

# Elements of Statistical Learning

Andrew Tullloch



## Contents

Chapter 2. Overview of Supervised Learning	4
Chapter 3. Linear Methods for Regression	12
Chapter 4. Linear Methods for Classification	23
Chapter 5. Basis Expansions and Regularization	28
Chapter 13. Support Vector Machines and Flexible Discriminants	29

## Overview of Supervised Learning

**Exercise 2.1.** Suppose that each of  $K$ -classes has an associated target  $t_k$ , which is a vector of all zeroes, except a one in the  $k$ -th position. Show that classifying 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.

PROOF. The assertion is equivalent to showing that

$$\arg \max_i \hat{y}_i = \arg \min_k \|t_k - \hat{y}\| = \arg \min_k \|\hat{y} - t_k\|^2$$

by monotonicity of  $x \mapsto x^2$  and symmetry of the norm.

WLOG, let  $\|\cdot\|$  be the Euclidean norm  $\|\cdot\|_2$ . Let  $k = \arg \max_i \hat{y}_i$ , with  $\hat{y}_k = \max y_i$ . Note that then  $\hat{y}_k \geq \frac{1}{K}$ , since  $\sum \hat{y}_i = 1$ .

Then for any  $k' \neq k$  (note that  $y_{k'} \leq y_k$ ), we have

$$\begin{aligned} \|y - t_{k'}\|_2^2 - \|y - t_k\|_2^2 &= y_k^2 + (y_{k'} - 1)^2 - (y_{k'}^2 + (y_k - 1)^2) \\ &= 2(y_k - y_{k'}) \\ &\geq 0 \end{aligned}$$

since  $y_{k'} \leq y_k$  by assumption.

Thus we must have

$$\arg \min_k \|t_k - \hat{y}\| = \arg \max_i \hat{y}_i$$

as required. □

**Exercise 2.2.** Show how to compute the Bayes decision boundary for the simulation example in Figure 2.5.

PROOF. The Bayes classifier is

$$\hat{G}(X) = \arg \max_{g \in \mathcal{G}} P(g|X = x).$$

In our two-class example ORANGE and BLUE, the decision boundary is the set where

$$P(g = \text{BLUE}|X = x) = P(g = \text{ORANGE}|X = x) = \frac{1}{2}.$$

By the Bayes rule, this is equivalent to the set of points where

$$P(X = x|g = \text{BLUE})P(g = \text{BLUE}) = P(X = x|g = \text{ORANGE})P(g = \text{ORANGE})$$

And since we know  $P(g)$  and  $P(X = x|g)$ , the decision boundary can be calculated.  $\square$

**Exercise 2.3.** Derive equation (2.24)

PROOF. TODO  $\square$

**Exercise 2.4.** Consider  $N$  data points uniformly distributed in a  $p$ -dimensional unit ball centered at the origin. Show the the median distance from the origin to the closest data point is given by

$$d(p, N) = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/p}$$

PROOF. Let  $r$  be the median distance from the origin to the closest data point. Then

$$P(\text{All } N \text{ points are further than } r \text{ from the origin}) = \frac{1}{2}$$

by definition of the median.

Since the points  $x_i$  are independently distributed, this implies that

$$\frac{1}{2} = \prod_{i=1}^N P(\|x_i\| > r)$$

and as the points  $x_i$  are uniformly distributed in the unit ball, we have that

$$\begin{aligned} P(\|x_i\| > r) &= 1 - P(\|x_i\| \leq r) \\ &= 1 - \frac{Kr^p}{K} \\ &= 1 - r^p \end{aligned}$$

Putting these together, we obtain that

$$\frac{1}{2} = (1 - r^p)^N$$

and solving for  $r$ , we have

$$r = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/p}$$

$\square$

**Exercise 2.5.** Consider inputs drawn from a spherical multivariate-normal distribution  $X \sim N(0, \mathbf{1}_p)$ . The squared distance from any sample point to the origin has a  $\chi_p^2$  distribution with mean  $p$ . Consider a prediction point  $x_0$  drawn from this distribution, and let  $a = \frac{x_0}{\|x_0\|}$  be an associated unit vector. Let  $z_i = a^T x_i$  be the projection of each of the training points on this direction.

Show that the  $z_i$  are distributed  $N(0, 1)$  with expected squared distance from the origin 1, while the target point has expected squared distance  $p$  from the origin. Hence for  $p = 10$ , a randomly drawn test point is about 3.1 standard deviations from the origin, while all the training points are on average one standard deviation along direction  $a$ . So most prediction points see themselves as lying on the edge of the training set.

PROOF. Let  $z_i = a^T x_i = \frac{x_0^T}{\|x_0\|} x_i$ . Then  $z_i$  is a linear combination of  $N(0, 1)$  random variables, and hence normal, with expectation zero and variance

$$\text{Var}(z_i) = \|a^T\|^2 \text{Var}(x_i) = \text{Var}(x_i) = 1$$

as the vector  $a$  has unit length and  $x_i \sim N(0, 1)$ .

For each target point  $x_i$ , the squared distance from the origin is a  $\chi_p^2$  distribution with mean  $p$ , as required.  $\square$

**Exercise 2.6.** (a) Derive equation (2.27) in the notes.  
 (b) Derive equation (2.28) in the notes.

PROOF. (i) We have

$$\begin{aligned} EPE(x_0) &= E_{y_0|x_0} E_{\mathcal{T}}(y_0 - \hat{y}_0)^2 \\ &= \text{Var}(y_0|x_0) + E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}\hat{y}_0]^2 + [E_{\mathcal{T}} - x_0^T \beta]^2 \\ &= \text{Var}(y_0|x_0) + \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0). \end{aligned}$$

We now treat each term individually. Since the estimator is unbiased, we have that the third term is zero. Since  $y_0 = x_0^T \beta + \epsilon$  with  $\epsilon$  an  $N(0, \sigma^2)$  random variable, we must have  $\text{Var}(y_0|x_0) = \sigma^2$ .

