Exercises in Elements of Statistical Learning by Hastie, Tibshirani, and Friedman (2009)

Takao Noguchi
tkngch@hotmail.com

May 7, 2020

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Retrieved from `https://web.stanford.edu/~hastie/ElemStatLearn/`

Contents

Chapter 2. Overview of Supervised Learning **2**

Chapter 3. Linear Methods for Regression **7**

Chapter 4. Linear Methods for Classification **24**

Chapter 7. Model Assessment and Selection **33**

Chapter 10. Boosting and Additive Trees **39**

Chapter 2. Overview of Supervised Learning

Ex. 2.1

Suppose each of K-classes has an associated target t_k, which is a vector of all zeros, except a one in the kth position. Show that classifying to the largest element of $\hat{y}$ amounts to choosing the closest target, $\min_k \|t_k - \hat{y}\|$, if the elements of $\hat{y}$ sum to one.

To work on this exercise, I need to make the following assumptions:

$$\hat{y} \in \mathbb{R}^{\mathrm{K}}, \quad \hat{\mathrm{y}}_{\mathrm{i}} \geq 0 \, \forall \mathrm{i}, \quad \sum_{\mathrm{i}} \hat{\mathrm{y}}_{\mathrm{i}} = 1.$$

Then,

$$\begin{aligned}
\arg\min_k \|t_k - \hat{y}\| &= \arg\min_k \sum_i [t_{ki} - \hat{y}_i]^2 \\
&= \arg\min_k \sum_i \left[t_{ki}^2 + \hat{y}_i^2 - 2t_{ki}\hat{y}_i\right] \\
&= \arg\min_k \sum_i [-2t_{ki}\hat{y}_i] \\
&= \arg\max_k \sum_i [t_{ki}\hat{y}_i] \\
&= \arg\max_k \hat{y}_k
\end{aligned}$$

because t_k is a vector of all zeros, except a one in the kth position.

Ex. 2.2

Show how to compute the Bayes decision boundary for the simulation example in Figure 2.5.

The boundary represents x^* where

$$\begin{aligned}
&\Pr(g = \texttt{blue} | X = x^*) = \Pr(g = \texttt{orange} | X = x^*) \\
\Leftrightarrow &\Pr(g = \texttt{blue}) \Pr(X = x^* | g = \texttt{blue}) = \Pr(g = \texttt{orange}) \Pr(X = x^* | g = \texttt{orange}) \\
\Leftrightarrow &\Pr(X = x^* | g = \texttt{blue}) = \Pr(X = x^* | g = \texttt{orange}),
\end{aligned}$$

assuming $\Pr(g = \texttt{blue}) = \Pr(g = \texttt{orange})$.

Now from the description in pages 16–17, we know

$$\Pr(X = x^* | g = \texttt{blue}) = \mathcal{N}_{pdf}\left(x^* \middle| \mu, \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right), \quad \text{and} \quad \mu = \mathcal{N}_{pdf}\left(\mu \middle| \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right).$$

Similarly,

$$\Pr(X = x^* | g = \texttt{orange}) = \mathcal{N}_{pdf}\left(x^* \middle| \nu \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right), \quad \text{and} \quad \nu = \mathcal{N}_{pdf}\left(\nu \middle| \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right).$$

Then,

$$\begin{aligned}
&\Pr(X = x^* | g = \texttt{blue}) = \Pr(X = x^* | g = \texttt{orange}) \\
&\Leftrightarrow \int \mathcal{N}_{pdf}\left(x^* \middle| \mu, \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right) \mathcal{N}_{pdf}\left(\mu \middle| \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) d_\mu \\
&\quad = \int \mathcal{N}_{pdf}\left(x^* \middle| \nu, \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right) \mathcal{N}_{pdf}\left(\nu \middle| \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) d_\nu.
\end{aligned}$$

The computation of the above integrals is intractable. Perhaps to circumvent the intractability, the authors assumed that μ and ν are known: 10 samples were drawn for μ and ν, and these 20 samples are assumed equally likely. Then,

$$\Pr(X = x^* | g = \texttt{blue}) = \Pr(X = x^* | g = \texttt{orange})$$

is approximated as

$$\begin{aligned}
\sum_{i=1}^{10} \mathcal{N}_{pdf}\left(x^* \middle| \mu^{(i)}, \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right) &= \sum_{j=1}^{10} \mathcal{N}_{pdf}\left(x^* \middle| \nu^{(j)}, \begin{bmatrix} 1/5 & 0 \\ 0 & 1/5 \end{bmatrix}\right) \\
\Leftrightarrow \sum_{i=1}^{10} \left\| x^* - \mu^{(i)} \right\| &= \sum_{j=1}^{10} \left\| x^* - \nu^{(j)} \right\|
\end{aligned}$$

Here the ith sample of μ is denoted as $\mu^{(i)}$.

Ex. 2.6

Consider a regression problem with inputs x_i and outputs y_i, and a parameterized model $f_\theta(x)$ to be fit by least squares. Show that if there are observations with tied or identical values of x, then the fit can be obtained from a reduced weighted least squares problem.

The squared loss function, L, is given by

$$L = \sum_i (y_i - f_\theta(x_i))^2.$$

Now, let us assume n observations with identical values of x: $x_1 = x_2 = \cdots = x_n$. Then,

$$\begin{aligned}
L &= \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \sum_{i=n+1} (y_i - f_\theta(x_i))^2 \\
&= n\left(\frac{1}{n}\sum_{i=1}^{n} y_i - f_\theta(x_1)\right)^2 + c + \sum_{i=n+1} (y_i - f_\theta(x_i))^2,
\end{aligned}$$

where c depends only on y and independent of f_θ. Thus, minimizing the least square is equivalent to minimizing the following:

$$n\left(\frac{1}{n}\sum_{i=1}^{n} y_i - f_\theta(x_1)\right)^2 + \sum_{i=n+1} (y_i - f_\theta(x_i))^2.$$

Ex. 2.7

Suppose we have a sample of N pairs x_i, y_i drawn i.i.d. from the distribution characterized as follows:

$$\begin{aligned} x_i &\sim h(x), \quad \text{the design density} \\ y_i &= f(x_i) + \epsilon_i, \quad f \text{ is the regression function} \\ \epsilon_i &\sim (0, \sigma^2). \quad \text{(mean zero, variance } \sigma^2\text{)} \end{aligned}$$

We construct an estimator for f linear in the y_i,

$$\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0; \mathcal{X}) y_i,$$

where the weights $l_i(x_0; \mathcal{X})$ do not depend on the y_i, but do depend on the entire training sequence of x_i, denoted here by $\mathcal{X}$.

(a) Show that linear regression and k-nearest-neighbour regression are members of this class of estimators. Describe explicitly the weights $l_i(x_0; \mathcal{X})$ in each of these cases.

(b) Decompose the conditional mean-squared error

$$E_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2$$

into a conditional squared bias and a conditional variance component. Like $\mathcal{X}$, $\mathcal{Y}$ represents the entire training sequence of y_i.

(c) Decompose the (unconditional) mean-squared error

$$E_{\mathcal{Y},\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2$$

into a squared bias and a variance component.

First, I clarify the notations.

$$x_i \in \mathbb{R}^{\mathrm{p}} \quad \mathrm{y_i} \in \mathbb{R}^{\mathrm{p}} \quad \mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{p}}$$

(a)

The linear regression is given by $\hat{f}(x_0) = x_0\beta$, where $\beta = (X^TX)^{-1}X^Ty$. Then,

$$\hat{f}(x_0) = x_0(X^TX)^{-1}X^Ty = \sum_{i=1}^{N} x_0(X^TX)^{-1}x_iy_i$$

Thus, $l_i(x_0; \mathcal{X}) = x_0(X^TX)^{-1}x_i$.

For the k-nearest-neighbour,

$$l_i(x_0; \mathcal{X}) = \begin{cases} \frac{1}{k} & \text{if } x_i \text{ is among the k nearest observation to } x_0 \\ 0 & \text{otherwise.} \end{cases}$$

(b)

Note $E_{\mathcal{Y}|\mathcal{X}}[f(x_0)] = f(x_0)$. Then we have the following:

$$
\begin{aligned}
&E_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2 \\
&= E_{\mathcal{Y}|\mathcal{X}}\left[f(x_0)^2 + \hat{f}(x_0)^2 - 2f(x_0)\hat{f}(x_0)\right] \\
&= f(x_0)^2 + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)^2\right] - 2f(x_0)E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right] \\
&= \left[f(x_0) - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]\right]^2 - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]^2 + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)^2\right] \\
&= \left[f(x_0) - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]\right]^2 + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)^2\right] - 2E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right] + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]^2 \\
&= \left[f(x_0) - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]\right]^2 + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)^2 - 2\hat{f}(x_0)E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right] + E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]^2\right] \\
&= \left[f(x_0) - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]\right]^2 + E_{\mathcal{Y}|\mathcal{X}}\left[\left(\hat{f}(x_0) - E_{\mathcal{Y}|\mathcal{X}}\left[\hat{f}(x_0)\right]\right)^2\right] \\
&= \text{Bias}^2 + \text{Variance}.
\end{aligned}
$$

(c)

Note $E_{\mathcal{Y},\mathcal{X}}[f(x_0)] = f(x_0)$. Then the derivation is basically the same as the one in above (b).

$$
\begin{aligned}
&E_{\mathcal{Y},\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2 \\
&= E_{\mathcal{Y},\mathcal{X}}\left[f(x_0)^2 + \hat{f}(x_0)^2 - 2f(x_0)\hat{f}(x_0)\right] \\
&= f(x_0)^2 + E_{\mathcal{Y},\mathcal{X}}\left[\hat{f}(x_0)^2\right] - 2f(x_0)E_{\mathcal{Y},\mathcal{X}}\left[\hat{f}(x_0)\right] \\
&= \dots \\
&= \left[f(x_0) - E_{\mathcal{Y},\mathcal{X}}\left[\hat{f}(x_0)\right]\right]^2 + E_{\mathcal{Y},\mathcal{X}}\left[\left(\hat{f}(x_0) - E_{\mathcal{Y},\mathcal{X}}\left[\hat{f}(x_0)\right]\right)^2\right] \\
&= \text{Bias}^2 + \text{Variance}.
\end{aligned}
$$

Ex. 2.9

Consider a linear regression model with p parameters, fit by least squares to a set of training data $(x_1, y_1), \dots, (x_N, y_N)$ drawn at random from a population. Let β be the least squares estimate. Suppose we have some test data $(\tilde{x}_1, \tilde{y}_1), \dots, (\tilde{x}_M, \tilde{y}_M)$ drawn at random from the same population as the training data. If $R_{tr}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i\beta)^2$ and $R_{te}(\beta) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \tilde{x}_i\beta)^2$, prove that

$$E\left[R_{tr}(\hat{\beta})\right] \leq E\left[R_{te}(\hat{\beta})\right]$$

where the expectations are over all that is random in each expression.

From the description, we know

$$\hat{\beta} = \arg\min_{\beta}\left[\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i\beta)^2\right] \quad \text{and} \quad R_{tr}(\hat{\beta}) = \min_{\beta} R_{tr}(\beta).$$

Now, let $\tilde{\beta}$ be the least square estimate on the N test data.

$$\tilde{\beta} = \arg\min_{\beta} \left[\frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i - \tilde{x}_i \beta)^2 \right] \quad \text{and} \quad R_{te}(\tilde{\beta}) = \min_{\beta} R_{te}(\beta).$$

Then given that (x_i, y_i) and $(\tilde{x}_i, \tilde{y}_i)$ are from the same distribution, we have

$$E\left[R_{tr}(\hat{\beta})\right] = E\left[\frac{1}{N} \sum_{i=1}^{N} (y_i - x_i \hat{\beta})^2 \right] = E\left[\frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i - \tilde{x}_i \tilde{\beta})^2 \right]$$

Here I note that the right-most term is independent of N:

$$E\left[\frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i - \tilde{x}_i \beta)^2 \right] = E\left[(\tilde{y}_1 - \tilde{x}_1 \beta)^2 \right] = E\left[\frac{1}{M} \sum_{i=1}^{M} (\tilde{y}_i - \tilde{x}_i \beta)^2 \right] = E\left[R_{te}(\beta) \right].$$

Then we have

$$E\left[R_{tr}(\hat{\beta})\right] = E\left[R_{te}(\tilde{\beta})\right] \leq E\left[R_{te}(\hat{\beta})\right].$$

The equality holds when $\hat{\beta} = \tilde{\beta}$.

Chapter 3. Linear Methods for Regression

Ex. 3.2

Given data on two variables X and Y, consider fitting a cubic polynomial regression model $f(X) = \sum_{j=0}^{3} \beta_j X^j$. In addition to plotting the fitted curve, you would like a 95% confidence band about the curve. Consider the following two approaches.

1. At each point x_0, form a 95% confidence interval for the linear function $\alpha^T \beta = \sum_{j=0}^{3} \beta_j x_0^j$.
2. Form a 95% confidence set for β as in (3.15), which in turn generates confidence intervals for $f(x_0)$.

How do these approaches differ? Which band is likely to be wider? Conduct a small simulation experiment to compare the two methods.

First, I set $n = 100$ and $\beta = [-1, 2, 1, 3]^T$. Then x and y are randomly generated as below.

Approach 1.

We would like to estimate the variance of prediction, $Var\left(\sum_{j=0}^{3} \hat{\beta}_j x_0^j\right)$. For convenience, let $X_0 = [1, x_0, x_0^2, x_0^3]$. Then,

$$\begin{aligned} Var\left(\sum_{j=0}^{3} \hat{\beta}_j x_0^j\right) &= Var\left(X_0\,\hat{\beta}\right) \\ &= X_0\,\hat{\beta}\,\hat{\beta}^T\,{X_0}^T \\ &= X_0\,((X^TX)^{-1}X^Ty)\,((X^TX)^{-1}X^Ty)^T\,{X_0}^T \\ &= yy^T\,X_0\,(X^TX)^{-1}\,{X_0}^T \end{aligned}$$

where $yy^T = \sigma^2$ can be estimated with the equation immediately after equation (3.8) in page 47.

Approach 2.

We would like to form the confidence set for β. This is achieved by first drawing 100 samples from the multivariate normal distribution:

$$\tilde{\beta} \sim \mathcal{N}\left(\hat{\beta},\, (X^TX)^{-1}\hat{\sigma}^2\right)$$

(equation 3.10) and then retaining the samples which satisfy

$$(\hat{\beta} - \tilde{\beta})^T X^T X (\hat{\beta} - \tilde{\beta}) \leq \hat{\sigma}^2 \chi_5^{2(1-0.05)}$$

(equation 3.15), where $\chi_5^{2(1-0.05)} = 11.1$. The 100 $\tilde{\beta}s$ are organized into a 4×100 matrix, $\bar{\beta}$. Then for each x_0, I took the maximum and minimum of $X_0\bar{\beta}$ as the upper and lower bounds. The results are summarized in Figure 1.

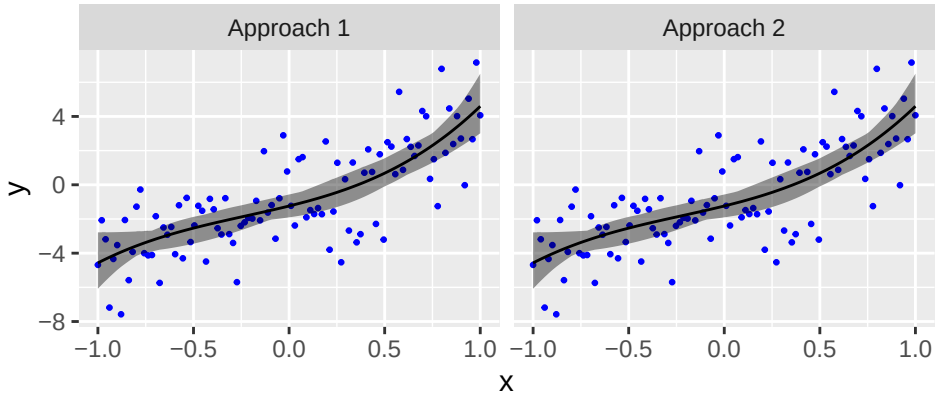


Figure 1: Results for Ex.3.2. The black solid line represents $\hat{y}$, the shaded area represents 95% confidence interval, and the blue dots represents data.

Ex. 3.5

Consider the ridge regression problem (3.41). Show that this problem is equivalent to the problem

$$\hat{\beta}^c = \arg\min_{\beta^c} \left\{ \sum_{i=1}^{N} \left[y_i - \beta_0^c - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^c \right]^2 + \lambda \sum_{j=1}^{p} \beta_j^{c2} \right\}$$

Give the correspondence between β^c and the original β in (3.41). Characterize the solution to this modified criterion. Show that a similar result holds for the lasso.

Equation (3.41) is

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

and

$$\hat{\beta}^c = \arg\min_{\beta^c} \left\{ \sum_{i=1}^{N} \left[y_i - \beta_0^c + \sum_{j=1}^{p} \bar{x}_j\beta_j^c - \sum_{j=1}^{p} x_{ij}\beta_j^c \right]^2 + \lambda \sum_{j=1}^{p} \beta_j^{c2} \right\}$$

These two equations are equivalent, when

$$\beta_0 = \beta_0^c - \sum_{j=1}^{p} \bar{x}_j\beta_j^c \quad \text{and} \quad \beta_j = \beta_j^c$$

For the lasso, Equation (3.52) shows

$$\begin{aligned} \hat{\beta}^{\text{lasso}} &= \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \\ &= \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \bar{x_j}\beta_j - \sum_{j=1}^{p} (x_{ij} - \bar{x_j})\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \end{aligned}$$

which is equivalent to

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0^c - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

when

$$\beta_0 = \beta_0^c - \sum_{j=1}^{p} \bar{x}_j \beta_j.$$

Ex. 3.6

Show that the ridge regression estimate is the mean (and mode) of the posterior distribution, under a Gaussian prior $\beta \sim N(0, \tau I)$, and Gaussian sampling model $y \sim N(X\beta, \sigma^2 I)$. Find the relationship between the regularization parameter λ in the ridge formula, and the variances τ and σ^2.

