{1.7}$$

where compared with (1.6), we have taken a further expectation over the training data D.

We still have no way of computing (1.7) directly, but we can attempt to estimate it. The idea of v-fold cross-validation is to split the data into v groups or folds of roughly equal size. Let D_{-k} be all the data except that in the kth fold, and let $A_k \subset \{1, \ldots, n\}$

be the observation indices corresponding to the kth fold. For each j we apply $\hat{h}^{(j)}$ to data D_{-k} to obtain hypothesis $\hat{h}^{(j)}_{-k} := \hat{h}^{(j)}_{D_{-k}}$. We choose the value of j that minimises

$$\mathrm{CV}(j) := \frac{1}{n}\sum_{k=1}^{v}\sum_{i\in A_k} \ell(\hat{h}^{(j)}_{-k}(X_i), Y_i).$$

Writing $\hat{j}$ for the minimiser, we may take final selected hypothesis to be $\hat{h}^{(\hat{j})}_D$.

Note that for each $i \in A_k$,

$$\mathbb{E}\ell(\hat{h}^{(j)}_{-k}(X_i), Y_i) = \mathbb{E}[\mathbb{E}\{\ell(\hat{h}^{(j)}_{-k}(X_i), Y_i)|D_{-k}\}]. \tag{1.8}$$

This is precisely the expected loss in (1.7) but with training data D replaced with a training data set of smaller size. If all the folds have the same size, then $\mathrm{CV}(j)$ is an average of n identically distributed quantities, each with expected value as in (1.8). However, the quantities being averaged are not independent as they share the same data.

Thus cross-validation gives a biased estimate of the expected prediction error. The amount of the bias depends on the size of the folds, the case when the $v = n$, known as *leave-one-out cross-validation* typically giving the least bias, though using this often comes with an increased computational cost. Typical choices of v are 5 or 10.

2 Popular machine learning methods I

2.1 Decision trees

We now have a way to select an appropriate subset of basis functions to use from a larger collection, but how should we choose this collection in the first place? Decision trees (also known as regression trees in the regression context we study here; there are also variants for classification which we will not discuss) form a highly popular class of methods for doing this in a data-driven fashion.

Regression trees use a set of basis functions consisting of indicator functions on rectangular regions and take the form

$$T(x) = \sum_{j=1}^{J} \gamma_j \mathbb{1}_{R_j}(x); \tag{2.1}$$

here R_j are rectangular regions that form a partition of $\mathbb{R}^p$ and the γ_j are coefficients in $\mathbb{R}$.

The regions and coefficients are typically computed from data $(X_i, Y_i)_{i=1}^n$ using the following recursive binary partitioning algorithm.

1. Input maximum number of regions J. Initialise $\hat{\mathcal{R}} = \{\mathbb{R}^p\}$.

2. We now split one of the regions in $\hat{\mathcal{R}}$ using an axis aligned split such that a particular splitting criterion is minimised. In the regression case, it often makes sense to aim to minimise the overall residual sum of squares (RSS) as follows.

   (a) For each region $R \in \hat{\mathcal{R}}$ such that $I := \{i : X_i \in R\}$ has $|I| > 1$, perform the following. For each $j = 1, \ldots, p$, let $\mathcal{S}_j$ be the set of mid-points between adjacent $\{X_{ij}\}_{i \in I}$. Find the predictor $\hat{j}_R$ and split point $\hat{s}_R$ to minimise over $j \in \{1, \ldots, p\}$ and $s \in \mathcal{S}_j$,

$$\underbrace{\min_{\gamma_L \in \mathbb{R}} \sum_{i \in I : X_{ij} \leq s} (Y_i - \gamma_L)^2 + \min_{\gamma_R \in \mathbb{R}} \sum_{i \in I : X_{ij} > s} (Y_i - \gamma_R)^2}_{\text{RSS on } I \text{ when splitting at } s} - \underbrace{\min_{c \in \mathbb{R}} \sum_{i \in I} (Y_i - c)^2}_{\text{RSS on } I \text{ without splitting}} \quad . \quad (2.2)$$

