Statistical Applications

APPM 4580/5580, Spring 2016

William Kleiber[1]

1 Introduction

Statistical learning refers to a broad set of tools that can be used to explore and model complex datasets. There are overlaps with computer science and machine learning, where statistical science often places a greater emphasis on identifying and quantifying sources of *uncertainty*.

[The spam dataset consists of word frequencies and whether or not an e-mail is marked as spam] In this case, the word frequencies are *input variables*, while spam/not spam is the *output variable*. Our goal is to characterize a relationship between inputs and outputs.

Notation 1. *X will refer to* input variables*, which also go by names* predictors, independent variables, features *or* variables*. The* output variable *will be denoted by Y, and is sometimes called the* response *or* dependent variable*.*

With multiple inputs, we distinguish between them using adorned Xs, e.g., X_1 =free, X_2=address and X_3=length of longest capital word in e-mail.

Given p types of predictors $X_1, \ldots, X_p$, we assume there is some type of relationship

$$Y = f(X_1, \ldots, X_p) + \varepsilon$$

where f is a fixed but unknown function of the inputs and ε is a random, mean zero error term. In this model, f represents the *systematic* information that $X_1, \ldots, X_p$ provide about Y, while ε represents stochastic (or random) error that cannot be explained using $X_1, \ldots, X_p$.

[1]Department of Applied Mathematics, University of Colorado, Boulder, CO. Author e-mail: william.kleiber@colorado.edu

1.1 Typical Goals

The two major goals are to

- Find relationships between a group of *explanatory* variables and a *response* variable that provides good predictive performance
- Reduce the *size* of a group of variables for scientific or computational purposes.

Some possibilities are:

- Classification and regression
- Algorithms for large data analysis
- Recommender systems
- Spam filters
- Text processing
- Disease monitoring

Generally, most of the above reduce to estimating f in

$$Y = f(X_1, \ldots, X_p) + \varepsilon.$$

We may wish to do this for many reasons, which can be roughly categorized into *prediction* and *inference*.

Prediction

In many cases, the input variables $X_1, \ldots, X_p$ are easily available (or are variables that we can control), whereas Y is quantity of main interest. If we could estimate f, say using $\hat{f}$, and we knew $X_1, \ldots, X_p$, then we could *predict* Y using

$$\hat{Y} = \hat{f}(X_1, \ldots, X_p).$$

For example, we might like to know if an e-mail with a 20-letter sequence of capital letters is likely to be spam or not.

The accuracy of $\hat{Y}$ as a predictor for Y depends on two quantities of *reducible error* and *irreducible error*. Notice the expected squared error between our predictor and the response is

$$\begin{aligned}
\mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}(f(X_1, \ldots, X_p) + \varepsilon - \hat{f}(X_1, \ldots, X_p))^2 \\
&= \mathbb{E}(f + \varepsilon - \hat{f})^2 \\
&= \mathbb{E}\big((f - \hat{f})^2 + \varepsilon^2 - 2\varepsilon(f - \hat{f})\big) \\
&= \mathbb{E}(f - \hat{f})^2 + \operatorname{Var}\varepsilon.
\end{aligned}$$

The first term is *reducible* while the second term is *irreducible*. In other words, by using more data or better learning techniques, we can improve the estimate of $\hat{f} \approx f$, whereas ε is random and cannot be predicted. For prediction, our goal will be to *minimize* this expected squared error $\mathbb{E}(Y - \hat{Y})^2$.

Inference

Inference refers to the act of estimating f and characterizing/ quantifying the unpredictable error ε using a stochastic model. Major inferential questions are:

- Which predictors are associated with the response? (Does frequency of "budget" imply an e-mail is spam?)
- What is the relationship between Y and each X_i? (Does frequency of "budget" increase or decrease likelihood an e-mail is spam?)
- Are these relationships linear or nonlinear? (Can we use $f(X) = \beta_0 + \beta_1 X$ or do we need something more complicated?)