The posterior probability is given by

$$\begin{aligned} \Pr(\beta|y, X) &\propto \Pr(\beta)\ \Pr(y|\beta, X) = \mathcal{N}_{pdf}(\beta|0, \tau I)\, \mathcal{N}_{pdf}(y|X\beta, \sigma^2 I) \\ &\propto \exp\left\{ -\frac{\beta^T(\tau^{-1}I)\beta + (y - X\beta)^T(\sigma^{-2}I)(y - X\beta)}{2} \right\} \end{aligned}$$

The mode of this posterior is

$$\begin{aligned} \hat{\beta} &= \arg\max_{\beta} \exp\left\{ -\frac{\beta^T(\tau^{-1}I)\beta + (y - X\beta)^T(\sigma^{-2}I)(y - X\beta)}{2} \right\} \\ &= \arg\min_{\beta} \left[\beta^T(\tau^{-1}I)\beta + (y - X\beta)^T(\sigma^{-2}I)(y - X\beta) \right] \\ &= \arg\min_{\beta} \left[\tau^{-1} \sum_{j=1}^{p} \beta_j^2 + \sigma^{-2} \sum_{i=1}^{N} (y_i - X_{i\cdot}\beta)^2 \right] \\ &= \arg\min_{\beta} \left[\sum_{i=1}^{N} (y_i - X_{i\cdot}\beta)^2 + \frac{\sigma^2}{\tau} \sum_{j=1}^{p} \beta_j^2 \right] \end{aligned}$$

Then, the regularization parameter $\lambda = \frac{\sigma^2}{\tau}$.

Ex. 3.7

Assume $y_i \sim N(\beta_0 + x_i^T\beta, \sigma^2)$, $i = 1, 2, \ldots, N$, and the parameters β_j, $j = 1, \ldots, p$ are each distributed as $N(0, \tau^2)$, independently of one another. Assuming σ^2 and τ^2 are known, and β_0 is not governed by a prior (or has a flat improper prior), show that the (minus) log-posterior density of β is proportional to $\sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$ where $\lambda = \sigma^2/\tau^2$.

The likelihood is given by

$$\log \Pr(y|\beta, X) = -\log\left(\sigma\sqrt{2\pi}\right) - \frac{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2}{2\sigma^2}.$$

The log-prior density of β is

$$\log \Pr(\beta) = \sum_{j=1}^{p} \log \Pr(\beta_j) = \sum_{j=1}^{p} \log \left[\frac{1}{\tau\sqrt{2\pi}} e^{-\frac{\beta_j^2}{2\tau^2}} \right] = \sum_{j=1}^{p} \left[-\log\left(\tau\sqrt{2\pi}\right) - \frac{\beta_j^2}{2\tau^2} \right].$$

Assuming the improper flat prior, $\Pr(\beta_0) = c$, $\Pr(\beta_0)$ just contribute the constant to the log-posterior density. The negative log-posterior density of β is

$$\begin{aligned}
&- \log \Pr(\beta|y, X) \\
&= - \log \Pr(y|\beta, X) - \log \Pr(\beta) - \log \Pr(\beta_0) \\
&= \frac{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2}{2\sigma^2} + \sum_{j=1}^{p} \left[\frac{\beta_j^2}{2\tau^2} \right] + \log\left(\sigma\sqrt{2\pi}\right) + \sum_{j=1}^{p} \left[-\log\left(\tau\sqrt{2\pi}\right) \right] - \log c \\
&\propto \frac{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2}{2\sigma^2} + \frac{1}{2\tau^2} \sum_{j=1}^{p} \beta_j^2 \\
&\propto \frac{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 + \frac{\sigma^2}{\tau^2} \sum_{j=1}^{p} \beta_j^2}{2\sigma^2} \\
&\propto \sum_{i=1}^{N}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 + \frac{\sigma^2}{\tau^2} \sum_{j=1}^{p} \beta_j^2.
\end{aligned}$$

Ex. 3.10

Backward stepwise regression. Suppose we have the multiple regression fit of y on X, along with the standard errors and Z-scores as in Table 3.2. We wish to establish which variable, when dropped, will increase the residual sum-of-squares the least. How would you do this?

Let $RSS_{\bar{j}} - RSS$ denote the increase in the residual sum-of-squares after dropping x_j from the model. Then, we wish to find

$$\begin{aligned}
j &= \arg\min_j RSS_{\bar{j}} - RSS \\
&= \arg\min_j \frac{RSS_{\bar{j}} - RSS}{RSS/(N-p-1)},
\end{aligned}$$

where N and p are the number of rows and the number of columns in X. Thus, the variable to drop will have the minimum F static. From Ex. 3.1, this F score is the square of the corresponding z-score. Therefore, I would look for a coefficient with the smallest absolute z-score.

Ex. 3.11

Show that the solution to the multivariate linear regression problem (3.40) is given by (3.39). What happens if the covariance matrices Σ_i are different for each observation?

The equation (3.40) gives us the multivariate weighted criterion:

$$\mathrm{RSS}(B; \Sigma) = \sum_{i=1}^{N} (y_i - f(x_i))^T \Sigma^{-1} (y_i - f(x_i))$$

First, let me clarify the notation.

$Y \in \mathbb{R}^{\mathrm{N}\times\mathrm{K}}$	Matrix of outputs
$X \in \mathbb{R}^{\mathrm{N}\times\mathrm{p}+1}$	Matrix of inputs
$\Sigma \in \mathbb{R}^{\mathrm{p}+1\times\mathrm{p}+1}$	Covariance matrix
$B \in \mathbb{R}^{\mathrm{p}+1\times\mathrm{K}}$	Matrix of parameters

Now, in Equation (3.40), y_i is the ith row of Y, transposed to be in $\mathbb{R}^{\mathrm{K}\times 1}$, and similarly, x_i is the ith row of X, transposed to be in $\mathbb{R}^{\mathrm{p}+1\times 1}$. Then, $f(x_i) \in \mathbb{R}^{\mathrm{K}\times 1}$.

With the above notation, I calculate the residual sum of squares for the multivariate linear regression problem, where $f(x_i) = B^T x_i$. Thus

$$\begin{aligned}
\mathrm{RSS}(B;\Sigma) &= \sum_{i=1}^{N}(y_i - B^T x_i)^T \Sigma^{-1}(y_i - B^T x_i) \\
&= \mathrm{tr}((Y^T - B^T X^T)^T \Sigma^{-1}(Y^T - B^T X^T)) \\
&= \mathrm{tr}((Y - XB)\Sigma^{-1}(Y^T - B^T X^T)) \\
&= \mathrm{tr}(Y\Sigma^{-1}Y^T - Y\Sigma^{-1}B^T X^T - XB\Sigma^{-1}Y^T + XB\Sigma^{-1}B^T X^T) \\
&= \mathrm{tr}(Y\Sigma^{-1}Y^T) - 2\,\mathrm{tr}(XB\Sigma^{-1}Y^T) + \mathrm{tr}(XB\Sigma^{-1}B^T X^T)
\end{aligned}$$

Please note that here

$$\mathrm{tr}(Y\Sigma^{-1}B^T X^T) = \mathrm{tr}((Y\Sigma^{-1}B^T X^T)^T) = \mathrm{tr}(XB\Sigma^{-1}Y^T).$$

Now, I take the derivative of $\mathrm{RSS}(B;\Sigma)$ with respect to B, noting that

$$\frac{d\,\mathrm{tr}(AXB)}{dX} = BA.$$

Setting the derivative to zero, I obtain the following:

$$\begin{aligned}
\frac{\mathrm{RSS}(dB;\Sigma)}{dB} = 0 \quad &\Leftrightarrow \quad -2(\Sigma^{-1}Y^T X) + (\Sigma^{-1}B^T X^T X) + (X^T XB\Sigma^{-1})^T = 0 \\
&\Leftrightarrow \quad \Sigma^{-1}B^T X^T X = \Sigma^{-1}Y^T X \\
&\Leftrightarrow \quad B^T X^T X = Y^T X \\
&\Leftrightarrow \quad B^T = Y^T X(X^T X)^{-1} \\
&\Leftrightarrow \quad B = (X^T X)^{-1}X^T Y,
\end{aligned}$$

which is Equation (3.39).

The above derivation is not applicable when the covariance matrices are different for each observation

Ex. 3.12

Show that the ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set. We augment the centered matrix X with p additional rows $\sqrt{\lambda}I$, and augment y with p zeros. By introducing artificial data having response value zero, the fitting procedure is forced to shrink the coefficients toward zero. This is related to the idea of hints due to Abu-Mostafa (1995), where

model constraints are implemented by adding artificial data examples that satisfy them.

We let

$$\tilde{X} = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \quad \text{and} \quad \tilde{y} = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

Then the regression estimates are given by

$$\begin{aligned}(\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T\tilde{y} &= \left(\begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} \\ &= \left(X^T X + \lambda I\right)^{-1} X^T y\end{aligned}$$

which is identical to the ridge regression estimates (see equation 3.44 in page 64).

Ex. 3.13

Derive the expression (3.62), and show that $\hat{\beta}^{\text{pcr}}(p) = \hat{\beta}^{\text{ls}}$.

Equation (3.62) is

$$\begin{aligned}\hat{\beta}^{\text{pcr}}(M) &= \sum_{m=1}^{M} \hat{\theta}_m v_m \\ &= \sum_{m=1}^{M} \frac{\langle z_m, y \rangle}{\langle z_m, z_m \rangle} v_m\end{aligned}$$

where z_m is the mth principal components of X, Xv_m. Recall the principal components are given by the singular value decomposition $X = UDV^T$. Then, $Z = XV = UD$, and

$$\begin{aligned}\hat{\beta}^{\text{pcr}}(p) &= V(Z^T Z)^{-1} Z^T Y \\ &= V(V^T X^T X V)^{-1} V^T X^T Y \\ &= (X^T X)^{-1} X^T Y \\ &= \hat{\beta}^{\text{ls}}.\end{aligned}$$

Ex. 3.14

Show that in the orthogonal case, PLS stops after $m = 1$ steps, because subsequent $\hat{\phi}_{mj}$ in step 2 in Algorithm 3.3 are zero.

Follow the notation in Algorithm 3.3, I have

$$\begin{aligned}\hat{y}^{(0)} &= \bar{y}1 \\ x_j^{(0)} &= x_j \quad (j = 1, \ldots, p)\end{aligned}$$

Then,

$$\begin{aligned}
\hat{\phi}_{1j} &= \langle x_j^{(0)}, y \rangle = \langle x_j, y \rangle \\
z_1 &= \sum_{j=1}^{p} \hat{\phi}_{1j} x_j^{(0)} = \sum_{j=1}^{p} \hat{\phi}_{1j} x_j \\
x_j^{(1)} &= x_j^{(0)} - \frac{\langle z_1, x_j^{(0)} \rangle}{\langle z_1, z_1 \rangle} z_1 = x_j - \frac{\langle z_1, x_j \rangle}{\langle z_1, z_1 \rangle} z_1
\end{aligned}$$

Thus,

$$\begin{aligned}
\hat{\phi}_{2j} &= \langle x_j^{(1)}, y \rangle \\
&= \left\langle x_j - \frac{\langle z_1, x_j \rangle}{\langle z_1, z_1 \rangle} z_1, y \right\rangle \\
&= x_j^T y - \frac{z_1^T x_j}{z_1^T z_1} z_1^T y \\
&= \hat{\phi}_{1j} - \frac{\hat{\phi}_{1j}}{\sum_{i=1}^{p} \hat{\phi}_{1i}^2} \sum_{i=1}^{p} \hat{\phi}_{1i} \, x_i^T y \\
&= \hat{\phi}_{1j} - \frac{\hat{\phi}_{1j}}{\sum_{i=1}^{p} \hat{\phi}_{1i}^2} \sum_{i=1}^{p} \hat{\phi}_{1i} \, \hat{\phi}_{1i} \\
&= 0.
\end{aligned}$$

Then the algorithm must stop.

Ex. 3.16

Derive the entries in Table 3.4, the explicit forms for estimators in the orthogonal case.

Table 3.4 lists three transformations of the least squares estimate $\hat{\beta}_j$ in the case of orthonormal columns of X. Then,

$$X^T X = I \quad \text{and} \quad X X^T = I$$

and thus,

$$\hat{\beta} = (X^T X)^{-1} X^T y = X^T y, \quad \text{and} \quad z_j^2 = \frac{\hat{\beta}_j^2}{\hat{\sigma}^2},$$

where $\hat{\sigma}^2$ is the estimated variance of y.

(1) Best subset

Given that the columns of X are orthogonal, the best subset of size M, which gives the smallest residual sum of squares, include M variables whose squared z score is the largest. Letting $z_{(M)}^2$ the Mth largest squared z score, the best subset of size M is given by

$$\hat{\beta}_j \cdot I\left(\hat{z}_j^2 \geq \hat{z}_{(M)}^2\right) \quad \Leftrightarrow \quad \hat{\beta}_j \cdot I\left(\hat{\beta}_j^2 \geq \hat{\beta}_{(M)}^2\right) \quad \Leftrightarrow \quad \hat{\beta}_j \cdot I\left(\left|\hat{\beta}_j\right| \geq \left|\hat{\beta}_{(M)}\right|\right).$$

(2) Ridge

The parameter of ridge regression is given by

$$\begin{aligned}\hat{\beta}^{\text{ridge}} &= (X^T X + \lambda I)^{-1} X^T y \\ &= (I + \lambda I)^{-1} X^T y \\ &= (I + \lambda I)^{-1} \hat{\beta} \\ &= \frac{1}{1+\lambda} \hat{\beta}.\end{aligned}$$

(3) Lasso

The parameter is given by Equation (3.52):

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left[\frac{1}{2} \sum_{i=1}^{N} (y - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right].$$

Then

$$\begin{aligned}\hat{\beta}^{\text{lasso}} &= \arg\min_{\beta} \left[\frac{1}{2} (y - X\beta)^T (y - X\beta) + \lambda |\beta| \right] \\ &= \arg\min_{\beta} \left[\frac{1}{2} (y - X\beta)^T X X^T (y - X\beta) + \lambda |\beta| \right] \\ &= \arg\min_{\beta} \left[\frac{1}{2} (\hat{\beta} - \beta)^T (\hat{\beta} - \beta) + \lambda |\beta| \right] \\ &= \arg\min_{\beta} \left[\frac{1}{2} \left(\hat{\beta}^T \hat{\beta} + \beta^T \beta - \hat{\beta}^T \beta - \beta^T \hat{\beta} \right) + \lambda |\beta| \right] \\ &= \arg\min_{\beta} \left[\frac{1}{2} \left(\beta^T \beta - \hat{\beta}^T \beta - \beta^T \hat{\beta} \right) + \lambda |\beta| \right].\end{aligned}$$

Now let

$$f(\beta) = \frac{1}{2} \left(\beta^T \beta - \hat{\beta}^T \beta - \beta^T \hat{\beta} \right) + \lambda |\beta|$$

Then the derivative of f is given by

$$\frac{df(\beta)}{d\beta_j} = \begin{cases} \beta_j - \hat{\beta}_j + \lambda & \text{if } \beta > 0 \\ \beta_j - \hat{\beta}_j - \lambda & \text{if } \beta < 0. \end{cases}$$

By setting the derivative to be zero, I obtain $\hat{\beta}^{\text{lasso}}$:

$$\hat{\beta}_j^{\text{lasso}} = \begin{cases} \hat{\beta}_j - \lambda & \text{if } \hat{\beta}_j - \lambda > 0 \\ \hat{\beta}_j + \lambda & \text{if } \hat{\beta}_j + \lambda < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Ex. 3.17

Repeat the analysis of Table 3.3 on the spam data discussed in Chapter 1.

Table 3.3 summarizes the coefficients estimated with the least squares, the best subset, the ridge, the lasso, the PCR and the PLS methods.

One column in the spam data indicates whether an email was a spam (1) or not (0). This is a classification problem, but here, I apply linear regressions. The columns are normalised so that each column has zero mean and unit variance. Then, I took the 3220 rows for training the models and the remaining 1380 for testing. I used `glmnet` package in R to estimate coefficients. The shrinkage parameters for the ridge and lasso regressions are estimated with 10-fold cross-validation.

The results are shown in Tables C3.1 and C3.2. The spam data has 57 columns, and Table C3.1 list the coefficients which are estimated to the largest values for the least square method.

Table C3.1: A few of the estimated coefficients for different methods on the spam data.

Term	LS	Ridge	Lasso
(Intercept)	0.482	0.495	0.491
X0.778	0.108	0.080	0.104
X0.32.1	0.059	0.048	0.057
X0.3	0.058	0.044	0.058
X0.96	0.056	0.045	0.058
X1.93	0.044	0.040	0.044
X0.14	0.043	0.033	0.042
X0.13	0.040	0.028	0.034
X0.32	0.039	0.030	0.034
X0.23	0.037	0.007	0.000

Table C3.2: Test error results for different methods on the spam data. MAE stands for mean absolute error, and MSE stands for mean squared error.

Term	LS	Ridge	Lasso
Test Error (MAE)	0.347	0.363	0.351
Test Error (MSE)	0.182	0.176	0.179

Ex. 3.19

Show that $\|\hat{\beta}^{\text{ridge}}\|$ increases as its tuning parameter $\lambda \to 0$. Does the same property hold for the lasso and partial least squares estimates? For the latter, consider the "tuning parameter" to be the successive steps in the algorithm.

