INFERENCE IN STATISTICAL MODELLING AND MACHINE LEARNING SOLUTIONS MANUAL STUDENT VERSION

James Burridge
Nick Tosh

This book was compiled using cspmA.cls 2009/09/17, v2.00

Contents

1

Orientation

■ ***Exercise*** 1.1 Hypothesis TT implies that Hilda spoke truly. Given what Hilda said, this implies that at least one of Hilda and Godiva is a liar, contradicting TT. Hypotheses LT implies that Hilda spoke falsely; this implies that it is *not* the case that at least one of Hilda and Godiva is a liar, contradicting LT. Hypothesis LL is inconsistent with the evidence for the same reason. Hypothesis TL implies that Hilda spoke truly, which in turn implies that at least one of Hilda and Godiva is a liar. But here there is no contradiction, because Godiva can be a liar (as the hypothesis states).

■ ***Exercise*** 1.2 Here we will speak of *events* rather than hypotheses, to match the terminology in Appendix A. Let $\mathbf{Pr}$ be the probability measure representing your degrees of belief before you receive Hilda's answer to the question 'is at least one of you a liar?' Let E be the event that Hilda replies 'yes'. We wish to calculate $\mathbf{Pr}(TL|E)$. By Bayes' theorem, we have

$$\mathbf{Pr}(TL|E) = \mathbf{Pr}(E|TL)\frac{\mathbf{Pr}(TL)}{\mathbf{Pr}(E)}. \tag{1.1}$$

Since TT, TL, LT and LL are mutually exclusive events and $\mathbf{Pr}(TT \cup TL \cup LT \cup LL) = 1$, the law of total probability gives

$$\mathbf{Pr}(E) = \mathbf{Pr}(E|TT)\mathbf{Pr}(TT) + \mathbf{Pr}(E|TL)\mathbf{Pr}(TL) + \mathbf{Pr}(E|LT)\mathbf{Pr}(LT) + \mathbf{Pr}(E|LL)\mathbf{Pr}(LL).$$

We are supposing that 'people act against type 20% of the time'. That means

$$\mathbf{Pr}(E|TT) = \mathbf{Pr}(E|LT) = \mathbf{Pr}(E|LL) = \tfrac{1}{5},$$

while $\mathbf{Pr}(E|TL) = \frac{4}{5}$. Finally, we are given that

$$\mathbf{Pr}(TT) = \mathbf{Pr}(TL) = \mathbf{Pr}(LT) = \mathbf{Pr}(LL) = \tfrac{1}{4}.$$

Plugging everything into equation (1.1), we have

$$\mathbf{Pr}(TL|E) = \tfrac{4}{5} \times \frac{\frac{1}{4}}{(\frac{1}{5} + \frac{4}{5} + \frac{1}{5} + \frac{1}{5}) \times \frac{1}{4}} = \tfrac{4}{7}.$$

■ ***Exercise*** 1.3 'Given a series of clinical measurements from a single individual, including blood sugar levels, weight, height, blood pressure and so on, predict whether that person has diabetes': this is a *classification* task.

'Given a photograph of a plant, predict the plant's species': this is a *classification* task.

'Given an audio recording of a person speaking, predict the person's age': this is a *regression* task.

■ ***Exercise*** 1.4 Here we have two variables: volume of gunpowder, x, and distance travelled by the cannonball, y. The question asks us to predict how much gunpowder is needed to make the cannonball travel a distance $y = 7$. There are two possible approaches: we could treat y as the predictor and x as the response, or vice-versa, corresponding to two different prediction functions. Writing these $\hat{f}$ and $\hat{g}$ we have

$$\hat{x} = \hat{f}(y)$$
$$\hat{y} = \hat{g}(x).$$

In the second case we'd need to invert the prediction function in order to predict x given y. That is $\hat{x} = \hat{g}^{-1}(y)$. If we think that the distance travelled by the ball is equal to some deterministic function of x, plus some random noise which doesn't depend on x, then there are principled reasons (to be discussed in later chapters) why we might prefer to treat x as the predictor. The question indicates that $\hat{g}$ is a member of the family $g(x) = \beta x$ where $\beta > 0$ is parameter which we must choose.

One way to find β is to minimise the *residual sum of squares* between predictions and data. For an arbitrary prediction function g, this is

$$\mathrm{RSS}(g) = \sum_{(\mathrm{x},\mathrm{y})} (y - g(x))^2.$$

In our example,

$$\mathrm{RSS}(\beta) = (\beta - 5)^2 + (2\beta - 11)^2 + (3\beta - 14)^2 = 342 - 138\beta + 14\beta^2.$$

The smallest value of the RSS function is achieved when $\mathrm{RSS}'(\beta) = -138 + 28\beta = 0$ so $\beta = 69/14 \approx 4.93$. Hence our prediction function is $\hat{y} = 69x/14$. Inverting this relationship, our prediction for the volume of gunpowder needed to fire the ball a distance 7 is $\hat{x} = 7 \times 14/69 = 98/69 \approx 1.420$.

Now suppose we treat y as the predictor and let $f(y) = \alpha x$, so

$$\mathrm{RSS}(\alpha) = (5\alpha - 1)^2 + (11\alpha - 2)^2 + (14\alpha - 3)^2 = 14 - 138\alpha + 342\alpha^2$$

which is minimised when $\alpha = 138/684 \approx 0.202$ so $\hat{x} = 138y/684$. Using this model, the predicted gunpowder volume to fire the ball 7 distance units is $\hat{x} = 161/114 \approx 1.412$. We will see in later chapters that these two approaches correspond to different probabilistic models of the relationship between y and x.

■ ***Exercise*** 1.5 (a) Let $\boldsymbol{U}$ and $\boldsymbol{U}'$ be i.i.d. d-dimensional random vectors with each coordinate selected uniformly and independently at random from $[0, 1]$. The square of the distance between them is the random variable

$$\|\boldsymbol{U} - \boldsymbol{U}'\|^2 = \sum_{i=1}^{d} (U_i - U_i')^2.$$

We are given that the expectation of each $(U_i - U_i')^2$ is μ; thus,

$$\mathbb{E}[\|\boldsymbol{U} - \boldsymbol{U}'\|^2] = \mu d.$$

We are also given that the variance of each $(U_i - U_i')^2$ is σ^2. Since the variance of the sum of independent random variables is the sum of the individual variances, we have

$$\mathrm{Var}[\|\boldsymbol{U} - \boldsymbol{U}'\|^2] = \sigma^2 d,$$

corresponding to a standard deviation of $\sigma\sqrt{d}$.

(b) We can divide all distances by d without affecting distance ratios. The above results imply that the random variable $\|\boldsymbol{U} - \boldsymbol{U}'\|^2/d$, representing an arbitrary pair of points, has mean μ and standard deviation $\sigma/\sqrt{d}$. As we increase d, the standard deviation collapses; thus the ratio of the largest to the smallest inter-point distance approaches 1 as $d \to \infty$.

2

Supervised learning warm-up

■ ***Exercise*** 2.1 The residual sum of squares $\mathrm{RSS}(c; D)$ is a quadratic function of c, and the coefficient of the c^2 term is positive. It therefore has a single stationary point which is the global minimum. We find it by solving $\frac{d\mathrm{RSS}}{dc} = 0$ for c.

Since

$$\frac{d\mathrm{RSS}}{dc} = -2\sum_{i=1}^{n}(y_i - c),$$

the RSS-minimising value of c, which we denote $\hat{c}$, satisfies

$$\sum_{i=1}^{n}(y_i - \hat{c}) = 0.$$

Rearranging, we obtain

$$\hat{c} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

■ ***Exercise*** 2.2 (a) The residual sum of squares $\mathrm{RSS}(\beta; D)$ is a quadratic function of β, and the coefficient of the β^2 term is positive. It therefore has a single stationary point which is the global minimum. We find it by solving $\frac{d\mathrm{RSS}}{d\beta} = 0$ for β.

Since

$$\frac{d\mathrm{RSS}}{d\beta} = -2\sum_{i=1}^{n}(y_i - \beta x_i)x_i,$$

the RSS-minimising value of β, which we denote $\hat{\beta}$, satisfies

$$\sum_{i=1}^{n} x_i(y_i - \hat{\beta}x_i) = 0.$$

Rearranging, we obtain

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}.$$

(b) $\hat{\beta} = 0.63$.

(c) (to do)

■ ***Exercise*** 2.3 Define function f by

$$f(x) = \begin{cases} y & \text{if } (x, y) \in D, \\ -1000 & \text{if there is no } y \text{ such that } (x, y) \in D, \end{cases}$$

where D is the Queen's inventors' dataset. This definition works — i.e., it successfully picks out a function — because dataset D happens not to include any pairs of points of the form (x, y), (x, y') with $y \neq y'$. (Although we referred to D to pick out f, this is not essential: f really does exist as a mathematical function, and we could in principle have specified it without mentioning D.)

Now let SILLY be the singleton function family $\{f\}$. What happens when we look for the element of SILLY that minimises RSS on D, or on some portion of D, or (in fact) on any set of datapoints whatsoever? We get f, because that is the *only* element of SILLY. Since f maps every x value that occurs in D to the co-occurring y value, SILLY's cross-validation error on D is zero. Nevertheless, f maps any x value that does *not* occur in D to the ludicrous y value of -1000.

If you prefer an example in which the 'fitting' procedure is non-trivial, you can instead define the parameterised function $f(-; \theta)$ by

$$f(x; \theta) = \begin{cases} y + \theta & \text{if } (x, y) \in D, \\ -1000 + \theta & \text{if there is no } y \text{ such that } (x, y) \in D. \end{cases}$$

Let SILLY2 be the associated function family, i.e. $\{f(-; \theta) : \theta \in \mathbb{R}\}$. When we look for the element of SILLY2 that minimises RSS on D, or on some portion of D, we obtain $f(-; 0)$ (that is, we find $\hat{\theta} = 0$).

3

Unsupervised learning warm-up

■ ***Exercise*** 3.1 The trial could be defined as follows.

1 Pick a digging location (x, y) uniformly at random from a very large rectangle (large enough to contain all the buried hoards).
2 Dig a deep hole one spade-width wide at this location.
3 If you find a hoard here, return location (x, y) as the result of the trial; halt.
4 Otherwise, go to 1.

(That is a trial one could imagine actually conducting. But if we are content with a purely mathematical definition, matters are simpler. There is some finite set S of buried hoards. Pick a member of S uniformly at random, and return its location (x, y).)

■ ***Exercise*** 3.2 This is a 'classic' integral which is worth knowing how to evaluate. We present the standard method, which involves squaring the integral and changing to polar coordinates. Making the change of variables $z = (x - \mu)/\sigma$ we have

$$\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-z^2/2} dz = \frac{J}{\sqrt{2\pi}},$$

where J is the value of the z-integral. Squaring J we have

$$J^2 = \int_{-\infty}^{\infty} e^{-x^2/2} dx \int_{-\infty}^{\infty} e^{-y^2/2} dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)/2} dx dy.$$

Making the change of variables $x = r \sin\theta$, $y = r \cos\theta$ where $r \in [0, \infty)$ and $\theta \in [0, 2\pi]$ we have

$$J^2 = \int_0^{\infty} \int_0^{2\pi} e^{-r^2/2} r \, dr \, d\theta = 2\pi \left[-e^{-r^2/2}\right]_0^{\infty} = 2\pi.$$

Hence $J = \sqrt{2\pi}$ and

$$\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = \frac{J}{\sqrt{2\pi}} = 1.$$

■ ***Exercise*** 3.3 $L_{\text{prod}}(\theta)$ is the joint probability density function for n i.i.d. samples from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.

■ ***Exercise*** 3.4 We have

$$\ell(\mu, \sigma) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 - n \log \sigma - \frac{n}{2} \log(2\pi).$$

In the degenerate case where $x_i = \mu$ for every i, there is no maximum: $\ell(\mu, \sigma)$ increases without bound as $\sigma \to 0$. We'll assume we're not dealing with the degenerate case.

The partial derivatives are

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu)$$

and

$$\frac{\partial \ell}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} (x_i - \mu)^2.$$

Solving for $\frac{\partial \ell}{\partial \mu} = 0$ and $\frac{\partial \ell}{\partial \sigma} = 0$, we obtain

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2.$$

This is the only stationary point, and from the form of $\ell(\mu, \sigma)$ it is clear it must be a maximum. ($\ell(\mu, \sigma)$ is a continuous function which tends to $-\infty$ when σ becomes very small or very large, and also when μ tends to $+\infty$ or $-\infty$.)

■ ***Exercise*** 3.5 Each of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_3$ is a two-dimensional vector, while σ_1, σ_2, σ_3, w_1 and w_2 are all scalars (one-dimensional). When all these parameters are bundled into a single vector $\boldsymbol{\theta}$, this vector has $3 \times 2 + 5 \times 1 = 11$ components.

■ ***Exercise*** 3.6 Each of $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ is a two-dimensional vector, while $\sigma_1, \ldots, \sigma_k$ and $w_1, \ldots, w_{k-1}$ are all scalars. When all these parameters are bundled into a single vector $\boldsymbol{\theta}_k$, this vector has $2k + k + k - 1 = 4k - 1$ components.

■ ***Exercise*** 3.7 This exercise is answered in the main text.

4

Interlude: probability, likelihood and Bayes

■ ***Exercise*** 4.1 A false alarm would be somewhat surprising, as they occur on only 1% of non-invasion days — but an actual alien invasion would be much more surprising. Intuitively, therefore, it would not be wise to conclude that an invasion is in progress simply on the grounds that the invasion alarm bell is ringing.

■ ***Exercise*** 4.2 The measure of fit we used in Chapter 3 was

$$\ell(\boldsymbol{\theta}; D) = \sum_{\boldsymbol{x} \in D} \log f(\boldsymbol{x}; \boldsymbol{\theta}),$$

where f was a probability density parameterised by $\boldsymbol{\theta}$. Each possible value of $\boldsymbol{\theta}$ corresponds to a hypothesis: namely, that the elements of D are i.i.d. observations from the density $f(-; \boldsymbol{\theta})$. Since we have, equivalently,

$$\ell(\boldsymbol{\theta}; D) = \log \prod_{\boldsymbol{x} \in D} f(\boldsymbol{x}; \boldsymbol{\theta}),$$

and since $\prod_{x \in D} f(\boldsymbol{x}; \boldsymbol{\theta})$ is the conditional probability density of the data given hypothesis $\boldsymbol{\theta}$, we recognise ℓ as the logarithm of the relevant likelihood function.

■ ***Exercise*** 4.3 Given a dataset D, inventing hypotheses whose likelihoods are 100% will always be trivial task. For example, one can hypothesise that an omnipotent deity ensured that the outcome of the experiment must be (precisely) D. The conditional probability of the data *given this silly hypothesis* is 1. And of course one can invent many other hypotheses that have this feature. (Replace omnipotent deity by mysterious alien technology, etc.)