1.2 Some Jargon

Parametric vs. Nonparametric

Learning methods for estimating f can be roughly categorized as *parametric* or *nonparametric* (or a mix between the two which is sometimes called *semiparametric*). For example, a parametric model for $Y =$ global temperature given $X =$ CO2 concentration might use

$$f(X) = \beta_0 + \beta_1 X$$

or in other words

$$Y \approx \beta_0 + \beta_1 X.$$

The major obstacle is then to *estimate* the unknown *parameters* β_0 and β_1, for example using least squares. Once we have estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we can predict Y by

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X.$$

A nonparametric model makes no explicit assumption about the form of the relationship between Y and $X_1, \ldots, X_p$, but seeks an f that gets close to a set of training data points without being too wiggly or rough. Typically nonparametric methods are more flexible, but are not as easy to interpret as parametric models, and the modeler must weigh between these two as techniques are applied.

Supervised vs. Unsupervised

Learning techniques can be either *supervised* or *unsupervised*. Supervised techniques apply to the cases where we have a response of interest Y, given some covariates $X_1, \ldots, X_p$. Most of this class is devoted to supervised learning. Unsupervised learning applies when no response Y is available, but we are interested in learning about (for example) the *relationship* between covariates. For example, given $X_1, \ldots, X_n$, *cluster analysis* seeks to determine if the observations fall into distinct groups.

[E.g., given a set of e-mails, determine which is spam, or given a set of Netflix ratings on a shared account, how to tell which rating comes from which user]

Regression vs. Classification

A variable X (or Y) is either *quantitative*, $X \in \mathbb{R}$, or *qualitative*, $X \in \{s_1, \ldots, s_m\}$. Sometimes qualitative variables are known as *categorical*. For instance, salary would be a quantitative variable whereas gender would be a categorical variable as it breaks up into *classes* or categories. Problems with a quantitative response are referred to as *regressions* while qualitative responses are known as *classifications*.

A Common Theme: Bias-Variance Tradeoff

A common way to assess the quality of fit of a model is by examining the *predictive mean squared error*. That is, for a new observation $Y = f(X) + \varepsilon$ and given an estimator $\hat{f}(X)$, we want to minimize

$$\mathbb{E}(Y - \hat{f}(X))^2 = \mathbb{E}(Y - \hat{f})^2.$$

Before we get data, $\hat{f}$ is random (because it depends on unobserved random data $Y_1, \ldots, Y_n$). Assuming ε and $\hat{f}$ are uncorrelated, and that f is fixed, we have

$$\begin{aligned}
\mathbb{E}(Y - \hat{f})^2 &= \mathbb{E}(f + \varepsilon - \hat{f})^2 \\
&= \mathbb{E}(f - \hat{f})^2 + \mathbb{E}\varepsilon^2 + 2\mathbb{E}(\varepsilon(f - \hat{f}))^2 \\
&= \mathbb{E}(f - \mathbb{E}\hat{f} + \mathbb{E}\hat{f} - \hat{f})^2 + \mathrm{Var}\varepsilon \\
&= (f - \mathbb{E}\hat{f})^2 + \mathbb{E}(\hat{f} - \mathbb{E}\hat{f})^2 + \mathrm{Var}\varepsilon \\
&= (f - \mathbb{E}\hat{f})^2 + \mathrm{Var}\hat{f} + \mathrm{Var}\varepsilon \\
&= (\text{bias of } \hat{f})^2 + \text{variance of } \hat{f} + \text{random error}
\end{aligned}$$

since $\mathbb{E}(\mathbb{E}\hat{f} - \hat{f}) = 0$.

[Picture of squared-bias/variance tradeoff with sum as MSE]

The predictive mean squared error breaks into the squared bias of $\hat{f}$, the variance of $\hat{f}$ and an irreducible and unpredictable variance of the error term. As the model for $\hat{f}$ becomes more flexible, we reduce the bias but increase the variance, whereas more rigid models exhibit smaller variance at the cost of increased bias. [Picture of MSE on y-axis as function of model complexity on x-axis with squared bias decreasing variance increasing]

2 Linear Regression

[R example (RegressionExamples.R)]

Linear regression is a simple but powerful approach for supervised learning. Given a set of predictors (or features or covariates or explanatory variables) $X_1, \ldots, X_p$ and a response (or supervisor) Y, we should be able to answer the following questions:

- Is there a relationship between Y and X_i? If so, how strong is it?
- How accurately can we quantify a relationship?
- How accurately can we predict Y, given a set of covariates?
- Is the relationship linear, and are there interactions between $X_1, \ldots, X_p$?