(1) Ridge

Let the singular value decomposition of X be UDV^T ($X = UDV^T$), where $U^TU =$

$V^TV = VV^T = I$ and D is a diagonal matrix. Then

$$\begin{aligned}
\|\hat{\beta}^{\text{ridge}}\| &= \|(X^TX + \lambda I)^{-1}X^Ty\| \\
&= \|(VDU^TUDV^T + \lambda VV^T)^{-1}VDU^Ty\| \\
&= \|(VD^2V^T + \lambda VV^T)^{-1}VDU^Ty\| \\
&= \|[V(D^2 + \lambda I)V^T]^{-1}VDU^Ty\| \\
&= \|V^{-T}(D^2 + \lambda I)^{-1}V^{-1}VDU^Ty\| \quad \because (AB)^{-1} = B^{-1}A^{-1} \\
&= \|V(D^2 + \lambda I)^{-1}DU^Ty\| \quad \because (V)^{-T} = V.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\hat{\beta}^{\text{ridge}}\|^2 &= \left(V(D^2 + \lambda I)^{-1}DU^Ty\right)^T \left(V(D^2 + \lambda I)^{-1}DU^Ty\right) \\
&= y^TUD(D^2 + \lambda I)^{-1}V^TV(D^2 + \lambda I)^{-1}DU^Ty \\
&= y^TUD(D^2 + \lambda I)^{-2}DU^Ty \\
&= \sum_i y^TU_i \frac{d_{ii}^2}{\left(d_{ii}^2 + \lambda\right)^2} U_i^Ty \\
&= \sum_i \frac{d_{ii}^2}{\left(d_{ii}^2 + \lambda\right)^2} y^Tu_iu_i^Ty \\
&= \sum_i \frac{d_{ii}^2}{\left(d_{ii}^2 + \lambda\right)^2} \left(y^Tu_i\right)^2.
\end{aligned}$$

Then, $\|\hat{\beta}^{\text{ridge}}\|^2$ strictly, monotonically increases as λ approaches to 0.

(2) Lasso

From Table 3.4 and Ex. 3.16, we know that when X is an orthonormal matrix, the lasso estimate is given by

$$\hat{\beta}_j^{\text{lasso}} = \begin{cases} \hat{\beta}_j - \lambda & \text{if } \hat{\beta}_j - \lambda > 0 \\ \hat{\beta}_j + \lambda & \text{if } \hat{\beta}_j + \lambda < 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $\hat{\beta}$ is the least square estimates. Then,

$$\|\hat{\beta}_j^{\text{lasso}}\| = \begin{cases} \|\hat{\beta}_j - \lambda\| & \text{if } \hat{\beta}_j - \lambda > 0 \\ \|\hat{\beta}_j + \lambda\| & \text{if } \hat{\beta}_j + \lambda < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus, as λ approaches to 0, $\|\hat{\beta}^{\text{lasso}}\|$ weakly, monotonically increases from 0 to $\|\hat{\beta}\|$.

Ex. 3.23

Consider a regression problem with all variables and response having mean zero and standard deviation one. Suppose also that each variable has identical absolute correlation with the response:

$$\frac{1}{N}|\langle x_j, y\rangle| = \lambda, \quad j = 1, \ldots, p.$$

Let $\hat{\beta}$ be the least-squares coefficient of y on X, and let $u(\alpha) = \alpha X\hat{\beta}$ for $\alpha \in [0, 1]$ be the vector that moves a fraction α toward the least squares fit u. Let RSS be the residual sum-of-squares from the full least squares fit.

(a) Show that

$$\frac{1}{N}|\langle x_j, y - u(\alpha)\rangle| = (1-\alpha)\lambda, \quad j = 1, \ldots, p,$$

and hence the correlations of each x_j with the residuals remain equal in magnitude as we progress toward u.

(b) Show that these correlations are all equal to

$$\lambda(\alpha) = \frac{(1-\alpha)}{\sqrt{(1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N}RSS}}\lambda,$$

and hence they decrease monotonically to zero.

(c) Use these results to show that the LAR algorithm in Section 3.4.4 keeps the correlations tied and monotonically decreasing, as claimed in (3.55).

(a)

First, I note

$$\begin{aligned} X^T(y - u(\alpha)) &= X^T(y - \alpha X\beta) \\ &= X^T\left(y - \alpha X\left(X^TX\right)^{-1}X^Ty\right) \\ &= X^Ty - \alpha X^Ty \\ &= (1-\alpha)X^Ty. \end{aligned}$$

As $\langle x_j, y - u(\alpha)\rangle$ is the jth element of vector $X^T(y - u(\alpha))$,

$$\begin{aligned} |\langle x_j, y - u(\alpha)\rangle| &= \left|x_j^T(y - u(\alpha))\right| \\ &= \left|\left[X^T(y - u(\alpha))\right]_j\right| \\ &= \left|(1-\alpha)\left[X^Ty\right]_j\right| \\ &= (1-\alpha)|\langle x_j, y\rangle| \\ &= (1-\alpha)N\lambda \end{aligned}$$

Therefore,

$$\frac{1}{N}|\langle x_j, y - u(\alpha)\rangle| = (1-\alpha)\lambda.$$

(b)

The correlation between x_j and $y - u(\alpha)$ is given by

$$\frac{Cov(x_j, y - u(\alpha))}{\sqrt{Var(x_j)}\sqrt{Var(y - u(\alpha))}} = \frac{Cov(x_j, y - u(\alpha))}{\sqrt{Var(y - u(\alpha))}}$$

because x_j has zero mean and unit standard deviation.

Given that both x_j and y have zero mean, the covariance between x_j and $y - u(\alpha)$ is given by

$$Cov(x_j, y - u(\alpha)) = \frac{1}{N} x_j^T (y - u(\alpha)) = (1 - \alpha)\lambda.$$

Then,

$$\frac{Cov(x_j, y - u(\alpha))}{\sqrt{Var(y - u(\alpha))}} = \frac{(1 - \alpha)\lambda}{\sqrt{Var(y - u(\alpha))}}.$$

Before calculating the variance of $y - u(\alpha)$, we consider the residual sum of square, which is given by

$$RSS = (y - X\beta)^T (y - X\beta),$$

and as β is the least-square coefficient,

$$\begin{aligned} &\frac{dRSS}{d\beta} = 0 \\ &\Leftrightarrow X^T(y - X\beta) = 0 \\ &\Leftrightarrow X^T y = X^T X\beta. \end{aligned}$$

Then,

$$\begin{aligned} RSS &= y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta \\ &= y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T y \\ &= y^T y - y^T X\beta. \end{aligned}$$

Now, the variance is given by

$$\begin{aligned} Var(y - u(\alpha)) &= \frac{1}{N}(y - u(\alpha))^T (y - u(\alpha)) \\ &= \frac{1}{N}(y - \alpha X\beta)^T (y - \alpha X\beta) \\ &= \frac{1}{N}\left[y^T y - \alpha y^T X\beta - \alpha\beta^T X^T y + \alpha^2 \beta^T X^T X\beta\right] \\ &= \frac{1}{N}\left[y^T y - \alpha y^T X\beta - \alpha\beta^T X^T y + \alpha^2 \beta^T X^T y\right] \\ &= \frac{1}{N}\left[y^T y - 2\alpha y^T X\beta + \alpha^2 y^T \beta X\right] \\ &= \frac{1}{N}\left[y^T y + (\alpha^2 - 2\alpha) y^T X\beta\right] \\ &= \frac{1}{N}\left[y^T y + (\alpha^2 - 2\alpha)(y^T y - RSS)\right] \\ &= \frac{1}{N}\left[(1 - \alpha)^2 y^T y + \alpha(2 - \alpha) RSS\right] \\ &= \left[(1 - \alpha)^2 + \frac{\alpha(2 - \alpha)}{N} RSS\right], \end{aligned}$$

because that y has zero mean and unit standard deviation.

Therefore, the correlation is given by

$$\frac{Cov(x_j, y-u(\alpha))}{\sqrt{Var(x_j)}\sqrt{Var(y-u(\alpha))}} = \frac{Cov(x_j, y-u(\alpha))}{\sqrt{Var(y-u(\alpha))}} = \frac{(1-\alpha)\lambda}{\sqrt{Var(y-u(\alpha))}} = \frac{(1-\alpha)\lambda}{\sqrt{(1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N}RSS}}.$$

To prove that the correlation monotonically decreases to zero, as α approaches to 1, I rewrite the above correlation:

$$\frac{(1-\alpha)\lambda}{\sqrt{(1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N}RSS}} = \frac{\lambda}{\sqrt{1 + \frac{\alpha(2-\alpha)}{(1-\alpha)^2}\frac{RSS}{N}}}.$$

The term, α, $\frac{\alpha(2-\alpha)}{(1-\alpha)^2}$, monotonically increases as α approaches to 1. To see this, I differentiate the term with respect to α:

$$\begin{aligned}
\frac{d}{d\alpha}\frac{\alpha(2-\alpha)}{(1-\alpha)^2} &= \left(\frac{d}{d\alpha}\alpha(2-\alpha)\right)\frac{1}{(1-\alpha)^2} + \alpha(2-\alpha)\left(\frac{d}{d\alpha}\frac{1}{(1-\alpha)^2}\right) \\
&= \frac{2(1-\alpha)}{(1-\alpha)^2} + \alpha(2-\alpha)\frac{2}{(1-\alpha)^3} \\
&= \frac{2(1-\alpha)(1-\alpha) + 2\alpha(2-\alpha)}{(1-\alpha)^3} \\
&= \frac{2}{(1-\alpha)^3} > 0 \quad \because \alpha \in [0,1].
\end{aligned}$$

Then, the correlation monotonically decreases from λ to 0, as α approaches from 0 to 1.

(c)

In the LAR algorithm, $y - u(\alpha)$ is gradually decreased by increasing α. From (a) above, the correlation between between x_j and $y - u(\alpha)$ is tied (i.e., the correlation is identical for all j) and from (b) above, the correlation monotonically decreases as α increases.

Ex. 3.28

Suppose for a given t in (3.51), the fitted lasso coefficient for variable X_j is $\hat{\beta}_j = a$. Suppose we augment our set of variables with an identical copy $X_j^* = X_j$. Characterize the effect of this exact collinearity by describing the set of solutions for $\hat{\beta}_j$ and $\hat{\beta}_j^*$, using the same value of t.

The notation in this question is a bit different from others. Let me clarify the notations first:

$$\begin{aligned}
&X \in \mathbb{R}^{\mathrm{N}\times\mathrm{p}} \\
&x_j \quad \text{the } j\text{th column of } X \\
&X_{-j} \in \mathbb{R}^{\mathrm{N}\times\mathrm{p}-1} \quad X \text{ without } x_j \\
&X^* = [X, x_j] \in \mathbb{R}^{\mathrm{N}\times\mathrm{p}+1} \\
&\hat{\beta} \quad \text{the coefficients for } X \\
&\hat{\beta}^* \quad \text{the coefficients for } X^*.
\end{aligned}$$

From the description, we know

$$\hat{\beta} = \arg\min_{\beta}(y - X\beta)^T(y - X\beta) \quad \text{subject to} \quad \sum_i |\beta_i| \leq t.$$

Then $\hat{\beta}^*$ is given by

$$\hat{\beta}^* = \arg\min_{\beta}(y - X^*\beta)^T(y - X^*\beta) \quad \text{subject to} \quad \sum_i |\beta_i| \leq t.$$

Because

$$(y - X^*\beta)^T(y - X^*\beta) = (y - X_{-j}\beta_{-j} - x_j(\beta_j + \beta_{p+1}))^T(y - X_{-j}\beta_{-j} - x_j(\beta_j + \beta_{p+1})),$$

we have

$$\hat{\beta}_i = \hat{\beta}_i^* \quad \text{for } i = 1, \ldots, j-1, j+1, \ldots, p \quad \text{and} \quad \hat{\beta}_j = \hat{\beta}_j^* + \hat{\beta}_{p+1}^*,$$

and then

$$\begin{aligned}
\sum_i |\hat{\beta}_i^*| \leq \sum_i |\hat{\beta}_i| &\Leftrightarrow |\hat{\beta}_j^*| + |\hat{\beta}_{p+1}^*| \leq |\hat{\beta}_j| \\
&\Leftrightarrow |\hat{\beta}_j^*| + |\hat{\beta}_{p+1}^*| \leq |\hat{\beta}_j^* + \hat{\beta}_{p+1}^*| \quad \because \hat{\beta}_j = \hat{\beta}_j^* + \hat{\beta}_{p+1}^* \\
&\Leftrightarrow \left(|\hat{\beta}_j^*| + |\hat{\beta}_{p+1}^*|\right)^2 \leq \left(\hat{\beta}_j^* + \hat{\beta}_{p+1}^*\right)^2 \\
&\Leftrightarrow |\hat{\beta}_j^*| \cdot |\hat{\beta}_{p+1}^*| \leq \hat{\beta}_j^* \cdot \hat{\beta}_{p+1}^* \\
&\Leftrightarrow 0 \leq \hat{\beta}_j^* \cdot \hat{\beta}_{p+1}^*.
\end{aligned}$$

Therefore,

$$\hat{\beta}_j = a = \hat{\beta}_j^* + \hat{\beta}_{p+1}^* \quad \text{and} \quad 0 \leq \hat{\beta}_j^* \cdot \hat{\beta}_{p+1}^*.$$

Ex. 3.29

Suppose we run a ridge regression with parameter λ on a single variable X, and get coefficient a. We now include an exact copy $X^* = X$, and refit our ridge regression. Show that both coefficients are identical, and derive their value. Show in general that if m copies of a variable X_j are included in a ridge regression, their coefficients are all the same.

Before solving this problem, let me revise the notation to clarify.

$$X = [x; x] \quad \text{and} \quad x \in \mathbb{R}^N.$$

Then,

$$
\begin{aligned}
\hat{\beta}^{\text{ridge}} &= (X^T X + \lambda I)^{-1} X^T y \\
&= \begin{bmatrix} x^T x + \lambda & x^T x \\ x^T x & x^T x + \lambda \end{bmatrix}^{-1} \begin{bmatrix} x^T y \\ x^T y \end{bmatrix} \\
&= \begin{bmatrix} -\left[x^T x + \lambda - x^T x (x^T x + \lambda) x^T x\right]^{-1} x^T x (x^T x + \lambda)^{-1} & \left[x^T x + \lambda - x^T x (x^T x + \lambda) x^T x\right]^{-1} \\ \left[x^T x + \lambda - x^T x (x^T x + \lambda) x^T x\right]^{-1} & -\left[x^T x + \lambda - x^T x (x^T x + \lambda) x^T x\right]^{-1} x^T x (x^T x + \lambda)^{-1} \end{bmatrix} \begin{bmatrix} x^T y \\ x^T y \end{bmatrix} \\
&\qquad\qquad \because \text{block-wise inversion}^{1} \\
&= \begin{bmatrix} (x^T x + \lambda)(\lambda^2 + 2x^T x \lambda)^{-1} & (-x^T x)(\lambda^2 + 2x^T x \lambda)^{-1} \\ (-x^T x)(\lambda^2 + 2x^T x \lambda)^{-1} & (x^T x + \lambda)(\lambda^2 + 2x^T x \lambda)^{-1} \end{bmatrix} \begin{bmatrix} x^T y \\ x^T y \end{bmatrix} \\
&= \begin{bmatrix} (\lambda + 2x^T x)^{-1} x^T y \\ (\lambda + 2x^T x)^{-1} x^T y \end{bmatrix}.
\end{aligned}
$$

Therefore,

$$\hat{\beta}_1^{\text{ridge}} = \hat{\beta}_2^{\text{ridge}}.$$

To consider a more general case, let $\tilde{X}$ be a matrix with m identical columns:

$$\tilde{X}^{(m)} = [x; x; \ldots; x] \in \mathbb{R}^{\mathrm{N} \times \mathrm{m}}.$$

Then its ridge coefficient is given by

$$\tilde{\beta}^{(m)} = \begin{bmatrix} (\lambda + m x^T x)^{-1} x^T y \\ (\lambda + m x^T x)^{-1} x^T y \\ \vdots \\ (\lambda + m x^T x)^{-1} x^T y \end{bmatrix}. \tag{C3.1}$$

The earlier derivation proves Equation C3.1 for $m = 2$. For $m > 2$, I provide proof by induction on m. Assume Equation C3.1 holds for $m - 1$:

$$\tilde{\beta}^{(m-1)} = \begin{bmatrix} (\lambda + (m-1) x^T x)^{-1} x^T y \\ (\lambda + (m-1) x^T x)^{-1} x^T y \\ \vdots \\ (\lambda + (m-1) x^T x)^{-1} x^T y \end{bmatrix} = \left(\tilde{X}^{(m-1)T} \tilde{X}^{(m-1)} + \lambda I\right)^{-1} \tilde{X}^{(m-1)T} y.$$

[1] https://en.wikipedia.org/wiki/Invertible_matrix#Blockwise_inversion

Then,

$$\begin{aligned}
\tilde{\beta}^{(m)} &= \left(\tilde{X}^{(m)T}\tilde{X}^{(m)} + \lambda I\right)^{-1}\tilde{X}^{(m)T}y \\
&= \left([\tilde{X}^{(m-1)}; x]^T[\tilde{X}^{(m-1)}; x] + \lambda I\right)^{-1}[\tilde{X}^{(m-1)}; x]^T y \\
&= \begin{bmatrix} \tilde{X}^{(m-1)T}\tilde{X}^{(m-1)} + \lambda I & \tilde{X}^{(m-1)T}x \\ x^T\tilde{X}^{(m-1)} & x^Tx + \lambda I \end{bmatrix}^{-1} \begin{bmatrix} \tilde{X}^{(m-1)T}y \\ x^Ty \end{bmatrix} \\
&= \dots \\
&= \begin{bmatrix} (\lambda + (m-1)x^Tx)(\lambda + mx^Tx)^{-1}\tilde{\beta}^{(m-1)} \\ (\lambda + mx^Tx)^{-1}x^Ty \end{bmatrix} \\
&= \begin{bmatrix} (\lambda + mx^Tx)^{-1}x^Ty \\ (\lambda + mx^Tx)^{-1}x^Ty \\ \vdots \\ (\lambda + mx^Tx)^{-1}x^Ty \end{bmatrix}.
\end{aligned}$$

Note that by repeating the identical columns, the bias of estimator decreases. The fit ($\hat{y}$) is given by

$$\hat{y}^{(m)} = \tilde{X}^{(m)}\tilde{\beta}^{(m)} = x\sum_{i=1}^{m}\tilde{\beta}_i^{(m)}$$

and without repeated columns, it is

$$\hat{y}^{(1)} = \tilde{X}^{(1)}\tilde{\beta}^{(1)} = x\tilde{\beta}^{(1)}.$$

Here,

$$\sum_{i=1}^{m}\tilde{\beta}_i^{(m)} = m(\lambda + mx^Tx)^{-1}x^Ty = (\lambda/m + x^Tx)^{-1}x^Ty > (\lambda + x^Tx)^{-1}x^Ty = \tilde{\beta}^{(1)}$$

because $\lambda > 0$, $x^Tx > 0$, and $m > 1$. Therefore, repeating the column m times is equivalent to dividing the shrinkage parameter by m.