■ ***Exercise*** 4.4 From the definition of conditional probability, we have

$$\mathbf{Pr}(H|D) = \frac{\mathbf{Pr}(H \cap D)}{\mathbf{Pr}(D)}$$

and

$$\mathbf{Pr}(D|H) = \frac{\mathbf{Pr}(H \cap D)}{\mathbf{Pr}(H)}.$$

Together, these imply

$$\mathbf{Pr}(H|D) = \mathbf{Pr}(D|H)\frac{\mathbf{Pr}(H)}{\mathbf{Pr}(D)}.$$

■ ***Exercise*** 4.5 Here we take D to be the observation that our (single) invasion alarm has rung today. Applying the Bayesian updating rule followed by Bayes' theorem, we have

$$\begin{aligned}\mathbf{Pr}'(H_{\text{inv}}) &= \mathbf{Pr}(H_{\text{inv}}|D)\\ &= \mathbf{Pr}(D|H_{\text{inv}})\frac{\mathbf{Pr}(H_{\text{inv}})}{\mathbf{Pr}(D)}.\end{aligned}$$

We know that $\mathbf{Pr}(D|H_{\text{inv}}) = 1$, because the alarm is sure to sound if there is an invasion. We also know that $\mathbf{Pr}(H_{\text{inv}}) = 10^{-6}$, since that was specified as the prior. To calculate $\mathbf{Pr}(D)$, we use the rule of total probability:

$$\begin{aligned}\mathbf{Pr}(D) &= \mathbf{Pr}(D|H_{\text{inv}})P(H_{\text{inv}}) + \mathbf{Pr}(D|H_{\text{null}})P(H_{\text{null}})\\ &= 1 \times 10^{-6} + 0.01 \times (1 - 10^{-6})\\ &\approx 0.01.\end{aligned}$$

Notice that we put $\mathbf{Pr}(D|H_{\text{null}}) = 0.01$ here, since there is a 1% chance that the invasion alarm rings if H_{null} is true. Putting it all together, we have

$$\mathbf{Pr}'(H_{\text{inv}}) \approx 1 \times \frac{10^{-6}}{0.01} = 10^{-4}.$$

■ ***Exercise*** 4.6 Now we suppose that we have fifty independent invasion alarms, and we take D to be the observation that all of them have rung today. The Bayesian analysis and calculation go through as in the previous exercise, except that now $\mathbf{Pr}(D|H_{\text{null}}) = 0.01^{50} = 10^{-100}$, a vanishingly small probability. Thus

$$\begin{aligned}\mathbf{Pr}(D) &= \mathbf{Pr}(D|H_{\text{inv}})P(H_{\text{inv}}) + \mathbf{Pr}(D|H_{\text{null}})P(H_{\text{null}})\\ &= 1 \times 10^{-6} + 10^{-100} \times (1 - 10^{-6})\\ &\approx 10^{-6},\end{aligned}$$

and

$$\mathbf{Pr}'(H_{\text{inv}}) \approx 1 \times \frac{10^{-6}}{10^{-6}} = 1.$$

Fifty independent alarms ringing is such strong evidence that it overwhelms the strong prior against the invasion hypothesis. In the scenario envisaged in this exercise, you should be almost certain that aliens are indeed invading.

5

Probabilistic modelling

■ ***Exercise*** 5.1 (a)

$$\log\left(\prod_{i=1}^{n} a_i\right) = \sum_{i=1}^{n} \log a_i$$

(b)

$$\log\left(\prod_{i=1}^{n} \frac{e^{\frac{a_i}{b}}}{c}\right) = \sum_{i=1}^{n} \log\left(\frac{e^{\frac{a_i}{b}}}{c}\right) = \frac{S}{b} - n\log c$$

■ ***Exercise*** 5.2

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{k=1}^{n} \log f_{Y|X}(y_k|x_k;\boldsymbol{\theta}) \\
&= \sum_{k=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_k - g(x_k;\boldsymbol{\beta}))^2}{2\sigma^2}\right)\right) \\
&= -\frac{1}{2\sigma^2}\sum_{k=1}^{n}(y_k - g(x_k;\boldsymbol{\beta}))^2 - n\log(\sqrt{2\pi}\sigma)
\end{aligned}$$

■ ***Exercise*** 5.3 (a) The parameters of the model are β and σ. We can bundle these two parameters into a single parameter vector $\boldsymbol{\theta} = (\beta, \sigma)$.

(b) We have

$$\mathcal{L}(\boldsymbol{\theta}; D) = f_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x}) = \prod_{k=1}^{n} f_{Y|X}(y_k|x_k;\boldsymbol{\theta}),$$

so

$$\begin{aligned}\ell(\boldsymbol{\theta}; D) = \log \mathcal{L}(\boldsymbol{\theta}; D) &= \sum_{k=1}^{n} \log f_{Y|X}(y_k|x_k; \boldsymbol{\theta}) \\ &= \sum_{k=1}^{n} \log \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k - \beta x_k^2)^2}{2\sigma^2}} \right) \\ &= -\frac{1}{2\sigma^2} \sum_{k=1}^{n} (y_k - \beta x_k^2)^2 - n \log(\sqrt{2\pi}\sigma).\end{aligned}$$

(c) From the form of the log-likelihood, we see that maximising ℓ with respect to β is equivalent to minimising $\mathrm{RSS}(\beta) = \sum_{k=1}^{n} (y_k - \beta x_k^2)^2$. The RSS is a quadratic function of β, and the coefficient of the β^2 term is positive. It therefore has a single stationary point which is the global minimum. We find it by solving $\frac{d\mathrm{RSS}}{d\beta} = 0$ for β. Since

$$\frac{d\mathrm{RSS}}{d\beta} = -2 \sum_{k=1}^{n} (y_k - \beta x_k^2) x_k^2,$$

the RSS-minimising value of β, which we denote $\hat{\beta}$, satisfies

$$\sum_{k=1}^{n} (y_k - \hat{\beta} x_k^2) x_k^2 = 0.$$

Rearranging, we obtain

$$\hat{\beta} = \frac{\sum_{k=1}^{n} y_k x_k^2}{\sum_{k=1}^{n} x_k^4}.$$

(d) $\hat{\beta} \approx 1.058$. (plot to do)

6

Frequentist and Bayesian uncertainty

■ ***Exercise*** 6.1 The random dataset now takes the form $\mathcal{D} = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$.

■ ***Exercise*** 6.2 (a) If $D = \{1\}$, i.e. the single coin toss yielded heads, then the likelihood function is $\mathcal{L}(\theta) = 1$.

(b) If $D = \{0\}$, i.e. the single coin toss yielded tails, then the likelihood function is $\mathcal{L}(\theta) = 0$.

(c) Neither of the constant functions above can be interpreted as a probability density function over possible values of θ, because neither satisfies $\int_{-infty}^{+\infty} \mathcal{L}(\theta) d\theta = 1$.

■ ***Exercise*** 6.3 From the compact definition, we have $\mathbf{Pr}(X = 1; p) = f_X(1; p) = p$ and $\mathbf{Pr}(X = 0; p) = f_X(0; p) = 1 - p$, which matches the original definition of **Pr**.

■ ***Exercise*** 6.4 The assumption that the X_is are independent would be violated.

■ ***Exercise*** 6.5 If $X_1, \ldots, X_n$ are i.i.d. with $X_i \sim \text{Bernoulli}(p)$ for each i, then their sum $S_n = \sum_{i=1}^n X_i$ is binomially distributed, $S_n \sim \text{Binomial}(n, p)$. We have $\mathbb{E}[S_n] = np$ and $\text{Var}[S_n] = np(1 - p)$.

■ ***Exercise*** 6.6 The maximum likelihood estimate is $\hat{p} = \frac{14}{20} = 0.7$. The plug-in estimate of the estimator's standard error is $\sqrt{\frac{\hat{p}(1-\hat{p})}{20}} \approx 0.10$.

■ ***Exercise*** 6.7 We have

$$\begin{aligned} \mathbb{E}[t(\underline{X})] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] \\ &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}[X_i]. \end{aligned}$$

Since $\mathbb{E}[X_i] = p$ for each i, this implies

$$\begin{aligned}\mathbb{E}[t(\underline{X})] &= \frac{1}{n}\sum_{i=1}^{n} p \\ &= p.\end{aligned}$$

In other words, the expected value of the estimator is equal to the true sex ratio. The estimator is *unbiased*.

■ ***Exercise*** 6.8 Since θ lies somewhere on the real line, the three events $\theta < a(\underline{X})$, $\theta > b(\underline{X})$, and $\theta \in [a(\underline{X}), b(\underline{X})]$ are jointly exhaustive. They are also mutually exclusive. Thus, their probabilities must add to 1, and we have

$$\mathbf{Pr}(\theta \in [a(\underline{X}), b(\underline{X})]) = 1 - \mathbf{Pr}(\theta < a(\underline{X})) - \mathbf{Pr}(\theta > b(\underline{X})).$$

If $\mathbf{Pr}(\theta < a(\underline{X})) = \mathbf{Pr}(\theta > b(\underline{X})) = \alpha/2$, then it follows that $\mathbf{Pr}(\theta \in [a(\underline{X}), b(\underline{X})]) = 1 - \alpha$. The converse does not hold, however. We could in principle have an asymmetric $1 - \alpha$ confidence interval, where the probabilities $\mathbf{Pr}(\theta < a(\underline{X}))$ and $\mathbf{Pr}(\theta > b(\underline{X}))$ are unequal (but still sum to α).

■ ***Exercise*** 6.9 The maximum likelihood estimate of the sex ratio is $\hat{p} = \frac{549}{1000} = 0.549$. The plug-in estimate of the estimator's standard error is $\sqrt{\frac{\hat{p}(1-\hat{p})}{1000}} \approx 0.0157$. Using the '$\hat{\theta}(\underline{x}) \pm 2 \times \hat{\mathrm{se}}(\underline{x})$' method of constructing 95% confidence intervals presented in the text, we obtain the approximate 95% confidence interval for p of 0.549 ± 0.031 or, equivalently, $[0.518, 0.580]$.

■ ***Exercise*** 6.10 We have

$$\begin{aligned}\alpha &= \mathbf{Pr}(Z > z_\alpha) \\ &= 1 - \mathbf{Pr}(Z \leq z_\alpha) \\ &= 1 - \Phi(z_\alpha),\end{aligned}$$

so

$$\Phi(z_\alpha) = 1 - \alpha.$$

But since $\Phi : \mathbb{R} \to (0, 1)$ is a monotonic function, it is invertible. The previous equation therefore implies

$$z_\alpha = \Phi^{-1}(1 - \alpha).$$

■ ***Exercise*** 6.11 We are testing

$$H_0 : p < \frac{1}{2} \text{ versus } H_1 : p \geq \frac{1}{2}.$$

This is a one-sided test. We reject H_0 if

$$\frac{\hat{p}(\underline{X}) - \frac{1}{2}}{\hat{\text{se}}} > z_\alpha,$$

where α is the significance level. We have $\hat{p}(\underline{x}) = 0.549$ and $\hat{\text{se}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{1000}} \approx 0.0157$. Thus we reject H_0 if

$$\frac{0.049}{0.0157} \approx 3.114 > z_\alpha.$$

To test at the 5% significance level, we set $\alpha = 0.05$. Since $z_{0.05} = \Phi^{-1}(0.95) \approx 1.645$, we comfortably reject H_0 at this significance level.

The p-value is the strictest significance level at which we would reject H_0. We can find that threshold by solving $3.114 = z_\alpha = \Phi^{-1}(1-\alpha)$ for α. We have $1 - \alpha = \Phi(3.114)$, and so $\alpha = 1 - \Phi(3.114) = 0.00092$, or approximately 0.1‰.

■ ***Exercise*** 6.12

$$\begin{aligned} f_X(0) &= \int_0^1 f_{X,P}(0,p)dp \\ &= \int_0^1 2p(1-p)dp \\ &= \left[p^2 - \frac{2}{3}p^3\right]_0^1 \\ &= \frac{1}{3} \end{aligned}$$

$$\begin{aligned} f_X(1) &= \int_0^1 f_{X,P}(1,p)dp \\ &= \int_0^1 2p^2 dp \\ &= \left[\frac{2}{3}p^3\right]_0^1 \\ &= \frac{2}{3} \end{aligned}$$

■ ***Exercise*** 6.13

$$\begin{aligned} f_X(1) &= \int_0^1 f_{X,P}(1,p)dp \\ &= \int_0^1 f_P(p) f_{X|P}(1|p)dp \\ &= \int_0^1 f_{X|P}(1|p)dp \\ &= \int_0^1 p dp \\ &= \left[\frac{1}{2}p^2\right]_0^1 \\ &= \frac{1}{2} \end{aligned}$$

■ ***Exercise*** 6.14 We have already seen that the posterior is

$$f_{P|X}(p|x_1) = 2p.$$

Since $\hat{p}^{\text{MAP}}$ is the value of p (in $[0,1]$) that maximises this quantity, it is clear that $\hat{p}^{\text{MAP}} = 1$.

Computing $\hat{p}^{\text{PM}}$ requires an integral. We have

$$\begin{aligned} \hat{p}^{\text{PM}} &= \int_0^1 p f_{P|X}(1,p)dp \\ &= \int_0^1 2p^2 dp \\ &= \left[\frac{2}{3}p^3\right]_0^1 \\ &= \frac{2}{3}. \end{aligned}$$

■ ***Exercise*** 6.15 (a) The likelihood $\mathcal{L}(\boldsymbol{p})$ is the probability of observing $\underline{X} = \underline{x}$ according to hypothesis $H_{\boldsymbol{p}}$. That is

$$\mathcal{L}(\boldsymbol{p}) = \prod_{i=1}^{\tilde{n}} p_i^{m_i},$$

so the log-likelihood is

$$\ell(\boldsymbol{p}) = \sum_{i=1}^{\tilde{n}} m_i \log p_i.$$

To prove the final equality, we'll need the definition of the Kullback-Leibler divergence (Appendix C) and a bit of algebraic juggling. Starting from the right-hand side, we have

$$\begin{aligned}\sum_{i=1}^{\tilde{n}} m_i \log m_i - n \log n - n\mathbb{KL}(\boldsymbol{p}^* || \boldsymbol{p}) &= \sum_{i=1}^{\tilde{n}} m_i \log m_i - n \log n - n \sum_{i=1}^{\tilde{n}} p_i^* \log \left(\frac{p_i^*}{p_i} \right) \\ &= \sum_{i=1}^{\tilde{n}} m_i \log m_i - n \log n - \sum_{i=1}^{\tilde{n}} m_i \log \left(\frac{m_i}{n p_i} \right) \\ &= -n \log n + \sum_{i=1}^{\tilde{n}} m_i (\log n + \log p_i) \\ &= \sum_{i=1}^{\tilde{n}} m_i \log p_i,\end{aligned}$$

where in the last step we have used the fact that $\sum_{i=1}^{\tilde{n}} m_i = n$.

(b) Maximising the likelihood is equivalent to maximising the log-likelihood. The only term in the expression for $\ell(\boldsymbol{p})$ that depends on $\boldsymbol{p}$ is the term involving the Kullback-Leibler divergence. To maximise $\ell(\boldsymbol{p})$ with respect to $\boldsymbol{p}$, we seek the value of $\boldsymbol{p}$ that *minimises* $\mathbb{KL}(\boldsymbol{p}^* || \boldsymbol{p})$. As shown in Appendix C, this value is just $\boldsymbol{p}^*$.

■ ***Exercise*** 6.16 An overfitted model mistakes noise for signal and therefore underestimates the variability of the dataset-generating process. If we sample from the overfitted model, the replica datasets we obtain will all look rather like the actual dataset (i.e., they will resemble it more closely than datasets obtained from genuine reruns of the experiment would). Thus, when we use the parametric bootstrap to measure uncertainty in the overfitted model's parameters, we will get false reassurance. The non-parametric bootstrap is potentially more useful in this situation because it lets us estimate sampling distributions in a way that does not rely on the overfitted model.

■ ***Exercise*** 6.17 To implement the parametric bootstrap in the lizard sex ratio case, we would generate replica datasets by drawing samples of size n from Bernoulli($\hat{p}$), where n is the size of the original dataset and $\hat{p}$ is the fraction of male lizards in it. To implement the *non*-parametric bootstrap, we would generate replica datasets by drawing (with replacement) samples of size n from the actual dataset. Since the probability of any particular resampled lizard being male is precisely $\hat{p}$, this second procedure is mathematically equivalent to the first.

As discussed in the text, the plug-in method lets us calculate exactly what the variance of our sex ratio estimator would be if the true fraction of males was $\hat{p}$. Since the exact calculation is straightforward in this particular case, it is a better bet than the computationally expensive and inexact bootstrap procedure. (In practice, though, using the bootstrap would also work fine.)

■ ***Exercise*** 6.18 (a) We have

$$\begin{aligned}\int \frac{\partial \log f_X(x;\theta)}{\partial \theta} f_X(x;\theta)dx &= \int \frac{1}{f_X(x;\theta)} \frac{\partial f_X(x;\theta)}{\partial \theta} f_X(x;\theta)dx \\ &= \int \frac{\partial f_X(x;\theta)}{\partial \theta} dx \\ &= \frac{\partial}{\partial \theta} \int f_X(x;\theta)dx \\ &= \frac{\partial}{\partial \theta} 1 \\ &= 0,\end{aligned}$$

where to obtain the third equality we have assumed that the density family is sufficiently well behaved (see postscript below) that we can swap the order of differentiation with respect to θ and integration with respect to x.