2.1 Simple Linear Regression

In the simple linear regression case, we have a single quantitative response Y and a single predictor X. We assume that, up to some random error, Y and X share a linear relationship,

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{1}$$

where ε is the random error term. Sometimes we say Y is being *regressed on* X. This is our first example of a *statistical model*. Here, β_0 and β_1 are unknown *coefficients* or *parameters* which must be estimated. Note this is equivalent to our typical model, where now $f(X) = \beta_0 + \beta_1 X$. This function is known as the *population regression line*, and ε is the *residual*. Of course, β_0 is the estimated value of Y for $X = 0$ and β_1 is the increase in Y for a unit increase of X.

In practice we have a set of n *observation pairs*

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

of both covariates x_i and responses y_i. If the relationship (1) holds, then we have a set of observation equations

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_1 + \varepsilon_1 \\ &\vdots \\ y_n &= \beta_0 + \beta_1 x_n + \varepsilon_n \end{aligned}$$

where the ε_i are now fixed numbers. Equivalently, a priori these are random variables so

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where ε_i are random.

There are differing levels of assumptions that are typically made in practice regarding (1).

A1
1. The relationship (1) holds
2. ε_i are iid (independent and identically distributed) $N(0, \sigma^2)$ for all $i = 1, \ldots, n$

A2
1. The relationship (1) holds
2. $\mathbb{E}\varepsilon_i = 0$
3. $\mathrm{Var}\varepsilon_i = \sigma^2$ (homoskedasticity)
4. ε_i and ε_j are iid

A3
1. The relationship (1) holds
2. $\mathbb{E}\varepsilon_i = 0$
3. $\mathrm{Var}\varepsilon_i = \sigma^2$ (homoskedasticity)
4. ε_i and ε_j are uncorrelated for $i \neq j$

A4
1. The relationship (1) holds
2. $\mathbb{E}\varepsilon_i = 0$
3. $\mathrm{Var}\varepsilon_i < \infty$.

[Picture of A1: what would we expect plot to look like with $n \gg 0$?]

What is common to all levels of assumptions is that the posited relationship (1) holds. The differences between the three assumptions are about the assumed behavior of the residuals. A1 is the strongest in that we explicitly assume the residuals are normally distributed, whereas A2 relaxes this assumption to being iid from some unspecified probability distribution. Statistical independence is a stronger statement than correlation, and A4 is the weakest in that we only put assumptions on the first two *moments* of the variables. For instance, $\mathbb{E}\varepsilon_1^3$ may be different than $\mathbb{E}\varepsilon_2^3$ in A3, but not for A2 or A1.

The fourth assumption is the weakest, and relaxes the *homoskedasticity* property (equal variance among all residuals) by allowing for *heteroskedasticity* (nonconstant variance). We will work with A1. Our goals are usually

1. Estimate parameters β_0, β_1 and σ^2 (and quantify uncertainty in our estimates)
2. Assess the validity of our assumed model (1)
3. Predict response at a new covariate value $X = x$.

2.2 OLS

The heuristic for estimating the regression parameters β_0 and β_1 is that we should minimize the distance between y_i and its fitted value $\beta_0 + \beta_1 x_i$.

[Picture of data with candidate lines]

The *ordinary least squares* (OLS) estimators for β_0 and β_1 are found by minimizing

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

where RSS refers to the *residual sum of squares*. Note that other types of distance can be used, but least squares is the most popular, in part due to the simplicity of the solution (and also a connection to maximum likelihood).

Doing this minimization results in the *least squares estimators*

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2}$$
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

where $\overline{x} = n^{-1} \sum_{i=1}^n x_i$ and $\overline{y} = n^{-1} \sum_{i=1}^n y_i$. Notice that the estimators are *linear* in the observations. It can be shown that the OLS estimators are the *best linear unbiased estimators* (BLUEs) under squared loss. Note

$$\mathbb{E} y_i = \beta_0 + \beta_1 x_i$$

and

$$\mathbb{E} \overline{y} = \beta_0 + \beta_1 \overline{x}.$$

The OLS estimators are *unbiased* in that

$$\begin{aligned}
\mathbb{E}\hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \overline{x}) \mathbb{E}(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2} \\
&= \frac{\sum_{i=1}^n (x_i - \overline{x})(\beta_1 (x_i - \overline{x}))}{\sum_{i=1}^n (x_i - \overline{x})^2} \\
&= \beta_1 \frac{\sum_{i=1}^n (x_i - \overline{x})^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \\
&= \beta_1
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\hat{\beta}_0 &= \mathbb{E}\overline{y} - \overline{x} \mathbb{E}\hat{\beta}_1 \\
&= \beta_0 + \beta_1 \overline{x} - \overline{x} \beta_1 \\
&= \beta_0.
\end{aligned}$$