The middle term is more difficult. First, note that we have

$$\begin{aligned} \text{Var}_{\mathcal{T}}(\hat{y}_0) &= \text{Var}_{\mathcal{T}}(x_0^T \hat{\beta}) \\ &= x_0^T \text{Var}_{\mathcal{T}}(\hat{\beta}) x_0 \\ &= E_{\mathcal{T}} x_0^T \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} x_0 \end{aligned}$$

by conditioning (3.8) on  $\mathcal{T}$ .

(ii) TODO

$\square$

**Exercise 2.7.** Consider a regression problem with inputs  $x_i$  and outputs  $y_i$ , and a parameterized model  $f_{\theta}(x)$  to be fit with 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.

PROOF. This is relatively simple. WLOG, assume that  $x_1 = x_2$ , and all other observations are unique. Then our RSS function in the general least-squares estimation is

$$RSS(\theta) = \sum_{i=1}^N (y_i - f_{\theta}(x_i))^2 = \sum_{i=2}^N w_i (y_i - f_{\theta}(x_i))^2$$

where

$$w_i = \begin{cases} 2 & i = 2 \\ 1 & \text{otherwise} \end{cases}$$

Thus we have converted our least squares estimation into a reduced weighted least squares estimation. This minimal example can be easily generalised.  $\square$

**Exercise 2.8.** Suppose that we have a sample of  $N$  pairs  $x_i, y_i$ , drawn IID from the distribution such that

$$\begin{aligned} x_i &\sim h(x), \\ y_i &= f(x_i) + \epsilon_i, \\ E(\epsilon_i) &= 0, \\ \text{Var}(\epsilon_i) &= \sigma^2. \end{aligned}$$

We construct an estimator for  $f$  linear in the  $y_i$ ,

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

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

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

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

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

- (c) Decompose the (unconditional) MSE

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

into a squared bias and a variance component.

- (d) Establish a relationship between the square biases and variances in the above two cases.

PROOF. (a) Recall that the estimator for  $f$  in the linear regression case is given by

$$\hat{f}(x_0) = x_0^T \beta$$

where  $\beta = (X^T X)^{-1} X^T y$ . Then we can simply write

$$\hat{f}(x_0) = \sum_{i=1}^N (x_0^T (X^T X)^{-1} X^T)_i y_i.$$

Hence

$$\ell_i(x_0; \mathcal{X}) = (x_0^T (X^T X)^{-1} X^T)_i.$$

In the  $k$ -nearest-neighbour representation, we have

$$\hat{f}(x_0) = \sum_{i=1}^N \frac{y_i}{k} \mathbf{1}_{x_i \in N_k(x_0)}$$

where  $N_k(x_0)$  represents the set of  $k$ -nearest-neighbours of  $x_0$ . Clearly,

$$\ell_i(x_0; \mathcal{X}) = \frac{1}{k} \mathbf{1}_{x_i \in N_k(x_0)}$$

(b) TODO

(c) TODO

(d) TODO

□

**Exercise 2.9.** Compare the classification performance of linear regression and  $k$ -nearest neighbour classification on the `zipcode` data. In particular, consider on the 2's and 3's, and  $k = 1, 3, 5, 7, 15$ . Show both the training and test error for each choice.

PROOF. Our implementation in R and graphs are attached.



```

library('ProjectTemplate')
load.project()

## Linear Regression
mod <- lm(Y ~ ., data = zip.train.filtered)

# Round predictions
category_f <- function(x) { if (x > 2.5) 3 else 2 }
predictions.lm.test <- as.character(sapply(predict(mod, zip.test.filtered),
                                           category_f))
predictions.lm.train <- as.character(sapply(predict(mod, zip.train.filtered),
                                             category_f))

## KNN
knn.train <- zip.train.filtered[, 2:257]
knn.test <- zip.test.filtered[, 2:257]

knn.train.Y <- as.factor(zip.train.filtered$Y)
knn.test.Y <- as.factor(zip.test.filtered$Y)

# KNN Predictions
predictions.knn.test <- sapply(1:15, function(k) {
  knn(train = knn.train,
      test = knn.test,
      cl = knn.train.Y,
      k = k)
})
predictions.knn.train <- sapply(1:15, function(k) {
  knn(train = knn.train,
      test = knn.train,
      cl = knn.train.Y,
      k = k)
})

# Compute error rates
errors.xs <- 1:15

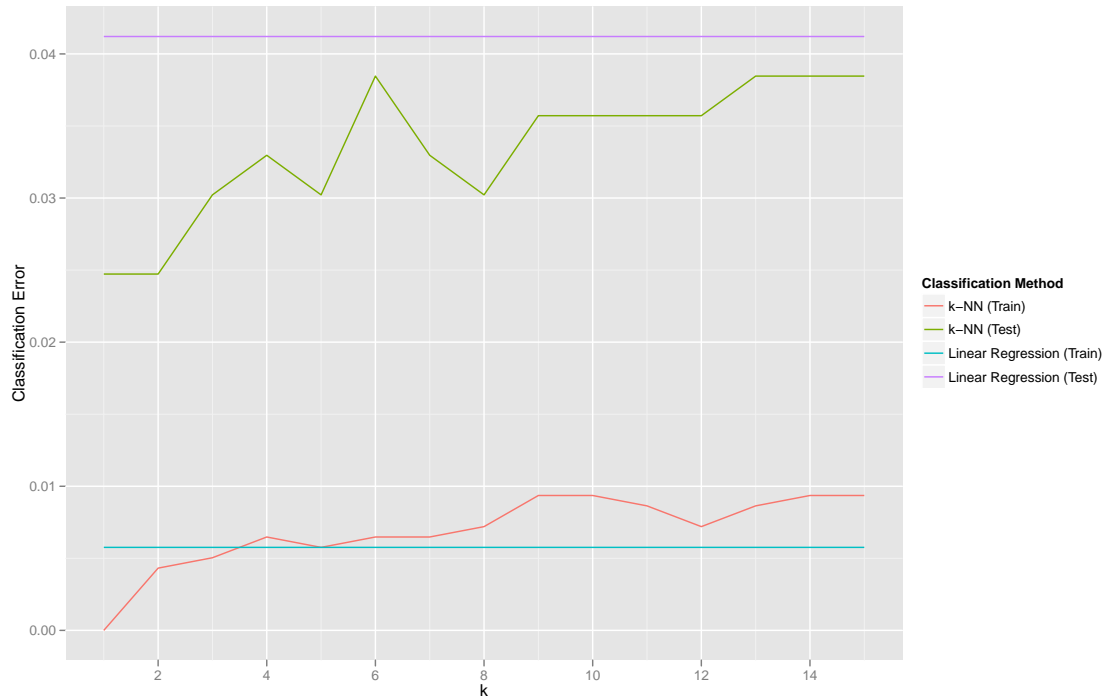
errors.knn.test <- apply(predictions.knn.test, 2, function(prediction) {
  classError(prediction, as.factor(zip.test.filtered$Y))$errorRate
})
errors.knn.train <- apply(predictions.knn.train, 2, function(prediction) {
  classError(prediction, as.factor(zip.train.filtered$Y))$errorRate
})
errors.lm.test <- sapply(errors.xs, function(k) {
  classError(predictions.lm.test, as.factor(zip.test.filtered$Y))$errorRate
})
errors.lm.train <- sapply(errors.xs, function(k) {
  classError(predictions.lm.train, as.factor(zip.train.filtered$Y))$errorRate
})

errors <- data.frame("K"=errors.xs,
                    "KNN.Train"=errors.knn.train,
                    "KNN.Test"=errors.knn.test,
                    "LR.Train"=errors.lm.train,