(b) We have already shown that

$$\int \frac{\partial \log f_X(x;\theta)}{\partial \theta} f_X(x;\theta)dx = 0.$$

We now differentiate that equation with respect to θ, again assuming that we can pass the differentiation operator through the integral sign:

$$\int \frac{1}{\partial \theta} \left(\frac{\partial \log f_X(x;\theta)}{\partial \theta} f_X(x;\theta) \right) dx = 0,$$

i.e.

$$\int \left(\frac{\partial^2 \log f_X(x;\theta)}{\partial \theta^2} f_X(x;\theta) + \frac{\partial \log f_X(x;\theta)}{\partial \theta} \frac{\partial f_X(x;\theta)}{\partial \theta} \right) dx = 0.$$

But since $\frac{\partial \log f_X(x;\theta)}{\partial \theta} = \frac{1}{f_X(x;\theta)} \frac{\partial f_X(x;\theta)}{\partial \theta}$ and hence $\frac{\partial f_X(x;\theta)}{\partial \theta} = f_X(x;\theta) \frac{\partial \log f_X(x;\theta)}{\partial \theta}$, we can rewrite the last displayed equation as

$$\int \left(\frac{\partial^2 \log f_X(x;\theta)}{\partial \theta^2} + \left(\frac{\partial \log f_X(x;\theta)}{\partial \theta} \right)^2 \right) f_X(x;\theta)dx = 0,$$

from which the desired result follows immediately.

(c)

$$\begin{aligned}\mathbb{E}(s(X;\theta)) &= \int s(x;\theta) f_X(x;\theta)dx \\ &= \int \frac{\partial \log f_X(x;\theta)}{\partial \theta} f_X(x;\theta)dx \\ &= 0.\end{aligned}$$

$$\begin{aligned}\mathrm{Var}(s(X;\theta)) &= \mathbb{E}\left(s(X;\theta)^2\right)\\ &= \int s(x;\theta)^2 f_X(x;\theta)dx\\ &= \int \left(\frac{\partial \log f_X(x;\theta)}{\partial\theta}\right)^2 f_X(x;\theta)dx\\ &= -\int \frac{\partial^2 \log f_X(x;\theta)}{\partial\theta^2} f_X(x;\theta).\end{aligned}$$

Postscript. In the above solution, we on two occasions moved a differentiation operator through an integral sign. When is this justified? We will state (without proof) one simple sufficient condition. Let $\phi(-;-)$ be a function with two real arguments, where the first takes values anywhere on the real line and the second takes values in some set Θ. Then the equality

$$\frac{\partial}{\partial\theta}\int_{\mathbb{R}} \phi(x;\theta)dx = \int_{\mathbb{R}} \frac{\partial\phi(x;\theta)}{\partial\theta}dx$$

holds if $\frac{\partial\phi}{\partial\theta}$ can be bounded in absolute value by some integrable function of x; that is, if for some integrable function g we have

$$\left|\frac{\partial\phi(x;\theta)}{\partial\theta}\right| \leq g(x)$$

for all $x \in \mathbb{R}$ and $\theta \in \Theta$. (If the partial derivative fails to exist anywhere, we take it that this sufficient condition is not met.) The rough idea here is that interchanging differentiation and integration is safe if you could replace the integral by a large finite sum and still differentiate term-by-term without things going haywire — which is the case as long as the integrand's gradient in θ stays within reasonable bounds.

■ ***Exercise*** 6.19 Since the X_i are i.i.d., we have

$$\begin{aligned}\mathrm{Var}(t(\underline{X})) &= \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(X_i)\\ &= \frac{1}{n^2}n\sigma(\tau)^2\\ &= \frac{1}{n}\sigma(\tau)^2,\end{aligned}$$

where $\sigma(\tau)^2$ is the variance of an exponentially distributed random variable with scale parameter τ. Let X be such a random variable. It is straightforward to verify that

$$\mathbb{E}(X) = \int_0^\infty x\frac{1}{\tau}e^{-\frac{x}{\tau}} = \tau$$

and

$$\mathbb{E}(X^2) = \int_0^\infty x^2 \frac{1}{\tau} e^{-\frac{x}{\tau}} = 2\tau^2.$$

Therefore

$$\sigma(\tau)^2 = \mathrm{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2 = \tau^2,$$

and so

$$\mathrm{Var}(t(\underline{X})) = \frac{\tau^2}{n}.$$

Thus the standard error of estimator $t(\underline{X})$ is

$$\mathrm{se} = \frac{\tau}{\sqrt{n}}.$$

The only difference between this exact formula for the standard error and the approximation derived in the main text is that the latter has $\hat{\tau}$ (which can be calculated from data) in place of τ (which is unknown).

■ ***Exercise*** 6.20 (i) The score function is

$$\begin{aligned} s(X;p) &= \frac{\partial \log f_X(X;p)}{\partial p} \\ &= \frac{\partial}{\partial p}[X \log p + (1-X)\log(1-p)] \\ &= \frac{X}{p} - (1-X)\frac{1}{1-p} \\ &= \frac{1}{1-p}\left(\frac{X}{p} - 1\right). \end{aligned}$$

It is easy to check that $\mathbb{E}(s(X;p) = 0$, as expected. (Recall that in this context all expectations and variances are taken with respect to probability mass function $f_X(-;p)$.) The Fisher information is

$$\begin{aligned} \mathcal{I}(p) = \mathrm{Var}(s(X;p)) &= \mathbb{E}\left(s(X;p)^2\right) \\ &= \frac{1}{(1-p)^2}\mathbb{E}\left(\frac{X^2}{p^2} - \frac{2X}{p} + 1\right) \\ &= \frac{1}{(1-p)^2}\left(\frac{p}{p^2} - 2 + 1\right) \\ &= \frac{1}{(1-p)^2}\left(\frac{1}{p} - 1\right) \\ &= \frac{1}{(1-p)^2}\frac{1-p}{p} = \frac{1}{p(1-p)}. \end{aligned}$$

(ii) Given n observations of i.i.d. random variables $X_1, ..., X_n$ with common distribution Bernoulli(p), the maximum likelihood estimator of p is $\hat{p} = \frac{1}{n}\sum_{i=1}^n X_i$. The normal

approximation to the estimator's standard error is

$$\hat{\text{se}} = \frac{1}{\sqrt{n\mathcal{I}(p)}} = \sqrt{\frac{p(1-p)}{n}}.$$

This matches the exact expression derived in Section 6.4.1. (Here as there, we in practice would plug in $\hat{p}$ for p.)

■ ***Exercise*** 6.21 Since the Bernoulli mass function is

$$f_{X|\Theta}(x|\theta) = \theta^x(1-\theta)^{1-x},$$

we have

$$\begin{aligned}\log f_{X|\Theta}(x|\theta) &= x\log\theta + (1-x)\log(1-\theta)\\ &= x\log\frac{\theta}{1-\theta} + \log(1-\theta).\end{aligned}$$

Exponentiating both sides yields the desired result. We therefore recognise the Bernoulli family of distributions as a one-parameter exponential family with

$$\begin{aligned}h(x) &= 1,\\ \eta(\theta) &= \log\frac{\theta}{1-\theta},\\ T(x) &= x,\\ A(\theta) &= -\log(1-\theta).\end{aligned}$$

■ ***Exercise*** 6.22 Since the probability density function of an exponentially distributed random variable is

$$\begin{aligned}f_{X|\Theta}(x|\theta) &= \theta e^{-\theta x}\\ &= e^{-\theta x + \log\theta},\end{aligned}$$

we have a one-parameter exponential family with

$$\begin{aligned}h(x) &= 1,\\ \eta(\theta) &= -\theta,\\ T(x) &= x,\\ A(\theta) &= -\log\theta.\end{aligned}$$

■ ***Exercise*** 6.23 We have

$$\begin{aligned} f_{X|\Theta}(x|\theta) &= h(x)\exp\left(\eta(\theta)T(x) - A(\theta)\right) \\ &= \binom{N}{x}\exp\left(x\log\frac{\theta}{1-\theta} + N\log(1-\theta)\right) \\ &= \binom{N}{x}\left(\frac{\theta}{1-\theta}\right)^x (1-\theta)^N \\ &= \binom{N}{x}\theta^x(1-\theta)^{N-x}, \end{aligned}$$

which we recognise as the binomial distribution.

■ ***Exercise*** 6.24 The posterior (given a Gamma(1, 1) prior) is

$$\Theta|\{\underline{X} = \underline{x}\} \sim \text{Gamma}\left(1 + \sum_{i=1}^{n} x_i, 1 + n\right).$$

Using the data given, we find $\sum_{i=1}^{n} x_i = 11$ and $n = 10$. Thus

$$\Theta|\{\underline{X} = \underline{x}\} \sim \text{Gamma}(12, 11).$$

The posterior mean estimate is therefore $\hat{\theta} = \frac{12}{11} \approx 1.09$.

■ ***Exercise*** 6.25 As noted in Exercise 6.23, the binomial family for fixed n is a one-parameter exponential family with $\eta(\theta) = \log(\theta/(1-\theta))$ and $A(\theta) = -n\log(1-\theta)$. The conjugate prior is therefore

$$\begin{aligned} f_\Theta(\theta) &\propto \exp\left(\nu\log\left(\frac{\theta}{1-\theta}\right) + \lambda n\log(1-\theta)\right) \\ &= \left(\frac{\theta}{1-\theta}\right)^\nu (1-\theta)^{\lambda n} \\ &= \theta^\nu(1-\theta)^{\lambda n - \nu}, \end{aligned}$$

which we recognise as a beta distribution (see Box 6.6).

7

Frequentist linear regression

■ ***Exercise*** 7.1 Starting from the definition of variance, we have

$$\begin{aligned}
\mathrm{Var}\left(\sum_{i=1}^{n} X_i\right) &= \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left(\sum_{i=1}^{n} X_i\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mathbb{E}(X_i)\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} (X_i - \mathbb{E}(X_i))\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n} (X_i - \mathbb{E}(X_i)) \sum_{j=1}^{n} \left(X_j - \mathbb{E}(X_j)\right)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[(X_i - \mathbb{E}(X_i))\left(X_j - \mathbb{E}(X_j)\right)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}(X_i, X_j) \\
&= \sum_{i=1}^{n} \mathrm{Var}(X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathrm{Cov}(X_i, X_j),
\end{aligned}$$

where in the final equality we have split $\sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}(X_i, X_j)$ into two terms, the first covering cases where $i = j$, and the second the remaining ($i \neq j$) cases.

■ ***Exercise*** 7.2 (a) The likelihood is

$$\begin{aligned}
\mathcal{L}(\beta_0, \beta_1, \sigma) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right] \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2\right].
\end{aligned}$$

(b) The log-likelihood is

$$\ell(\beta_0, \beta_1, \sigma) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2.$$

The values of β_0 and β_1 that maximise the log-likelihood (and hence also the likelihood) are, irrespective of σ, the values that minimise $\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$. We can also write that quantity as $\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)^2$ (this makes the calculus in (c) slightly tidier). Either way, we recognise it as $\mathrm{RSS}(\beta_0, \beta_1)$.

(c) The RSS is an upward curving quadratic function of β_0 and β_1, so any stationary point will be a global minimum. We have

$$\frac{\partial \mathrm{RSS}}{\partial \beta_0} = 2\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i),$$

so the maximum likelihood estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ satisfy

$$\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i - y_i) = 0,$$

or equivalently (dividing by n and rearranging)

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

We also have

$$\frac{\partial \mathrm{RSS}}{\partial \beta_1} = 2\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)x_i,$$

so the maximum likelihood estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ also satisfy

$$\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i - y_i)x_i = 0.$$

Substituting in our expression for $\hat{\beta}_0$ in terms of $\hat{\beta}_1$, we have

$$\sum_{i=1}^{n}(\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i - y_i)x_i = 0,$$

which we can rewrite

$$\sum_{i=1}^{n}\left(\bar{y} - y_i + \hat{\beta}_1(x_i - \bar{x})\right)x_i = 0,$$

or equivalently

$$\sum_{i=1}^{n}\left(\bar{y} - y_i + \hat{\beta}_1(x_i - \bar{x})\right)(x_i - \bar{x}) = 0,$$

where in the last step we have exploited the fact that the sum of deviations from the sample mean is zero. Rearranging the previous equation, we obtain

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

■ ***Exercise*** 7.3 (a) The partial derivatives are

$$\frac{\partial \text{RSS}}{\partial \beta_0} = 2\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)$$

and

$$\frac{\partial \text{RSS}}{\partial \beta_1} = 2\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)x_i.$$

Thus the maximum likelihood estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ satisfy

$$\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i - y_i) = 0$$

and

$$\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i - y_i)x_i = 0.$$

Since for each i we have $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, the last two equations can be written

$$\sum_{i=1}^{n}(\hat{y}_i - y_i) = \sum_{i=1}^{n} x_i(\hat{y}_i - y_i) = 0.$$

It follows that

$$\sum_{i=1}^{n}\hat{y}_i(y_i - \hat{y}_i) = 0,$$

because

$$\begin{aligned}\sum_{i=1}^{n}\hat{y}_i(y_i - \hat{y}_i) &= \sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i)(y_i - \hat{y}_i)\\ &= -\hat{\beta}_0\sum_{i=1}^{n}(\hat{y}_i - y_i) - \hat{\beta}_1\sum_{i=1}^{n}x_i(\hat{y}_i - y_i)\\ &= -\hat{\beta}_0 \times 0 - \hat{\beta}_1 \times 0\\ &= 0.\end{aligned}$$

(b)

$$\begin{aligned}\sum_{i=1}^{n}(y_i-\hat{y}_i)(\hat{y}_i-\bar{y}) &= \sum_{i=1}^{n}\hat{y}_i(y_i-\hat{y}_i)-\bar{y}\sum_{i=1}^{n}(y_i-\hat{y}_i)\\ &= 0-\bar{y}\times 0\\ &= 0.\end{aligned}$$

■ ***Exercise*** 7.4 (a) The Pearson correlation between Y and X is

$$\hat{\rho}=\frac{\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{\sqrt{\sum_{k=1}^{n}(x_k-\bar{x})^2\sum_{k=1}^{n}(y_k-\bar{y})^2}}=\frac{\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{n\hat{\sigma}_x\hat{\sigma}_y},$$

and from equation (7.2) in the main text we have

$$\hat{\beta}_1=\frac{\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{\sum_{k=1}^{n}(x_k-\bar{x})^2}=\frac{\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{n\hat{\sigma}_x^2}.$$

Thus

$$\hat{\beta}_1=\frac{\hat{\rho}\hat{\sigma}_y}{\hat{\sigma}_x}.$$

(b) We have

$$\begin{aligned}\text{RSS} &= \sum_{i=1}^{n}(\hat{y}_i-y_i)^2\\ &= \sum_{i=1}^{n}(\hat{\beta}_0+\hat{\beta}_1x_i-y_i)^2.\end{aligned}$$

But also, from equation (7.3) in the main text,

$$\hat{\beta}_0=\bar{y}-\hat{\beta}_1\bar{x}.$$

Thus

$$\begin{aligned}\text{RSS} &= \sum_{i=1}^{n}(\bar{y}-\hat{\beta}_1\bar{x}+\hat{\beta}_1x_i-y_i)^2\\ &= \sum_{i=1}^{n}\left(\hat{\beta}_1(x_i-\bar{x})-(y_i-\bar{y})\right)^2.\end{aligned}$$

