Elements of Statistical Learning Solutions

Daniel Mitsutani

2 Overview of Supervised Learning

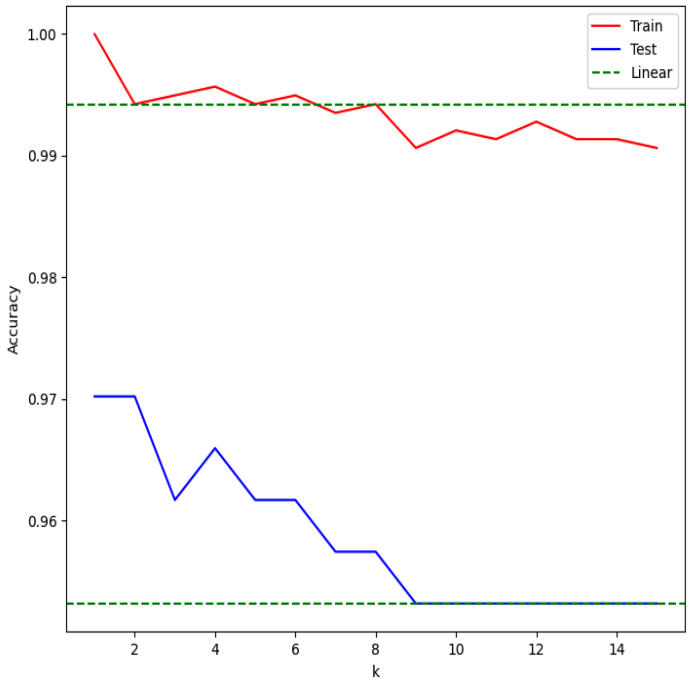
Exercise 2.1. *Derive equation (2.24).*

Solution. A ball of radius r has volume $r^p \operatorname{Vol} B(1)$, where $B(1)$ is the unit ball. Hence the probability that a given point lies inside it is r^p. The probability that a given point lies outside it is $1 - r^p$; the probability that all points lie outside it is $(1 - r^p)^N$. The median smallest distance $d(p, N)$ is the radius r such that the probability above is $1/2$. Solving for r gives

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

Exercise 2.2. *Compare the classification performance of linear regression and k–nearest neighbor classification on the zipcode data. In particular, consider only the 2's and 3's, and $k = 1, 3, 5, 7$ and 15. Show both the training and test error for each choice. The zipcode data are available from the book website www-stat.stanford.edu/ElemStatLearn.*