```

```
      "LR.Test"=errors.lm.test)

# Create Plot
plot.data <- melt(errors, id="K")
ggplot(data=plot.data,
       aes(x=K, y=value, colour=variable)) +
  geom_line() +
  xlab("k") +
  ylab("Classification Error") +
  opts(title="Classification Errors for different methods on zipcode data")
  scale_colour_hue(name="Classification Method",
                  labels=c("k-NN (Train)",
                           "k-NN (Test)",
                           "Linear Regression (Train)",
                           "Linear Regression (Test)"))
)
ggsave(file.path('graphs', 'exercise_2_8.pdf'))
ggsave(file.path('graphs', 'exercise_2_8.png'))
```



□

**Exercise 2.10.** Consider a linear regression model with  $p$  parameters, fitted by OLS to a set of training data  $(x_i, y_i)_{1 \leq i \leq N}$  drawn at random from a population. Let  $\hat{\beta}$  be the least squares estimate. Suppose we have some test data  $(\tilde{x}_i, \tilde{y}_i)_{1 \leq i \leq M}$  drawn at random from the same population as the training data.

If  $R_{tr}(\beta) = \frac{1}{N} \sum_{i=1}^N (y_i - \beta^T x_i)^2$  and  $R_{te}(\beta) = \frac{1}{M} \sum_{i=1}^M (\tilde{y}_i - \beta^T \tilde{x}_i)^2$ , prove that

$$E(R_{tr}(\hat{\beta})) \leq E(R_{te}(\hat{\beta}))$$

where the expectation is over all that is random in each expression.

## Linear Methods for Regression

**Exercise 3.1.** Show that the  $F$  statistic for dropping a single coefficient from a model is equal to the square of the corresponding  $z$ -score.

PROOF. Recall that the  $F$  statistic is defined by the following expression

$$\frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N - p_1 - 1)}.$$

where  $RSS_0, RSS_1$  and  $p_0 + 1, p_1 + 1$  refer to the residual sum of squares and the number of free parameters in the smaller and bigger models, respectively. Recall also that the  $F$  statistic has a  $F_{p_1 - p_0, N - p_1 - 1}$  distribution under the null hypothesis that the smaller model is correct.

Next, recall that the  $z$ -score of a coefficient is

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma} \sqrt{v_j}}$$

and under the null hypothesis that  $\beta_j$  is zero,  $z_j$  is distributed according to a  $t$ -distribution with  $N - p - 1$  degrees of freedom.

Hence, by dropping a single coefficient from a model, our  $F$  statistic has a  $F_{1, N - p - 1}$  where  $p + 1$  are the number of parameters in the original model. Similarly, the corresponding  $z$ -score is distributed according to a  $t_{N - p - 1}$  distribution, and thus the square of the  $z$ -score is distributed according to an  $F_{1, N - p - 1}$  distribution, as required.

Thus both the  $z$ -score and the  $F$  statistic test identical hypotheses under identical distributions. Thus they must have the same value in this case.  $\square$

**Exercise 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  $a^T \beta = \sum_{j=0}^3 \beta_j x_0^j$ .
- (2) Form a 95% confidence set for  $\beta$  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.

PROOF. The key distinction is that in the first case, we form the set of points such that we are 95% confident that  $\hat{f}(x_0)$  is within this set, whereas in the second method, we are 95% confident that an arbitrary point is within our confidence interval. This is the distinction between a *pointwise* approach and a *global* confidence estimate.

In the pointwise approach, we seek to estimate the variance of an individual prediction - that is, to calculate  $\text{Var}(\hat{f}(x_0)|x_0)$ . Here, we have

$$\begin{aligned}\sigma_0^2 &= \text{Var}(\hat{f}(x_0)|x_0) = \text{Var}(x_0^T \hat{\beta}|x_0) \\ &= x_0^T \text{Var}(\hat{\beta})x_0 \\ &= \hat{\sigma}^2 x_0^T (X^T X)^{-1} x_0.\end{aligned}$$