   (b) Let $\hat{R}$ be the region yielding the lowest value of (2.2) and define

$$\hat{R}_L := \{x \in \hat{R} : x_{\hat{j}_{\hat{R}}} \leq \hat{s}_{\hat{R}}\}, \quad \hat{R}_R := \hat{R} \setminus \hat{R}_L.$$

   Refine the partition via $\hat{\mathcal{R}} \leftarrow (\hat{\mathcal{R}} \setminus \{\hat{R}\}) \cup \{\hat{R}_L, \hat{R}_R\}$.

3. Repeat step 2 until $|\hat{\mathcal{R}}| = J$.

4. Writing $\hat{\mathcal{R}} = \{\hat{R}_1, \ldots, \hat{R}_J\}$, let $\hat{I}_j = \{i : X_i \in \hat{R}_j\}$ and

$$\hat{\gamma}_j := \frac{1}{|\hat{I}_j|} \sum_{i \in \hat{I}_j} Y_i.$$

Output $\hat{T} : \mathbb{R}^p \to \mathbb{R}$ such that $\hat{T}(x) = \sum_{j=1}^{J} \hat{\gamma}_j \mathbb{1}_{\{x \in \hat{R}_j\}}$.

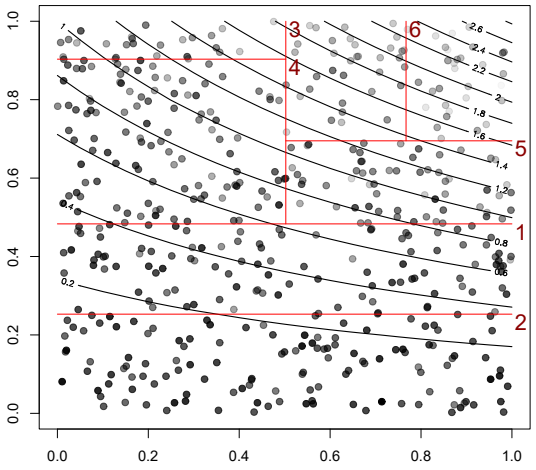
Note that $\hat{T}$ is the ERM over the class of functions

$$\left\{ T : T(x) = \sum_{j=1}^{J} \gamma_j \mathbb{1}_{\hat{R}_j}(x) \ : \gamma \in \mathbb{R}^J \right\},$$

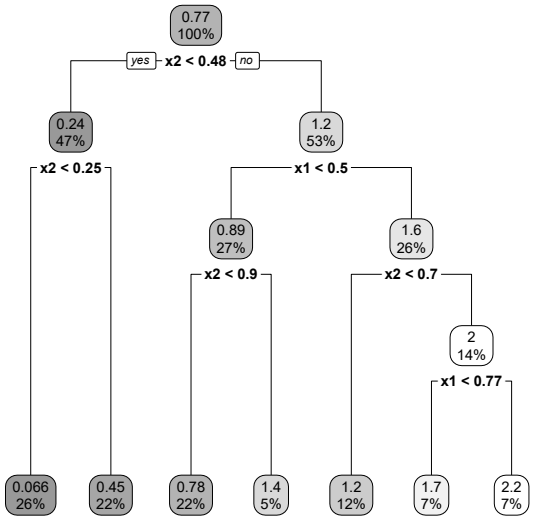
with the regions $\hat{R}_1, \ldots, \hat{R}_J$ fixed. Note that although the regions were constructed in a data-driven fashion, they were chosen greedily to minimise the RSS at each stage. Thus in general, the fitted $\hat{T}$ will not coincide with the RSS-minimising function of the form (2.1).

The fitted $\hat{T}$ can be conveniently visualised in terms of a tree as indicated in Figure 1b. The regions $\hat{R}_j$ correspond to the so-called *leaves*, those bottom nodes with only a single edge emanating from them.