Solution. The results for different k and linear regression are shown below.



```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier as KNClassifier
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score as accuracy
from matplotlib import pyplot as plt

train_data = np.loadtxt('training_data')
train_df = train_data[np.where
           ((train_data[:, 0] == 2) | (train_data[:, 0] == 3))]

test_data = np.loadtxt('test_data')
test_df = test_data[np.where
          ((test_data[:, 0] == 2) | (test_data[:, 0] == 3))]

X_train, X_test = train_df[:, 1:], test_df[:, 1:]
y_train, y_test = train_df[:, 0].reshape(-1), test_df[:, 0].reshape(-1)

k_list = range(1,16)
classifiers = []
for k in k_list:
    classifier = KNClassifier(k, n_jobs = -1)
    classifier.fit(X_train, y_train)
    classifiers.append(classifier)

accs_train = []
accs_test = []
for i in range(len(k_list)):
    y_train_predict = classifiers[i].predict(X_train)
    y_test_predict = classifiers[i].predict(X_test)
    accs_train.append(accuracy(y_train_predict, y_train))
    accs_test.append(accuracy(y_test_predict, y_test))

lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
linear_train_acc = accuracy(y_train, lin_model.predict(X_train).round())
linear_test_acc = accuracy(y_test, lin_model.predict(X_test).round())
```

```
plt.plot(k_list, accs_train, color = 'red', label = 'Train')
plt.plot(k_list, accs_test, color = 'blue', label = 'Test')
plt.axhline(
    linear_train_acc, color = 'green', label = 'Linear', ls = '--')
plt.axhline(linear_test_acc, color = 'green', ls = '--')

plt.xlabel('k')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

Exercise 2.3. *Suppose we have a sample of N pairs $x_i, y_i i$ drawn i.i.d. from the distribution characterized as follows:*

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

We construct an estimator for f linear in the y_i,

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

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

(a) Show that linear regression and k-nearest-neighbor 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

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

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.

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

Solution.

(a) For linear regression, $\hat{f}(x_0) = x_0^T \beta$, where $\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$, so that

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

For k-nearest-neighbors, we may write:

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i,$$

so that:

$$\ell_i(x_0, \mathcal{X}) = \frac{1}{k} I_{N_k(x_0)}(x_i),$$

and $I_{N_k(x_0)}$ is the indicator function of the k-nearest neighbors of x_0.

(b)

$$\begin{aligned}
\mathrm{E}_{\mathcal{Y}|\mathcal{X}}[(f(x_0) - \hat{f}(x_0))^2] &= \mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)^2] - 2f(x_0)\mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)] + f(x_0)^2 \\
&= \mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)^2] - \mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)]^2 \\
&+ \mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)]^2 - 2f(x_0)\mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)] + f(x_0)^2 \\
&= \mathrm{Var}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0)) + (\mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)] - f(x_0))^2