Ex. 3.30

Consider the elastic-net optimization problem:

$$\min_{\beta} \|y - X\beta\|^2 + \lambda[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1]. \tag{3.91}$$

Show how one can turn this into a lasso problem, using an augmented version of X and y.

From Ex. 3.12, the ridge regression on X with the penalty parameter $\tilde{\lambda}$ is equivalent to the least square regression on $\tilde{X}$ where

$$\tilde{X} = \begin{bmatrix} X \\ \sqrt{\tilde{\lambda}}I \end{bmatrix} \quad \text{and} \quad \tilde{y} = \begin{bmatrix} y \\ 0 \end{bmatrix}.$$

Now, consider the lasso regression on $\tilde{X}$ with the penalty parameter θ, where the coefficient is given by

$$\begin{aligned}
\hat{\beta} &= \min_{\beta} (\tilde{y} - \tilde{X}\beta)^T (\tilde{y} - \tilde{X}\beta) + \theta\|\beta\|_1 \\
&= \min_{\beta} \left(\begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\tilde{\lambda}} I \end{bmatrix} \beta \right)^T \left(\begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\tilde{\lambda}} I \end{bmatrix} \beta \right) + \theta\|\beta\|_1 \\
&= \min_{\beta} (y - X\beta)^T (y - X\beta) + \tilde{\lambda}\|\beta\|_2^2 + \theta\|\beta\|_1 \\
&= \min_{\beta} \|(y - X\beta)\|^2 + \tilde{\lambda}\|\beta\|_2^2 + \theta\|\beta\|_1,
\end{aligned}$$

which equals to Equation 3.91 when $\tilde{\lambda} = \lambda\alpha$ and $\theta = \lambda(1 - \alpha)$.

Chapter 4. Linear Methods for Classification

Ex. 4.1

Show how to solve the generalized eigenvalue problem

$$\max_a a^T B a \quad \text{subject to} \quad a^T W a = 1$$

by transforming to a standard eigenvalue problem.

To solve the constrained maximization problems, I use the idea of Lagrangian multipliers, where the Lagrangian is

$$\mathcal{L} = a^T B a - \lambda(a^T W a - 1).$$

Then

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a} &= 0 \\ \Leftrightarrow 2Ba - \lambda 2Wa &= 0 \\ \Leftrightarrow Ba &= \lambda Wa \end{aligned}$$

If W is nonsingular, then we have $W^{-1}Ba = \lambda a$, which is a standard eigenvalue problem. The vector a is an eigenvector of the matrix $W^{-1}B$ and λ is the corresponding eigenvalue.

Ex. 4.2

Suppose we have features $x \in \mathbb{R}^p$, a two-class response, with class sizes N_1, N_2, and the target coded as $-N/N_1$, and N/N_2.

(a) Show that the LDA rule classifies to class 2 if

$$x^T \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2}(\hat{\mu}_2 + \hat{\mu}_1)^T \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) - \log N_2/N_1$$

and class 1 otherwise.

(b) Consider minimization of the least squares criterion

$$\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T \beta)^2.$$

Show that the solution $\hat{\beta}$ satisfies

$$\left[(N-2)\hat{\Sigma} + N\hat{\Sigma}_B\right]\beta = N(\hat{\mu}_2 - \hat{\mu}_1)$$

(after simplification), where $\hat{\Sigma}_B = \frac{N_1 N_2}{N^2}(\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T$.

(c) Hence show that $\hat{\Sigma}_B \beta$ is in the direction $(\hat{\mu}_2 - \hat{\mu}_1)$ and thus

$$\hat{\beta} \propto \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1).$$

Therefore the least-squares regression coefficient is identical to the LDA coefficient, up to a scalar multiple.

(d) Show that this result holds for any (distinct) coding of the two classes.

(e) Find the solution $\hat{\beta}^0$ (up to the same scalar multiple as in (c), and hence the predicted value $\hat{f}(x) = \hat{\beta}_0 + x^T\hat{\beta}$. Consider the following rule: classify to class 2 if $\hat{f}(x) > 0$ and class 1 otherwise. Show this is not the same as the LDA rule unless the classes have equal numbers of observations.

(a)

In the LDA, the ratio of two classes is given by

$$\frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} = \frac{N_2}{N_1}\frac{f_2(x)}{f_1(x)}$$

where the likelihood of class k is defined as

$$f_k(x) \propto \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right\}$$

Then,

$$\begin{aligned}
\log\frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} &= \log\frac{N_2}{N_1} + \log\frac{f_2(x)}{f_1(x)} \\
&= \log\frac{N_2}{N_1} + \log f_2(x) - \log f_1(x) \\
&= \log\frac{N_2}{N_1} - \frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\
&= \log\frac{N_2}{N_1} - \frac{1}{2}\left(-x^T\Sigma^{-1}\mu_2 - \mu_2^T\Sigma^{-1}x + \mu_2^T\Sigma^{-1}\mu_2 + x^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma^{-1}x - \mu_1^T\Sigma^{-1}\mu_1\right) \\
&= \log\frac{N_2}{N_1} - \frac{1}{2}\left(-x^T\Sigma^{-1}\mu_2 - x^T\Sigma^{-1}\mu_2 + \mu_2^T\Sigma^{-1}\mu_2 + x^T\Sigma^{-1}\mu_1 + x^T\Sigma^{-1}\mu_1 - \mu_1^T\Sigma^{-1}\mu_1\right) \\
&= \log\frac{N_2}{N_1} + \left(x^T\Sigma^{-1}(\mu_2-\mu_1)\right) - \frac{1}{2}(\mu_2+\mu_1)^T\Sigma^{-1}(\mu_2-\mu_1)
\end{aligned}$$

Therefore, x is classified into class 2 if

$$\begin{aligned}
&\frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} > 1 \\
\Leftrightarrow &\log\frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} > 0 \\
\Leftrightarrow &\log\frac{N_2}{N_1} + \left(x^T\Sigma^{-1}(\mu_2-\mu_1)\right) - \frac{1}{2}(\mu_2+\mu_1)^T\Sigma^{-1}(\mu_2-\mu_1) > 0 \\
\Leftrightarrow &\left(x^T\Sigma^{-1}(\mu_2-\mu_1)\right) > \frac{1}{2}(\mu_2+\mu_1)^T\Sigma^{-1}(\mu_2-\mu_1) - \log\frac{N_2}{N_1}.
\end{aligned}$$

(b)

First, the covariance matrix is estimated as

$$(N-2)\hat{\Sigma} = \sum_{i=1}^{N_1}(x_i-\mu_1)(x_i-\mu_1)^T + \sum_{i=N_1+1}^{N}(x_i-\mu_2)(x_i-\mu_2)^T$$

(see page 109).

Now for brevity, I organize a matrix X such that the first N_1 rows are of class 1 and the remaining N_2 rows are of class 2. I also add a vector of 1 as the first column:

$$X = \begin{bmatrix} 1 & X^{(1)} \\ 1 & X^{(2)} \end{bmatrix}$$

where $X \in \mathbb{R}^{\mathrm{N} \times (\mathrm{p}+1)}$, $X^{(1)} \in \mathbb{R}^{\mathrm{N}_1 \times \mathrm{p}}$ and $X^{(2)} \in \mathbb{R}^{\mathrm{N}_2 \times \mathrm{p}}$.

Then, the normal equation gives us

$$X^T X \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} = X^T y$$

The left hand-side of the normal equation is

$$\begin{aligned}
& X^T X \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 \\ X^{(1)T} & X^{(2)T} \end{bmatrix} \begin{bmatrix} 1 & X^{(1)} \\ 1 & X^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} \\
&= \begin{bmatrix} N_1 + N_2 & N_1\hat{\mu}_1^T + N_2\hat{\mu}_2^T \\ N_1\hat{\mu}_1 + N_2\hat{\mu}_2 & X^{(1)T}X^{(1)} + X^{(2)T}X^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} \\
&= \begin{bmatrix} \hat{\beta}_0(N_1 + N_2) + (N\hat{\mu}_1 + N_2\hat{\mu}_2)^T\hat{\beta}_1 \\ (N\hat{\mu}_1 + N_2\hat{\mu}_2)\hat{\beta}_0 + \left(X^{(1)T}X^{(1)} + X^{(2)T}X^{(2)}\right)\hat{\beta} \end{bmatrix}
\end{aligned}$$

Now,

$$\begin{aligned}
& X^{(1)T}X^{(1)} + X^{(2)T}X^{(2)} \\
&= \sum_{i=1}^{N_1} x_i x_i^T + \sum_{i=N_1+1}^{N} x_i x_i^T \\
&= \sum_{i=1}^{N_1} \left[(x_i - \hat{\mu}_1)(x_i - \hat{\mu}_1)^T + x_i\hat{\mu}_1^T + \hat{\mu}_1 x_i^T - \hat{\mu}_1\hat{\mu}_1^T\right] \\
&\quad + \sum_{i=N_1+1}^{N} \left[(x_i - \hat{\mu}_2)(x_i - \hat{\mu}_2)^T + x_i\hat{\mu}_2^T + \hat{\mu}_2 x_i^T - \hat{\mu}_2\hat{\mu}_2^T\right] \\
&= \sum_{i=1}^{N_1} \left[(x_i - \hat{\mu}_1)(x_i - \hat{\mu}_1)^T\right] + N_1\hat{\mu}_1\hat{\mu}_1^T + N_1\hat{\mu}_1\hat{\mu}_1^T - N_1\hat{\mu}_1\hat{\mu}_1^T \\
&\quad + \sum_{i=N_1+1}^{N} \left[(x_i - \hat{\mu}_2)(x_i - \hat{\mu}_2)^T\right] + N_2\hat{\mu}_2\hat{\mu}_2^T + N_2\hat{\mu}_2\hat{\mu}_2^T - N_2\hat{\mu}_2\hat{\mu}_2^T \\
&= (N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T
\end{aligned}$$

Therefore,

$$X^T X \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \hat{\beta}_0 + (N_1\hat{\mu}_1 + N_2\hat{\mu}_2)^T\hat{\beta}_1 \\ (N\hat{\mu}_1 + N_2\hat{\mu}_2)\hat{\beta}_0 + \left((N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T\right)\hat{\beta} \end{bmatrix}$$

The right hand-side of the normal equation is

$$X^T y = \begin{bmatrix} 1 & 1 \\ X^{(1)T} & X^{(2)T} \end{bmatrix} \begin{bmatrix} -N/N_1 \\ -N/N_1 \\ \vdots \\ -N/N_1 \\ N/N_2 \\ N/N_2 \\ \vdots \\ N/N_2 \end{bmatrix} = \begin{bmatrix} -N+N \\ -N\hat{\mu}_1 + N\hat{\mu}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ N(\hat{\mu}_2 - \hat{\mu}_1) \end{bmatrix},$$

because from the description, $y_1 = y_2 = \cdots = y_{N_1} = -N/N_1$ and $y_{N_1+1} = y_{N_1+2} = \cdots = y_N = N/N_2$.

Putting them together,

$$\begin{aligned} & X^T X \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} = X^T y \\ \Leftrightarrow & \begin{bmatrix} \hat{\beta}_0 + (N_1\hat{\mu}_1 + N_2\hat{\mu}_2)^T\hat{\beta}_1 \\ (N\hat{\mu}_1 + N_2\hat{\mu}_2)\hat{\beta}_0 + \left((N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T\right)\hat{\beta} \end{bmatrix} = \begin{bmatrix} 0 \\ N(\hat{\mu}_2 - \hat{\mu}_1) \end{bmatrix} \\ \Leftrightarrow & \begin{cases} \hat{\beta}_0 + (N_1\hat{\mu}_1 + N_2\hat{\mu}_2)^T\hat{\beta}_1 = 0 \\ (N_1\hat{\mu}_1 + N_2\hat{\mu}_2)\hat{\beta}_0 + \left((N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T\right)\hat{\beta} = N(\hat{\mu}_2 - \hat{\mu}_1) \end{cases} \end{aligned}$$

The solution is given by

$$\left[(N-2)\hat{\Sigma} + N\hat{\Sigma}_B\right]\beta = N(\hat{\mu}_2 - \hat{\mu}_1)$$

where $\hat{\Sigma}_B = \frac{N_1 N_2}{N^2}(\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T$.

(c)

Given the results above (b),

$$\hat{\Sigma}_B\hat{\beta} = \frac{N_1 N_2}{N^2}(\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T\hat{\beta} = Z(\hat{\mu}_2 - \hat{\mu}_1)$$

where Z is a scalar, $\frac{N_1 N_2}{N^2}(\hat{\mu}_2 - \hat{\mu}_1)^T\hat{\beta}$. Thus, $\hat{\beta}$ is in the direction of $(\hat{\mu}_2 - \hat{\mu}_1)$.

Ex. 4.3

Suppose we transform the original predictors X to $\hat{Y}$ via linear regression. In detail, let $\hat{Y} = X(X^TX)^{-1}X^TY = X\hat{B}$, where Y is the indicator response matrix. Similarly for any input $x \in \mathbb{R}^p$, we get a transformed vector $\hat{y} = \hat{B}^T x \in \mathbb{R}^K$. Show that LDA using $\hat{Y}$ is identical to LDA in the original space.

For brevity, I only consider two class case.

When LDA is applied on the original space, the log ratio is given by

$$\log\frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} = \log\frac{N_2}{N_1} + \left(x^T\Sigma^{-1}(\mu_2 - \mu_1)\right) - \frac{1}{2}(\mu_2 + \mu_1)^T\Sigma^{-1}(\mu_2 - \mu_1)$$

where the covariance matrix is

$$\Sigma = \sum_{k=1}^{K} \sum_{g_i=k} (x_i - \mu_k)(x_i - \mu_k)^T/(N-K)$$

(see page 109). Then when LDA is applied on the transformed space, the covariance is given by

$$\begin{aligned}
\hat{\Sigma} &= \sum_{k=1}^{K} \sum_{g_i=k} (\hat{B}^T x_i - \hat{B}^T \mu_k)(\hat{B}^T x_i - \hat{B}^T \mu_k)^T/(N-K) \\
&= \sum_{k=1}^{K} \sum_{g_i=k} \hat{B}^T (x_i - \mu_k)(x_i - \mu_k)^T \hat{B}/(N-K) \\
&= \hat{B}^T \Sigma \hat{B}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\log \frac{p(\text{class } 2|x)}{p(\text{class } 1|x)} &= \log \frac{N_2}{N_1} + \left((\hat{B}^T x)^T \hat{\Sigma}^{-1} (\hat{B}^T \mu_2 - \hat{B}^T \mu_1) \right) - \frac{1}{2} (\hat{B}^T \mu_2 + \hat{B}^T \mu_1)^T \hat{\Sigma}^{-1} (\hat{B}^T \mu_2 - \hat{B}^T \mu_1) \\
&= \log \frac{N_2}{N_1} + \left(x^T \hat{B} \hat{\Sigma}^{-1} \hat{B}^T (\mu_2 - \mu_1) \right) - \frac{1}{2} (\mu_2 + \mu_1)^T \hat{B} \hat{\Sigma}^{-1} \hat{B}^T (\mu_2 - \mu_1) \\
&= \log \frac{N_2}{N_1} + \left(x^T \Sigma^{-1} (\mu_2 - \mu_1) \right) - \frac{1}{2} (\mu_2 + \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1)
\end{aligned}$$

because

$$\begin{aligned}
\hat{B} \hat{\Sigma}^{-1} \hat{B}^T &= \hat{B} (\hat{B}^T \Sigma \hat{B})^{-1} \hat{B}^T \\
&= \hat{B} \hat{B}^{-1} \Sigma^{-1} \hat{B}^{-T} \hat{B}^T \\
&= \Sigma^{-1}.
\end{aligned}$$

Ex. 4.4

Consider the multilogit model with K classes (4.17). Let β be the $(p+1)(K-1)$-vector consisting of all the coefficients. Define a suitably enlarged version of the input vector x to accommodate this vectorized coefficient matrix. Derive the Newton-Raphson algorithm for maximizing the multinomial log-likelihood, and describe how you would implement this algorithm.