At first sight, it might appear that the minimisation in 2 (a) is computationally intensive as it involves both a loop over j and for each $s \in \mathcal{S}_j$ performing a least squares regression. To see how the computations may be arranged efficiently, let us consider, for notational simplicity, the first split, so $I = \{1, \ldots, n\}$, and where $p = 1$.



(a) Rectangular regions constructed using the regression tree algorithm fitted to a dataset with two predictors with numbers indicating the order in which the splits were made. Also shown are the contours of the true regression function $\mathbb{E}(Y \mid X = x)$.



(b) Visualisation of the fitted regression tree. The percentages give the proportion of data in the corresponding region and also given is the average of the responses corresponding to those points.

Suppose that the $\{X_i\}_{i=1}^n$ are sorted so $X_1 < X_2 < \cdots < X_n$. The minimisation problem above is equivalent to finding m to minimise $Q_m + P_m$ where

$$Q_m := \min_{\gamma_L \in \mathbb{R}} \sum_{i \le m} (Y_i - \gamma_L)^2 \qquad \text{and} \qquad P_m := \min_{\gamma_R \in \mathbb{R}} \sum_{i > m} (Y_i - \gamma_R)^2.$$

Note that

$$Q_m = \sum_{i \le m} \left(Y_i - \frac{1}{m} \sum_{i \le m} Y_i \right)^2 = \sum_{i \le m} Y_i^2 - \frac{1}{m} \left(\sum_{i \le m} Y_i \right)^2,$$

with a similar decomposition for P_m. Thus

$$P_m + Q_m = \sum_{i=1}^n Y_i^2 - \frac{1}{m} \left(\sum_{i \le m} Y_i \right)^2 - \frac{1}{n-m} \left(\sum_{i > m} Y_i \right)^2.$$

As the first term does not depend on m, we may equivalently maximise

$$\frac{1}{m} \left(\sum_{i \le m} Y_i \right)^2 + \frac{1}{n-m} \left(\sum_{i > m} Y_i \right)^2$$

over m. Let $A_m := \sum_{i \le m} Y_i$ and $B_m := \sum_{i > m} Y_i$. Then $A_{m+1} = A_m + Y_{m+1}$ and $B_{m+1} = B_m - Y_{m+1}$. Thus all $A_1, \ldots, A_{n-1}$ and $B_1, \ldots, B_{n-1}$ may be computed in $O(n)$ operations. Thus we may compute the display above for all $m = 1, \ldots, n-1$ in $O(n)$ operations, and hence we may minimise it over m with the same cost.

A further important point is that the minimisation in 2 (a) need only be performed for the two new regions constructed in 2 (b) during the previous operation. The reason is that we can store the value of the objective in 2 (a), as well as $\hat{j}_R$ and $\hat{s}_R$ for all other regions R, and these do not need to be recomputed.

In order to use a decision tree is practice, one must choose the number of regions J: a large J might result in overfitting, while a small J may result in a large bias. Choosing J may be done via cross-validation. An alternative (typically preferred) approach is to grow a very large tree, and then collapse regions together according to a pruning strategy; we do not discuss this here.

2.2 Random forests

Whilst decision trees as above are a useful machine learning method in their own right, they have a few disadvantages:

- The piecewise constant estimated regression functions they fit, while useful for visualisation purposes (see Figure 1b) might not always deliver the best prediction error particularly when the true regression function varies smoothly with the predictors.
- The process of building a tree is greedy and unstable. As a consequence, small changes in the training data may lead to a very different tree; that is a fitted tree can have high variance (over the training data).

The *Random forest* procedure is a highly successful algorithm that aims to remedy these two deficiencies, though as we shall see, it does sacrifice interpretability of the fitted regression function.