\end{aligned}$$

The second term is the conditional square bias.

(c) Similarly:

$$\mathrm{E}_{\mathcal{Y},\mathcal{X}}[(f(x_0) - \hat{f}(x_0))^2] = \mathrm{Var}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0)) + (\mathrm{E}_{\mathcal{Y},\mathcal{X}}[\hat{f}(x_0)] - f(x_0))^2$$

(d) We now use the linearity assumption. For simplicity, we write y and $L = \ell(x_0, \mathcal{X})$ to be the coresponding vectors, so that $\hat{f}(x_0) = L^T y$. Further, we let $f(\mathcal{X}) = (f(x_1), \ldots, f(x_n))$ and $\varepsilon = f(\mathcal{X}) - y$. Then

$$\mathrm{E}_{\mathcal{Y}|\mathcal{X}}[\hat{f}(x_0)] - f(x_0) = \mathrm{E}_{\mathcal{Y}|\mathcal{X}}[L^T(f(\mathcal{X}) + \varepsilon)] - f(x_0) = L^T f(\mathcal{X}) - f(x_0)$$

and

$$\mathrm{Var}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0)) = \mathrm{Var}_{\varepsilon}(L^T\varepsilon) = \sigma^2 L^T L$$

Exercise 2.4. *Consider a linear regression model with p parameters, fit by least squares to a set of training data $(x_1, y_1), ..., (x_N, y_N)$ drawn at random from a population. Let $\hat{\beta}$ be the least squares estimate. Suppose we have some test data $(\tilde{x}_1, \tilde{y}_1), ..., (\tilde{x}_M, \tilde{y}_M)$ drawn at random from the same population as the training data. If $R_{tr}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2$ 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 expectations are over all that is random in each expression. [This exercise was brought to our attention by Ryan Tibshirani, from a homework assignment given by Andrew Ng.]

Solution. Note that the terms $(\tilde{y}_i - \hat{\beta}^T \tilde{x}_i)^2$ are i.i.d. in $(\tilde{x}_i, \tilde{y}_i)$ since $\hat{\beta}$ only depends on the training data. Thus, the RHS of the inequality is independent of M and we may set $M = N$.

For the test data $(\tilde{x}_1, \tilde{y}_1), ..., (\tilde{x}_N, \tilde{y}_N)$, let $\tilde{\beta}$ be the least squares estimate of β. By definition $R_{te}(\hat{\beta}) \geq R_{te}(\tilde{\beta})$ so that:

$$E[R_{te}(\hat{\beta})] \geq E[R_{te}(\tilde{\beta})].$$

On the other hand, since the number of samples and the distribution of the test data and training data are the exact same, $R_{te}(\tilde{\beta})$ is equal in distribution to $R_{tr}(\hat{\beta})$. Thus the RHS of the above inequality is in fact equal to $E[R_{tr}(\hat{\beta})]$, and we are done.

3 Linear Methods for Regression

Exercise 3.1. *Show that the F statistic (3.13) for dropping a single coefficient from a model is equal to the square of the corresponding z-score (3.12).*

Solution. Let $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_1$ be the estimators of $\mathbf{y}$ with the j-th variable removed and the full model, respectively. Then $\hat{\mathbf{y}}_0$ is the projection of $\mathbf{y}$ onto the span of the $\mathbf{x}_i$ with $i \neq j$ and $\hat{\mathbf{y}}_1$ is the projection of $\mathbf{y}$ onto the span of the $\mathbf{x}_i$. By the Pythagorean Theorem:

$$\mathrm{RSS}_0 = \|\hat{\mathbf{y}}_0 - \mathbf{y}\|^2 = \|\hat{\mathbf{y}}_1 - \mathbf{y}\|^2 + \frac{\langle \mathbf{y}, \mathbf{z}_j \rangle^2}{\|\mathbf{z}_j\|^2} = \mathrm{RSS}_1 + \frac{\langle \mathbf{y}, \mathbf{z}_j \rangle^2}{\|\mathbf{z}_j\|^2},$$

where $\mathbf{z}_j$ is the residual of $\mathbf{x}_j$ regressed on the remaining $\mathbf{x}_i$.