Before describing the enlarged input matrix, let me consider the log-likelihood function and its derivatives. Letting $\beta^{(j)}$ be the coefficients for the log-odds of the class j, I have

$$\begin{aligned}
L(\beta) &= \sum_{i=1}^{N} \log \Pr_{y_i} \left(x_i \beta^{(j)} \right) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{K-1} \mathbb{1}_{y_i=j} \log \Pr_j \left(x_i \beta^{(j)} \right) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{K-1} \left[\mathbb{1}_{y_i=j} \left(\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(j)} \right) - \log \left(1 + \sum_{k=1}^{K-1} \exp \left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\} \right) \right]
\end{aligned}$$

Here, $\mathbb{1}$ is an indicator function. Then, the derivative is

$$\begin{aligned}
\frac{\partial L(\beta)}{\partial \beta^{(k)}} &= \sum_{i=1}^{N} \left[\mathbb{1}_{y_i=k} \left(\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \right) - \frac{\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\}}{1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\}} \right] \\
&= \sum_{i=1}^{N} \left[\mathbb{1}_{y_i=k} \left(\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \right) - \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \Pr_k \left(x_i \beta^{(k)} \right) \right] \\
&= \sum_{i=1}^{N} \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \left(\mathbb{1}_{y_i=k} - \Pr_k \left(x_i \beta^{(k)} \right) \right)
\end{aligned}$$

The second derivative is

$$\begin{aligned}
&\frac{\partial^2 L(\beta)}{\partial \beta^{(k)} \partial \beta^{(k)T}} \\
&= \frac{\partial}{\partial \beta^{(k)T}} \sum_{i=1}^{N} \frac{- \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\}}{1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\}} \\
&= \sum_{i=1}^{N} - \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \left(\frac{\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\}}{1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\}} - \frac{\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\} \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\}}{\left(1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\} \right)^2} \right) \\
&= - \sum_{i=1}^{N} \Pr_k \left(x_i, \beta^{(k)} \right) \left(1 - \Pr_k \left(x_i, \beta^{(k)} \right) \right) \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T
\end{aligned}$$

and when $a \neq k$,

$$\begin{aligned}
\frac{\partial^2 L(\beta)}{\partial \beta^{(k)} \partial \beta^{(a)T}} &= \frac{\partial}{\partial \beta^{(a)T}} \sum_{i=1}^{N} \frac{- \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\}}{1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\}} \\
&= \sum_{i=1}^{N} - \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T \left(- \frac{\begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(k)} \right\} \exp\left\{ \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \beta^{(a)} \right\}}{\left(1 + \sum_{j=1}^{K-1} \exp\left\{ \begin{bmatrix} 1 & x_{j,\cdot} \end{bmatrix} \beta^{(j)} \right\} \right)^2} \right) \\
&= \sum_{i=1}^{N} \Pr_k \left(x_i, \beta^{(k)} \right) \Pr_a \left(x_i, \beta^{(a)} \right) \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix} \begin{bmatrix} 1 & x_{i,\cdot} \end{bmatrix}^T .
\end{aligned}$$

Now I define the enlarged input matrix, $\tilde{X}$, as below.

$$\tilde{X} = \begin{bmatrix} 1 & X & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & X & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & X \end{bmatrix} \quad \in \mathbb{R}^{\mathrm{N(K-1)} \times \mathrm{(p+1)(K-1)}}.$$

I also define the enlarged output matrices:

$$\tilde{y} = \begin{bmatrix} \mathbb{1}_{y=1} \\ \mathbb{1}_{y=2} \\ \vdots \\ \mathbb{1}_{y=K-1} \end{bmatrix} \quad \in \mathbb{R}^{\mathrm{N(K-1)}}, \quad \text{and} \quad \tilde{\Pr} = \begin{bmatrix} \Pr_1 \\ \Pr_2 \\ \vdots \\ \Pr_{K-1} \end{bmatrix} \quad \in \mathbb{R}^{\mathrm{N(K-1)}}.$$

Then, the first derivative can be written as

$$\frac{\partial L(\beta)}{\partial \beta} = \tilde{X}^T(\tilde{y} - \tilde{\Pr})$$

and the second derivative can be written as

$$\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} = -\tilde{X}^T \tilde{W} \tilde{X},$$

where $\tilde{W} \in \mathbb{R}^{\mathrm{N(K-1)\times N(K-1)}}$ is given by

$$\tilde{W} = \begin{bmatrix} \mathrm{diag}(\Pr_1 \odot (1 - \Pr_1)) & \mathrm{diag}(-\Pr_1 \odot \Pr_2) & \cdots & \mathrm{diag}(-\Pr_1 \odot \Pr_{K-1}) \\ \mathrm{diag}(-\Pr_2 \odot \Pr_1) & \mathrm{diag}(\Pr_2 \odot (1 - \Pr_2)) & \cdots & \mathrm{diag}(-\Pr_2 \odot \Pr_{K-1}) \\ \vdots & & & \\ \mathrm{diag}(-\Pr_{K-1} \odot \Pr_1) & \mathrm{diag}(-\Pr_{K-1} \odot \Pr_2) & \cdots & \mathrm{diag}(\Pr_{K-1} \odot (1 - \Pr_{K-1})) \end{bmatrix}.$$

Here $\odot$ indicates the element-wise multiplication.

Thus, a single Newton update[2]is

$$\begin{aligned} \beta^{\mathrm{new}} &= \beta^{\mathrm{old}} - \left(\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial L(\beta)}{\partial \beta} \\ &= \beta^{\mathrm{old}} + \left(\tilde{X}^T \tilde{W} \tilde{X}\right)^{-1} \tilde{X}^T (\tilde{y} - \tilde{\Pr}) \\ &= \left(\tilde{X}^T \tilde{W} \tilde{X}\right)^{-1} \tilde{X}^T \tilde{W} \left(\tilde{X} \beta^{\mathrm{old}} + \tilde{W}^{-1} (\tilde{y} - \tilde{\Pr})\right) \end{aligned}$$

Please note that when $K = 2$, this update is identical to Equation 4.26 in the book.

Ex. 4.5

Consider a two-class logistic regression problem with $x \in \mathbb{R}$. Characterize the maximum-likelihood estimates of the slope and intercept parameter if the sample x_i for the two classes are separated by a point $x_0 \in \mathbb{R}$. Generalize this result to (a) $x \in \mathbb{R}^{\mathrm{p}}$ (see Figure 4.16), and (b) more than two classes.

As the two classes are separated by a point x_0, without the loss of generalisation I can write

$$y = \begin{cases} 1 & \text{if } 0 < x_0 - x \\ 2 & \text{if } x_0 - x < 0. \end{cases}$$

Now, let

$$\Pr_1 = \frac{\exp\{\beta_0 + \beta_1 x\}}{1 + \exp\{\beta_0 + \beta_1 x\}} \quad \text{and} \quad \Pr_2 = \frac{1}{1 + \exp\{\beta_0 + \beta_1 x\}.}$$

[2]see also

Böhning, D. (1992). Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44, 197–200. `https://www.ism.ac.jp/editsec/aism/pdf/044_1_0197.pdf`.

Hasan, A., Zhiyu, W., & Mahani, A. S. (2014). Fast estimation of multinomial logit models: R package mnlogit. arXiv:1404.3177 [stat.CO]. `https://arxiv.org/abs/1404.3177`.

Then, at $a > 0$,

$$\begin{cases} \frac{\exp\{ax_0 - ax\}}{1+\exp\{ax_0-ax\}} > \frac{1}{1+\exp\{ax_0-ax\}} \Leftrightarrow \Pr_1 > \Pr_2 & \text{when } 0 < x_0 - x \Leftrightarrow y = 1 \\ \frac{\exp\{ax_0 - ax\}}{1+\exp\{ax_0-ax\}} \leq \frac{1}{1+\exp\{ax_0-ax\}} \Leftrightarrow \Pr_1 < \Pr_2 & \text{when } x_0 - x < 0 \Leftrightarrow y = 2. \end{cases}$$

Therefore, the logistic regression gives correct classification when $\beta_0 = ax_0$ and $\beta_1 = -a$.

Now, assume $N_1 + N_2$ observations: N_1 with $x_i < x_0$ and $y = 1$; and N_2 where $x_j > x_0$ and $y = 2$. Then the log-likelihood is given by

$$\begin{aligned} L &= \sum_{i=1}^{N_1} \log\left[\frac{\exp\{ax_0 - ax_i\}}{1+\exp\{ax_0 - ax_i\}}\right] + \sum_{i=N_1+1}^{N_1+N_2} \log\left[\frac{1}{1+\exp\{ax_0 - ax_i\}}\right] \\ &= \sum_{i=1}^{N_1} \left[ax_0 - ax_i - \log\left(1+\exp\{ax_0 - ax_i\}\right)\right] - \sum_{i=N_1+1}^{N_1+N_2} \log\left(1+\exp\{ax_0 - ax_i\}\right) \end{aligned}$$

Its derivative with respect to a is

$$\begin{aligned} \frac{\partial L}{\partial a} &= \sum_{i=1}^{N_1} \left[x_0 - x_i - \frac{(x_0 - x_i)\exp\{ax_0 - ax_i\}}{1+\exp\{ax_0 - ax_i\}}\right] - \sum_{i=N_1+1}^{N_1+N_2} \frac{(x_0 - x_i)\exp\{ax_0 - ax_i\}}{1+\exp\{ax_0 - ax_i\}} \\ &= \sum_{i=1}^{N_1} \left[|x_0 - x_i|\left(1 - \frac{\exp\{ax_0 - ax_i\}}{1+\exp\{ax_0 - ax_i\}}\right)\right] + \sum_{i=N_1+1}^{N_1+N_2} \left[|x_0 - x_i|\frac{\exp\{ax_0 - ax_i\}}{1+\exp\{ax_0 - ax_i\}}\right] \\ &> 0. \end{aligned}$$

Thus, as a increases the log-likelihood monotonically increases. Here, the maximum likelihood estimate does not exist, and as a approaches the infinity,

$$\begin{aligned} \beta_0 &\to \begin{cases} -\infty & \text{when } x_0 < 0 \\ 0 & \text{when } x_0 = 0 \\ \infty & \text{when } x_0 > 0 \end{cases} \\ \beta_1 &\to \infty. \end{aligned}$$

(a)

In general, when two classes are separated by $x^* \in \mathbb{R}^p$, there exists a vector γ which satisfies the followings:

$$ax_i\gamma > 0 \quad \text{when } y_i = 1 \quad \text{and} \quad ax_i\gamma < 0 \quad \text{when } y_i = 2$$

with $a > 0$. In the earlier consideration where $x^* = x_0 \in \mathbb{R}$, $\gamma = [x_0, -1]^T$.

Then the derivative of log-likelihood with respect to a is always positive. Thus the maximum likelihood estimate does not exist, and the parameters are estimated to be zero or approaches to the positive or negative infinity. The derivation is essentially identical to the one given above.

(b)

When $K > 2$ classes are separated, there exist vectors that satisfy

$$ax_i\gamma^{(k)} > 0 \quad \text{when } y_i = k \quad \text{for } k = 1, 2, \ldots, K-1.$$

Then again, the derivative of log-likelihood with respect to a is always positive. As the log-likelihood increases, the parameters stay at zero or approaches to the positive or negative infinity. The derivation is again essentially the same as the one above.

Ex. 4.6

Suppose we have N points x_i in $\mathbb{R}^p$ in general position, with class labels $y_i \in \{-1, 1\}$. Prove that the perceptron learning algorithm converges to a separating hyperplane in a finite number of steps:

(a) Denote a hyperplane by $f(x) = \beta_1^T x + \beta_0 = 0$, or in more compact notation $\beta^T x^* = 0$, where $x^* = (x, 1)$ and $\beta = (\beta_1, \beta_0)$. Let $z_i = x_i^*/\|x_i^*\|$. Show that separability implies the existence of a β_{sep} such that $y_i \beta_{sep}^T z_i \geq 1 \forall i$.

(b) Given a current β_{old}, the perceptron algorithm identifies a point z_i that is misclassified, and produces the update $\beta_{new} \leftarrow \beta_{old} + y_i z_i$. Show that $\|\beta_{new} - \beta_{sep}\|^2 \leq \|\beta_{old} - \beta_{sep}\|^2 - 1$, and hence that the algorithm converges to a separating hyperplane in no more than $\|\beta_{start} - \beta_{sep}\|^2$ steps (Ripley, 1996).

(a)

The separability implies that

$$\beta^T z_i > 0 \quad \text{if } y_i = 1 \quad \text{and} \quad \beta^T z_i < 0 \quad \text{if } y_i = -1$$

and thus,

$$y_i \beta^T z_i > 0.$$

Then, there exists $a \in \mathbb{R}$ such that

$$y_i \beta^T z_i \geq a > 0.$$

Therefore,

$$y_i \frac{1}{a} \beta^T z_i \geq 1$$

and by setting $\beta_{sep} = \frac{1}{a}\beta$, we obtain

$$y_i \beta_{sep}^T z_i \geq 1.$$

(b)

From $\beta_{new} = \beta_{old} - y_i z_i$, we obtain

$$\beta_{new} - \beta_{sep} = \beta_{old} - \beta_{sep} - y_i z_i.$$

Then

$$\begin{aligned}
(\beta_{new} - \beta_{sep})^T (\beta_{new} - \beta_{sep}) &= (\beta_{old} - \beta_{sep} - y_i z_i)^T (\beta_{old} - \beta_{sep} - y_i z_i) \\
\Leftrightarrow \|\beta_{new} - \beta_{sep}\|^2 &= (\beta_{old} - \beta_{sep})^T (\beta_{old} - \beta_{sep}) - 2 y_i z_i^T (\beta_{old} - \beta_{sep}) + y_i^2 z_i^T z_i \\
\Leftrightarrow \|\beta_{new} - \beta_{sep}\|^2 &= (\beta_{old} - \beta_{sep})^T (\beta_{old} - \beta_{sep}) - 2 y_i (\beta_{old} - \beta_{sep})^T z_i + 1
\end{aligned}$$

because $y_i \in \{-1, 1\}$ and $z_i^T z_i = 1$. Now as z_i is misclassified, we have $y_i \beta_{old}^T z_i < 0$ and from (a), we have $y_i \beta_{sep} z_i \geq 1$. Thus,

$$y_i (\beta_{old} - \beta_{sep})^T z_i = y_i \beta_{old}^T z_i - y_i \beta_{sep}^T z_i \geq 1.$$

Therefore,

$$\begin{aligned}
\|\beta_{new} - \beta_{sep}\|^2 &= (\beta_{old} - \beta_{sep})^T (\beta_{old} - \beta_{sep}) - 2 y_i (\beta_{old} - \beta_{sep})^T z_i + 1 \\
&\leq \|\beta_{old} - \beta_{sep}\|^2 - 1.
\end{aligned}$$

Now, because $0 \leq \|\beta_{new} - \beta_{sep}\|^2$, the algorithm has to converge in no more than $\|\beta_{start} - \beta_{sep}\|^2$ steps.

Chapter 7. Model Assessment and Selection

Ex. 7.1

Derive the estimate of in-sample error (7.24).

Equation (7.24) is

$$\mathbb{E}_y\left(\mathrm{Err}_{\mathrm{in}}\right) = \mathbb{E}_y\left(\overline{\mathrm{err}}\right) + 2\frac{d}{N}\sigma_\epsilon^2$$

for a linear additive error model: $f(X) = X\beta$ and $y = f(X) + \epsilon$.

To derive this equation, I start with the expected optimism:

$$\begin{aligned}
\mathbb{E}_y\omega &= \mathbb{E}_y\left(\mathrm{Err}_{\mathrm{in}}\right) - \mathbb{E}_y\left(\overline{\mathrm{err}}\right) \\
&= \mathbb{E}_y\left(\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{y^0}\left[L(y_i^0, \hat{f}(x_i))\right]\right) - \mathbb{E}_y\left(\frac{1}{N}\sum_{i=1}^{N}\left[L(y_i, \hat{f}(x_i))\right]\right) \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}_y\mathbb{E}_{y^0}\left[L(y_i^0, \hat{f}(x_i))\right] - \mathbb{E}_y\left[L(y_i, \hat{f}(x_i))\right]\right)
\end{aligned}$$

where y^0 indicates a new response value at the training point x_i (page 228).

For the squared loss function, the expected optimism is given by

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}_y\mathbb{E}_{y^0}\left[L(y_i^0, \hat{f}(x_i))\right] - \mathbb{E}_y\left[L(y_i, \hat{f}(x_i))\right]\right) \\
&= \frac{1}{N}\left(\mathbb{E}_y\mathbb{E}_{y^0}\left[\left(y^0 - \hat{y}\right)^T\left(y^0 - \hat{y}\right)\right] - \mathbb{E}_y\left[(y - \hat{y})^T(y - \hat{y})\right]\right) \\
&= \frac{1}{N}\left(\mathbb{E}_{y^0}\left[y^{0T}y^0\right] - 2\mathbb{E}_y\mathbb{E}_{y^0}\left[y^{0T}\hat{y}\right] + \mathbb{E}_y\left[\hat{y}^T\hat{y}\right] - \mathbb{E}_y\left[y^Ty\right] + 2\mathbb{E}_y\left[y^T\hat{y}\right] - \mathbb{E}_y\left[\hat{y}^T\hat{y}\right]\right) \\
&= \frac{2}{N}\left(\mathbb{E}_y\left[y^T\hat{y}\right] - \mathbb{E}_y\mathbb{E}_{y^0}\left[y^{0T}\hat{y}\right]\right) \\
&= \frac{2}{N}\left(\mathbb{E}_y\left[y^T\hat{y}\right] - \mathbb{E}_y\mathbb{E}_{y^0}\left[y^{0T}\right]\mathbb{E}_y\mathbb{E}_{y^0}\left[\hat{y}\right]\right) \\
&= \frac{2}{N}\left(\mathbb{E}_y\left[y^T\hat{y}\right] - \mathbb{E}_{y^0}\left[y^{0T}\right]\mathbb{E}_y\left[\hat{y}\right]\right) \\
&= \frac{2}{N}\left(\mathbb{E}_y\left[y^T\hat{y}\right] - \mathbb{E}_y\left[y^T\right]\mathbb{E}_y\left[\hat{y}\right]\right) \\
&= \frac{2}{N}\sum_{i=1}^{N}\mathrm{Cov}(\hat{y}_i, y_i),
\end{aligned}$$

which proves Equations (7.21) and (7.22).

Now, I assume the model is linear regression, where $\hat{y} = X(X^TX)^{-1}X^Ty$ and $\mathrm{Cov}(y_i, y_i) = \sigma_\epsilon^2$ for all i. Then,

$$\begin{aligned}
\mathrm{Cov}(\hat{y}, y) &= \mathrm{Cov}(X(X^TX)^{-1}X^Ty, y) \\
&= X(X^TX)^{-1}X^T\mathrm{Cov}(y, y) \\
&= X(X^TX)^{-1}X^T\sigma_\epsilon^2.
\end{aligned}$$

Thus,

$$\begin{aligned}
\frac{2}{N}\sum_{i=1}^{N}\mathrm{Cov}(\hat{y}_i, y_i) &= \frac{2}{N}\mathrm{trace}(\mathrm{Cov}(\hat{y}, y)) \\
&= \frac{2}{N}\mathrm{trace}(X(X^TX)^{-1}X^T\sigma_\epsilon^2) \\
&= \frac{2}{N}\mathrm{trace}(X(X^TX)^{-1}X^T)\sigma_\epsilon^2 \\
&= \frac{2}{N}\mathrm{trace}((X^TX)^{-1}X^TX)\sigma_\epsilon^2 \quad \because \text{the cyclic property of trace} \\
&= \frac{2}{N}\mathrm{trace}(I_d)\sigma_\epsilon^2 \quad \because X \in \mathbb{R}^{\mathrm{N}\times\mathrm{d}} \\
&= \frac{2}{N}d\sigma_\epsilon^2.
\end{aligned}$$

To summarise, the above derivation proves

$$\mathbb{E}_y\left(\mathrm{Err}_{\mathrm{in}}\right) - \mathbb{E}_y\left(\overline{\mathrm{err}}\right) = \frac{2}{N}d\sigma_\epsilon^2$$

and hence Equation (7.24) for a linear regression.