Consider the regression setting where $Y_i \in \mathbb{R}$ and we are using squared error loss. Let $\hat{T}_D$ be a decision tree trained on data $D := (X_i, Y_i)_{i=1}^n$. Also let $\bar{T}$ be given by $\bar{T}(x) = \mathbb{E}\hat{T}_D(x)$ and let (X, Y) be independent of D with $(X, Y) \stackrel{d}{=} (X_1, Y_1)$.

Recall the decomposition of the expected risk (1.5) in Section 1.4:

$$\mathbb{E}R(\hat{T}_D) = \underbrace{\mathbb{E}\{\mathbb{E}(Y \,|\, X) - \bar{T}(X)\}^2}_{\text{squared bias}} + \underbrace{\mathbb{E}\mathrm{Var}(\hat{T}_D(X) \,|\, X)}_{\text{variance of the tree}} + \underbrace{\mathbb{E}\mathrm{Var}(Y \,|\, X)}_{\text{irreducible variance}} .$$

If the number of regions J used by $\hat{T}_D$ is large, some of these regions will contain only small numbers of observations in them so the corresponding coefficients $\hat{\gamma}_j$ will by highly variable and consequently $\mathbb{E}\mathrm{Var}(\hat{T}_D(X) \,|\, X)$ will tend to be large. On the other hand, the squared bias above and hence $R(\bar{T})$ may be low as a large J would allow $\bar{T}$ to approximate $x \mapsto \mathbb{E}(Y \,|\, X = x)$ well.

Random forest effectively attempts to 'estimate' $\bar{T}$ and so improve upon the variance of a single tree. If we had multiple independent datasets $D_1, \ldots, D_B$, we could form an unbiased estimate via $\sum_{b=1}^B \hat{T}_{D_b}$. Random forest samples the data D with replacement to form new datasets $D_1^*, \ldots, D_B^*$ and performs the following.

1. For each $b = 1, \ldots, B$, grow a decision tree $\hat{T}^{(b)} := \hat{T}_{D_b^*}$ but when searching for the best predictor to split on, randomly sample (without replacement) m_{try} of the p predictors and choose the best split from among these variables.

2. Output $f_{\text{rf}} = \frac{1}{B}\sum_{b=1}^{B} \hat{T}^{(b)}$.

One reason for sampling predictors is to try to make the $\hat{T}^{(b)}$ more independent. To see why this would be useful, suppose for $b_1 \neq b_2$ and some $x \in \mathbb{R}^p$ that $\text{Corr}(\hat{T}^{(b_1)}(x), \hat{T}^{(b_2)}(x)) = \rho \geq 0$. Then

$$\begin{aligned}\text{Var}(f_{\text{rf}}(x)) &= \frac{1}{B}\text{Var}(\hat{T}^{(1)}(x)) + \frac{\rho B(B-1)}{B^2}\text{Var}(\hat{T}^{(1)}(x)) \\ &= \frac{1-\rho}{B}\text{Var}(\hat{T}^{(1)}(x)) + \rho\text{Var}(\hat{T}^{(1)}(x)).\end{aligned}$$

Whilst the first term can be made small for large B, the second term does not depend on B, so we would like ρ to be small. The extra randomisation in the form of sampling predictors can help to achieve this. On the other hand, we would expect the squared bias to increase as m_{try} is decreased. An appropriate value of m_{try} may be selected using cross-validation.

3 Statistical learning theory

In a regression setting, using OLS with a set of d basis functions as in Example 1 to give $\hat{h}_D$ (where $D = (X_{1:n}, Y_{1:n})$ is the training data) yields

$$\mathbb{E}R(\hat{h}_D) - \mathbb{E}R(\widetilde{h}_{X_{1:n}}) \approx \frac{\sigma^2 d}{n}, \tag{3.1}$$

assuming $\sigma^2 := \text{Var}(Y \mid X = x)$ is constant in x (see example sheet and the discussion in Section 1.4).