Further, $\mathrm{RSS}_1 = (N - p_1 - 1)\hat{\sigma}^2$ by (3.8) so that:

$$F = (N - p_1 - 1) \cdot \frac{\mathrm{RSS}_0 - \mathrm{RSS}_1}{\mathrm{RSS}_1} = \frac{\langle \mathbf{y}, \mathbf{z}_j \rangle^2}{\hat{\sigma}^2 \|\mathbf{z}_j\|^2} = \frac{\hat{\beta}_j^{\,2} \|\mathbf{z}_j\|^2}{\hat{\sigma}^2},$$

where we used (3.28). To conclude, recall that:

$$\mathrm{Var}(\hat{\beta}_j) = v_j \sigma^2 = \frac{\sigma^2}{\|z_j\|^2},$$

and use the definition of the z-score.

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 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.

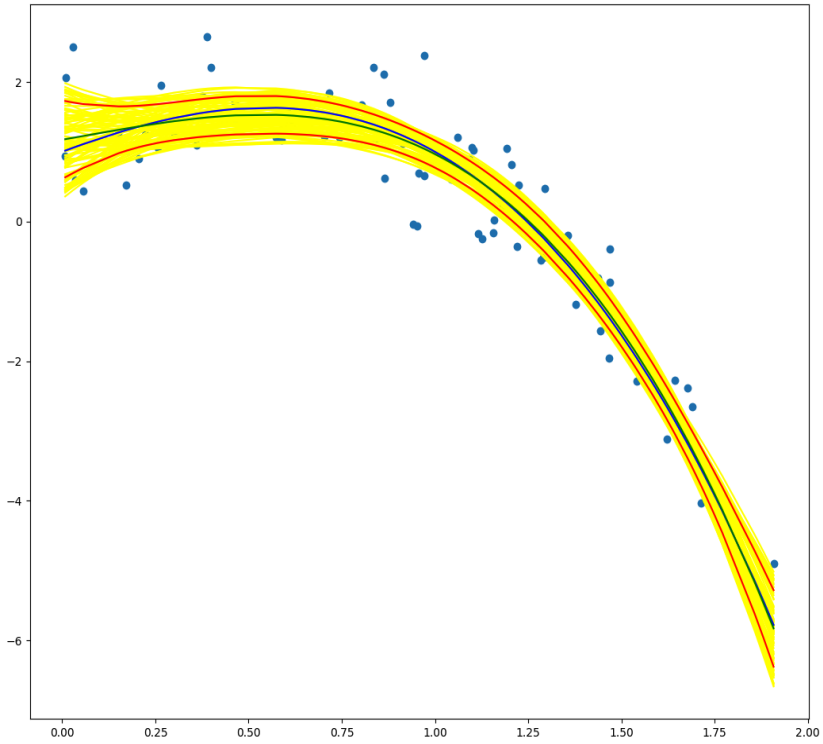


Figure 1: The blue and green lines are the actual and predicted (by OLS) curves for a cubic polynomial. The yellow shaded region are in fact the graphs of 100 $\beta's$ obtained by method 2 for different δ uniformly distributed on the sphere of appropriate radius. The red lines are the error bands for method 1.

Solution. For method 1, since $\hat{\beta} \sim N(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$ then $a^T\hat{\beta} \sim N(a^T\beta, a^T(\mathbf{X}^T\mathbf{X})^{-1}a\sigma^2)$. Hence the 95% confidence interval for $x_0^T\beta$ is:

$$x_0^T\hat{\beta} \pm 1.96 \cdot \sigma\sqrt{x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0}.$$

For method 2, the confidence set for β is given by:

$$C_\beta = \{\beta : (\beta - \hat{\beta})^T\mathbf{X}^T\mathbf{X}(\beta - \hat{\beta}) \leq \hat{\sigma}^2\chi^2(0.95, 4)\}$$

Letting LL^T be the Cholesky decomposition of $\mathbf{X}^T\mathbf{X}$, any β on the boundary of C_β is of the form:

$$\beta = \hat{\beta} + (L^T)^{-1}\delta,$$

where δ is a vector on the sphere of radius $\hat{\sigma}\sqrt{\chi^2(0.95, 4)}$. The error on y is then given by $x_0^T(L^T)^{-1}\delta$. If δ is the most expanded direction (in the SVD) of $(L^T)^{-1}$, this may exceed the error of the former method. We illustrate this in the image above generated from the code that follows.

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.stats import chi2

x_min = 0
x_max = 2
n = 100
sigma = 0.7
p = 0.95
df = 4
num = 100

f = lambda x: 1 + 2 * x - x ** 2 - x ** 3
f_vec = np.vectorize(f)

def make_X(n, x_min, x_max):
    x0 = np.ones(n)
    x1 = np.sort((x_max-x_min) * np.random.random(n) + x_min)
    x2 = np.square(x1)
    x3 = np.power(x1, 3)
    return np.column_stack((x0, x1, x2, x3))

X = make_X(n, x_min, x_max)

y_actual = f_vec(X[: , 1].T)
y_observed = y_actual + sigma*np.random.normal(0, sigma, n)
y_var = np.sqrt(np.diag(X @ np.linalg.inv(X.T @ X) @ X.T))

beta_ols = np.linalg.inv(X.T @ X) @ X.T @ y_observed
y_predict = X @ beta_ols

y_max_1 = y_predict + 1.96 * sigma * y_var
y_min_1 = y_predict - 1.96 * sigma * y_var

L = np.linalg.cholesky(X.T @ X)
U = L.T
U_inv = np.linalg.inv(U)
for i in range(num):
    delta = np.random.normal(0,1,4)
    delta = delta * sigma * np.sqrt(chi2.ppf(p, df))
            /(np.linalg.norm(delta, ord = 2))
    delta = U_inv @ delta
    beta_sim = beta_ols + delta
    plt.plot(X[:, 1], X @ beta_sim, color = 'yellow')
```

```
plt.scatter(X[:, 1], y_observed)
plt.plot(X[:, 1], y_actual, color = 'blue')
plt.plot(X[:, 1], y_predict, color = 'green')
plt.plot(X[:, 1], y_max_1, color = 'red')
plt.plot(X[:, 1], y_min_1, color = 'red')
plt.show()
```