(c)

$$\begin{aligned}
\mathrm{RSS} &= \sum_{i=1}^{n}\left(\frac{\hat{\rho}\hat{\sigma}_y}{\hat{\sigma}_x}(x_i-\bar{x})-(y_i-\bar{y})\right)^2 \\
&= \frac{\hat{\rho}^2\hat{\sigma}_y^2}{\hat{\sigma}_x^2}\sum_{i=1}^{n}(x_i-\bar{x})^2+\sum_{i=1}^{n}(y_i-\bar{y})^2-2\frac{\hat{\rho}\hat{\sigma}_y}{\hat{\sigma}_x}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) \\
&= \frac{\hat{\rho}^2\hat{\sigma}_y^2}{\hat{\sigma}_x^2}n\hat{\sigma}_x^2+n\hat{\sigma}_y^2-2\frac{\hat{\rho}\hat{\sigma}_y}{\hat{\sigma}_x}n\hat{\rho}\hat{\sigma}_x\hat{\sigma}_y \\
&= n\hat{\rho}^2\hat{\sigma}_y^2+n\hat{\sigma}_y^2-2n\hat{\rho}^2\hat{\sigma}_y^2 \\
&= n\hat{\sigma}_y^2(1-\hat{\rho}^2).
\end{aligned}$$

(d)

$$R^2 = 1-\frac{\mathrm{RSS}}{\mathrm{TSS}} = 1-\frac{n\hat{\sigma}_y^2(1-\hat{\rho}^2)}{n\hat{\sigma}_y^2} = 1-(1-\hat{\rho}^2) = \hat{\rho}^2.$$

■ ***Exercise*** 7.5 Here we flesh out the sketch provided in Box 7.3. We have

$$\begin{aligned}
t(\underline{X}) &= c\sum_{i=1}^{n}\left(X_i-\frac{1}{n}\sum_{j=1}^{n}X_j\right)^2 \\
&= c\sum_{i=1}^{n}\left(X_i^2+\frac{1}{n^2}\left(\sum_{j=1}^{n}X_j\right)^2-\frac{2}{n}X_i\sum_{j=1}^{n}X_j\right) \\
&= c\left(\sum_{i=1}^{n}X_i^2+\frac{1}{n}\left(\sum_{j=1}^{n}X_j\right)^2-\frac{2}{n}\left(\sum_{j=1}^{n}X_j\right)^2\right) \\
&= c\left(\sum_{i=1}^{n}X_i^2-\frac{1}{n}\left(\sum_{i=1}^{n}X_i\right)^2\right).
\end{aligned}$$

The expectation of this estimator is

$$\mathbb{E}(t(\underline{X})) = c\left[\sum_{i=1}^{n}\mathbb{E}(X_i^2)-\frac{1}{n}\mathbb{E}\left((\sum_{i=1}^{n}X_i)^2\right)\right].$$

Since any random variable Y with well-defined variance satisfies $\mathbb{E}(Y^2) = \mathrm{Var}(Y) + \mathbb{E}(Y)^2$, for each i we have

$$\mathbb{E}(X_i^2) = \sigma^2+\mu^2.$$

Also

$$\begin{aligned}\mathbb{E}\left((\sum_{i=1}^{n} X_i)^2\right) &= \mathrm{Var}(\sum_{i=1}^{n} X_i) + (n\mu)^2 \\ &= n\sigma^2 + (n\mu)^2,\end{aligned}$$

where to derive the last equality we used the fact that variance is additive for independent random variables. Putting it all together, we obtain

$$\begin{aligned}\mathbb{E}(t(\underline{X})) &= c\left[n(\sigma^2+\mu^2) - \frac{1}{n}(n\sigma^2 + n^2\mu^2)\right] \\ &= c(n-1)\sigma^2.\end{aligned}$$

■ ***Exercise*** 7.6 We have

$$\begin{aligned}\mathbb{E}\left((g(\boldsymbol{x};\boldsymbol{t}(\mathcal{D})) - \mathfrak{g}(\boldsymbol{x}) - \mathcal{E})^2|\mathcal{D}\right) &= (g(\boldsymbol{x};\boldsymbol{t}(\mathcal{D})) - \mathfrak{g}(\boldsymbol{x}))^2 \\ &\quad + \mathbb{E}(\mathcal{E}^2|\mathcal{D}) - 2(g(\boldsymbol{x};\boldsymbol{t}(\mathcal{D})) - \mathfrak{g}(\boldsymbol{x}))\mathbb{E}(\mathcal{E}|\mathcal{D}).\end{aligned}$$

But since test noise is independent of training data, we have

$$\mathbb{E}(\mathcal{E}^2|\mathcal{D}) = \mathbb{E}(\mathcal{E}^2) = \sigma^2, \quad \mathbb{E}(\mathcal{E}|\mathcal{D}) = \mathbb{E}(\mathcal{E}) = 0.$$

Thus

$$\mathbb{E}\left((g(\boldsymbol{x};\boldsymbol{t}(\mathcal{D})) - \mathfrak{g}(\boldsymbol{x}) - \mathcal{E})^2|\mathcal{D}\right) = (g(\boldsymbol{x};\boldsymbol{t}(\mathcal{D})) - \mathfrak{g}(\boldsymbol{x}))^2 + \sigma^2.$$

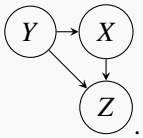
8

Directed graphical models

■ ***Exercise*** 8.1 A third factorisation is

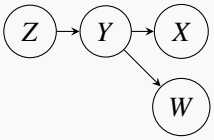
$$f_{X,Y,Z}(x,y,z) = f_{Z|X,Y}(z|x,y) f_{X|Y}(x|y) f_Y(y);$$

the corresponding DAG is



.

■ ***Exercise*** 8.2



■ ***Exercise*** 8.3

■ ***Exercise*** 8.4

$$f_{W,X,Y,Z}(w,x,y,z) = f_{W|Z}(w|z) f_{Z|X,Y}(z|x,y) f_Y(y) f_X(x)$$

■ ***Exercise*** 8.5 (a) Starting from the definition of conditional probability, we have

$$\begin{aligned} f_{X|Y,Z}(x|y,z) &= \frac{f_{X,Y,Z}(x,y,z)}{f_{Y,Z}(y,z)} \\ &= \frac{f_{X,Y|Z}(x,y|z)f_Z(z)}{f_{Y,Z}(y,z)} \\ &= \frac{f_{X,Y|Z}(x,y|z)}{f_{Y,Z}(y,z)/f_Z(z)} \\ &= \frac{f_{X,Y|Z}(x,y|z)}{f_{Y|Z}(y|z)}. \end{aligned}$$

Alternatively, one can observe that for any fixed z, $f_{X,Y|Z}(x,y|z)$ is a well-defined joint probability density for X and Y. The desired result is then just the definition of the associated conditional probability density of X given Y.

(b) From the result in (a), we have

$$f_{X,Y|Z}(x,y|z) = f_{X|Y,Z}(x|y,z)f_{Y|Z}(y|z).$$

Therefore, if $f_{X|Y,Z}(x|y,z) = f_{X|Z}(x|z)$, it follows that

$$f_{X,Y|Z}(x,y|z) = f_{X|Z}(x|z)f_{Y|Z}(y|z).$$

(c) Substituting $f_{X|Z}(x|z)f_{Y|Z}(y|z)$ for $f_{X,Y|Z}(x,y|z)$ in the result proved in (a) yields

$$f_{X|Y,Z}(x|y,z) = f_{X|Z}(x|z).$$

■ ***Exercise*** 8.6 (a) We need to show that the node sets X and Y are d-separated by the conditioning set W. There are only two undirected paths from X to Y, one via W and the other via Z. The path via W contains the fork $X \leftarrow W \rightarrow Y$. Since $W \in W$, that path is blocked (i.e., is d-separated) by the conditioning set. The path via Z contains the collider $X \rightarrow Z \leftarrow Y$. Since neither Z not any of its descendents lies in W, that path is also blocked. Thus X and Y are indeed d-separated by W, and we conclude that $X \perp\!\!\!\perp Y|W$.

(b) We need to show that the node sets W and Z are d-separated by the conditioning set X, Y. There are only two undirected paths from W to Z, one via X and the other via Y. Both paths contain a pipe whose middle node lies in X, Y, which established d-separation.

(c) We need to show that the node sets V and Y are d-separated by the empty conditioning set $\emptyset$. That means we need to show that every undirected path from V to Y contains a collider. There are only two undirected paths from V to Y, one via W and the other via Z. The path via W contains the collider $V \rightarrow X \leftarrow W$. The path via Z contains the collider $X \rightarrow Z \leftarrow Y$ — so we are done.

9

Bayesian linear regression, priors, and regularisation

■ ***Exercise*** 9.1 (a) $\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} = \underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})^{-1}\underline{\boldsymbol{x}}^T\underline{y} = \underline{\boldsymbol{x}}^T\underline{y}$

(b) We begin by expanding the scalar product,

$$\begin{aligned}(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta})^T(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta}) &= \underline{y}^T\underline{y} - \underline{y}^T\underline{\boldsymbol{x}}\boldsymbol{\beta} - \boldsymbol{\beta}^T\underline{\boldsymbol{x}}^T\underline{y} + \boldsymbol{\beta}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\boldsymbol{\beta} - 2\underline{y}^T\underline{\boldsymbol{x}}\boldsymbol{\beta} + \underline{y}^T\underline{y},\end{aligned}$$

where in the second equality we have used the fact that $\underline{y}^T\underline{\boldsymbol{x}}\boldsymbol{\beta}$ is a scalar, and hence equal to its transpose $\boldsymbol{\beta}^T\underline{\boldsymbol{x}}^T\underline{y}$. Next we use the multivariate 'completing the square' identity from Box 8.2, with $\boldsymbol{\beta}$, $\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{x}}^T\underline{y}$ serving as z, $\underline{\boldsymbol{S}}$ and $\boldsymbol{b}$, respectively. That gives us

$$(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta})^T(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta}) = (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) - \underline{y}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} + \underline{y}^T\underline{y}.$$

But from (a), we have $\hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{y} = \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}$. Since this quantity is a scalar, we also have $\underline{y}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}$. We can therefore write the scalar product as

$$(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta})^T(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta}) = (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + \underline{y}^T\underline{y} - \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}.$$

(c) Expanding the right-hand side of the equation we wish to prove gives

$$\begin{aligned}(\underline{y} - \underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}})^T(\underline{y} - \underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}) &= \underline{y}^T\underline{y} + \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} - 2\underline{y}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} \\ &= \underline{y}^T\underline{y} - \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}},\end{aligned}$$

where in the second equality we have used the fact established in (b) that $\underline{y}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}$. Using this result, we can rewrite the scalar product in (b) as

$$(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta})^T(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta}) = (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + (\underline{y} - \underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}})^T(\underline{y} - \underline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}),$$

which is the desired identity.

■ ***Exercise*** 9.2 Starting from the standard form of the multivariate normal density, we

have

$$\begin{aligned}\mathcal{N}(\boldsymbol{\beta}|\hat{\boldsymbol{\beta}}, \sigma^2(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})^{-1}) &= \frac{1}{(2\pi)^{\frac{p+1}{2}}\sqrt{\det(\sigma^2(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})^{-1})}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right) \\ &= \frac{1}{(2\pi)^{\frac{p+1}{2}}\sqrt{\sigma^{2(p+1)}/\det(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right) \\ &= \frac{\sqrt{\det(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})}}{(2\pi)^{\frac{p+1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^T\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right),\end{aligned}$$

where in the second equality we have made use of the properties of determinants mentioned in the question. We can therefore rewrite equation (9.14) as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \sigma) &= \frac{1}{(2\pi)^{n/2}\sigma^n}\exp\left(-\frac{\nu s^2}{2\sigma^2}\right)\frac{(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}}{\sqrt{\det(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})}}\mathcal{N}(\boldsymbol{\beta}|\hat{\boldsymbol{\beta}}, \sigma^2(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})^{-1}) \\ &= \frac{1}{(2\pi)^{\nu/2}\sigma^{\nu}\sqrt{\det(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})}}\exp\left(-\frac{\nu s^2}{2\sigma^2}\right)\mathcal{N}(\boldsymbol{\beta}|\hat{\boldsymbol{\beta}}, \sigma^2(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}})^{-1}),\end{aligned}$$

which matches equation (9.15). (We used the definition $\nu = n - p - 1$ in the final step.)

■ ***Exercise*** 9.3 The densities referred to in this solution are conditional densities with $\underline{Y}$ and $\underline{\boldsymbol{X}}$ given. To keep notation simple, we will suppress the conditions, writing e.g. $f_\Sigma(\sigma)$ rather than $f_{\Sigma|\underline{Y},\underline{\boldsymbol{X}}}(\sigma|\underline{y},\underline{\boldsymbol{x}})$.

Applying the change of variable technique described in Appendix A, we have

$$\begin{aligned}f_Z(z) &= f_\Sigma(\sqrt{z})\frac{d}{dz}(\sqrt{z}) \\ &= cz^{-\frac{\nu+1}{2}}\exp\left(-\frac{\nu s^2}{2z}\right)\frac{1}{2}z^{-1/2} \\ &= \frac{c}{2}z^{-(\frac{\nu}{2}+1)}\exp\left(-\frac{\nu s^2}{2z}\right).\end{aligned}$$

■ ***Exercise*** 9.4 We have

$$\begin{aligned}f_{\Sigma|\underline{Y},\underline{\boldsymbol{X}}}(\sigma|\underline{y},\underline{\boldsymbol{x}}) &= 2\sigma\mathrm{IG}\left(\sigma^2\middle|\frac{\nu}{2}, \frac{\nu s^2}{2}\right) \\ &= 2\sigma\frac{\left(\frac{\nu s^2}{2}\right)^{\nu/2}}{\Gamma(\nu/2)}(\sigma^2)^{-(1+\frac{\nu}{2})}\exp\left(-\frac{\nu s^2}{2\sigma^2}\right) \\ &= \frac{2\left(\frac{\nu s^2}{2}\right)^{\nu/2}}{\Gamma(\nu/2)}\sigma^{-(\nu+1)}\exp\left(-\frac{\nu s^2}{2\sigma^2}\right).\end{aligned}$$

Comparing this to equation (9.19), we obtain an expression for the normalising constant

c in the latter:

$$c = \frac{2\left(\frac{\nu s^2}{2}\right)^{\nu/2}}{\Gamma(\nu/2)}.$$

■ ***Exercise*** 9.5 We begin by noting that the numerator in the exponent of equation (9.25) can be written

$$(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta})^T(\underline{y} - \underline{\boldsymbol{x}}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\underline{\boldsymbol{I}}_\epsilon\boldsymbol{\beta} = \boldsymbol{\beta}^T(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)\boldsymbol{\beta} - 2\underline{y}^T\underline{\boldsymbol{x}}\boldsymbol{\beta} + \underline{y}^T\underline{y}.$$

Next we use the multivariate 'completing the square' identity from Box 8.2, with $\boldsymbol{\beta}$, $\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon$ and $\underline{\boldsymbol{x}}^T\underline{y}$ serving as $\boldsymbol{z}$, $\underline{\boldsymbol{S}}$ and $\boldsymbol{b}$, respectively. That lets us rewrite the expression above as

$$\left(\boldsymbol{\beta} - (\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)^{-1}\underline{\boldsymbol{x}}^T\underline{y}\right)^T(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)\left(\boldsymbol{\beta} - (\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)^{-1}\underline{\boldsymbol{x}}^T\underline{y}\right) + \text{const},$$

where the term denoted 'const' has no dependence on $\boldsymbol{\beta}$. It follows that the density (9.25) is multivariate normal with inverse covariance matrix $\underline{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\sigma^2}(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)$, hence $\underline{\boldsymbol{\Sigma}} = \sigma^2(\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}} + \lambda\underline{\boldsymbol{I}}_\epsilon)^{-1}$, and mean $\boldsymbol{\mu} = \frac{1}{\sigma^2}\underline{\boldsymbol{\Sigma}}\underline{\boldsymbol{x}}^T\underline{y}$.