Our goal now is to study a roughly analogous quantity to the LHS of (3.1) in the classification setting. For an ERM $\hat{h}$ over a class $\mathcal{H}$, in general, $x \mapsto \mathbb{E}(\hat{h}_D(x) \mid X_{1:n})$ will not be a classifier. Instead, we may compare the risk or expected risk of $\hat{h}_D = \hat{h}$ to

$$h^* := \arg\min_{h \in \mathcal{H}} R(h),$$

the best[3] hypothesis in $\mathcal{H}$.

The quantity $R(\hat{h}) - R(h^*)$ is sometimes known as the *excess risk*. Some questions of interest are:

- How does the 'complexity' of $\mathcal{H}$ influence the excess risk?

[3]If there is no h^* that achieves the associated infimum, we can consider an approximate minimiser with $R(h^*) < \inf_{h \in \mathcal{H}} R(h) + \epsilon$ for arbitrary $\epsilon > 0$ and all our analysis to follow will carry through. In fact similar reasoning is applicable to the ERM $\hat{h}$.

- How does a change in the size n of the data affect the excess risk?

Statistical learning theory is the branch of machine learning devoted to these sorts of considerations and in this course we aim to provide an introduction to some of the key ideas in this area. Our starting point is the following decomposition of the excess risk:

$$\begin{aligned} R(\hat{h}) - R(h^*) &= R(\hat{h}) - \hat{R}(\hat{h}) + \underbrace{\hat{R}(\hat{h}) - \hat{R}(h^*)}_{\leq 0} + \hat{R}(h^*) - R(h^*) \\ &\leq \sup_{h \in \mathcal{H}} \{R(h) - \hat{R}(h)\} + \hat{R}(h^*) - R(h^*). \end{aligned}$$

We wish to bound either the tail probability or the expectation of the excess risk. To motivate the developments to follow, consider the former case, for which it would be helpful to upper bound

$$\mathbb{P}\left(\sup_{h \in \mathcal{H}} \{R(h) - \hat{R}(h)\} > t \right)$$

for a given $t \geq 0$. Consider, for the time being, the setting where $|\mathcal{H}|$ is finite; ultimately we would like to tackle the case where $|\mathcal{H}|$ is infinite. A *union bound* gives

$$\begin{aligned} \mathbb{P}\left(\max_{h \in \mathcal{H}} \{R(h) - \hat{R}(h)\} > t \right) &= \mathbb{P}(\cup_{h \in \mathcal{H}} \{R(h) - \hat{R}(h) > t\}) \\ &\leq \sum_{h \in \mathcal{H}} \mathbb{P}(R(h) - \hat{R}(h) > t). \end{aligned} \tag{3.2}$$

Now for each fixed $h \in \mathcal{H}$,

$$R(h) - \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} [\mathbb{E}\{\ell(h(X_i), Y_i)\} - \ell(h(X_i), Y_i)]$$

is an average of n i.i.d. mean-zero random variables. The central limit theorem (CLT) would suggest that $\sqrt{n}\{R(h) - \hat{R}(h)\}$ should behave like a $N(0, \mathrm{Var}(\ell(h(X_1), Y_1)))$-distributed random variable. However, in order to make use of this to bound (3.2), we would need a *uniform* limiting result for all $h \in \mathcal{H}$. In order to trade off bias and variance favourably, we may wish to increase the complexity of $\mathcal{H}$, i.e. the size of $|\mathcal{H}|$, for large n, so it is not at all clear that such a uniform result should hold. Moreover, in order for (3.2) to be small, we would need to consider t fairly large, so we would need such a limiting result to provide a good approximation in the far right tail of the distribution of $\sqrt{n}\{R(h) - \hat{R}(h)\}$. Such desiderata go far beyond what is offered by the CLT, and instead we turn to concentration inequalities, an important area of probability theory that (for example) can provide nonasymptotic tail bounds that mimic what we would have liked to obtain from the CLT, for averages of certain types of independent random variables.