Exercise 3.3. *Gauss–Markov theorem:*

(a) Prove the Gauss–Markov theorem: the least squares estimate of a parameter $a^T\beta$ has variance no bigger than that of any other linear unbiased estimate of $a^T\beta$ (Section 3.2.2).

(b) The matrix inequality $B \preceq A$ holds if $B - A$ is positive semidefinite. Show that if $\hat{\boldsymbol{V}}$ is the variance-covariance matrix of the least squares estimate of β and $\tilde{\boldsymbol{V}}$ is the variance-covariance matrix of any other linear unbiased estimate, then $\hat{\boldsymbol{V}} \preceq \tilde{\boldsymbol{V}}$.

Solution.

(a) Write $c^T = a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T + \Delta^T$. Then:

$$\mathrm{E}(c^T y) = \mathrm{E}((a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T + \Delta^T)y) = a^T\beta + \mathrm{E}(\Delta^T X\beta),$$

and since $c^T y$ is unbiased we have $\Delta^T X = 0$.

For simplicity, let $a^T(\mathbf{X}^T\mathbf{X})^{-1} = M$. Now we compute the variance:

$$\begin{aligned}
\mathrm{Var}(c^T y) &= \mathrm{E}(y^T c c^T y) - (a^T\beta)^2 \\
&= \mathrm{E}(y^T(M\mathbf{X}^T + \Delta^T)^T(M\mathbf{X}^T + \Delta^T)y) - (a^T\beta)^2 \\
&= \mathrm{E}(y^T(M\mathbf{X}^T)^T(M\mathbf{X}^T)y) - (a^T\beta)^2 + \mathrm{E}(y^T\Delta\Delta^T y) \\
&= \mathrm{Var}_{OLS} + \mathrm{E}((\Delta^T y)^2),
\end{aligned}$$

where on the third line we used $\Delta^T X = 0$. Since $\mathrm{E}((\Delta^T y)^2) \geq 0$, this concludes the proof.

(b) The proof is the same but in matrix form.

Exercise 3.4. *Show how the vector of least squares coefficients can be obtained from a single pass of the Gram-Schmidt procedure (Algorithm 3.1). Represent your solution in terms of the QR decomposition of $\boldsymbol{X}$.*

Solution. By (3.32), $\hat{\beta} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y}$. Since $\mathbf{Q} = \mathbf{Z}\mathbf{D}^{-1}$, $\mathbf{Q}$ can be computed as one obtains $\mathbf{z}_i$ in the Gram-Schmidt algorithm by entering the normalized $\mathbf{z}_i$. Similarly, $\mathbf{R} = \mathbf{D}\boldsymbol{\Gamma}$, is immediately found as the entries of $\hat{\gamma}_{kj}$ with the normalized $\mathbf{z}_i$. Since $\mathbf{R}$ is upper triangular, it can be inverted quickly.

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

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

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

Solution. Letting $\beta_0 = \beta_0^c - \sum_{j=1}^{p} \bar{x}_j\beta_j^c$, and $\beta_i = \beta_i^c$ for $i \neq 0$, the problem above becomes the usual lasso

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

From the relations between β and β^c, we see that the slope coefficients for the centered problem are the same, whereas the intercept $\beta_0^c = \beta_0 + \bar{\mathbf{x}}^T\beta$ is the predicted value (by the usual lasso) at the mean of the data. The analysis for ridge is identical, since it only changes the penalty term.

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

Solution. By Bayes' formula:

$$f(\beta|\mathbf{y}, \mathbf{X}) \propto f(\mathbf{y}, \mathbf{X}|\beta)f(\beta) \propto f(\mathbf{y}|\mathbf{X}, \beta)f(\beta),$$

since $f(\mathbf{X})$ does not depend on β. Thus:

$$\begin{aligned} f(\beta|\mathbf{y}, \mathbf{X}) &\propto \exp\left(-\frac{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}{2\sigma^2} - \frac{1}{2\tau^2}\beta^T\beta \right) \\ &\propto \exp\left(\frac{-1}{2\sigma^2} \left(-\beta^T \left(\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I} \right) \beta + \mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{y} \right) \right) \end{aligned}$$

Recall in general that a multivariate Gaussian in β with mean μ and covariance matrix Σ has pdf proportional to

$$\exp\left(-\frac{(\beta - \mu)^T\Sigma^{-1}(\beta - \mu)}{2} \right) \propto \exp\left(-\frac{\beta^T\Sigma^{-1}\beta + \beta^T\Sigma^{-1}\mu + \mu^T(\Sigma^{-1})^T\beta}{2} \right)$$

Comparing the two, we see that the pdf $f(\beta|\mathbf{y}, \mathbf{X})$ is indeed Gaussian with

$$\Sigma^{-1} = \frac{1}{\sigma^2}\left(\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right),$$

and moreover we see that $\mathbf{X}^T\mathbf{y} = \sigma^2\Sigma^{-1}\mu$ by comparing the third term in $f(\beta|\mathbf{y}, \mathbf{X})$ with the second term in the general Gaussian pdf. Solving for μ and plugging Σ^{-1} found as above gives:

$$\mu = \left(\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y},$$

which is precisely β^{ridge} with $\lambda = \sigma^2/\tau^2$, as desired.