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

R code and graphs of the simulation are attached.

```
library('ProjectTemplate')
load.project()

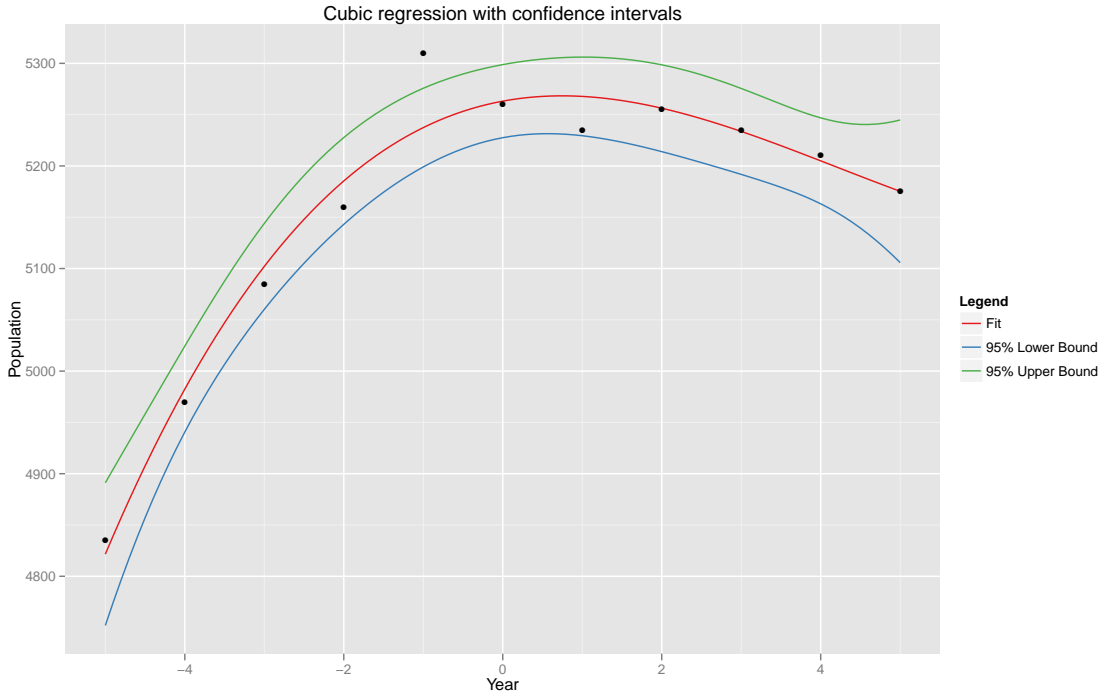
# Raw data
simulation.xs <- c(1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969)
simulation.ys <- c(4835, 4970, 5085, 5160, 5310, 5260, 5235, 5255, 5235, 5210, 5175)
simulation.df <- data.frame(pop = simulation.ys, year = simulation.xs)

# Rescale years
simulation.df$year = simulation.df$year - 1964

# Generate regression, construct confidence intervals
fit <- lm(pop ~ year + I(year^2) + I(year^3), data=simulation.df)
xs = seq(-5, 5, 0.1)
fit.confidence = predict(fit, data.frame(year=xs), interval="confidence", level=0.95)

# Create data frame containing variables of interest
df = as.data.frame(fit.confidence)
df$year <- xs
df = melt(df, id.vars="year")

p <- ggplot() + geom_line(aes(x=year, y=value, colour=variable), df) +
  geom_point(aes(x=year, y=pop), simulation.df)
p <- p + scale_x_continuous('Year') + scale_y_continuous('Population')
p <- p + opts(title="Cubic regression with confidence intervals")
p <- p + scale_color_brewer(name="Legend",
  labels=c("Fit",
    "95% Lower Bound",
    "95% Upper Bound"),
  palette="Set1")
ggsave(file.path('graphs', 'exercise_3_2.pdf'))
ggsave(file.path('graphs', 'exercise_3_2.png'))
```



TODO: Part 2. □

**Exercise 3.3** (The Gauss-Markov Theorem). (1) Prove the Gauss-Markov theorem: the least squares estimate of a parameter  $a^T \beta$  has a variance no bigger than that of any other linear unbiased estimate of  $a^T \beta$ .

(2) Secondly, show that if  $\hat{V}$  is the variance-covariance matrix of the least squares estimate of  $\beta$  and  $\tilde{V}$  is the variance covariance matrix of any other linear unbiased estimate, then  $\hat{V} \leq \tilde{V}$ , where  $B \leq A$  if  $A - B$  is positive semidefinite.

PROOF. Let  $\hat{\theta} = a^T \hat{\beta} = a^T (X^T X)^{-1} X^T y$  be the least squares estimate of  $a^T \beta$ . Let  $\tilde{\theta} = c^T y$  be any other unbiased linear estimator of  $a^T \beta$ . Now, let  $d^T = c^T - a^T (X^T X)^{-1} X^T$ . Then as  $c^T y$  is unbiased, we must have

$$\begin{aligned}
 E(c^T y) &= E(a^T (X^T X)^{-1} X^T + d^T) y \\
 &= a^T \beta + d^T X \beta \\
 &= a^T \beta
 \end{aligned}$$

as  $c^T y$  is unbiased, which implies that  $d^T X = 0$ .

Now we calculate the variance of our estimator. We have

$$\begin{aligned}
\text{Var}(c^T y) &= c^T \text{Var}(y) c \\
&= \sigma^2 c^T c \\
&= \sigma^2 (a^T (X^T X)^{-1} X^T + d^T) (a^T (X^T X)^{-1} X^T + d^T)^T \\
&= \sigma^2 (a^T (X^T X)^{-1} X^T + d^T) (X (X^T X)^{-1} a + d) \\
&= \sigma^2 \left( a^T (X^T X)^{-1} X^T X (X^T X)^{-1} a + a^T (X^T X)^{-1} \underbrace{X^T d}_{=0} + \underbrace{d^T X}_{=0} (X^T X)^{-1} a + d^T d \right) \\
&= \sigma^2 \left( \underbrace{a^T (X^T X)^{-1} a}_{\text{Var}(\hat{\theta})} + \underbrace{d^T d}_{\geq 0} \right)
\end{aligned}$$

Thus  $\text{Var}(\hat{\theta}) \leq \text{Var}(\tilde{\theta})$  for all other unbiased linear estimators  $\tilde{\theta}$ .

The proof of the matrix version is almost identical, except we replace our vector  $d$  with a matrix  $D$ . It is then possible to show that  $\tilde{V} = \hat{V} + D^T D$ , and as  $D^T D$  is a positive semidefinite matrix for any  $D$ , we have  $\hat{V} \leq \tilde{V}$ .  $\square$

**Exercise 3.4.** Show how the vector of least square coefficients can be obtained from a single pass of the Gram-Schmidt procedure. Represent your solution in terms of the QR decomposition of  $X$ .