Ex. 7.2

(a) For 0-1 loss with $Y \in \{0,1\}$ and $\Pr(Y = 1|x_0) = f(x_0)$, show that

$$\begin{aligned}
\mathrm{Err}(x_0) &= \Pr(Y \neq \hat{G}(x_0)|X = x_0) \\
&= \mathrm{Err}_B(x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0),
\end{aligned}$$

where $\hat{G}(x_0) = I(\hat{f}(x_0) > \frac{1}{2})$, $G(x_0) = I(f(x_0) > \frac{1}{2})$ is the Bayes classifier, and $\mathrm{Err}_B(x_0) = \Pr(Y \neq G(x_0)|X = x_0)$, the irreducible Bayes error at x_0.

(b) Using the approximation $\hat{f}(x_0) \sim N(\mathbb{E}\hat{f}(x_0), \mathrm{Var}\hat{f}(x_0))$, show that

$$\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0) \approx \Phi\left(\frac{\mathrm{sign}(\frac{1}{2} - f(x_0))(\mathbb{E}\hat{f}(x_0) - \frac{1}{2})}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right).$$

In the above,

$$\Phi(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t}\exp(-t^2/2)dt,$$

the cumulative Gaussian distribution function. This is an increasing function, with value 0 at $t = -\infty$ and value 1 and $t = \infty$.

We can think of $\mathrm{sign}(\frac{1}{2} - f(x_0))(\mathbb{E}\hat{f}(x_0) - \frac{1}{2})$ as a kind of boundary-bias term, as it depends on the true $f(x_0)$ only through which side of the boundary ($\frac{1}{2}$) that it lies. Notice also that the bias and variance combine in a multiplicative rather than additive fashion. If $\mathbb{E}\hat{f}(x_0)$ is on the same side of $\frac{1}{2}$ as $f(x_0)$, then the bias is negative, and decreasing the variance will decrease the misclassification error. On the other hand, if $\mathbb{E}\hat{f}(x_0)$ is on the opposite side of $\frac{1}{2}$ to $f(x_0)$, then the bias is positive and it pays to increase the variance! Such an increase will improve the chance that $\hat{f}(x_0)$ falls on the correct side of $\frac{1}{2}$ (Friedman, 1997).

(a)

Note that

$$\begin{aligned}
&\mathrm{Err}(x_0)\\
&= \Pr(Y \neq \hat{G}(x_0)|X = x_0)\\
&= \Pr(Y = 0|X = x_0)\Pr(G(x_0) = 0|X = x_0)\Pr(\hat{G}(x_0) = 1|X = x_0)\\
&\quad + \Pr(Y = 0|X = x_0)\Pr(G(x_0) = 1|X = x_0)\Pr(\hat{G}(x_0) = 1|X = x_0)\\
&\quad + \Pr(Y = 1|X = x_0)\Pr(G(x_0) = 1|X = x_0)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&\quad + \Pr(Y = 1|X = x_0)\Pr(G(x_0) = 0|X = x_0)\Pr(\hat{G}(x_0) = 0|X = x_0).
\end{aligned}$$

Now I consider the above derivation in two separate cases: when $f(x_0) > \frac{1}{2}$ and when $f(x_0) < \frac{1}{2}$.

When $f(x_0) > \frac{1}{2} \Leftrightarrow G(x_0) = 1$,

$$\begin{aligned}
&\mathrm{Err}(x_0)\\
&= \Pr(Y = 0|X = x_0)\Pr(\hat{G}(x_0) = 1|X = x_0) + \Pr(Y = 1|X = x_0)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&= (1 - f(x_0))(1 - \Pr(\hat{G}(x_0) = 0|X = x_0)) + f(x_0)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&= 1 - f(x_0) + (2f(x_0) - 1)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&= \Pr(Y = 0|X = x_0) + (2f(x_0) - 1)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&= \Pr(Y \neq G(x_0)|X = x_0) + (2f(x_0) - 1)\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)\\
&= \Pr(Y \neq G(x_0)|X = x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)\\
&= \mathrm{Err}_B(x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)
\end{aligned}$$

because $f(x_0) > \frac{1}{2}$.

Similarly when $f(x_0) < \frac{1}{2} \Leftrightarrow G(x_0) = 0$,

$$\begin{aligned}
&\mathrm{Err}(x_0)\\
&= \Pr(Y = 0|X = x_0)\Pr(\hat{G}(x_0) = 1|X = x_0) + \Pr(Y = 1|X = x_0)\Pr(\hat{G}(x_0) = 0|X = x_0)\\
&= (1 - f(x_0))\Pr(\hat{G}(x_0) = 1|X = x_0) + f(x_0)(1 - \Pr(\hat{G}(x_0) = 1|X = x_0))\\
&= f(x_0) + (1 - 2f(x_0))\Pr(\hat{G}(x_0) = 1|X = x_0)\\
&= \Pr(Y = 1|X = x_0) + (1 - 2f(x_0))\Pr(\hat{G}(x_0) = 1|X = x_0)\\
&= \Pr(Y \neq G(x_0)|X = x_0) + (1 - 2f(x_0))\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)\\
&= \Pr(Y \neq G(x_0)|X = x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)\\
&= \mathrm{Err}_B(x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)
\end{aligned}$$

because $f(x_0) < \frac{1}{2}$.

Therefore, $\mathrm{Err}(x_0) = \mathrm{Err}_B(x_0) + |2f(x_0) - 1|\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0)$.

(b)

When $f(x_0) > \frac{1}{2} \Leftrightarrow G(x_0) = 1$,

$$\begin{aligned}
\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0) &= \Pr(\hat{G}(x_0) = 0|X = x_0) \\
&= \Pr\left(\hat{f}(x_0) < \frac{1}{2}\right) \\
&= \Pr\left(\frac{\hat{f}(x_0) - \mathbb{E}\hat{f}(x_0)}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}} < \frac{\frac{1}{2} - \mathbb{E}\hat{f}(x_0)}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right) \\
&\approx \Phi\left(\frac{\frac{1}{2} - \mathbb{E}\hat{f}(x_0)}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right) \\
&= \Phi\left(\frac{\mathrm{sign}(\frac{1}{2} - f(x_0))(\mathbb{E}\hat{f}(x_0) - \frac{1}{2})}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right).
\end{aligned}$$

Similarly when $f(x_0) < \frac{1}{2} \Leftrightarrow G(x_0) = 0$,

$$\begin{aligned}
\Pr(\hat{G}(x_0) \neq G(x_0)|X = x_0) &= \Pr(\hat{G}(x_0) = 1|X = x_0) \\
&= \Pr\left(\hat{f}(x_0) > \frac{1}{2}\right) \\
&= \Pr\left(\frac{\mathbb{E}\hat{f}(x_0) - \hat{f}(x_0)}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}} < \frac{\mathbb{E}\hat{f}(x_0) - \frac{1}{2}}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right) \\
&\approx \Phi\left(\frac{\mathbb{E}\hat{f}(x_0) - \frac{1}{2}}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right) \\
&= \Phi\left(\frac{\mathrm{sign}(\frac{1}{2} - f(x_0))(\mathbb{E}\hat{f}(x_0) - \frac{1}{2})}{\sqrt{\mathrm{Var}(\hat{f}(x_0))}}\right).
\end{aligned}$$

Ex. 7.4

Consider the in-sample prediction error (7.18) and the training error err in the case of squared-error loss:

$$\mathrm{Err}_{in} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{Y^0}(Y_i^0 - \hat{f}(x_i))^2$$

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{f}(x_i))^2.$$

Add and subtract $f(x_i)$ and $\mathbb{E}\hat{f}(x_i)$ in each expression and expand. Hence establish that the average optimism in the training error is

$$\frac{2}{N}\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i)$$

as given in (7.21).

This exercise is solved as part of Ex. 7.1 above.

Ex. 7.6

Show that for an additive-error model, the effective degrees-of-freedom for the k-nearest-neighbors regression fit is N/k.

The k-nearest-neighbors regression method can be written as

$$\hat{y} = Sy$$

where S is a $N \times N$ square matrix:

$$S_{ij} = \begin{cases} 1/k & \text{if } x_j \text{ within the } k \text{ nearest neighbours of } x_i \\ 0 & \text{otherwise.} \end{cases}$$

Then, the effective degrees-of-freedom is given by

$$\mathrm{df}(S) = \mathrm{trace}(S) = N/k$$

(see equation (7.32) in page 232).

Ex. 7.7

Use the approximation $1/(1-x)^2 \approx 1+2x$ to expose the relationship between C_p / AIC (7.26) and GCV (7.52), the main difference being the model used to estimate the noise variance σ_ϵ^2.

Equation (7.26) in page 230 defines C_p statistic:

$$C_p = \overline{\mathrm{err}} + 2\frac{d}{N}\hat{\sigma}_\epsilon^2,$$

where $\overline{\mathrm{err}}$ indicates the training error:

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)).$$

For Gaussian models, this C_p statistics is equivalent to the AIC (page 231).

Equation (7.52) in page 244 defines GCV (generalised cross-validation) approximation:

$$GCV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{f}(x_i)}{1 - \mathrm{trace}(S)/N}\right]^2$$

where S is a hat matrix.

Using the approximation $1/(1-\mathrm{trace}(S)/N)^2 = 1 + 2\mathrm{trace}(S)/N$, the GCV is

$$\begin{aligned} GCV(\hat{f}) &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{f}(x_i)}{1 - \mathrm{trace}(S)/N}\right]^2 \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\left[\Big(y_i - \hat{f}(x_i)\Big)^2\left(1 + 2\frac{\mathrm{trace}(S)}{N}\right)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{f}(x_i)\Big)^2 + 2\frac{\mathrm{trace}(S)}{N}\frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{f}(x_i)\Big)^2. \end{aligned}$$

Then assuming the squared loss function, we obtain

$$GCV(\hat{f}) = \overline{\text{err}} + 2\frac{\text{trace}(S)}{N}\frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{f}(x_i)\Big)^2.$$

And assuming $\hat{f}$ is low-bias model, we obtain

$$GCV(\hat{f}) = \overline{\text{err}} + 2\frac{\text{trace}(S)}{N}\hat{\sigma}_\epsilon^2.$$

Thus when $\text{trace}(S) = d$, for example a linear regression (see Ex. 7.1), the GCV is approximately equivalent to the C_p statistic. The main difference is that in the GCV, the error variance σ_ϵ^2 is estimated with the model being evaluated, while in the C_p, the error variance can be estimated with another model.

Ex. 7.9

For the prostate data of Chapter 3, carry out a best-subset linear regression analysis, as in Table 3.3 (third column from left). Compute the AIC, BIC, five- and tenfold cross-validation, and bootstrap .632 estimates of prediction error. Discuss the results.

For this exercise, I took the best-subset linear regression model from Table 3.3, and calculated the following statistics. The bootstrap .632 estimate is based on 100 bootstrap samples.

AIC	0.606
BIC	76.614
5-hold cross-validation	0.583
10-hold cross-validation	0.595
leave-one-out cross-validation	0.605
bootstrap .632 estimate	0.601

Apart from BIC, all the statistics have somewhat similar values. Perhaps the most notable is the similarity between AIC and the leave-one-out cross-validation (LOO-CV). As discussed in Ex. 7.7, AIC and GCV, which approximates the LOO-CV, are closely related. Thus the similarity between AIC and LOO-CV is confirmatory to Ex. 7.7 result.

Chapter 10. Boosting and Additive Trees

Ex. 10.1

Derive expression (10.12) for the update parameter in AdaBoost.

Expression (10.12) is the

$$\beta_m = \frac{1}{2}\log\frac{1-\text{err}_m}{\text{err}_m},$$

where the error is given by

$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}$$

and the classifier is an additive model:

$$f(x) = \sum_{m=1}^M \beta_m G_m(x).$$

To obtain β_m we need to solve

$$\beta_m, G_m = \arg\min_{\beta,G} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta G(x_i)).$$

With the exponential loss $L(y, f(x)) = \exp(-yf(x))$, this minimisation can be expressed as

$$\begin{aligned}\beta_m, G_m &= \arg\min_{\beta,G} \sum_{i=1}^N \exp\left(-y_i[f_{m-1}(x_i) + \beta G(x_i)]\right) \\ &= \arg\min_{\beta,G} \sum_{i=1}^N w_i^{(m)} \exp\left(-\beta y_i G(x_i)\right)\end{aligned}$$

with $w_i^{(m)} = \exp\left(-y_i f_{m-1}(x_i)\right)$.

Now, note $y \in \{-1, 1\}$ and $G_m(x) \in \{-1, 1\}$, and thus,

$$y_i G(x_i) = \begin{cases} 1 & \text{if } y_i = G(x_i) \quad \Leftrightarrow \quad 1 - I(y_i \neq G(x_i)) = 1 \\ -1 & \text{if } y_i \neq G(x_i) \quad \Leftrightarrow \quad I(y_i \neq G(x_i)) = 1. \end{cases}$$

Then,

$$\begin{aligned}\sum_{i=1}^N w_i^{(m)} \exp\left(-\beta y_i G(x_i)\right) &= \sum_{i=1}^N w_i^{(m)} \left[I(y_i = G(x_i)) \exp(-\beta) + I(y_i \neq G(x_i)) \exp(\beta)\right] \\ &= \left(\exp(\beta) - \exp(-\beta)\right) \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) + \exp(-\beta) \sum_{i=1}^N w_i^{(m)}.\end{aligned}$$

By differentiating the above with respect to β and equating it to zero, we obtain

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \beta_m}\left[\sum_{i=1}^{N} w_i^{(m)} \exp\left(-\beta_m y_i G(x_i)\right)\right] \\
&\Leftrightarrow 0 = \beta_m \left(\exp(\beta_m) + \exp(-\beta_m)\right) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) - \beta_m \exp(-\beta_m) \sum_{i=1}^{N} w_i^{(m)} \\
&\Leftrightarrow 0 = \left(\exp(\beta_m) + \exp(-\beta_m)\right) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) - \exp(-\beta_m) \sum_{i=1}^{N} w_i^{(m)} \quad \because \beta_m \neq 0 \\
&\Leftrightarrow 0 = \exp(\beta_m) \left(\exp(\beta_m) + \exp(-\beta_m)\right) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) - \exp(\beta_m) \exp(-\beta_m) \sum_{i=1}^{N} w_i^{(m)} \\
&\qquad \because \exp(\beta_m) \neq 0 \\
&\Leftrightarrow 0 = \left(\exp(2\beta_m) + 1\right) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) - \sum_{i=1}^{N} w_i^{(m)} \\
&\Leftrightarrow \exp(2\beta_m) + 1 = \frac{\sum_{i=1}^{N} w_i^{(m)}}{\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i))} \\
&\Leftrightarrow \exp(2\beta_m) + 1 = \frac{1}{\mathrm{err}_m} \\
&\Leftrightarrow \exp(2\beta_m) = \frac{1 - \mathrm{err}_m}{\mathrm{err}_m} \\
&\Leftrightarrow \beta_m = \frac{1}{2} \log \frac{1 - \mathrm{err}_m}{\mathrm{err}_m}
\end{aligned}
$$

which is Expression (10.12).

Ex. 10.2

Prove result (10.16), that is, the minimizer of the population version of AdaBoost criterion, is one-half of the log odds.

Result (10.16) is

$$f^*(x) = \arg\min_{f(x)} \mathbb{E}_{Y|x}\left(e^{-Yf(x)}\right) = \frac{1}{2} \log \frac{\Pr(Y=1|x)}{\Pr(Y=-1|x).}$$

To prove this, I first note that

$$\mathbb{E}_{Y|x}\left(e^{-Yf(x)}\right) = \Pr(Y=1|x)\, e^{-f(x)} + \Pr(Y=-1|x)\, e^{f(x)}.$$

Then,

$$\frac{\partial}{\partial f(x)} \mathbb{E}_{Y|x}\left(e^{-Yf(x)}\right) = -\Pr(Y=1|x)\, e^{-f(x)} + \Pr(Y=-1|x)\, e^{f(x)}.$$

Thus by assuming the convexity of the expectation with respect to $f(x)$, I obtain

$$
\begin{aligned}
0 &= \frac{\partial}{\partial f^*(x)} \mathbb{E}_{Y|x} \left(e^{-Y f^*(x)} \right) \\
&\Leftrightarrow 0 = -\Pr(Y=1|x)\, e^{-f^*(x)} + \Pr(Y=-1|x)\, e^{f^*(x)} \\
&\Leftrightarrow \Pr(Y=1|x)\, e^{-f^*(x)} = \Pr(Y=-1|x)\, e^{f^*(x)} \\
&\Leftrightarrow \log \Pr(Y=1|x) - f^*(x) = \log \Pr(Y=-1|x) + f^*(x) \\
&\Leftrightarrow f^*(x) = \frac{1}{2} \left(\log \Pr(Y=1|x) - \log \Pr(Y=-1|x) \right) \\
&\Leftrightarrow f^*(x) = \frac{1}{2} \log \frac{\Pr(Y=1|x)}{\Pr(Y=-1|x)}
\end{aligned}
$$

which proves result (10.16).

Ex. 10.3

Show that the marginal average (10.47) recovers additive and multiplicative functions (10.50) and (10.51), while the conditional expectation (10.49) does not.

Here, we consider the subvector X of $l < p$ of the input predictor variables $X^T = (X_1, X_2, \ldots, X_p)$, indexed by $S \subset \{1, 2, \ldots, p\}$, and also let C the complement set, with $S \cup C = \{1, 2, \ldots, p\}$. Then, equation (10.47) expresses the marginal average as follows:

$$\bar{f}_S(X_S) = \frac{1}{N} \sum_{i=1}^{N} f(X_S, x_{iC}),$$

where $\{x_{1C}, x_{2C}, \ldots, x_{NC}\}$ are the values of X_C occurring in the training data. In contrast, the conditional expectation is given by equation (10.49):

$$\tilde{f}_S(X_S) = \mathbb{E}(f(X_S, X_C)|X_S).$$

One of the key differences between the marginal average and the conditional expectation is that, while the marginal average accounts for all the values of X_C in the training data, the conditional expectation accounts for only the subset of X_C in the training data.