Exercise 3.7. *Consider the decomposition of the uncentered $N \times (p+1)$ matrix $\boldsymbol{X}$ whose first column is all ones), and the SVD of the $N \times p$ centered matrix $\tilde{\boldsymbol{X}}$. Show that $\boldsymbol{Q}_2$ and $\boldsymbol{U}$ span the same subspace, where $\boldsymbol{Q}_2$ is the sub-matrix of $\boldsymbol{Q}$ with the first column removed. Under what circumstances will they be the same, up to sign flips?*

Solution. Recall that $\mathbf{Q} = \mathbf{Z}\mathbf{D}^{-1}$, where $\mathbf{Z}$ is the vector of residuals obtained in the Gram-Schmidt algorithm. Hence, $\mathbf{q}_0$ is a scalar multiple of $\mathbf{1}$, since $\mathbf{z}_0 = \mathbf{1}$. Since $\mathbf{Q}$ is orthogonal, the span of $\mathbf{Q}_2$ is equal to $\text{span}(\mathbf{Q}) \cap \mathbf{1}^\perp$, where $\perp$ denotes orthogonal complement. Moreover $\text{span}(\mathbf{Q}) = \text{span}(\mathbf{X})$, since $\mathbf{R}$ is invertible. Hence $\text{span}(\mathbf{Q}_2) = \text{span}(\mathbf{X}) \cap \mathbf{1}^\perp$.

Clearly also $\text{span}(\mathbf{U}) = \text{span}(\tilde{\mathbf{X}})$, since again $\mathbf{DV}$ is invertible. But note that $\tilde{\mathbf{x}}_i$ is simply the projection of $\mathbf{x}_i$ onto $\mathbf{1}^\perp$, since $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{x}_i\mathbf{1}$, where $\bar{x}_i = (1/N)\mathbf{x}_i^T\mathbf{1}$. Hence $\text{span}(\mathbf{U}) = \text{span}(\tilde{\mathbf{X}}) = \text{span}(\mathbf{X}) \cap \mathbf{1}^\perp$ and the first part of the exercise is complete.

For these to agree, we need $\mathbf{DV} = \mathbf{R}$, i.e., an orthogonal matrix is equal to an upper-triangular one. Thus both have to be the identity (up to sign flips) and $\mathbf{X} = \mathbf{Q}$ (up to sign flips), i.e. the feature input vectors $\mathbf{x}_i$ must have been orthogonal to begin with.

Exercise 3.8. *Forward stepwise regression. Suppose we have the QR decomposition for the $N \times p$ matrix $\boldsymbol{X}_1$ in a multiple regression problem with response $\boldsymbol{y}$, and we have an additional $p-q$ predictors in the matrix $\boldsymbol{X}_2$. Denote the current residual by $\boldsymbol{r}$. We wish to establish which one of these additional variables will reduce the residual sum-of-squares the most when included with those in $\boldsymbol{X}_1$. Describe an efficient procedure for doing this.*

Solution. Let $\mathbf{y}_0$ be the current predictor of $\mathbf{y}$, i.e., $\mathbf{y} = \mathbf{y}_0 + \mathbf{r}$. Then $\mathbf{y}_0$ is the orthogonal projection of $\mathbf{y}$ onto the column span of $\mathbf{Q}$. If we add a new column $\mathbf{x}_{p+1}$, the new residual will be given by the projection of $\mathbf{R}$ we on the span columns of $\mathbf{Q}$ and $\mathbf{x}$. Letting $\mathbf{z}_{p+1} = \mathbf{x}_{p+1} - \sum_{i=1}^{p}\langle \mathbf{x}_{p+1}, \mathbf{q}_i\rangle \mathbf{q}_i$ be the orthogonal component of $\mathbf{x}_{p+1}$ relative to the column span of $\mathbf{Q}$, we can write the residue $\mathbf{r}$ as