10

Bayesian methods

■ ***Exercise*** 10.1 For density estimation, normalisation requires that

$$\begin{aligned} c &= \int \mathcal{L}(\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int f_{\underline{\boldsymbol{X}}|\boldsymbol{\Theta}}(\underline{\boldsymbol{x}}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int f_{\boldsymbol{\Theta},\underline{\boldsymbol{X}}}(\boldsymbol{\theta}, \underline{\boldsymbol{x}}) d\boldsymbol{\theta} \\ &= f_{\underline{\boldsymbol{X}}}(\underline{\boldsymbol{x}}). \end{aligned}$$

For regression or classification the reasoning is similar, but the likelihood $\mathcal{L}(\boldsymbol{\theta})$ is now $f_{\underline{Y}|\underline{\boldsymbol{X}},\boldsymbol{\Theta}}(\underline{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})$ rather than $f_{\underline{\boldsymbol{X}}|\boldsymbol{\Theta}}(\underline{\boldsymbol{x}}|\boldsymbol{\theta})$. The result is

$$c = \int f_{\boldsymbol{\Theta},\underline{\boldsymbol{Y}}|\underline{\boldsymbol{X}}}(\boldsymbol{\theta}, \underline{\boldsymbol{y}}|\underline{\boldsymbol{x}}) d\boldsymbol{\theta} = f_{\underline{\boldsymbol{Y}}|\underline{\boldsymbol{X}}}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}}).$$

■ ***Exercise*** 10.2 The factorisation implied by the DAG is

$$f_{X,P,A,B}(x, p, \alpha, \beta) = f_{X|P}(x|p) f_{P|A,B}(p|\alpha, \beta) f_A(\alpha) f_B(\beta).$$

By the definition of conditional probability, we therefore have

$$f_{X,P|A,B}(x, p|\alpha, \beta) = \frac{f_{X|P}(x|p) f_{P|A,B}(p|\alpha, \beta) f_A(\alpha) f_B(\beta)}{f_{A,B}(\alpha, \beta)}.$$

But the DAG also implies that A and B are independent, i.e. that $f_{A,B}(\alpha, \beta) = f_A(\alpha) f_B(\beta)$. The previous equation therefore reduces to

$$f_{X,P|A,B}(x, p|\alpha, \beta) = f_{X|P}(x|p) f_{P|A,B}(p|\alpha, \beta).$$

■ ***Exercise*** 10.3 Since S is a Markov process, for every positive integer m we have

$$f_{\boldsymbol{X}_m|\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{m-1}}(\boldsymbol{x}_m|\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{m-1}) = f_{\boldsymbol{X}_m|\boldsymbol{X}_{m-1}}(\boldsymbol{x}_m|\boldsymbol{x}_{m-1}).$$

This is the Markov property. Using it, the factorisation given in the exercise simplifies

as follows:

$$\begin{aligned} f_{\boldsymbol{X}_0,\ldots,\boldsymbol{X}_n}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_n) &= f_{\boldsymbol{X}_n|\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{n-1}}(\boldsymbol{x}_n|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{n-1}) f_{\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{n-1}}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{n-1}) \\ &= f_{\boldsymbol{X}_n|\boldsymbol{X}_{n-1}}(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f_{\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{n-1}}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{n-1}). \end{aligned}$$

But we can repeat the trick, because the final factor in the above equation ($f_{\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{n-1}}$) factorises in a similar way. We therefore have a recursion, which bottoms out at

$$f_{\boldsymbol{X}_0,\ldots,\boldsymbol{X}_n}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_n) = f_{\boldsymbol{X}_0}(\boldsymbol{x}_0) \prod_{k=0}^{n-1} f_{\boldsymbol{X}_{k+1}|\boldsymbol{X}_k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k).$$

■ ***Exercise*** 10.4 Suppose first that $f^*_{\boldsymbol{X}}(\boldsymbol{x}) > f^*_{\boldsymbol{X}}(\boldsymbol{y})$. Then the acceptance probability rule in Box 10.4 gives

$$a(\boldsymbol{x},\boldsymbol{y}) = \frac{f^*_{\boldsymbol{X}}(\boldsymbol{y})}{f^*_{\boldsymbol{X}}(\boldsymbol{x})}$$

and

$$a(\boldsymbol{y},\boldsymbol{x}) = 1.$$

Consequently,

$$a(\boldsymbol{y},\boldsymbol{x}) f^*_{\boldsymbol{X}}(\boldsymbol{y}) = a(\boldsymbol{x},\boldsymbol{y}) f^*_{\boldsymbol{X}}(\boldsymbol{x}) = f^*_{\boldsymbol{X}}(\boldsymbol{y}).$$

Now suppose instead that $f^*_{\boldsymbol{X}}(\boldsymbol{x}) \leq f^*_{\boldsymbol{X}}(\boldsymbol{y})$. Then the acceptance probability rule in Box 10.4 gives

$$a(\boldsymbol{x},\boldsymbol{y}) = 1$$

and

$$a(\boldsymbol{y},\boldsymbol{x}) = \frac{f^*_{\boldsymbol{X}}(\boldsymbol{x})}{f^*_{\boldsymbol{X}}(\boldsymbol{y})}.$$

Consequently,

$$a(\boldsymbol{y},\boldsymbol{x}) f^*_{\boldsymbol{X}}(\boldsymbol{y}) = a(\boldsymbol{x},\boldsymbol{y}) f^*_{\boldsymbol{X}}(\boldsymbol{x}) = f^*_{\boldsymbol{X}}(\boldsymbol{x}).$$

■ ***Exercise*** 10.5 If the current location is $\boldsymbol{x}$ and the step size is b, then the candidate random vector is

$$\boldsymbol{Y} = \boldsymbol{x} + \boldsymbol{Z}_b,$$

where $\boldsymbol{Z}_b \sim \mathcal{N}(\boldsymbol{0}, b^2\underline{\boldsymbol{I}})$. We need to show that if $f^*_{\boldsymbol{X}}$ is continuous, then for any $\epsilon > 0$,

$$\lim_{b\to 0} \mathbf{Pr}\left(\frac{f^*_{\boldsymbol{X}}(\boldsymbol{x}+\boldsymbol{Z}_b)}{f^*_{\boldsymbol{X}}(\boldsymbol{x})} > 1-\epsilon\right) = 1.$$

To do that, two observations suffice. First, for any $\delta > 0$,

$$\lim_{b\to 0} \mathbf{Pr}(|\boldsymbol{Z}_b| < \delta) = 1.$$

Second, continuity of $f^*_{\boldsymbol{X}}$ implies that for any given $\epsilon > 0$, a $\delta > 0$ can be fixed such that $|\boldsymbol{Z}_b| < \delta$ implies $|f^*_{\boldsymbol{X}}(\boldsymbol{x} + \boldsymbol{Z}_b) - f^*_{\boldsymbol{X}}(\boldsymbol{x})| < \epsilon f^*_{\boldsymbol{X}}(\boldsymbol{x})$.

■ ***Exercise*** 10.6 (a) We have

$$\mathbb{E}(S) = \mathbb{E}\left(\sum_{i=1}^{N} \Delta_i\right) = \sum_{i=1}^{N} \mathbb{E}(\Delta_i) = 0.$$

Since by definition $\mathrm{Var}(S) = \mathbb{E}(S^2) - \mathbb{E}(S)^2$, the previous result implies that $\mathrm{Var}(S) = \mathbb{E}(S^2)$. Finally, since $\Delta_1, \Delta_2, \ldots, \Delta_N$ are i.i.d., we have

$$\mathrm{Var}(S) = \sum_{i=1}^{N} \mathrm{Var}(\Delta_i) = Nb^2.$$

(b) Passing now to the multidimensional case, we have

$$\begin{aligned}
\mathbb{E}(\boldsymbol{S}^T\boldsymbol{S}) &= \mathbb{E}\left[\left(\sum_{i=1}^{N} \boldsymbol{\Delta}_i^T\right)\left(\sum_{j=1}^{N} \boldsymbol{\Delta}_j\right)\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{E}(\boldsymbol{\Delta}_i^T\boldsymbol{\Delta}_j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} db^2\delta_{ij} = \sum_{i=1}^{N} db^2 = Ndb^2.
\end{aligned}$$

■ ***Exercise*** 10.7 Using Bienaymé's identity and the fact that

$$\mathrm{Cov}(\Theta_i, \Theta_j) = \rho(j-i)\mathrm{Var}(\Theta),$$

we have

$$\begin{aligned}
\mathrm{Var}\left(n^{-1}\sum_{i=1}^{n} \Theta_i\right) &= n^{-2}\,\mathrm{Var}\left(\sum_{i=1}^{n} \Theta_i\right) \\
&= n^{-2}\left[n\mathrm{Var}(\Theta) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathrm{Cov}(\Theta_i, \Theta_j)\right] \\
&= n^{-2}\,\mathrm{Var}(\Theta)\left[n + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \rho(j-i)\right].
\end{aligned}$$

But

$$\begin{aligned}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\rho(j-i) &= \sum_{i=1}^{n-1}\sum_{\tau=1}^{n-i}\rho(\tau)\\ &= \sum_{\tau=1}^{n-1}\sum_{i=1}^{n-\tau}\rho(\tau)\\ &= \sum_{\tau=1}^{n-1}(n-\tau)\rho(\tau),\end{aligned}$$

where in the second equality we have flipped the order of the two summations. Plugging this result into our previous expression for $\mathrm{Var}\left(n^{-1}\sum_{i=1}^{n}\Theta_i\right)$, we obtain

$$\begin{aligned}\mathrm{Var}\left(n^{-1}\sum_{i=1}^{n}\Theta_i\right) &= n^{-2}\,\mathrm{Var}(\Theta)\left[n+2\sum_{\tau=1}^{n-1}(n-\tau)\rho(\tau)\right]\\ &= \frac{\mathrm{Var}(\Theta)}{n}\left[1+2\sum_{\tau=1}^{n-1}(1-\frac{\tau}{n})\rho(\tau)\right].\end{aligned}$$

■ ***Exercise*** 10.8 We can write the desired variance as

$$\begin{aligned}\mathrm{Var}(X_A - X_B) &= \mathrm{Cov}(X_A - X_B, X_A - X_B)\\ &= \mathrm{Cov}(X_A, X_A) + \mathrm{Cov}(X_B, X_B) - 2\mathrm{Cov}(X_A, X_B)\\ &= \sigma_A^2 + \sigma_B^2 - 2\mathrm{Cov}(X_A, X_B).\end{aligned}$$

But by the definition of correlation,

$$\rho = \frac{\mathrm{Cov}(X_A, X_B)}{\sigma_A\sigma_B},$$

and so $\mathrm{Cov}(X_A, X_B) = \rho\sigma_A\sigma_B$. Thus

$$\mathrm{Var}(X_A - X_B) = \sigma_A^2 + \sigma_B^2 - 2\rho\sigma_A\sigma_B.$$

■ ***Exercise*** 10.9 The probability that R lies between r and $r+dr$ while Ψ lies between ψ and $\psi + d\psi$ is $f_{R,\Psi}(r,\psi)drd\psi$. But using the given normal density and the fact that the region we have described has area $rd\psi dr$, this probability is also

$$\frac{e^{-\frac{r^2}{2\sigma^2}}}{2\pi\sigma^2}rd\psi dr.$$

It follows that

$$f_{R,\Psi}(r,\psi) = \frac{re^{-\frac{r^2}{2\sigma^2}}}{2\pi\sigma^2},$$

and that the marginal density of R is

$$f_R(r) = \int_0^{2\pi} f_{R,\Psi}(r,\psi)d\psi = \frac{r}{\sigma^2}e^{-\frac{r^2}{2\sigma^2}},$$

i.e. $R \sim \text{Rayleigh}(\sigma)$.

■ ***Exercise*** 10.10 If $R \sim \text{Rayleigh}(\sigma)$, then

$$\begin{aligned}
\mathbb{E}(R) &= \int_0^\infty \frac{r^2}{\sigma^2}e^{-\frac{r^2}{2\sigma^2}}dr \\
&= \sqrt{2\pi}\sigma\frac{1}{\sqrt{2\pi}\sigma}\int_0^\infty \frac{r^2}{\sigma^2}e^{-\frac{r^2}{2\sigma^2}}dr \\
&= \frac{\sqrt{2\pi}}{\sigma}\frac{1}{2}\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^\infty r^2 e^{-\frac{r^2}{2\sigma^2}}dr \\
&= \frac{\sqrt{2\pi}}{\sigma}\frac{1}{2}\sigma^2 \\
&= \sqrt{\frac{\pi}{2}}\sigma.
\end{aligned}$$

■ ***Exercise*** 10.11 Since $R \sim \text{Rayleigh}(\sigma)$, we have

$$f_R(r) = \frac{r}{\sigma^2}e^{-\frac{r^2}{2\sigma^2}}.$$

We now wish to change variables to $Z = R^2$. Applying the change of variables technique described in Appendix A, we find

$$\begin{aligned}
f_Z(z) &= f_R(z^{1/2})\frac{d}{dz}z^{1/2} \\
&= \frac{z^{1/2}}{\sigma^2}e^{-\frac{z}{2\sigma^2}}\frac{1}{2}z^{-1/2} \\
&= \frac{1}{2\sigma^2}e^{-\frac{z}{2\sigma^2}}.
\end{aligned}$$

We recognise this as an exponential density. Specifically, $Z \sim \text{Exponential}\left(\frac{1}{2\sigma^2}\right)$.

■ ***Exercise*** 10.12 If $X \sim \text{Gamma}(a,b)$, then $\mathbb{E}(X) = \frac{a}{b}$ and $\text{Var}(X) = \frac{a}{b^2}$. The standard deviation is of course the square root of the variance. We therefore have

- M_0: prior mean 2, prior standard deviation $\sqrt{2} \approx 1.41$
- M_1: prior mean 2, prior standard deviation $\sqrt{20} \approx 4.47$
- Λ: prior mean 2, prior standard deviation $\sqrt{40} \approx 6.32$

■ ***Exercise*** 10.13 (a) Using Box 9.3, we have $\mu = a/b$ and $\sigma^2 = a/b^2$. These formulae imply

$$a = \frac{\mu^2}{\sigma^2}, \quad b = \frac{\mu}{\sigma^2}.$$

(b) Box 9.3 also tells us that $m = (a-1)/b$. Combining that with the variance formula, $\sigma^2 = a/b^2$, we deduce

$$\frac{\sigma^2}{m^2} = \frac{a}{(a-1)^2},$$

or equivalently

$$a^2 - \left(\frac{m^2}{\sigma^2} + 2\right) a + 1 = 0.$$

The roots of this quadratic are

$$a = 1 + \frac{m^2}{2\sigma^2}\left(1 \pm \sqrt{1 + \frac{4\sigma^2}{m^2}}\right).$$

However, we know that $a > 1$ (because the mode is non-zero). We can therefore discard one of the roots, concluding that

$$a = 1 + \frac{m^2}{2\sigma^2}\left(1 + \sqrt{1 + \frac{4\sigma^2}{m^2}}\right)$$

and

$$b = \frac{a-1}{m} = \frac{m}{2\sigma^2}\left(1 + \sqrt{1 + \frac{4\sigma^2}{m^2}}\right).$$

■ ***Exercise*** 10.14 The likelihood function is

$$\mathcal{L}(\lambda) = \prod_{j=1}^{n_i} \left(\lambda \exp(-\lambda z_{ij})\right) = \lambda^{n_i} \exp\left(-\lambda \sum_{j=1}^{n_i} z_{ij}\right).$$

Combining this with the $\Lambda \sim \text{Gamma}(a, b)$ prior and recalling that

$$\text{posterior} \propto \text{prior} \times \text{likelihood},$$

we deduce that

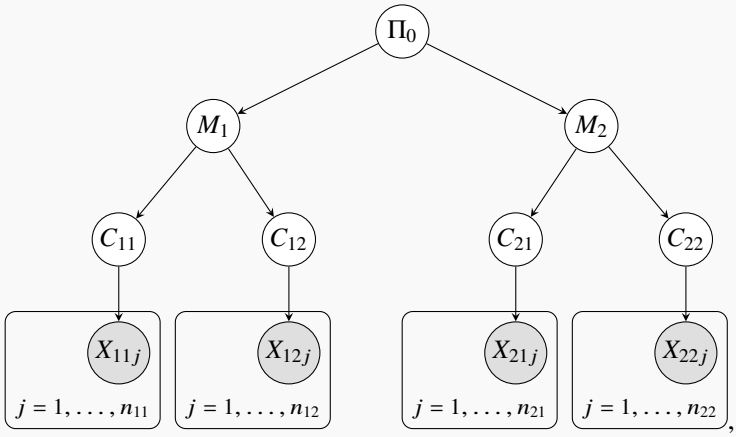
$$f_{\Lambda|\underline{Z}}(\lambda|\underline{z}_i) \propto \lambda^{a-1} \exp(-b\lambda) \times \lambda^{n_i} \exp\left(-\lambda \sum_{j=1}^{n_i} z_{ij}\right)$$

$$= \lambda^{n_i+a-1} \exp\left(-\lambda\left(\sum_{j=1}^{n_i} z_{ij} + b\right)\right).$$

We recognise this as the density of a Gamma $(n_i + a, n_i\bar{z}_i + b)$-distributed random variable (see Box 9.3).