PROOF. Recall that by a single pass of the Gram-Schmidt procedure, we can write our matrix  $X$  as

$$X = Z\Gamma,$$

where  $Z$  contains the orthogonal columns  $z_j$ , and  $\Gamma$  is an upper-diagonal matrix with ones on the diagonal, and  $\gamma_{ij} = \frac{\langle z_i, x_j \rangle}{\|z_i\|^2}$ . This is a reflection of the fact that by definition,

$$x_j = z_j + \sum_{k=0}^{j-1} \gamma_{kj} z_k.$$

Now, by the QR decomposition, we can write  $X = QR$ , where  $Q$  is an orthogonal matrix and  $R$  is an upper triangular matrix. We have  $Q = ZD^{-1}$  and  $R = D\Gamma$ , where  $D$  is a diagonal matrix with  $D_{jj} = \|z_j\|$ .

Now, by definition of  $\hat{\beta}$ , we have

$$(X^T X) \hat{\beta} = X^T y.$$



Now, using the  $QR$  decomposition, we have

$$\begin{aligned}(R^T Q^T)(QR)\hat{\beta} &= R^T Q^T y \\ R\hat{\beta} &= Q^T y\end{aligned}$$

As  $R$  is upper triangular, we can write

$$\begin{aligned}R_{pp}\hat{\beta}_p &= \langle q_p, y \rangle \\ \|z_p\|\hat{\beta}_p &= \|z_p\|^{-1}\langle z_p, y \rangle \\ \hat{\beta}_p &= \frac{\langle z_p, y \rangle}{\|z_p\|^2}\end{aligned}$$

in accordance with our previous results. Now, by back substitution, we can obtain the sequence of regression coefficients  $\hat{\beta}_j$ . As an example, to calculate  $\hat{\beta}_{p-1}$ , we have

$$\begin{aligned}R_{p-1,p-1}\hat{\beta}_{p-1} + R_{p-1,p}\hat{\beta}_p &= \langle q_{p-1}, y \rangle \\ \|z_{p-1}\|\hat{\beta}_{p-1} + \|z_{p-1}\|\gamma_{p-1,p}\hat{\beta}_p &= \|z_{p-1}\|^{-1}\langle z_{p-1}, y \rangle\end{aligned}$$

and then solving for  $\hat{\beta}_{p-1}$ . This process can be repeated for all  $\beta_j$ , thus obtaining the regression coefficients in one pass of the Gram-Schmidt procedure.  $\square$

**Exercise 3.5.** Consider the ridge regression problem (3.41). Show that this problem is equivalent to the problem

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

PROOF. Consider rewriting our objective function above as

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

Note that making the substitutions

$$\begin{aligned}\beta_0 &\mapsto \beta_0^c - \sum_{j=1}^p \hat{x}_j \beta_j \\ \beta_j &\mapsto \beta_j^c, j = 1, 2, \dots, p\end{aligned}$$

that  $\hat{\beta}$  is a minimiser of the original ridge regression equation if  $\hat{\beta}^c$  is a minimiser of our modified ridge regression.

The modified solution merely has a shifted intercept term, and all other coefficients remain the same.  $\square$

**Exercise 3.6.** Show that the ridge regression estimate is the mean (and mode) of the posterior distribution, under a Gaussian prior  $\beta \sim N(0, \tau \mathbf{I})$ , and Gaussian sampling model  $y \sim N(X\beta, \sigma^2 \mathbf{I})$ . Find the relationship between the regularization parameter  $\lambda$  in the ridge formula, and the variances  $\tau$  and  $\sigma^2$ .

**Exercise 3.7.** Assume

$$y_i \sim N(\beta_0 + x_i^T \beta, \sigma^2), i = 1, 2, \dots, N$$

and the parameters  $\beta_j$  are each distributed as  $N(0, \tau^2)$ , independently of one another. Assume  $\sigma^2$  and  $\tau^2$  are known, show that the minus log-posterior density of  $\beta$  is proportional to

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

where  $\lambda = \frac{\sigma^2}{\tau^2}$ .

**Exercise 3.8.** Consider the QR decomposition of the uncentred  $N \times (p+1)$  matrix  $X$ , whose first column is all ones, and the SVD of the  $N \times p$  centred matrix  $\tilde{X}$ . Show that  $Q_2$  and  $U$  share the same subspace, where  $Q_2$  is the submatrix of  $Q$  with the first column removed. Under what circumstances will they be the same, up to sign flips?

PROOF. Denote the columns of  $X$  by  $x_0, \dots, x_p$ , the columns of  $Q$  by  $z_0, \dots, z_p$ , the columns of  $\tilde{X}$  by  $\tilde{x}_1, \dots, \tilde{x}_p$ , and the columns of  $U$  by  $u_1, \dots, u_p$ . Without loss of generality, we can assume that for all  $i$ ,  $\|x_i\| = 1$  and that  $X$  is non-singular (this cleans up the proof somewhat).

First, note that by the QR decomposition, we have that  $\text{span}(x_0, \dots, x_j) = \text{span}(z_0, \dots, z_j)$  for any  $0 \leq j \leq p$ .

By our assumption, we have that  $\tilde{x}_i = x_i - \bar{x}_i \mathbf{1}$  for  $i = 1, \dots, p$ . Thus we can write  $\tilde{x}_i = \sum_{j \leq i} \alpha_j z_j$ , and as the  $z_j$  are orthogonal, we must be able to write  $\tilde{x}_i$  in terms of  $z_j$  for  $j = 1, 2, \dots, i$ . Thus  $\text{span}(\tilde{x}_1, \dots, \tilde{x}_i) = \text{span}(z_1, \dots, z_i)$ .

Finally, we calculate  $\text{span}(u_1, \dots, u_p)$ . We have that  $U$  is a unitary  $N \times p$  matrix, and thus the columns of  $U$  span the column space of  $\tilde{X}$ , and thus the span of  $Q_2$  is equal to the span of  $U$ .

TODO: When is  $Q_2$  equal to  $U$  up to parity? Is it where columns of □

**Exercise 3.9** (Forward stepwise regression). Suppose that we have the QR decomposition for the  $N \times q$  matrix  $X_1$  in a multiple regression problem with response  $y$ , and we have an additional  $p - q$  predictors in matrix  $X_2$ . Denote the current residual by  $r$ . We wish to establish which one of these additional variables will reduce the residual-sum-of-squares the most when included with those in  $X_1$ . Describe an efficient procedure for doing this.