$$\mathbf{r} = \mathbf{r}' + \frac{\langle \mathbf{r}, \mathbf{z}_{p+1}\rangle}{\langle \mathbf{z}_{p+1}, \mathbf{z}_{p+1}\rangle}\mathbf{z}_{p+1},$$

where $\mathbf{r}'$ will be the new residues after adding $\mathbf{x}_{p+1}$. Hence the difference in RSS is given by $|\langle \mathbf{r}, \mathbf{z}_{p+1} \rangle| / \|\mathbf{z}_{p+1}\|$. We can perform this procedure for all columns of $\mathbf{X}_2$ and find the largest value.

Note that this also allows us to update the QR decomposition for the next step as well via $\mathbf{q}_{p+1} = \mathbf{z}_{p+1} / \|\mathbf{z}_{p+1}\|$.

Exercise 3.9. *Backward stepwise regression. Suppose we have the multiple regression fit of $\boldsymbol{X}$ on $\boldsymbol{y}$, along with the standard errors and Z-scores as in Table 3.2. We wish to establish which variable, when dropped, will increase residual sum-of-squares the least. How would you do this?*

Solution. By Exercise 3.1, the residual sum of squares difference (F-score) is proportional to the Z-score when dropping a single variable. Hence we simply pick the variable with smallest Z-score to drop.

Exercise 3.10. *Show that the ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set. We augment the centered matrix $\boldsymbol{X}$ with p additional rows $\sqrt{r}\mathbf{I}$, 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.*

Solution. The ordinary least squares regression solves:

$$\hat{\beta}^{OLS} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2$$

For the augmented matrix $\mathbf{X}$ and the augmented $\mathbf{y}$, by the Pythagorean theorem this is simply:

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \|\mathbf{0} - \sqrt{\lambda}\beta\|^2 = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2,$$

which is the Ridge regression estimator.

Exercise 3.11. *Suppose for a given t in (3.51) the fitted lasso coefficient for the 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.*

Solution. We will consider adding a copy of the last column $\mathbf{x}_p$. The constrained form

of lasso regression is

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p-1} x_{ij}\beta_j - x_{ip}\beta_p \right)^2.$$

subject to $\sum_{j=1}^{p} |\beta_j| \leq t$. By augmenting with a copy as explained, we add a column $x_{i(p+1)} = x_{ip}$ with coefficient β_{p+1}, so the new optimization problem is:

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

subject to $\sum_{j=1}^{p+1} |\beta_j| \leq t$. Let β_p^0 be the estimator for β_p for the original problem without the copied variable. For the new problem, as long as $\beta_p + \beta_{p+1} = \beta_p^0$ and $|\beta_p| + |\beta_{p+1}| = |\beta_p^0|$ with other coefficients being equal then we have a solution to the new optimization problem which is on the boundary.

Therefore adding the copy of a variable creates ambiguity on the coefficients of the copied variable and its copy, i.e., any choice of the form $\beta_p = t\beta_p^0$, $\beta_{p+1} = (1-t)\beta_p^0$ for $t \in [0, 1]$ will be a solution to lasso regression.

7 Model Assesment and Selection

Exercise 7.1. *Derive the estimate of in-sample error (7.24).*

Solution. The estimate (7.24) is obtained by plugging (7.23) into (7.22). It remains to derive (7.23), assuming additive errors $Y = f(X) + \varepsilon$ and a linear model $\hat{Y} = \mathbf{S}Y$. In what follows since we deal with in-sample errors all expectations are taken over Y, i.e., X is assumed to be fixed. Note that