To illustrate, consider $\tilde{f}_S(x_{1S})$, assuming $x_{1S} = x_{iS}$ for $i = 2, \ldots, q$ and $x_{1S} \neq x_{jS}$ for $j = q+1, \ldots, N$. To calculate the conditional expectation, we select the subset of X_C that was observed with the same value of x_{1S} in X_S: $\{x_{1S}, x_{2S}, \ldots, x_{qS}\}$. We then take the average value of f:

$$\tilde{f}(x_{1S}) = \frac{1}{q} \sum_{i=1}^{q} f(x_{iS}, x_{iC}).$$

Now, to answer the exercise question, we consider the additive function, where $f(X) =$

$h_1(X_S) + h_2(X_C)$. Then,

$$\begin{aligned}
\bar{f}_S(X_S) &= \frac{1}{N}\sum_{i=1}^{N} f(X_S, x_{iC}) \\
&= \frac{1}{N}\sum_{i=1}^{N} h_1(X_S) + h_2(x_{iC}) \\
&= h_1(X_S) + \frac{1}{N}\sum_{i=1}^{N} h_2(x_{iC}) \\
&= h_1(X_S) + a,
\end{aligned}$$

where a is independent of X_S and can be considered to be a constant. Thus, the marginal average recovers the additive function up to an additive constant. On the other hand,

$$\begin{aligned}
\tilde{f}_S(X_S) &= \mathbb{E}(f(X_S, X_C)|X_S) \\
&= \mathbb{E}(h_1(X_S) + h_2(X_C)|X_S) \\
&= h_1(X_S) + \mathbb{E}(h_2(X_C)|X_S) \\
&= h_1(X_S) + g(X_S, X_C),
\end{aligned}$$

where g depends on X_S and cannot be considered to be a constant. Thus, the conditional expectation does not recover the additive function.

Similarly with the multiplicative function $f(X) = h_1(X_S)\, h_2(X_C)$,

$$\begin{aligned}
\bar{f}_S(X_S) &= \frac{1}{N}\sum_{i=1}^{N} f(X_S, x_{iC}) \\
&= \frac{1}{N}\sum_{i=1}^{N} h_1(X_S) h_2(x_{iC}) \\
&= h_1(X_S)\frac{1}{N}\sum_{i=1}^{N} h_2(x_{iC}) \\
&= h_1(X_S)\, a.
\end{aligned}$$

As a is a constant, the marginal average recovers the multiplicative function to to a multiplicative constant. On the other hand,

$$\begin{aligned}
\tilde{f}_S(X_S) &= \mathbb{E}(f(X_S, X_C)|X_S) \\
&= \mathbb{E}(h_1(X_S) h_2(X_C)|X_S) \\
&= h_1(X_S)\mathbb{E}(h_2(X_C)|X_S). \\
&= h_1(X_S) g(X_S, X_C)
\end{aligned}$$

As g is not a constant, the conditional expectation does not recover the multiplicative function.

Ex. 10.4

(a) Write a program implementing AdaBoost with trees.

(b) Redo the computations for the example of Figure 10.2. Plot the training error as well as test error, and discuss its behavior.

(c) Investigate the number of iterations needed to make the test error finally start to rise.

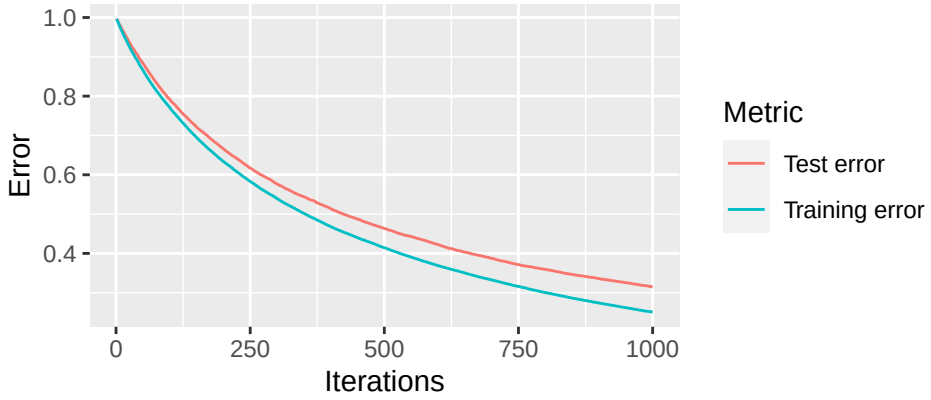
(d) Change the setup of this example as follows: define two classes, with the features in Class 1 being $X_1, X_2, \ldots, X_{10}$, standard independent Gaussian variates. In Class 2, the features $X_1, X_2, \ldots, X_{10}$ are also independent Gaussian, but conditioned on the event $\sum_j X_j^2 > 12$. Now the classes have significant overlap in feature space. Repeat the AdaBoost experiments as in Figure 10.2 and discuss the results.

(a)

I use the R package, "gbm".

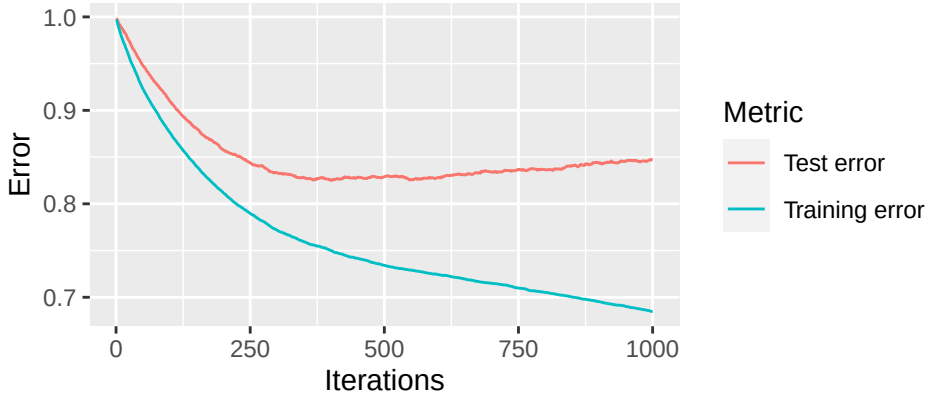
(b) and **(c)**

I have tested up to 1,000 boosting iterations (the main text examined up to 400 iterations), and both training and test errors decreased. See the figure below.



(d)

When the two classes have significant overlap in feature space, the AdaBoost shows the sign of over-fitting. After about 300 boosting iterations, the test error starts to increase while the training error continues to descrease. See the figure below.



Ex. 10.5

Multiclass exponential loss (Zhu et al., 2005). For a K-class classification problem, consider the coding $Y = (Y_1, \ldots, Y_K)^T$ with

$$Y_k = \begin{cases} 1 & \text{if } G = \mathcal{G}_k \\ -\frac{1}{K-1} & \text{otherwise.} \end{cases}$$

Let $f = (f_1, \ldots, f_K)^T$ with $\sum_{k=1}^K f_k = 0$, and define

$$L(Y, f) = \exp\left(-\frac{1}{K} Y^T f\right).$$

(a) Using Lagrange multipliers, derive the population minimizer f^* of $L(Y, f)$, subject to the zero-sum constraint, and relate these to the class probabilities.

(b) Show that a multiclass boosting using this loss function leads to a reweighting algorithm similar to Adaboost, as in Section 10.4.

(a)

The population minimizer is given by

$$f^* = \arg\min_f \mathbb{E}_Y L(Y, f) \quad \text{subject to } \sum_{k=1}^K f_k = 0.$$

The equivalent Lagrangeian form is

$$f^* = \arg\min_f \mathbb{E}_Y L(Y, f) + \lambda \sum_{k=1}^K f_k.$$

To derive the population minimizer, let me first consider the loss function. Without the loss of generality, I assume $G = G_j$.

$$\begin{aligned} L(Y, f) &= \exp\left(-\frac{1}{K} Y^T f\right) \\ &= \exp\left(-\frac{1}{K} \sum_{k=1}^K Y_j f_j\right) \\ &= \exp\left(-\frac{1}{K}\left[-\frac{K}{K-1}\sum_{k=1}^K f_j + \left(\frac{1}{K-1} + 1\right) f_j\right]\right) \\ &= \exp\left(-\frac{1}{K}\left[\frac{K}{K-1} f_j\right]\right) \\ &= \exp\left(-\frac{1}{K-1} f_j\right). \end{aligned}$$

Then, its expectation is given by

$$\mathbb{E}_Y L(Y, f) = \sum_{k=1}^K \Pr(G = \mathcal{G}_k) \exp\left(-\frac{1}{K-1} f_k\right).$$

Therefore, the derivative of Lagrangeian function can be written as

$$\frac{\partial}{\partial f_k}\left[\mathbb{E}_Y L(Y,f)+\lambda\sum_{k=1}^{K} f_k\right] = -\frac{\Pr(G=\mathcal{G}_k)}{K-1}\exp\left(-\frac{1}{K-1}f_k\right)+\lambda.$$

Setting the derivative to zero, we obtain

$$\begin{aligned}
&\frac{\partial}{\partial f_k^*}\left[\mathbb{E}_Y L(Y,f)+\lambda\sum_{k=1}^{K} f_k^*\right] = 0\\
&\Leftrightarrow \frac{\Pr(G=\mathcal{G}_k)}{K-1}\exp\left(-\frac{1}{K-1}f_k^*\right)=\lambda\\
&\Leftrightarrow \exp\left(-\frac{1}{K-1}f_k^*\right)=\lambda\frac{K-1}{\Pr(G=\mathcal{G}_k)}\\
&\Leftrightarrow -\frac{1}{K-1}f_k^* = \log\lambda+\log(K-1)-\log\Pr(G=\mathcal{G}_k)\\
&\Leftrightarrow f_k^* = \left[\log\Pr(G=\mathcal{G}_k)-\log\lambda-\log(K-1)\right](K-1).
\end{aligned}$$

Now recall $\sum_k f_k^* = 0$. Thus,

$$\begin{aligned}
&\sum_{k=1}^{K}\left[\log\Pr(G=\mathcal{G}_k)-\log\lambda-\log(K-1)\right](K-1)=0\\
&\Leftrightarrow \sum_{k=1}^{K}\left[\log\Pr(G=\mathcal{G}_k)-\log\lambda-\log(K-1)\right]=0\\
&\Leftrightarrow \sum_{k=1}^{K}\log\Pr(G=\mathcal{G}_k) = K\left(\log\lambda+\log(K-1)\right).
\end{aligned}$$

Plugging in to the earlier equation, we get

$$f_k^* = \left[\log\Pr(G=\mathcal{G}_k)-\frac{1}{K}\sum_{k=1}^{K}\log\Pr(G=\mathcal{G}_k)\right](K-1).$$

Finally I note

$$\arg\max_k f_k^* = \arg\max_k \Pr(G=\mathcal{G}_k).$$

(b)

For a multiclass boosting, the basis functions are the individual classifiers $f^{(m)}$ where $\sum_{k=1}^{K} f_k^{(m)} = 0$:

$$g^{(m)}(x) = \sum_{m=1}^{M}\beta_m f^{(m)}(x)$$

In this exercise, we solve

$$\beta_m, f^{(m)} = \arg\min_{\beta,f}\sum_{i=1}^{N} L\left(Y^{(i)},\, g^{(m-1)}+\beta f(x_i)\right).$$

Letting $w_i^{(m)} = L\left(Y^{(i)},\, g^{(m-1)}(x_i)\right)$, we have

$$\begin{aligned}\sum_{i=1}^{N} L\left(Y^{(i)},\, g^{(m-1)}(x_i) + \beta f(x_i)\right) &= \sum_{i=1}^{N} w_i^{(m)} L\left(Y^{(i)},\, \beta f(x_i)\right) \\ &= \sum_{i=1}^{N} w_i^{(m)} \exp\left(-\frac{1}{K} Y^{(i)T} \beta f(x_i)\right).\end{aligned}$$

Now, following Zhu et al. (2005).[3], I define

$$f(x_i)_k = \begin{cases} 1 & \text{if } \mathcal{G}_k \text{ is predicted: } \hat{G}^{(i)} = \mathcal{G}_k^{(i)} \\ -\frac{1}{K-1} & \text{otherwise.} \end{cases}$$

Then,

$$\begin{aligned}&\sum_{i=1}^{N} w_i^{(m)} \exp\left(-\frac{1}{K} Y^{(i)T} \beta f(x_i)\right) \\ &= \sum_{i=1}^{N} w_i^{(m)} \left[\exp\left(\frac{-\beta}{K-1}\right) I\left(\hat{G}^{(i)} = G^{(i)}\right) + \exp\left(\frac{\beta}{(K-1)^2}\right) I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\ &= \sum_{i=1}^{N} w_i^{(m)} \left[\exp\left(\frac{-\beta}{K-1}\right) \left(1 - I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right) + \exp\left(\frac{\beta}{(K-1)^2}\right) I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\ &= \sum_{i=1}^{N} w_i^{(m)} \left[\exp\left(\frac{-\beta}{K-1}\right) \left(1 - I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right) + \exp\left(\frac{\beta}{(K-1)^2}\right) I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\ &= \sum_{i=1}^{N} w_i^{(m)} \left[\exp\left(\frac{-\beta}{K-1}\right) + \left[\exp\left(\frac{\beta}{(K-1)^2}\right) - \exp\left(\frac{-\beta}{K-1}\right)\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right].\end{aligned}$$

Now the error is defined as follows:

$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i^{(m)} I\left(\hat{G}^{(i)} \neq G^{(i)}\right)}{\sum_{i=1}^{N} w_i^{(m)}.}$$

[3]Zhu, J., Zou, H., Rosset, S., & Hastie, T. (2009). Multi-class adaboost. *Statistics and Its Interface, 2*(3), 349-360.

Then by setting the derivative of the minimisation function to zero, I get

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \beta_m} \sum_{i=1}^{N} L\left(Y^{(i)},\, g^{(m-1)} + \beta_m f(x_i)\right) \\
\Leftrightarrow 0 &= \frac{\partial}{\partial \beta_m} \sum_{i=1}^{N} w_i^{(m)} \left[\exp\left(\frac{-\beta_m}{K-1}\right) + \left[\exp\left(\frac{\beta_m}{(K-1)^2}\right) - \exp\left(\frac{-\beta_m}{K-1}\right)\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\
\Leftrightarrow 0 &= -\frac{\beta_m}{K-1} \exp\left(\frac{-\beta_m}{K-1}\right) \sum_{i=1}^{N} w_i^{(m)} \\
&\quad + \left[\frac{\beta_m}{(K-1)^2} \exp\left(\frac{\beta_m}{(K-1)^2}\right) + \frac{\beta_m}{K-1} \exp\left(\frac{-\beta_m}{K-1}\right)\right] \sum_{i=1}^{N} w_i^{(m)} I\left(\hat{G}^{(i)} \neq G^{(i)}\right) \\
\Leftrightarrow 0 &= -\exp\left(\frac{-\beta_m}{K-1}\right) \sum_{i=1}^{N} w_i^{(m)} + \left[\frac{1}{K-1} \exp\left(\frac{\beta_m}{(K-1)^2}\right) + \exp\left(\frac{-\beta_m}{K-1}\right)\right] \sum_{i=1}^{N} w_i^{(m)} I\left(\hat{G}^{(i)} \neq G^{(i)}\right) \\
\Leftrightarrow 0 &= -\sum_{i=1}^{N} w_i^{(m)} + \left[\frac{1}{K-1} \exp\left(\frac{\beta_m}{(K-1)^2} + \frac{\beta_m}{K-1}\right) + 1\right] \sum_{i=1}^{N} w_i^{(m)} I\left(\hat{G}^{(i)} \neq G^{(i)}\right) \\
\Leftrightarrow & \frac{\sum_{i=1}^{N} w_i^{(m)}}{\sum_{i=1}^{N} w_i^{(m)} I\left(\hat{G}^{(i)} \neq G^{(i)}\right)} - 1 = \frac{1}{K-1} \exp\left(\frac{K\beta_m}{(K-1)^2}\right) \\
\Leftrightarrow & \frac{1}{\mathrm{err}_m} - 1 = \frac{1}{K-1} \exp\left(\frac{K\beta_m}{(K-1)^2}\right) \\
\Leftrightarrow & (K-1)\frac{1-\mathrm{err}_m}{\mathrm{err}_m} = \exp\left(\frac{K\beta_m}{(K-1)^2}\right) \\
\Leftrightarrow & \log\left[(K-1)\frac{1-\mathrm{err}_m}{\mathrm{err}_m}\right] \frac{(K-1)^2}{K} = \beta_m \\
\Leftrightarrow & \beta_m = \frac{(K-1)^2}{K}\left[\log\frac{1-\mathrm{err}_m}{\mathrm{err}_m} + \log(K-1)\right].
\end{aligned}
$$

When $k = 2$, this derivation of β_m is equal to the one in Section 10.4 (Expression 10.12 in page 344). This equality indicates that the multiclass exponential loss reduces to the exponential loss (AdaBoost) for binary classification problems.

Therefore, we have

$$
\begin{aligned}
g^{(m)}(x) &= g^{(m-1)}(x) + \beta_m f^{(m)}(x) \\
&= g^{(m-1)}(x) + \frac{(K-1)^2}{K}\left[\log\frac{1-\mathrm{err}_m}{\mathrm{err}_m} + \log(K-1)\right] f^{(m)}(x) \\
&= g^{(m-1)}(x) + \frac{(K-1)^2}{K} \alpha_m f^{(m)}(x)
\end{aligned}
$$

with $\alpha_m = \left[\log\frac{1-\mathrm{err}_m}{\mathrm{err}_m} + \log(K-1)\right]$ and $\beta_m = \alpha_m \frac{(K-1)^2}{K}$.