PROOF. Select the vector  $x_{j'}$  where

$$x_{j'} = \arg \min_{j=q+1, \dots, p} \left| \left\langle \frac{x_q}{\|x_q\|}, r \right\rangle \right|$$

This selects the vector that explains the maximal amount of variance in  $r$  given  $X_1$ , and thus reduces the residual sum of squares the most. It is then possible to repeat this procedure by updating  $X_2$  as in Algorithm 3.1.  $\square$

**Exercise 3.10** (Backward stepwise regression). *Suppose that we have the multiple regression fit of  $y$  on  $X$ , along with standard errors and  $z$ -scores. We wish to establish which variable, when dropped, will increase the  $RSS$  the least. How would you do this?*

PROOF. By Exercise 3.1, we can show that the  $F$ -statistic for dropping a single coefficient from a model is equal to the square of the corresponding  $z$ -score. Thus, we drop the variable that has the lowest squared  $z$ -score from the model.  $\square$

**Exercise 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  $\Sigma_i$  are different for each observation?*

**Exercise 3.12.** *Show that the ridge regression estimates can be obtained by OLS on an augmented data set. We augment the centred matrix  $X$  with  $p$  additional rows  $\sqrt{\lambda}\mathbf{I}$ , and augment  $y$  with  $p$  zeroes.*

PROOF. For our augmented matrix  $X_1$ , equal to appending  $\sqrt{\lambda}\mathbf{I}$  to the original observation matrix  $X$ , we have that the  $RSS$  expression for OLS regression becomes

$$\begin{aligned} RSS &= \sum_{i=1}^{N+p} \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \\ &= \sum_{i=1}^N \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \sum_{i=N+1}^{N+p} \left( \sum_{j=1}^p x_{ij} \beta_j \right)^2 \\ &= \sum_{i=1}^N \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \sum_{j=1}^p \lambda \beta_j^2 \end{aligned}$$

which is the objective function for the ridge regression estimate.  $\square$

**Exercise 3.13.** *Derive expression (3.62), and show that  $\hat{\beta}^{pcr}(p) = \hat{\beta}^{ls}$ .*

**Exercise 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.*

**Exercise 3.15.** *Verify expression (3.64), and hence show that the PLS directions are a compromise between the OLS coefficients and the principal component directions.*

**Exercise 3.16.** *Derive the entries in Table 3.4, the explicit forms for estimators in the orthogonal case.*

**Exercise 3.17.** *Repeat the analysis of Table 3.3 on the spam data discussed in Chapter 1.*

PROOF. R code implementing this method is attached. We require the `MASS`, `lars`, and `pls` packages.

```

library("ProjectTemplate")
load.project()

library("lars") # For least-angle and lasso
library("MASS") # For ridge
library("pls") # For PLS and PCR

mod.ls <- lm(Y ~ . - 1, spam.train)
mod.ridge <- lm.ridge(Y ~ ., spam.train)
mod.pcr <- pcr(formula=Y ~ ., data=spam.train, validation="CV")
mod.plsr <- plsr(formula=Y ~ ., data=spam.train, validation="CV")
mod.lars <- lars(as.matrix(spam.train[,1:ncol(spam.train) - 1]),
                spam.train[,ncol(spam.train)],
                type="lar")
mod.lasso <- lars(as.matrix(spam.train[,1:ncol(spam.train) - 1]),
                 spam.train[,ncol(spam.train)],
                 type="lasso")

mods.coefs <- data.frame(ls=mod.ls$coef,
                        ridge=mod.ridge$coef,
                        lasso=mod.lasso$beta[10,],
                        pcr=mod.pcr$coef[,10],
                        plsr=mod.plsr$coef[,10]
                        )

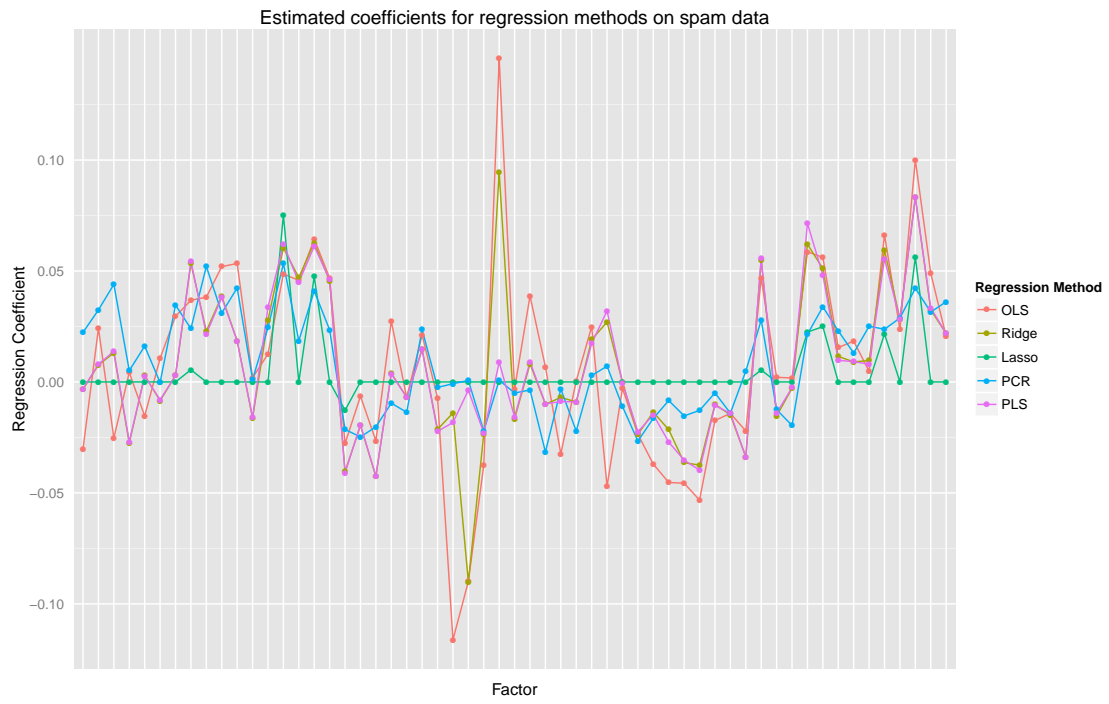
mods.coefs$xs = row.names(mods.coefs)
plot.data <- melt(mods.coefs, id="xs")

ggplot(data=plot.data,
       aes(x=factor(xs),
           y=value,
           group=variable,
           colour=variable)) +
  geom_line() +
  geom_point() +
  xlab("Factor") +
  ylab("Regression Coefficient") +
  opts(title = "Estimated coefficients for regression methods on spam data",
       axis.ticks = theme_blank(),
       axis.text.x = theme_blank()) +
  scale_colour_hue(name="Regression Method",
                  labels=c("OLS",
                          "Ridge",
                          "Lasso",
                          "PCR",
                          "PLS"))
)

ggsave(file.path('graphs', 'exercise_3_17.pdf'))
ggsave(file.path('graphs', 'exercise_3_17.png'))