■ ***Exercise*** 10.15 When $i \neq j$, all paths between $\{\underline{R}_i, M_i\}$ and $\{\underline{R}_j, M_j\}$ go via a fork whose middle node is A or B. Since both A and B are members of $\{A, B\}$, the d-separation theorem from Box 8.2 implies that $\{\underline{R}_i, M_i\} \perp\!\!\!\perp \{\underline{R}_j, M_j\} | \{A, B\}$.

■ ***Exercise*** 10.16 We'll represent the reliability of the ith manufacturer by latent random variable M_i, the reliability of that manufacturer's jth car model by latent random variable C_{ij}, and the reliability of the kth individual example of that model by random variable X_{ijk}. Suppose for concreteness that there are two manufacturers and that each manufacturer makes two models of car. The DAG is then



,

where Π_0 is a latent random variable representing the parameters of the distribution from which the manufacturer reliabilities M_i are drawn. (Π_0 plays the role (A, B) played in figure 10.19.)

11

Classification

■ ***Exercise*** 11.1 The cross-entropy loss $l(g; D)$ can never be negative because

$$\sum_{(\boldsymbol{x},y)\in D} \left(y \log g\left(\boldsymbol{x}\right) + (1-y) \log\left(1 - g(\boldsymbol{x})\right)\right)$$

can never be positive (i.e., can never exceed zero). Indeed, for each $(\boldsymbol{x}, y)$, we have $\log g(\boldsymbol{x}) \leq 0$ and $\log(1 - g(\boldsymbol{x})) \leq 0$, because both $g(\boldsymbol{x})$ and $1 - g(\boldsymbol{x})$ are probabilities. Since y is either 0 or 1, it follows that

$$y \log g\left(\boldsymbol{x}\right) + (1-y) \log\left(1 - g(\boldsymbol{x})\right) \leq 0.$$

The cross-entropy loss is precisely zero iff $g(\boldsymbol{x}) = y$ for every $(\boldsymbol{x}, y)$ in D, i.e. iff the classifier always assigns 100% probability to the correct class label. (We adopt the convention that $0 \times \log(0) = \lim_{x\to 0^+} x \log x = 0$.)

■ ***Exercise*** 11.2 We have

$$\begin{aligned}\mathbb{E}(l(g; \mathcal{D})) &= -\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\left(Y_i \log g\left(\boldsymbol{X}_i\right) + (1-Y_i) \log\left(1 - g(\boldsymbol{X}_i)\right)\right)\right] \\ &= -\frac{1}{n} \times n\mathbb{E}\left[\left(Y \log g\left(\boldsymbol{X}\right) + (1-Y) \log\left(1 - g(\boldsymbol{X})\right)\right)\right],\end{aligned}$$

which does not depend on n.

■ ***Exercise*** 11.3 Using natural (base e) logarithms and recording results to two decimal places, the log odds corresponding to the given probabilities are

p	$\text{logit}(p)$
10^{-12}	-27.63
0.01	-4.60
0.52	0.08
0.53	0.12

So a change in probability from 10^{-12} to 1% corresponds to a much greater change in the log odds than a change in probability from 52% to 53%.

■ ***Exercise*** 11.4 From equation (11.8) and the definition of the σ function in equation (11.7), we have

$$g(\boldsymbol{x}; \beta_0, \boldsymbol{\beta}) = \frac{1}{1 + e^{-(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})}}.$$

From that we deduce (after a little algebraic juggling) that

$$1 - g(\boldsymbol{x}; \beta_0, \boldsymbol{\beta}) = \frac{1}{1 + e^{\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}}}.$$

Plugging these results into equation (11.3), and remembering that $\log(1/x) = -\log(x)$, we obtain equation (11.9).

■ ***Exercise*** 11.5 (a) Figure S11.1 shows sketch plots of

$$g(x; \beta_0, \beta_1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

against x for two cases, $\beta_1 > 0$ and $\beta_1 < 0$.

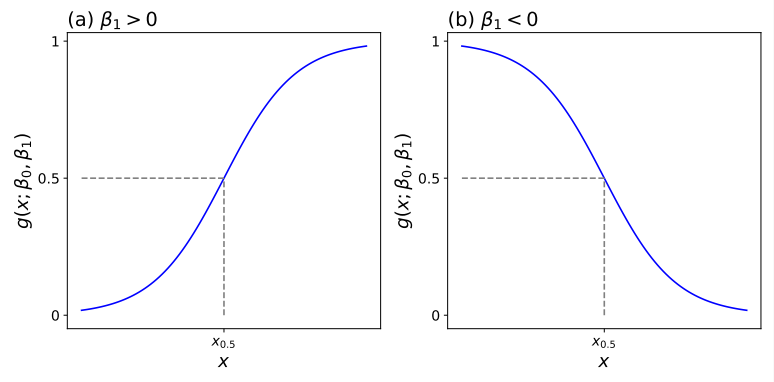


Figure S11.1: Sigmoid functions. The value of x for which the predicted probability is 0.5 is denoted $x_{0.5}$.

From the equation above we see that $g(x; \beta_0, \beta_1) = 0.5$ if and only if $\beta_0 + \beta_1 x = 0$. So assuming $\beta_1 \neq 0$, we have

$$x_{0.5} = -\frac{\beta_0}{\beta_1}.$$

(b) If $\beta_1 = 0$ then $g(x; \beta_0, 0) = \frac{1}{1+e^{-\beta_0}}$ is a constant function, i.e. the predicted probability does not depend on x.

(c) The log-odds is given by

$$\text{logit}\left(g(x; \beta_0, \beta_1)\right) = \beta_0 + \beta_1 x.$$

Thus the change in x required to increase the log-odds by one is

$$\delta_x = \frac{1}{\beta_1}.$$

Notice that δ_x has the same sign as β_1.

(d) The required probabilities are

$$g(x_{0.5} + \delta_x; \beta_0, \beta_1) = \frac{1}{1 + e^{-1}} \approx 0.731$$

and

$$g(x_{0.5} - \delta_x; \beta_0, \beta_1) = \frac{1}{1 + e} \approx 0.269,$$

as shown graphically in Figure S11.2.

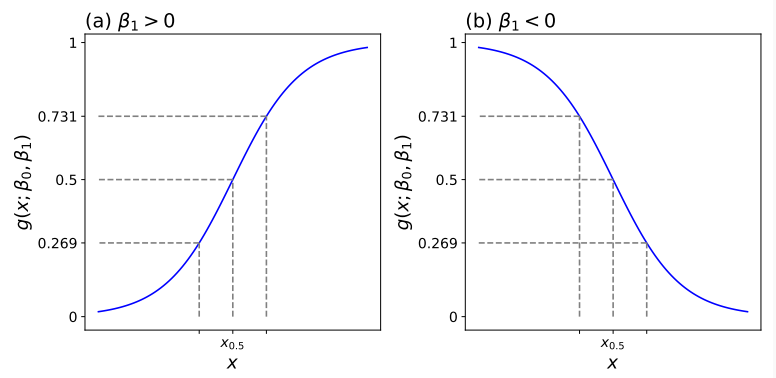


Figure S11.2: Sigmoid functions. The value of x for which the predicted probability is 0.5 is denoted $x_{0.5}$. The log-odds here is zero. Flanking tick marks show values of x corresponding to log-odds of -1 and 1.

■ ***Exercise*** 11.6

$$l(\beta_0, \beta_1; D) = \frac{1}{n} \sum_{(x,y) \in D} \left(y \log(1 + e^{-(\beta_0 + \beta_1 x)}) + (1 - y) \log(1 + e^{\beta_0 + \beta_1 x}) \right).$$

■ ***Exercise*** 11.7 If the initial predicted probability is p and the log-odds increase by Δ, the new predicted probability is

$$\begin{aligned} p' &= \frac{1}{1 + e^{-\left(\log \frac{p}{1-p} + \Delta\right)}} \\ &= \frac{p e^{\Delta}}{1 + p(e^{\Delta} - 1)}. \end{aligned}$$

Plugging $p = 0.4$ into this formula, we find that

(a) $\Delta = 1.08$ implies $p' \approx 0.663$, and
(b) $\Delta = 0.037$ implies $p' \approx 0.409$.

■ ***Exercise*** 11.8 Following the hint in the question, we note that

$$\begin{aligned}\exp\left(-nl^{\lambda}\right) &= \exp(-\lambda\beta_1^2)\exp\left(\sum_{(x,y)\in D}\left(y\log g(x;\beta_0,\beta_1) + (1-y)\log(1-g(x;\beta_0,\beta_1))\right)\right)\\ &= \exp(-\lambda\beta_1^2)\prod_{(x,y)\in D} g(x;\beta_0,\beta_1)^y(1-g(x;\beta_0,\beta_1))^{1-y}.\end{aligned}$$

Now, the product

$$\prod_{(x,y)\in D} g(x;\beta_0,\beta_1)^y(1-g(x;\beta_0,\beta_1))^{1-y}$$

is just the likelihood $\mathcal{L}(\beta_0,\beta_1;D)$, while $\exp(-\lambda\beta_1^2)$ is proportional to a Gaussian prior density on β_1 – specifically, a zero-mean prior with variance $\frac{1}{2\lambda}$. Maximising $\exp\left(-nl^{\lambda}\right)$ is therefore equivalent to doing maximum a posteriori inference with that particular prior.

■ ***Exercise*** 11.9 It would mean that there exists a $\boldsymbol{\beta}$ such that, for some scalar c, the hyperplane defined by the equation

$$\boldsymbol{\beta}^T\boldsymbol{x} = c$$

perfectly separates the two classes.

■ ***Exercise*** 11.10 Unregularised cross-entropy is a pure measure of predictive performance. Classifiers trained by minimising unregularised cross-entropy may overfit to their training data, meaning that their predictive performance is much worse on validation data than on training data. Moreover, when training data is linearly separable, the problem of minimising unregularised cross-entropy becomes ill-posed. To reduce the risk of overfitting and to ensure well-posed optimisation problems, we can use a *regularised* cross-entropy loss in training. However, there are no analogous reasons for including regularisation penalties in validation scores. When we are assessing a trained classifier's performance on validation data, pure predictive performance is what we care about.

■ ***Exercise*** 11.11 (a) In the $\lambda \to \infty$ limit the $\frac{\lambda}{n}\|\boldsymbol{\beta}\|^2$ term (which scales with λ) dominates the $l(\beta_0,\boldsymbol{\beta};D)$ term (which does not). So for any fixed β_0, we have

$$\lim_{\lambda\to\infty}\arg\min_{\boldsymbol{\beta}}\left[\frac{\lambda}{n}\|\boldsymbol{\beta}\|^2 + l(\beta_0,\boldsymbol{\beta};D)\right] = \mathbf{0}.$$

The desired result follows from this.

(b) The predicted probability $g(\boldsymbol{x}; \hat{\beta}_0, \mathbf{0})$ does not depend on $\boldsymbol{x}$; it is the same for every example in the training set. So, rather than first computing $\hat{\beta}_0 = \arg\min_{\beta_0} l(\beta_0, \mathbf{0}; D)$ and then using this to compute $g = g(\boldsymbol{x}; \hat{\beta}_0, \mathbf{0})$, we will just directly find the probability g that minimises

$$\begin{aligned} l(g; D) &= -\frac{1}{n} \sum_{(\boldsymbol{x}, y) \in D} (y \log g + (1 - y) \log(1 - g)) \\ &= -\pi \log g - (1 - \pi) \log(1 - g). \end{aligned}$$

It is clear from the second equality above that $l(g; D)$ is a convex (upward curving) function of g. Thus, any stationary point will be the unique global minimum. Differentiating, we find

$$l'(g; D) = -\frac{\pi}{g} + \frac{1 - \pi}{1 - g}.$$

Solving $l'(g; D) = 0$ therefore yields

$$g = \pi.$$

■ ***Exercise*** 11.12 The components of $\mathbb{E}(\boldsymbol{Y})$ are the base rates of the classes. That is,

$$\mathbb{E}(\boldsymbol{Y})_i = \mathbb{E}(Y_i)$$

is the probability that an individual drawn from the relevant population (i.e., a realisation of random vector $(\boldsymbol{X}, \boldsymbol{Y})$) belongs to the ith class.

The components of $\mathbb{E}(\boldsymbol{Y} | \boldsymbol{X} = \boldsymbol{x})$ are the conditional probabilities of the classes given $\boldsymbol{X} = \boldsymbol{x}$.

■ ***Exercise*** 11.13 Consider one term in the sum in equation (11.15). It is, for some particular $(\boldsymbol{x}, \boldsymbol{y})$ in D,

$$\boldsymbol{y}^T \log \boldsymbol{g}(\boldsymbol{x}) = \sum_{j=0}^{K-1} y_j \log g_j(\boldsymbol{x}) = \log g_y(\boldsymbol{x}),$$

where y (without boldface or subscript) denotes the unique integer $j \in \{0, \ldots, K-1\}$ for which $y_j = 1$; in other words, y is the integer class label corresponding to the one-hot vector $\boldsymbol{y}$. Equation (11.15) is therefore equivalent to equation (11.14).

■ ***Exercise*** 11.14 From the definition of softmax, we have

$$\sum_{i=0}^{K-1} \text{softmax}(\boldsymbol{s})_i = \frac{\sum_{i=0}^{K-1} e^{s_i}}{\sum_{j=0}^{K-1} e^{s_j}} = 1.$$

Moreover, it follows from the strict positivity of the exponential function that every com-

ponent of $\text{softmax}(\boldsymbol{s})$ is a non-negative number. Thus $\text{softmax}(\boldsymbol{s})$ is indeed a probability vector.

■ ***Exercise*** 11.15 We wish to show that for any scalar c,

$$\text{softmax}(\boldsymbol{s} + c\mathbf{1}) = \text{softmax}(\boldsymbol{s}),$$

where $\mathbf{1}$ is the K-dimensional 'all ones' vector. So, let i be an arbitrary component index in $\{0, \ldots, K-1\}$. We have

$$\begin{aligned}\text{softmax}(\boldsymbol{s} + c\mathbf{1})_i &= \frac{e^{s_i+c}}{\sum_{j=0}^{K-1} e^{s_j+c}} \\ &= \frac{e^c e^{s_i}}{e^c \sum_{j=0}^{K-1} e^{s_j}} \\ &= \frac{e^{s_i}}{\sum_{j=0}^{K-1} e^{s_j}} = \text{softmax}(\boldsymbol{s})_i.\end{aligned}$$

Since i was arbitrary, this establishes the desired result.