Then the weights for the next iteration are

$$
\begin{aligned}
w_i^{(m+1)} &= w_i^{(m)} \exp\left(-\frac{1}{K} Y^{(i)T} \beta_m f^{(m)}\right) \\
&= w_i^{(m)} \left[\exp\left(\frac{-\beta_m}{K-1}\right) + \left[\exp\left(\frac{\beta_m}{(K-1)^2}\right) - \exp\left(\frac{-\beta_m}{K-1}\right)\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\
&= w_i^{(m)} \exp\left(\frac{-\beta_m}{K-1}\right) \left[1 + \left[\exp\left(\frac{\beta_m}{(K-1)^2} + \frac{\beta_m}{K-1}\right) - 1\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\
&= w_i^{(m)} \exp\left(\frac{-\beta_m}{K-1}\right) \left[1 + \left[\exp\left(\frac{\alpha_m}{K} + \frac{\alpha_m (K-1)}{K}\right) - 1\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\
&= w_i^{(m)} \exp\left(\frac{-\beta_m}{K-1}\right) \left[1 + \left[\exp\left(\alpha_m\right) - 1\right] I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right] \\
&= w_i^{(m)} \exp\left(\frac{-\beta_m}{K-1}\right) \exp\left(\alpha_m I\left(\hat{G}^{(i)} \neq G^{(i)}\right)\right)
\end{aligned}
$$

which, when $K = 2$, is identical to AdaBoost's weights in Section 10.4 (Equation 10.15 in page 344).

Ex. 10.6

McNemar test (Agresti, 1996). We report the test error rates on the spam data to be 5.5% for a generalized additive model (GAM), and 4.5% for gradient boosting (GBM), with a test sample of size 1536.

(a) Show that the standard error of these estimates is about 0.6%.

Since the same test data are used for both methods, the error rates are correlated, and we cannot perform a two-sample t-test. We can compare the methods directly on each test observation, leading to the summary

	GBM	
GAM	Correct	Error
Correct	1434	18
Error	33	51

The McNemar test focuses on the discordant errors, 33 vs. 18.

(b) Conduct a test to show that GAM makes significantly more errors than gradient boosting, with a two-sided p-value of 0.036.

(a)

With the error rates and the sample size, we first calculate the variance of Binomial distributions. For the GAM, the variance on the number of misclassified cases is

$$\text{Var}_{GAM}^{(count)} = 1536 \cdot (1 - 0.055) \cdot 0.055,$$

thus, the variance of the misclassification rate is

$$\text{Var}_{GAM}^{(rate)} = \frac{\text{Var}_{GAM}^{(count)}}{1536^2},$$

and finally, the standard error is

$$\sqrt{\text{Var}_{GAM}^{(rate)}} \approx 0.006.$$

The similar calculation also shows the standard error of GBM misclassification rate is about 0.005.

(b)

The McNemar test statistics is given by

$$\chi^2 = \frac{(18-33)^2}{18+33} = 4.41$$

which, under the null hypothesis, follows the chi-squared distribution with 1 degree of freedom. Then, its p-value is 0.036.

Ex. 10.7

Derive expression (10.32).

Expression (10.32) provides the solution to the optimal constants for boosting trees. The optimal constants are the solution to the following minimisation (expression 10.30 in page 357):

$$\hat{\gamma}_{jm} = \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma_{jm}\right).$$

For the two-class classification with the exponential loss, the solution is given by expression (10.32) in page 357:

$$\hat{\gamma}_{jm} = \frac{1}{2}\log\frac{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = 1)}{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = -1).}$$

To derive expression (10.32), I first plug in the exponential loss function to expression (10.30):

$$\begin{aligned}
\hat{\gamma}_{jm} &= \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma_{jm}\right) \\
&= \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} \exp\left(-y_i(f_{m-1}(x_i) + \gamma_{jm})\right) \\
&= \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} \exp\left(-y_i f_{m-1}(x_i)\right)\exp\left(-y_i\gamma_{jm}\right) \\
&= \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} w_i^{(m)} \exp\left(-y_i\gamma_{jm}\right).
\end{aligned}$$

Then,

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \hat{\gamma}_{jm}} \sum_{x_i \in R_{jm}} w_i^{(m)} \exp\left(-y_i \hat{\gamma}_{jm}\right) \\
\Leftrightarrow 0 &= \sum_{x_i \in R_{jm}} w_i^{(m)} (-y_i) \exp\left(-y_i \hat{\gamma}_{jm}\right) \\
\Leftrightarrow 0 &= \sum_{x_i \in R_{jm}} \left[w_i^{(m)} \exp\left(\hat{\gamma}_{jm}\right) I(y_i = -1) - w_i^{(m)} \exp\left(-\hat{\gamma}_{jm}\right) I(y_i = 1) \right] \\
\Leftrightarrow 0 &= \sum_{x_i \in R_{jm}} \left[w_i^{(m)} \exp\left(2\hat{\gamma}_{jm}\right) I(y_i = -1) - w_i^{(m)} I(y_i = 1) \right] \\
\Leftrightarrow \exp\left(2\hat{\gamma}_{jm}\right) &= \frac{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = 1)}{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = -1)} \\
\Leftrightarrow \hat{\gamma}_{jm} &= \frac{1}{2} \log \frac{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = 1)}{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = -1),}
\end{aligned}
$$

which is expression (10.32).

Ex. 10.8

Consider a K-class problem where the targets y_{ik} are coded as 1 if observation i is in class k and zero otherwise. Suppose we have a current model $f_k(x), k = 1, \ldots, K$, with $\sum_{k=1}^{K} f_k(x) = 0$ (see (10.21) in Section 10.6). We wish to update the model for observations in a region R in predictor space, by adding constants $f_k(x) + \gamma_k$, with $\gamma_K = 0$.

(a) Write down the multinomial log-likelihood for this problem, and its first and second derivatives.

(b) Using only the diagonal of the Hessian matrix in (1), and starting from $\gamma_k = 0 \, \forall k$, show that a one-step approximate Newton update for γ_k is

$$
\gamma_k^1 = \frac{\sum_{x_i \in R} (y_{ik} - p_{ik})}{\sum_{x_i \in R} p_{ik}(1 - p_{ik})}, \quad k = 1, \ldots, K - 1,
$$

where $p_{ik} = \exp(f_k(x_i)) / \exp(\sum_{l=1}^{K} f_l(x_i))$.

(c) We prefer our update to sum to zero, as the current model does. Using symmetry arguments, show that

$$
\hat{\gamma}_k = \frac{K - 1}{K} \left(\gamma_k^1 - \frac{1}{K} \sum_{l=1}^{K} \gamma_l^1 \right), \quad k = 1, \ldots, K
$$

is an appropriate update, where γ_k^1 is defined as in (b) for all $k = 1, \ldots, K$.

I think the definition of p_{ik} above (as written in the book) is mistyped. In this exercise, I use the corrected definition of p_{ij}: $\exp(f_k(x_i)) / \sum_{l=1}^{K} \exp(f_l(x_i))$.

(a)

The multinomial probability is given by

$$p_{ik} = \frac{\exp(f_k(x_i))}{\sum_{l=1}^{K} \exp\left(f_l(x_i)\right)}.$$

Then, the log-likelihood is

$$\begin{aligned} LL(y, f) &= \sum_{x_i \in R} \sum_{k=1}^{K} y_{ik} \log(p_{ik}) \\ &= \sum_{x_i \in R} \left[\sum_{k=1}^{K} y_{ik} f_k(x_i) - \log\left(\sum_{l=1}^{K} \exp(f_l(x_i)) \right) \right]. \end{aligned}$$

Its first derivative is

$$\frac{\partial}{\partial f_k} LL(y, f) = \sum_{x_i \in R} \left[y_{ik} - \frac{\exp(f_k(x_i))}{\sum_{l=1}^{K} \exp(f_l(x_i))} \right] = \sum_{x_i \in R} \left(y_{ik} - p_{ik} \right),$$

and the second derivative (the diagonal of Hessian matrix) is

$$\begin{aligned} \frac{\partial^2}{\partial f_k \partial f_k} LL(y, f) &= - \sum_{x_i \in R} \left[\frac{\exp(f_k(x_i))}{\sum_{l=1}^{K} \exp(f_l(x_i))} - \left(\frac{\exp(f_k(x_i))}{\sum_{l=1}^{K} \exp(f_l(x_i))} \right)^2 \right] \\ &= - \sum_{x_i \in R} \left[p_{ik} - p_{ik}^2 \right] = - \sum_{x_i \in R} p_{ik}(1 - p_{ik}). \end{aligned}$$

(b)

The Newton update is given by

$$f^{new} = f - \left(\frac{\partial^2 LL(y, f)}{\partial f \partial f^T} \right)^{-1} \frac{\partial LL(y, f)}{\partial f}.$$

When we use only the diagonal of the Hessian matrix, this update simplifies to

$$\begin{aligned} f_k^{new} &= f_k - \left(\frac{\partial^2 LL(y, f)}{\partial f_k \partial f_k} \right)^{-1} \frac{\partial LL(y, f)}{\partial f_k} \\ &= f_k + \frac{\sum_{x_i \in R} \left(y_{ik} - p_{ik} \right)}{\sum_{x_i \in R} p_{ik}(1 - p_{ik}).} \end{aligned}$$

Therefore,

$$\gamma_k^1 = \frac{\sum_{x_i \in R} (y_{ik} - p_{ik})}{\sum_{x_i \in R} p_{ik}(1 - p_{ik})}, \quad k = 1, \ldots, K - 1.$$

(c)

Note that the adding an arbitrary constant a to γ leaves the model unchanged:

$$p_{ik} = \frac{\exp(f_k(x_i) + \gamma_k^1)}{\sum_{l=1}^{K} \exp\left(f_l(x_i) + \gamma_l^1\right)} = \frac{\exp(f_k(x_i) + \gamma_k^1 + a)}{\sum_{l=1}^{K} \exp\left(f_l(x_i) + \gamma_l^1 + a\right)}.$$

Thus, we can express the updates that sum to zero, $\tilde{\gamma}$, as follows:

$$\tilde{\gamma}_k = \gamma_k^1 + a \quad \text{and} \quad \sum_{l=1}^{K} \tilde{\gamma}_k = 0.$$

Then we have

$$\tilde{\gamma}_k = \gamma_k^1 - \frac{1}{K} \sum_{l=1}^{K} \gamma_l^1.$$

Now, recall that $\gamma_k^0 = 0 \, \forall k$ and that the updates are defined only for $k = 1, \ldots, K-1$. Thus,

$$\tilde{\gamma}_K = -\frac{1}{K} \sum_{l=1}^{K} \gamma_l^1 = -\frac{1}{K} \sum_{l=1}^{K-1} \gamma_l^1.$$

This calculation is not the same for the other non-base classes, and this choice of base class is arbitrary: we could use class $m(\neq K)$ as the base. Thus to retain the symmetry, we take the average of $\tilde{\gamma}_k$ for all the K possible base classes. Letting m index the base class and γ_K^1 be defined just as for $k = 1, \ldots$ in (b), we have

$$\hat{\gamma}_k = \frac{1}{K} \sum_{m=1}^{K} \left[I(k \neq m) \gamma_k^1 - \frac{1}{K} \sum_{l=1}^{K} I(l \neq m) \gamma_l^1 \right] = \frac{K-1}{K} \left(\gamma_k^1 - \frac{1}{K} \sum_{l=1}^{K} \gamma_l^1 \right).$$

Ex. 10.9

Consider a K-class problem where the targets y_{ik} are coded as 1 if observation i is in class k and zero otherwise. Using the multinomial deviance loss function (10.22) and the symmetric logistic transform, use the arguments leading to the gradient boosting Algorithm 10.3 to derive Algorithm 10.4.

Algorithm 10.3 lists the gradient boosting algorithm for a regression tree, and algorithm 10.4 shows the gradient boosting algorithm for a K-class classification tree. Here I focus on the calculation of updates (step 2(b)iii).

Following Ex. 10.8, I assume that the targets y_{ik} are coded as 1 if observation i is in class k and zero otherwise. Also

$$p_k(x) = \frac{\exp(f_k(x))}{\sum_{l=1}^{K} \exp(f_l(x))}, \quad k = 1, \ldots, K.$$

While we considered the maximisation of log-likelihood in Ex. 10.8, the gradient boosting algorithm concerns the minimisation of deviance loss function:

$$L(y, f(x)) = -LL(y, f(x)) = -\sum_{x_i \in R} \sum_{k=1}^{K} y_{ik} \log(p_{ik}).$$

The similar derivation to the one in Ex. 10.8(a) gives us the first derivative

$$-\frac{\partial L(y, f(x))}{\partial f_k(x)} = \sum_{x_i \in R} (y_{ik} - p_{ik}),$$

and the second derivative

$$-\frac{\partial^2 L(y, f(x))}{\partial f_k(x) \partial f_k(x)} = \sum_{x_i \in R} p_{ik}(1 - p_{ik}).$$

Then, the updates are given by

$$\begin{aligned}\tilde{f}_{km}(x) &= f_{k,m-1}(x) + \left(\frac{\partial^2 L(y, f(x))}{\partial f_k(x) \partial f_k(x)}\right)^{-1} \frac{\partial L(y, f(x))}{\partial f_k} \\ &= f_{k,m-1}(x) + \frac{\sum_{x_i \in R} (y_{ik} - p_{ik})}{\sum_{x_i \in R} p_{ik}(1 - p_{ik})}.\end{aligned}$$

As discussed in Ex. 10.8, an arbitrary constant can be added to this updates, without affecting the model. Thus we will force the updates of the base class to be zero. As in Ex. 10.8, the choice of base class is arbitrary, and we take the average of updates across all possible base classes. Then we obtain

$$f_{km}(x) = f_{k,m-1}(x) + \frac{K-1}{K} \frac{\sum_{x_i \in R} (y_{ik} - p_{ik})}{\sum_{x_i \in R} p_{ik}(1 - p_{ik})}.$$

To highlight the equivalence to Algorithm 10.4, I reparameterise the above updates. First, I define

$$r_{ik} = y_{ik} - p_{ik} = \begin{cases} 1 - p_{ik} > 0 & \text{if } y_{ik} = 1 \\ -p_{ik} < 0 & \text{if } y_{ik} = 0. \end{cases}$$

Then,

$$p_{ik} = y_{ik} - r_{ik} = \begin{cases} 1 - r_{ik} & \text{if } y_{ik} = 1 \\ -r_{ik} & \text{if } y_{ik} = 0, \end{cases}$$

and thus,

$$\begin{aligned} p_{ik}(1 - p_{ik}) &= \begin{cases} (1 - r_{ik}) r_{ik} & \text{if } y_{ik} = 1 \\ -r_{ik}(1 + r_{ik}) & \text{if } y_{ik} = 0 \end{cases} \\ &= |r_{ik}|(1 - |r_{ik}|), \end{aligned}$$

because when $y_{ik} = 1$, $r_{ik} > 0$, and when $y_{ik} = 0$, $r_{ik} < 0$. Therefore,

$$\begin{aligned} f_{km}(x) &= f_{k,m-1}(x) + \frac{K-1}{K} \frac{\sum_{x_i \in R} (y_{ik} - p_{ik})}{\sum_{x_i \in R} p_{ik}(1 - p_{ik})} \\ &= f_{k,m-1}(x) + \frac{K-1}{K} \frac{\sum_{x_i \in R} r_{ik}}{\sum_{x_i \in R} |r_{ik}|(1 - |r_{ik}|)} \end{aligned}$$

which is the updates as listed in Algorithm 10.4.

Note that this update calculation is identical to the one for Ex. 10.8. The major difference between the two algorithms is in what I did not discuss: how the model is trained to obtain regions R. Ex. 10.8 does not mention a model, but in Ex. 10.9 (the gradient boosting algorithm), a regression tree is fit to the residuals (r_{ik}) to give regions R. In the AdaBoost algorithm in constrast, a regression tree is fit to the weighted error.

Ex. 10.10

Show that for $K = 2$ class classification, only one tree needs to be grown at each gradient-boosting iteration.

At each iteration of the gradient boosting, regression trees are fit to the residuals. For K class classifications, we have K sets of residuals, each of which is fit by a regression tree, resulting in K trees.

For $K = 2$ class classification, however, we have

$$r_{i1} = I(y_i = \mathcal{G}_1) - p_{i1} \quad \text{and} \quad r_{i2} = I(y_i = \mathcal{G}_2) - p_{i2}.$$

Note that $y \in \{\mathcal{G}_1, \mathcal{G}_2\}$ and that $p_{i1} + p_{i2} = 1$. Thus,

$$\begin{aligned} r_{i2} &= (1 - I(y_i = \mathcal{G}_1)) - (1 - p_{i1}) \\ &= -(I(y_i = \mathcal{G}_1) - p_{i1}) \\ &= -r_{i1}. \end{aligned}$$

Thus, fitting a tree to r_{i1} is equivalent to fitting a tree to r_{i2}: The regions given by r_{i1} tree are identical to the regions given by r_{i2}. Therefore, only one tree needs to be fit.

Ex. 10.12

Referring to (10.49), let $S = \{1\}$ and $C = \{2\}$, with $f(X_1, X_2) = X_1$. Assume X_1 and X_2 are bivariate Gaussian, each with mean zero, variance one, and $\mathbb{E}(X_1 X_2) = \rho$. Show that $\mathbb{E}(f(X_1, X_2)|X_2) = \rho X_2$, even though f is not a function of X_2.

From the description,

$$\mathbb{E}(f(X_1, X_2)|X_2) = \mathbb{E}(X_1|X_2).$$

Let f_{X_1}, f_{X_2}, f_{X_1,X_2}, and $f_{X_1|X_2}$ represent the probability density functions. Given f_{X_1} and f_{X_2} are standard normal, the conditional density function is

$$\begin{aligned} & f_{X_1|X_2}(x_1, x_2) \\ &= \frac{f_{X_1,X_2}(x_1, x_2)}{f_{X_2}(x_2)} \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(\frac{-(x_1^2 + x_2^2 - 2\rho x_1 x_2)}{2(1-\rho^2)}\right) \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x_2^2}{2}\right)\right]^{-1} \\ &= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(\frac{-(x_1^2 + x_2^2 - 2\rho x_1 x_2)}{2(1-\rho^2)} - \frac{-x_2^2}{2}\right) \\ &= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(\frac{-1}{2} \frac{(x_1 - \rho x_2)^2}{1-\rho^2}\right). \end{aligned}$$

Thus, the conditional density function is of Gaussian distribution with mean ρx_2 and variance $1 - \rho^2$. Therefore,

$$\mathbb{E}(X_1|X_2) = \rho X_2.$$