```

□



## Linear Methods for Classification

**Exercise 4.1.** Show how to solve the generalised eigenvalue problem  $\max a^T B a$  subject to  $a^T W a = 1$  by transforming it to a standard eigenvalue problem.

PROOF. By Lagrange multipliers, we have that the function  $\mathcal{L}(a) = a^T B a - \lambda(a^T W a - 1)$  has a critical point where

$$\frac{d\mathcal{L}}{da} = 2a^T B^T - 2\lambda a^T W^T = 0,$$

that is, where  $B a = \lambda W a$ . If we let  $W = D^T D$  (Cholesky decomposition),  $C = D^{-1} B D^{-1}$ , and  $y = D a$ , we obtain that our solution becomes

$$C y = \lambda y,$$

and so we can convert our problem into an eigenvalue problem. It is clear that if  $y_m$  and  $\lambda_m$  are the maximal eigenvector and eigenvalue of the reduced problem, then  $D^{-1} y_m$  and  $\lambda_m$  are the maximal eigenvector and eigenvalue of the generalized problem, as required.  $\square$

**Exercise 4.2.** Suppose that we have features  $x \in \mathbb{R}^p$ , a two-class response, with class sizes  $N_1, N_2$ , and the target coded as  $-N/N_1, N/N_2$ .

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

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

(2) Consider minimization of the least squares criterion

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

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

$$\left( (N-2)\hat{\Sigma} + \frac{N_1 N_2}{N} \hat{\Sigma}_B \right) \beta = N(\hat{\mu}_2 - \hat{\mu}_1)$$

where  $\hat{\Sigma}_B = (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T$ .

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

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

and therefore the least squares regression coefficient is identical to the LDA coefficient, up to a scalar multiple.

- (4) Show that this holds for any (distinct) coding of the two classes.  
 (5) Find the solution  $\hat{\beta}_0$ , and hence the predicted values  $\hat{\beta}_0 + \hat{\beta}^T x$ . Consider the following rule: classify to class 2 if  $\hat{y}_i > 0$  and class 1 otherwise. Show that this is not the same as the LDA rule unless the classes have equal numbers of observations.

PROOF. We use the notation of Chapter 4.

- (1) Since in the two class case, we classify to class 2 if  $\delta_1(x) < \delta_2(x)$ . Substituting this into our equation for the Linear discriminant functions, we have

$$\delta_1(x) < \delta_2(x)$$

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

as required.

- (2) Let  $U_i$  be the  $n$  element vector with  $j$ -th element 1 if the  $j$ -th observation is class  $i$ , and zero otherwise. Then we can write our target vector  $Y$  as  $t_1 U_1 + t_2 U_2$ , where  $t_i$  are our target labels, and we have  $\mathbf{1} = U_1 + U_2$ . Note that we can write our estimates  $\hat{\mu}_1, \hat{\mu}_2$  as  $X^T U_i = N_i \hat{\mu}_i$ , and that  $X^T Y = t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2$ .

By the least squares criterion, we can write

$$RSS = \sum_{i=1}^N (y_i - \beta_0 - \beta^T X)^2 = (Y - \beta_0 \mathbf{1} - X\beta)^T (Y - \beta_0 \mathbf{1} - X\beta)$$

Minimizing this with respect to  $\beta$  and  $\beta_0$ , we obtain

$$2X^T X\beta - 2X^T Y + 2\beta_0 X^T \mathbf{1} = 0$$

$$2N\beta_0 - 2\mathbf{1}^T (Y - X\beta) = 0.$$

These equations can be solved for  $\beta_0$  and  $\beta$  by substitution as

$$\hat{\beta}_0 = \frac{1}{N} \mathbf{1}^T (Y - X\beta)$$

$$\left( X^T X - \frac{1}{N} X^T \mathbf{1} \mathbf{1}^T X \right) \hat{\beta} = X^T Y - \frac{1}{N} X^T \mathbf{1} \mathbf{1}^T Y$$

The RHS can be written as

$$X^T Y - \frac{1}{N} X^T \mathbf{1} \mathbf{1}^T Y = t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) (t_1 N_1 + t_2 N_2)$$

$$= \frac{N_1 N_2}{N} (t_1 - t_2) (\hat{\mu}_1 - \hat{\mu}_2)$$

where we use our relations for  $X^T U_i$  and the fact that  $\mathbf{1} = U_1 + U_2$ .



Similarly, the bracketed term on the LHS of our expression for  $\beta$  can be rewritten as

$$X^T X = (N - 2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T,$$

and by substituting in the above and the definition of  $\hat{\Sigma}_B$ , we can write

$$X^T X - \frac{1}{N}X^T \mathbf{1}\mathbf{1}^T X = (N - 2)\hat{\Sigma} + \frac{N_1 N_2}{N}\hat{\Sigma}_B$$

as required.