■ ***Exercise*** 11.16 The log odds between classes i and j is

$$\begin{aligned}\log\left(\frac{g_i(\boldsymbol{x}; \boldsymbol{\beta_0}, \underline{\boldsymbol{\beta}})}{g_j(\boldsymbol{x}; \boldsymbol{\beta_0}, \underline{\boldsymbol{\beta}})}\right) &= \log\left(\frac{\text{softmax}(\boldsymbol{\beta}_0 + \underline{\boldsymbol{\beta}}\,\boldsymbol{x})_i}{\text{softmax}(\boldsymbol{\beta}_0 + \underline{\boldsymbol{\beta}}\,\boldsymbol{x})_j}\right) \\ &= \log\left(\frac{e^{(\boldsymbol{\beta_0}+\underline{\boldsymbol{\beta}}\boldsymbol{x})_i}}{e^{(\boldsymbol{\beta_0}+\underline{\boldsymbol{\beta}}\boldsymbol{x})_j}}\right) \\ &= (\boldsymbol{\beta_0} + \underline{\boldsymbol{\beta}}\boldsymbol{x})_i - (\boldsymbol{\beta_0} + \underline{\boldsymbol{\beta}}\boldsymbol{x})_j.\end{aligned}$$

The change in log odds associated with a change $\boldsymbol{\Delta x}$ in $\boldsymbol{x}$ is therefore

$$(\underline{\boldsymbol{\beta}}\boldsymbol{\Delta x})_i - (\underline{\boldsymbol{\beta}}\boldsymbol{\Delta x})_j = \sum_l (\beta_{il} - \beta_{jl})\Delta x_l.$$

In particular, a unit increase in x_k (with other components of $\boldsymbol{x}$ held constant) corresponds to a shift in the log odds between classes i and j of $\beta_{ik} - \beta_{jk}$.

■ ***Exercise*** 11.17 (a) Consider an arbitrary point on the line segment joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. It can be written

$$\boldsymbol{x} = \alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$$

for some scalar $\alpha \in [0, 1]$. The ith component of the score vector at this point is

therefore

$$\begin{aligned} s_i(\boldsymbol{x}) &= \beta_0^{(i)} + \boldsymbol{\beta}^{(i)T}(\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2) \\ &= \alpha(\beta_0^{(i)} + \boldsymbol{\beta}^{(i)T}\boldsymbol{x}_1) + (1-\alpha)(\beta_0^{(i)} + \boldsymbol{\beta}^{(i)T}\boldsymbol{x}_2) \\ &= \alpha s_i(\boldsymbol{x}_1) + (1-\alpha)s_i(\boldsymbol{x}_2). \end{aligned}$$

In other words, scores for points on the line segment interpolate linearly between the scores at the end points. It follows that if

$$\arg\max_i s_i(\boldsymbol{x}_1) = \arg\max_i s_i(\boldsymbol{x}_2) = j,$$

then also

$$\arg\max_i s_i(\boldsymbol{x}) = j$$

for every $\boldsymbol{x}$ on the line segment joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. This shows that logistic regression decision regions are convex.

(b) At any point $\boldsymbol{x}$ lying on the boundary between the ith and jth decision regions, the ith and jth components of the score vector must be equal. That is

$$\beta_0^{(i)} + \boldsymbol{\beta}^{(i)T}\boldsymbol{x} = \beta_0^{(j)} + \boldsymbol{\beta}^{(j)T}\boldsymbol{x},$$

and therefore

$$(\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}^{(j)})^T\boldsymbol{x} = \beta_0^{(j)} - \beta_0^{(i)}.$$

When the feature space is two-dimensional, this implies that every point on the boundary between the ith and jth decision regions lies on the same line. By way of reasoning similar to that in (a), it is easy to show that the set of such boundary points must be convex. That means the boundary must be a line segment, a half line, or a line.

(c) In general, in a p-dimensional feature space, the equation we wrote down in (b) implies that every point on a boundary between two classes lies on a specific hyperplane. Hence, the boundary is a convex subset of that hyperplane. (In fact it is a *convex polytope*.)

■ ***Exercise*** 11.18 (a) No. It is clear from Figure 11.6 that no partitioning of the plane into convex decision regions (see previous exercise for definition of convexity) can possibly achieve 100% accuracy.

(b) Consider a multinomial prediction function $g(-; \boldsymbol{\beta_0}, \underline{\boldsymbol{\beta}})$ with $\boldsymbol{\beta_0}$ and $\underline{\boldsymbol{\beta}}$ chosen to achieve 100% accuracy on the simultaneously linearly separable dataset. Now consider making the updates

$$\begin{aligned} \boldsymbol{\beta_0} &\leftarrow c\boldsymbol{\beta_0}, \\ \underline{\boldsymbol{\beta}} &\leftarrow c\underline{\boldsymbol{\beta}}, \end{aligned}$$

where $c > 1$ is a scalar. The result is that for every $\boldsymbol{x}$, the score vector $\boldsymbol{\beta}_0 + \underline{\boldsymbol{\beta}}\,\boldsymbol{x}$ is inflated

by a factor of c. This does not affect decision boundaries (because $\arg\max_i s_i(\boldsymbol{x})$ remains the same for all $\boldsymbol{x}$), but it does sharpen the transitions in predicted probability as decision boundaries are traversed. It therefore reduces the unregularised cross-entropy loss. Since we can repeat this procedure ad infinitum, the unregularised cross-entropy loss minimisation problem is ill posed.

■ ***Exercise*** 11.19 Consider a training point that is taking its turn as a prediction target during cross validation. Suppose the point is an outlier in the sense that none of its nearest neighbours in the training folds match its class. A k-NN classifier fitted to the training folds will assign *zero* probability to the correct class in this case. However well the classifier does with the other points, the cross-validated cross-entropy loss will be infinite.

When working with large datasets and challenging classification tasks, the situation envisaged here – infinite cross-validated cross-entropy loss for k-NN – is not a rare edge case: it is the norm, because just one outlier point is enough to throw a spanner in the works. Cross-validated cross-entropy loss is therefore not a useful performance metric for k-NN.

■ ***Exercise*** 11.20 A k-NN classifier with k set too high *underfits* the training data.

■ ***Exercise*** 11.21 k-NN decision regions are *not* constrained to be connected. A decision region corresponding to a particular class could indeed consist of two or more disconnected blobs.

■ ***Exercise*** 11.22 Dialect is a nominal (as opposed to ordinal) categorical variable, meaning that there is no natural order on its six levels.

12

Unsupervised learning: a deeper dive

■ ***Exercise*** 12.1 The sum on the left-hand side of the equation we wish to prove can be rewritten as follows:

$$\begin{aligned}\sum_{i,j\in C_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= \sum_{i,j\in C_k} \|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k) - (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k)\|^2 \\ &= \sum_{i,j\in C_k} \left[\|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)\|^2 + \|(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k)\|^2 - 2(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k)\right] \\ &= 2|C_k| \sum_{i\in C_k} \|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)\|^2 - 2\sum_{i\in C_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T \sum_{j\in C_k} (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k) \\ &= 2|C_k| \sum_{i\in C_k} \|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)\|^2,\end{aligned}$$

where in the final step we have used the fact that

$$\sum_{j\in C_k} (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k) = \sum_{j\in C_k} \boldsymbol{x}_j - |C_k|\bar{\boldsymbol{x}}_k = 0.$$

It follows that

$$\frac{1}{2|C_k|} \sum_{i,j\in C_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{i\in C_k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|^2.$$

■ ***Exercise*** 12.2 We have

$$\begin{aligned}\mathbb{E}(U_i^2) &= \int_{-s/2}^{s/2} \frac{1}{s} u_i^2 du_i \\ &= \frac{1}{s}\left[\frac{1}{3}u_i^3\right]_{-s/2}^{s/2} \\ &= \frac{s^2}{12}.\end{aligned}$$

Passing now to the random vector, we have

$$\begin{aligned}\mathbb{E}\left(\|\boldsymbol{U}\|^2\right) &= \mathbb{E}\left(\sum_{i=1}^{p} U_i^2\right) \\ &= \sum_{i=1}^{p} \mathbb{E}(U_i^2) \\ &= \frac{ps^2}{12}.\end{aligned}$$

■ ***Exercise*** 12.3 As noted in the text,

$$f_X(x) \equiv f_X(x;\boldsymbol{\theta}),$$

and in general we have

$$f_{X|\boldsymbol{Z}}(x|\boldsymbol{z}) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \sigma_k^2)^{z_k}.$$

In the case specified in the question, $z_k = 1$ and $z_i = 0$ for all $i \neq k$. The previous displayed equation then reduces to

$$f_{X|\boldsymbol{Z}}(x|\boldsymbol{z}) = \mathcal{N}(x|\mu_k, \sigma_k^2).$$

Moreover, in this same special case, $f_Z(\boldsymbol{z}) = \pi_k$. Putting it all together, we have

$$\frac{f_{X|\boldsymbol{Z}}(x|\boldsymbol{z}) f_Z(\boldsymbol{z})}{f_X(x)} = \frac{\pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)}{f_X(x;\boldsymbol{\theta})}.$$

■ ***Exercise*** 12.4 If we treat K as a parameter alongside $\pi_1, \ldots, \pi_K$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and $\sigma_1, \ldots, \sigma_K$ and attempt to find a set of parameter values that maximises the likelihood, we will find that the likelihood can be increased without bound by setting $K = n$, placing one Gaussian blob on each training data point (say, $\boldsymbol{\mu}_i = \boldsymbol{x}_i$ for $i = 1, \ldots, n$), and shrinking every σ_i towards zero. Such models will perform disastrously on validation data, because they assign negligible probability density to points that are not in the training set.

■ ***Exercise*** 12.5 We derive the elementwise equivalent of equation (12.17) as follows:

$$\begin{aligned}[\underline{\boldsymbol{C}}]_{ij} &= \frac{1}{n}[\underline{\boldsymbol{x}}^T\underline{\boldsymbol{x}}]_{ij} \\ &= \frac{1}{n}\sum_{k=1}^{n}[\underline{\boldsymbol{x}}^T]_{ik}[\underline{\boldsymbol{x}}]_{kj} \\ &= \frac{1}{n}\sum_{k=1}^{n}[\underline{\boldsymbol{x}}]_{ki}[\underline{\boldsymbol{x}}]_{kj} \\ &= \frac{1}{n}\sum_{k=1}^{n}x_{ki}x_{kj}.\end{aligned}$$

■ ***Exercise*** 12.6 (a) Figure S12.1 covers part (a) and part (c) of this exercise.

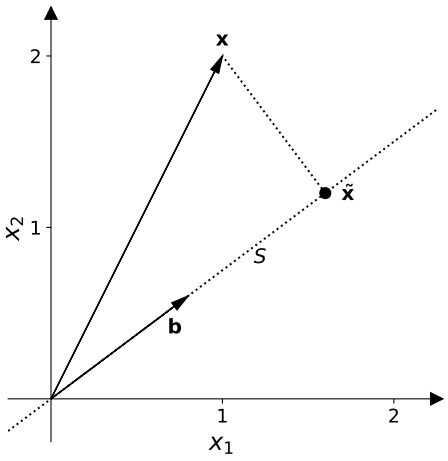


Figure S12.1: Orthogonal projection of the point $\boldsymbol{x}$ onto the one-dimensional subspace spanned by the unit vector $\boldsymbol{b}$.

(b)

$$\underline{\boldsymbol{P}} = \begin{pmatrix} 0.64 & 0.48 \\ 0.48 & 0.36 \end{pmatrix}$$

(c)

$$\tilde{\boldsymbol{x}} = (1.6, 1.2)^T$$

■ ***Exercise*** 12.7 (a) The entry in row i and column j of $\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}}$ is

$$[\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}}]_{ij} = \sum_{k=1}^{p} B_{ki}B_{kj},$$

which is the inner product of the ith and jth columns of $\underline{\boldsymbol{B}}$. Since the columns of $\underline{\boldsymbol{B}}$ are the basis vectors of an ONB, this inner product is one if $i = j$ and zero otherwise. So

$$[\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}}]_{ij} = \delta_{ij},$$

or equivalently

$$\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}} = \underline{\boldsymbol{I}},$$

where $\underline{\boldsymbol{I}}$ is the $M \times M$ identity matrix.

(b) We have

$$\begin{aligned}\tilde{\boldsymbol{x}}^T(\boldsymbol{x} - \tilde{\boldsymbol{x}}) &= (\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x})^T(\boldsymbol{x} - \underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x}) \\ &= \boldsymbol{x}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T(\boldsymbol{x} - \underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x}) \\ &= \boldsymbol{x}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x} - \boldsymbol{x}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x} \\ &= \boldsymbol{x}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x} - \boldsymbol{x}^T\underline{\boldsymbol{B}}\underline{\boldsymbol{B}}^T\boldsymbol{x} \\ &= 0,\end{aligned}$$

where in the penultimate equality we have used the result proved in (a).

■ ***Exercise*** 12.8 We have

$$\begin{aligned}\|\boldsymbol{u} - \boldsymbol{v}\|^2 &= \|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2 - 2\boldsymbol{v}^T\boldsymbol{u} \\ &= \|\boldsymbol{u}\|^2 - \|\boldsymbol{v}\|^2 + 2\|\boldsymbol{v}\|^2 - 2\boldsymbol{v}^T\boldsymbol{u} \\ &= \|\boldsymbol{u}\|^2 - \|\boldsymbol{v}\|^2 + 2\boldsymbol{v}^T(\boldsymbol{v} - \boldsymbol{u}),\end{aligned}$$

where in the last step we have used the fact that $\|\boldsymbol{u}\|^2 = \boldsymbol{v}^T\boldsymbol{v}$.

In the special case where $\boldsymbol{u} = \boldsymbol{x}_i$ and $\boldsymbol{v} = \tilde{\boldsymbol{x}}_i$, this becomes

$$\begin{aligned}\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|^2 &= \|\boldsymbol{x}_i\|^2 - \|\tilde{\boldsymbol{x}}_i\|^2 + 2\tilde{\boldsymbol{x}}_i^T(\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i) \\ &= \|\boldsymbol{x}_i\|^2 - \|\tilde{\boldsymbol{x}}_i\|^2,\end{aligned}$$

since $\tilde{\boldsymbol{x}}_i^T(\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i) = 0$ by the previous exercise.

■ ***Exercise*** 12.9 This is a simple consequence of the fact that $\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}} = \underline{\boldsymbol{I}}$ (see Exercise 12.7). For we have

$$\begin{aligned}\|\tilde{\boldsymbol{x}}\|^2 &= \|\underline{\boldsymbol{B}}\boldsymbol{z}\|^2 \\ &= \boldsymbol{z}^T\underline{\boldsymbol{B}}^T\underline{\boldsymbol{B}}\boldsymbol{z} \\ &= \boldsymbol{z}^T\underline{\boldsymbol{I}}\boldsymbol{z} \\ &= \boldsymbol{z}^T\boldsymbol{z} = \|\tilde{\boldsymbol{z}}\|^2.\end{aligned}$$

13

Neural networks and deep learning

■ ***Exercise*** 13.1 The product function $(x_1, x_2) \mapsto x_1 x_2$ cannot be computed by a single neuron. To see this, notice first that the output of a single neuron with two inputs is

$$y = \phi(b + w_1 x_1 + w_2 x_2)$$

for some activation function $\phi : \mathbb{R} \to \mathbb{R}$, some bias b and some weights w_1 and w_2. Now observe what happens when we try to pick ϕ, b, w_1 and w_2 so that

$$\phi(b + w_1 x_1 + w_2 x_2) = x_1 x_2$$

for all x_1 and x_2. By considering the case $x_1 = 0$, we conclude that

$$\phi(b + w_2 x_2) = 0$$

must hold for all x_2. So either ϕ is the zero function, in which case the neuron's output does not depend on either of its inputs, or $w_2 = 0$, in which case the output does not depend on x_2. Neither possibility is consistent with the desired behaviour of $y = x_1 x_2$ for all x_1 and x_2.

■ ***Exercise*** 13.2 We have

$$\begin{aligned}
\mathrm{TSS}(\boldsymbol{b}, \underline{\boldsymbol{W}}; D) &= n\,\mathrm{MSE}(\boldsymbol{b}, \underline{\boldsymbol{W}}; D) \\
&= \sum_{k=1}^{n} \|\boldsymbol{y}_k - \boldsymbol{b} - \underline{\boldsymbol{W}}\boldsymbol{x}_k\|^2 \\
&= \sum_{k=1}^{n} \sum_{i=1}^{m} \left(y_{ki} - b_i - (\underline{\boldsymbol{W}}\boldsymbol{x}_k)_i\right)^2 \\
&= \sum_{i=1}^{m} \sum_{k=1}^{n} \left(y_{ki} - b_i - (\underline{\boldsymbol{W}}\boldsymbol{x}_k)_i\right)^2,
\end{aligned}$$

where y_{ki} is the ith component of the kth response. The final expression above is the sum of m residual sum of squares (RSS) terms, one for each component of the prediction target.