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

so that it remains to expand the covariance matrix:

$$\begin{aligned}
\mathrm{trace}\,(\mathrm{Cov}(\hat{Y}, Y)) &= \mathrm{trace}\,(\mathrm{E}[(\hat{Y} - \mathrm{E}(\hat{Y}))(Y - \mathrm{E}[Y])^T]) \\
&= \mathrm{trace}\,(\mathrm{E}[(\mathbf{S}(f(X) + \varepsilon) - \mathrm{E}(\mathbf{S}(f(X) + \varepsilon)))(f(X) + \varepsilon - \mathrm{E}[f(x) + \varepsilon])^T]) \\
&= \mathrm{trace}\,(\mathrm{E}[(\mathbf{S}(f(X) + \varepsilon) - \mathrm{E}[\mathbf{S}f(X)])(f(X) + \varepsilon - \mathrm{E}[f(X)])^T]) \\
&= \mathrm{trace}\,(\mathrm{E}[(\mathbf{S}(f(X) + \varepsilon) - \mathrm{E}[\mathbf{S}f(X)])\varepsilon^T) \\
&= \mathrm{trace}\,(\mathrm{E}[\mathbf{S}\varepsilon\varepsilon^T]),
\end{aligned}$$

since all the terms not involving ε cancel out and $\mathrm{E}[\varepsilon] = 0$. Using the linearity of expectation:

$$\mathrm{trace}\,(\mathrm{E}[\mathbf{S}\varepsilon\varepsilon^T]) = \mathrm{trace}\,(\mathbf{S}\mathrm{E}[\varepsilon\varepsilon^T])] = \sigma_\varepsilon^2\,\mathrm{trace}(\mathbf{S}),$$

since ε is assumed to have mean 0 and variance $\sigma_\varepsilon^2 I_N$. By definition $d = \mathrm{trace}(\mathbf{S})$, which completes the proof.

Exercise 7.2. *Consider the in-sample prediction error (7.18) and the training error* $\overline{err}$ *in the case of squared-error loss:*

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{Y^0}(Y_i^0 - \hat{f}(x_i))^2$$

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

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

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

as given in (7.21).

Solution. Write $\hat{y}_i = \hat{f}(x_i)$. Adding and subtracting we can group:

$$Y_i^0 - \hat{f}(x_i) = (Y_i^0 - f(x_i)) + (f(x_i) - \mathrm{E}\hat{y}_i) + (\mathrm{E}\hat{y}_i - \hat{y}_i)$$

$$y_i - \hat{f}(x_i) = (y_i - f(x_i)) + (f(x_i) - \mathrm{E}\hat{y}_i) + (\mathrm{E}\hat{y}_i - \hat{y}_i)$$

Squaring these terms and subtracting and taking E_{Y^0} of the first gives:

$$\begin{aligned} E_{Y^0}(Y_i^0 - \hat{f}(x_i))^2 - (y_i - \hat{f}(x_i))^2 = E_{Y^0}(Y_i^0 - f(x_i))^2 - (y_i - f(x_i))^2 \\ - 2(y_i - f(x_i))[(f(x_i) - \mathrm{E}\hat{y}_i) + (\mathrm{E}\hat{y}_i - \hat{y}_i)], \end{aligned}$$

since $E_{Y^0}(Y_i^0 - \hat{f}(x_i)) = 0$ and the terms involving only the second and third elements of the sum cancel out. Hence the i-th term of the sum $\omega = E_{\mathbf{y}}(\mathrm{Err}_{in} - \overline{\mathrm{err}})$ is

$$E_{Y^0}(Y_i^0 - f(x_i))^2 - \mathrm{E}_{\mathbf{y}}(y_i - f(x_i))^2 + 2E_{\mathbf{y}}[(y_i - f(x_i))[(f(x_i) - \mathrm{E}\hat{y}_i) + (\hat{y}_i - \mathrm{E}\hat{y}_i)]]$$

The first and second term cancel since Y^0 is sampled from the same distribution as $\mathbf{y}$ and note that

$$2E_{\mathbf{y}}[(y_i - f(x_i))(f(x_i) - \mathrm{E}\hat{y}_i)] = 0,$$

since the second term in the product is constant and $E_{\mathbf{y}}(y_i - f(x_i)) = 0$. Hence what is left is:

$$2E_{\mathbf{y}}[(y_i - f(x_i))(\hat{y}_i - \mathrm{E}\hat{y}_i)] = \mathrm{Cov}\,(y_i, \hat{y}_i).$$

Adding these up finishes the proof.