Putting this together, we obtain our required result,

$$\left( (N - 2)\hat{\Sigma} + \frac{N_1 N_2}{N}\hat{\Sigma}_B \right) \hat{\beta} = \frac{N_1 N_2}{N}(t_1 - t_2)(\hat{\mu}_1 - \hat{\mu}_2),$$

and then substituting  $t_1 = -N/N_1, t_2 = N/N_2$ , we obtain our required result,

$$\left( (N - 2)\hat{\Sigma} + \frac{N_1 N_2}{N}\hat{\Sigma}_B \right) \hat{\beta} = N(\hat{\mu}_2 - \hat{\mu}_1)$$

- (3) All that is required is to show that  $\hat{\Sigma}_B \hat{\beta}$  is in the direction of  $(\hat{\mu}_2 - \hat{\mu}_1)$ . This is clear from the fact that

$$\hat{\Sigma}_B \hat{\beta} = (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T \hat{\beta} = \lambda(\hat{\mu}_2 - \hat{\mu}_1)$$

for some  $\lambda \in \mathbb{R}$ . Since  $\hat{\Sigma} \hat{\beta}$  is a linear combination of terms in the direction of  $(\hat{\mu}_2 - \hat{\mu}_1)$ , we can write

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

as required.

- (4) Since our  $t_1, t_2$  were arbitrary and distinct, the result follows.  
 (5) From above, we can write

$$\begin{aligned} \hat{\beta}_0 &= \frac{1}{N} \mathbf{1}^T (Y - X\hat{\beta}) \\ &= \frac{1}{N} (t_1 N_1 + t_2 N_2) - \frac{1}{N} \mathbf{1}^T X \hat{\beta} \\ &= -\frac{1}{N} (N_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2^T) \hat{\beta}. \end{aligned}$$

We can then write our predicted value  $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}^T x$  as

$$\begin{aligned} \hat{f}(x) &= \frac{1}{N} (Nx^T - N_1 \hat{\mu}_1^T - N_2 \hat{\mu}_2^T) \hat{\beta} \\ &= \frac{1}{N} (Nx^T - N_1 \hat{\mu}_1^T - N_2 \hat{\mu}_2^T) \lambda \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \end{aligned}$$

for some  $\lambda \in \mathbb{R}$ , and so our classification rule is  $\hat{f}(x) > 0$ , or equivalently,

$$\begin{aligned} Nx^T \lambda \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) &> (N_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2^T) \lambda \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) \\ x^T \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) &> \frac{1}{N} (N_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2^T) \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) \end{aligned}$$

which is different to the LDA decision rule unless  $N_1 = N_2$ .

□

**Exercise 4.3.** Suppose that we transform the original predictors  $X$  to  $\hat{Y}$  by taking the predicted values under linear regression. Show that LDA using  $\hat{Y}$  is identical to using LDA in the original space.

**Exercise 4.4.** Consider the multilogit model with  $K$  classes. Let  $\beta$  be the  $(p+1)(K-1)$ -vector consisting of all the coefficients. Define a suitable 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 the algorithm.

**Exercise 4.5.** Consider a two-class regression problem with  $x \in \mathbb{R}$ . Characterise the MLE of the slope and intercept parameter if the sample  $x_i$  for the two classes are separated by a point  $x_0 \in \mathbb{R}$ . Generalise this result to  $x \in \mathbb{R}^p$  and more than two classes.

**Exercise 4.6.** Suppose that 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.

- (1) Denote a hyperplane by  $f(x) = \beta^T x^* = 0$ . Let  $z_i = \frac{x_i^*}{\|x_i^*\|}$ . Show that separability implies the existence of a  $\beta_{\text{sep}}$  such that  $y_i \beta_{\text{sep}}^T z_i \geq 1$  for all  $i$ .
- (2) Given a current  $\beta_{\text{old}}$ , the perceptron algorithm identifies a point  $z_i$  that is misclassified, and produces the update  $\beta_{\text{new}} \leftarrow \beta_{\text{old}} + y_i z_i$ . Show that

$$\|\beta_{\text{new}} - \beta_{\text{sep}}\|^2 \leq \|\beta_{\text{old}} - \beta_{\text{sep}}\|^2 - 1$$

and hence that the algorithm converges to a separating hyperplane in no more than  $\|\beta_{\text{start}} - \beta_{\text{sep}}\|^2$  steps.

PROOF. Recall that the definition of separability implies the existence of a separating hyperplane - that is, a vector  $\beta_{\text{sep}}$  such that  $\text{sgn}(\beta_{\text{sep}}^T x_i^*) = y_i$ .

- (1) By assumption, there exists  $\epsilon > 0$  and  $\beta_{\text{sep}}$  such that

$$y_i \beta_{\text{sep}}^T z_i^* \geq \epsilon$$

for all  $i$ . Then the hyperplane  $\frac{1}{\epsilon} \beta_{\text{sep}}$  is a separating hyperplane that by linearity satisfies the constraint

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

(2) We have

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

Let  $\beta_k, k = 0, 1, 2, \dots$  be the sequence of iterates formed by this procedure, with  $\beta_0 = \beta_{\text{start}}$ . Let  $k^* = \lceil \|\beta_{\text{start}} - \beta_{\text{sep}}\|^2 \rceil$ . Then by the above result, we must have  $\|\beta_{k^*} - \beta_{\text{sep}}\|^2 = 0$ , and by properties of the norm we have that  $\beta_{k^*} = \beta_{\text{sep}}$ , and so we have reached a separating hyperplane in no more than  $k^*$  steps.

□

## Basis Expansions and Regularization

**Exercise 5.1.** Show that the truncated power basis functions in (5.3) represent a basis for a cubic spline with the two knots as indicated.

**Exercise 5.2.** Suppose that  $B_{i,M}(x)$  is an order- $M$  B-spline.

- (1) Show by induction that  $B_{i,M}(x) = 0$  for  $x \notin [\tau_i, \tau_{i+M}]$ . This shows, for example, that the support of cubic B-splines is at most 5 knots.
- (2) Show by induction that  $B_{i,M}(x) > 0$  for  $x \in (\tau_i, \tau_{i+M})$ . The B-splines are positive in the interior of their support.
- (3) Show by induction that  $\sum_{i=1}^{K+M} B_{i,M}(x) = 1$  for all  $x \in [\xi_0, \xi_{K+1}]$ .
- (4) Show that

**Exercise 5.3.**

CHAPTER 13

**Support Vector Machines and Flexible Discriminants**