■ ***Exercise*** 13.3 A network consisting of a single layer of m sigmoid neurons is

equivalent to m single-sigmoid-neuron networks in parallel — with each neuron hooked up to the same inputs $x_1, \ldots, x_p$. The output of each sigmoid neuron is a number in $(0, 1)$, so it can be interpreted as a probability. However, there is no reason why the m output probabilities should sum to 1, so we cannot interpret them as the probabilities of mutually exclusive and jointly exhaustive events. This sort of network, then, is not appropriate for multiclass classification, but might be appropriate for a task in which probabilities need to be assigned to events that are neither mutually exclusive nor jointly exhaustive.

■ ***Exercise*** 13.4 We begin by redrawing the network in Figure 13.6b to show its tunable parameters: four biases (b_1, b_2 and b_3 in the hidden layer and b in out output neuron) and three weights (w_1, w_2 and w_3).

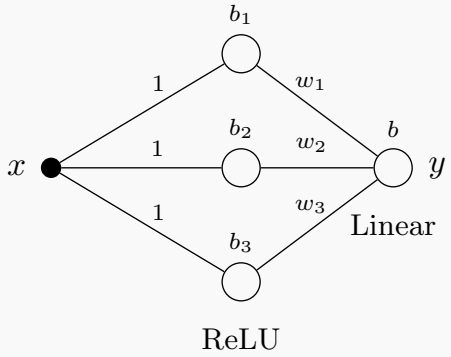


Figure S13.1: The network in Figure 13.6b redrawn with additional annotations.

It is easy to verify that the network computes the desired function if we set the tunable parameters as follows:

- $b_1 = 0$, $w_1 = 1$;
- $b_2 = -0.25$, $w_2 = -1$;
- $b_3 = -0.75$, $w_3 = -1$;
- $b = 0$.

This solution is not unique. For example, for any $\beta > 0$, we could instead set $b_1 = \beta$ and $b = -\beta w_1$, leaving the other parameters as above.

■ ***Exercise*** 13.5 We'll denote the weight of the connection from the input to the ith hidden unit $w_i^{(1)}$, the weight of the connection from the ith hidden unit to the output $w_i^{(2)}$, the bias of the ith hidden unit b_i, the bias of the output unit b, and the number of hidden units p.

The ith hidden unit has preactivation

$$z_i = b_i + w_i^{(1)} x,$$

which is also its activation since the activation function is linear (i.e., is the identity function). The network's output is therefore

$$\begin{aligned} y &= b + \sum_{i=1}^{p} w_i^{(2)} z_i \\ &= b + \sum_{i=1}^{p} w_i^{(2)} (b_i + w_i^{(1)} x). \end{aligned}$$

We obtain precisely the same dependence of y on x from a network consisting of a single linear neuron with bias $b + \sum_{i=1}^{p} w_i^{(2)} b_i$ and connection weight $\sum_{i=1}^{p} w_i^{(2)} w_i^{(1)}$. Thus, a single linear neuron can express any function that can be expressed by the network with a hidden layer of linear units. Moreover, the converse holds too. That is, if the bias b_{sing} and connection weight w_{sing} of a single linear neuron are specified, it will always be possible to replicate the function computed by this neuron using a linear hidden-layer network with $p \geq 1$ hidden units. The simplest (though not the only) way to do that is to exploit only one of the hidden units. For example, we can put $w_1^{(1)} = 1$, $w_i^{(1)} = 0$ for $i \neq 1$, $b_i = 0$ for all i, $w_1^{(2)} = w_{\text{sing}}$, and $b = b_{\text{sing}}$. (Weights $w_i^{(2)}$ for $i \neq 1$ then have no effect on the output, and can be set to any values we like.)

■ ***Exercise*** 13.6 The functions we define below both involve 'unruly behaviour' in a limit.

(a) The function $f : [0, \infty) \to \mathbb{R}$ defined by $f(x) = x^2$ gets steeper and steeper without bound as x increases, since $f'(x) = 2x$. Let $\hat{f}$ be a piecewise linear function intended to approximate f. Because $\hat{f}$ has only finitely many linear pieces, it has a largest slope. Thus $f(x) - \hat{f}(x) \to \infty$ as $x \to \infty$.

(b) The function $g : (0, 1] \to \mathbb{R}$ defined by $g(x) = 1/x$ becomes larger and larger without bound as $x \to 0$. Let $\hat{g}$ be a piecewise linear function intended to approximate g. Because $\hat{g}$ has only finitely many linear pieces, $\lim_{x \to 0} \hat{g}(x)$ is finite. Thus $g(x) - \hat{g}(x) \to \infty$ as $x \to 0$.

■ ***Exercise*** 13.7 Any continuous function f from a closed, bounded region $A \subset \mathbb{R}^p$ to the space of probability vectors over m categories can be represented as the composition of a continuous function $f_{\text{score}} : A \to \mathbb{R}^m$ that computes score vectors followed by the softmax operation, i.e.

$$f(\boldsymbol{x}) = \text{softmax}(f_{\text{score}}(\boldsymbol{x})).$$

Because of the fact proved in Exercise 11.5, the function f does not fix f_{score} uniquely.

If the function f_{score} satisfies the above equation, so does the function $\tilde{f}_{\text{score}}$ defined by

$$\tilde{f}_{\text{score}}(\boldsymbol{x}) = f_{\text{score}}(\boldsymbol{x}) + c(\boldsymbol{x})\mathbf{1},$$

where $\mathbf{1}$ is the vector of ones and c is any scalar function. But what matters for our purposes is existence, not uniqueness. Since we know from the UAT that a network with the architecture shown in Figure 13.5a can approximate any continuous function from A to score vectors, we can conclude that this same network followed by a softmax operation can approximate any continuous function from A to the space of probability vectors over m categories.

■ ***Exercise*** 13.8 $\underline{\boldsymbol{W}}^{(2)}$ is a $p_2 \times p_1$ matrix, $\boldsymbol{b}^{(2)}$ is a p_2-dimensional vector, $\underline{\boldsymbol{W}}^{(3)}$ is a $p_3 \times p_2$ matrix, and $\boldsymbol{b}^{(3)}$ is a p_3-dimensional vector.

Counting up all the elements of these matrices and vectors, we find that the network shown in Figure 13.7 has

$$p_1 p + p_1 + p_2 p_1 + p_2 + p_3 p_2 + p_3$$

parameters in total.

■ ***Exercise*** 13.9 (a) Consider chaining together *two* single-hidden-layer networks of the kind shown in Figure 13.5a, with the output $\boldsymbol{y}$ of the first network serving as the input to the second. We have

$$\boldsymbol{y} = \boldsymbol{b}^{(1)} + \underline{\boldsymbol{W}}^{(1)}\boldsymbol{h}^{(1)}$$

and

$$\boldsymbol{h}^{(2)} = \text{ReLU}(\boldsymbol{b}^{(2)} + \underline{\boldsymbol{W}}^{(2)}\boldsymbol{y}),$$

where $\boldsymbol{b}^{(1)}$ and $\underline{\boldsymbol{W}}^{(1)}$ are the biases and weights of the first network's linear output layer, $\boldsymbol{b}^{(2)}$ and $\underline{\boldsymbol{W}}^{(2)}$ are the biases and weights of the second network's hidden layer, and $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(2)}$ are the hidden layer activation vectors in the first and second networks. Substituting the first displayed equation into the second, we obtain

$$\begin{aligned}\boldsymbol{h}^{(2)} &= \text{ReLU}\left(\boldsymbol{b}^{(2)} + \underline{\boldsymbol{W}}^{(2)}(\boldsymbol{b}^{(1)} + \underline{\boldsymbol{W}}^{(1)}\boldsymbol{h}^{(1)})\right) \\ &= \text{ReLU}\left(\boldsymbol{b}^{(2)} + \underline{\boldsymbol{W}}^{(2)}\boldsymbol{b}^{(1)} + \underline{\boldsymbol{W}}^{(2)}\underline{\boldsymbol{W}}^{(1)}\boldsymbol{h}^{(1)}\right).\end{aligned}$$

But this same functional dependence of $\boldsymbol{h}^{(2)}$ on $\boldsymbol{h}^{(1)}$ can be obtained if the first hidden layer connects *directly* to the second; we just need to use connection weights

$$\underline{\boldsymbol{W}} = \underline{\boldsymbol{W}}^{(2)}\underline{\boldsymbol{W}}^{(1)}$$

and second hidden layer biases

$$\boldsymbol{b} = \boldsymbol{b}^{(2)} + \underline{\boldsymbol{W}}^{(2)}\boldsymbol{b}^{(1)}.$$

The desired result now follows by induction on the number of layers.

(b) Since single-hidden-layer networks can approximate the identity function, chains consisting of $L-1$ such networks (with the output of the ith serving as input to the $(i+1)$th) can also approximate identity. Thus, by the version of the UAT given in the text, chains of L such networks are universal approximators; thus, by the result proved in (a), layered networks with L hidden layers of ReLUs are universal approximators.

■ ***Exercise*** 13.10 For the envisaged task, 10^{30} loss function evaluations are required — a hopelessly impractical proposition.

■ ***Exercise*** 13.11 If $\nabla l(\boldsymbol{\theta}; D)$ has remained constant for many iterations, then repeated updating of $\boldsymbol{v}$ per equation (13.7) will have brought $\boldsymbol{v}$ close to a stable fixed point. We can find this fixed point by recasting the update rule as an equality and solving for $\boldsymbol{v}$. We have

$$\boldsymbol{v} = \beta \boldsymbol{v} + \nabla l,$$

which implies

$$\boldsymbol{v} = (1-\beta)^{-1} \nabla l.$$

The $\boldsymbol{\theta}$ update prescribed by equation (13.7) is then

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta(1-\beta)^{-1} \nabla l.$$

This is equivalent to a vanilla gradient descent update (equation (13.6)) with learning rate $\eta(1-\beta)^{-1}$.

■ ***Exercise*** 13.12 With vanilla gradient descent, the change made to the parameter vector $\boldsymbol{\theta}$ in any given iteration is proportional to $-\nabla l(\boldsymbol{\theta}; D)$, where $\nabla l(\boldsymbol{\theta}; D)$ is the current gradient of the loss function with respect to the parameter vector. So if $\boldsymbol{\theta}$ strays close to a bad local minimum during the training process, it will get 'sucked in'. By contrast, with gradient descent with momentum, the change made to the parameter vector in any given iteration is proportional to the current velocity. The velocity changes only gradually from iteration to iteration. Thus the parameter vector can stray close to a bad local minimum — indeed, it could in principle hit it exactly, so that $\nabla l(\boldsymbol{\theta}; D)$ is zero for one iteration — without getting stuck.

■ ***Exercise*** 13.13 The mean squared error of a prediction function $g(-; \boldsymbol{\theta})$ on dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is

$$\mathrm{MSE}(\boldsymbol{\theta}; D) = \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in D} \left(g(\boldsymbol{x}; \boldsymbol{\theta}) - y\right)^2,$$

conforming to the pattern laid down in equation (13.18). In this case, the pointwise loss is the squared error on a specific training example, and there is no regularisation penalty.

The unregularised cross-entropy loss of a probabilistic multiclass classifier $\boldsymbol{g}(-;\boldsymbol{\theta})$ on dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ (here the y_i are understood to be categorical) is

$$l(\boldsymbol{\theta}; D) = -\frac{1}{n} \sum_{(\boldsymbol{x},y)\in D} \log g_y(\boldsymbol{x};\boldsymbol{\theta}).$$

also conforming to the pattern laid down in equation (13.18). This time the pointwise loss is the negative logarithm of the probability assigned by the classifier to the correct label. To obtain a regularised cross-entropy loss, we would add a penalty term that depended only on $\boldsymbol{\theta}$. That corresponds to setting R in equation (13.18) to something other than the zero function.

■ ***Exercise*** 13.14 Let l be an unregularised loss function, and define the regularised loss function l^λ by

$$l^\lambda(\boldsymbol{\theta}; D) = l(\boldsymbol{\theta}; D) + \lambda \sum_{w\in W} w^2,$$

where λ is a hyperparameter controlling the strength of the regularisation. If we use l^λ instead of l in the gradient descent update rule (13.6), we find that the update for an arbitrary weight w is

$$w \leftarrow w - \eta\frac{\partial l^\lambda}{\partial w} - 2\eta\lambda w.$$

If $\frac{\partial l^\lambda}{\partial w}$ were to be zero, we would have

$$w \leftarrow (1 - 2\eta\lambda)w,$$

which (so long as $\eta\lambda < 0.5$) represents exponential decay of the weight.

■ ***Exercise*** 13.15 We consider a ReLU neuron with zero bias and the input weights shown in Figure 13.14b. As noted in the figure, this is a neuron that will respond to patches that are brighter on the right than on the left.

(a) $\mathrm{ReLU}(6) = 6$
(b) $\mathrm{ReLU}(-6) = 0$
(c) $\mathrm{ReLU}(0) = 0$

■ ***Exercise*** 13.16 The penultimate layer consists of 128 neurons that each receive inputs from all $16 \times 7 \times 7 = 784$ units in the preceding max-pooling layer. There are therefore 128 biases and 128×784 weights associated with the layer, for a total of

$$128 + 128 \times 784 = 100480$$

parameters. We now look at each of the other layers.

The first layer consists of eight 3×3 convolution filters. Associated with each filter is

one bias and $3 \times 3 = 9$ weights, so there are $8 \times 10 = 80$ parameters in total associated with this layer.

There are no parameters associated with the max-pooling layers, so the next layer to consider is the second convolution layer. Here we have sixteen filters, each of which processes a $3 \times 3 \times 8$ input block (corresponding to a 3×3 patch of image and eight channels). Associated with each filter is one bias and $3 \times 3 \times 8 = 72$ weights, so there are $16 \times 73 = 1168$ parameters in total associated with this layer.

The final softmax layer is equivalent to a layer of 10 linear neurons (each connected to all 128 neurons in the penultimate layer) followed by a fixed softmax operation. The parameters associated with this layer, therefore, are 10 biases and $128 \times 10 = 1280$ weights, for a total of 1290 parameters.

In the network as a whole, then, we have

$$80 + 1168 + 100480 + 1290 = 103018$$

parameters.

■ ***Exercise*** 13.17 The MNIST training set consists of 60,000 examples. This is much larger than the human speech sounds training set, which consists of 1897 examples (80% of 2371). To put the comparison another way, with the batch size set to 64, a single epoch of training on the MNIST data corresponds to 938 batches; but a single epoch of training on the speech sounds data corresponds to only 30 batches.

■ ***Exercise*** 13.18 Strictly speaking, if ϕ is ReLU then its gradient is

$$\phi_i'(z_i) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x < 0, \end{cases}$$

with the gradient undefined at $x = 0$. However, when doing gradient-based optimisation of neural networks it is normal to pretend either that $\phi_i'(0) = 0$ or that $\phi_i'(0) = 1$. (See discussion in Section 13.6.)

■ ***Exercise*** 13.19 Changing the dummy variable, the pointwise loss is

$$\boldsymbol{\ell} = \sum_{k=1}^{m} (y_k - h_{i(k)})^2.$$

Therefore

$$\begin{aligned} \frac{\partial \boldsymbol{\ell}}{\partial h_{i(j)}} &= \frac{\partial}{\partial h_{i(j)}} (y_j - h_{i(j)})^2 \\ &= -2(y_j - h_{i(j)}) \\ &= 2(h_{i(j)} - y_j), \end{aligned}$$

where in the first equality we have used the fact that identifiers are unique, so that $i(k) = i(j)$ implies $k = j$.

14

Expanding the toolkit

■ ***Exercise*** 14.1 If the vocabulary size is $|\mathcal{V}| = 10^4$, then there are $(10^4)^{10} = 10^{40}$ random 10-word sentences. If one in a billion billion (10^{18}) of these are grammatically correct, then there are

$$\frac{10^{40}}{10^{18}} = 10^{22}$$

grammatically correct 10-word sentences.