Based on our estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we have *fitted values*

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

and *estimated residuals*

$$\hat{\varepsilon}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i), \quad i = 1, \ldots, n.$$

[Draw picture of estimated residuals vs true residuals] Lastly, we must estimate σ^2. An unbiased estimate of σ^2 can be found by scaling the residual sum of squares (RSS) by its degrees of freedom,

$$\hat{\sigma}^2 = \frac{RSS}{n-2} = \frac{1}{n-2}\sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

where the -2 occurs since we are estimating two parameters β_0 and β_1 in the mean function. Recall the usual unbiased estimator for variance divides by $n-1$, which is when only β_0 is being estimated. Sometimes $\hat{\sigma}$ is known as the *standard error of regression.*

[R example (IntroLinearRegression.R)]

Given estimates β_0, β_1, and σ, can we assess the uncertainty in these? Note that $\hat{\beta}_0$ and $\hat{\beta}_1$ both involve y, so before the experiment is performed these can be considered random variables. Additionally note that if we work under A3, the ε_i are uncorrelated, mean zero and variance σ^2, so

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}_1) &= \mathrm{Var}\frac{\sum_{i=1}^{n}(x_i-\overline{x})\varepsilon_i}{\sum_{i=1}^{n}(x_i-\overline{x})^2} \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i-\overline{x})^2\right)^2}\mathrm{Var}\sum_{i=1}^{n}(x_i-\overline{x})\varepsilon_i \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i-\overline{x})^2\right)^2}\sum_{i=1}^{n}(x_i-\overline{x})^2\sigma^2 \\
&= \frac{\sigma^2}{\sum_{i=1}^{n}(x_i-\overline{x})^2}.
\end{aligned}$$

Similarly, we get

$$\mathrm{Var}(\hat{\beta}_0) = \frac{\sigma^2}{n} + \frac{\sigma^2\overline{x}^2}{\sum_{i=1}^{n}(x_i-\overline{x})^2}.$$

[Note that as $n \to \infty$ and $\sum_i (x_i - \overline{x})^2 \to \infty$ we get consistency]. Finally, the two estimators are negatively correlated with

$$\mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) = -\sigma^2\frac{\overline{x}}{\sum_{i=1}(x_i-\overline{x})^2}.$$

Note that if A1 held, i.e., the ε_i are iid $N(0, \sigma^2)$, then $\hat{\beta}_0$ and $\hat{\beta}_1$, as linear combinations

of normal random variables, are normal. In particular, *under A1*,

$$\hat{\beta}_0 \sim N\left(\beta_0, \frac{\sigma^2}{n} + \frac{\sigma^2 \overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2}\right)$$
$$\hat{\beta}_1 \sim N\left(\beta_1, \frac{\sigma^2}{\sum_{i=1}^n (x_i - \overline{x})^2}\right).$$

Additionally, it can be shown that

$$\frac{n-2}{\sigma^2}\hat{\sigma}^2 \sim \chi^2_{n-2}.$$

Confidence Intervals

An aside: If $X \sim N(\mu, \sigma^2)$, note that

$$P\left(-1.96 \leq \frac{X - \mu}{\sigma} \leq 1.96\right) \approx 0.9500042 \approx 95\%$$

so,

$$P\left(\mu - 1.96\sigma \leq X \leq \mu + 1.96\sigma\right) \approx 95\%$$

or,

$$P\left(X - 1.96\sigma \leq \mu \leq X + 1.96\sigma\right) \approx 95\%.$$

The middle statement says that $\mu \pm 1.96\sigma$ contains X 95% of the time. The last statement says that $X \pm 1.96\sigma$ will contain μ 95% of the time, and is the basis for constructing a *confidence interval*. Sometimes this is approximated as $X \pm 2\sigma$.

Let's simplify notation just a bit. Call

$$C = \frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2}$$

so that

$$\hat{\beta}_0 \sim N(\beta_0, \sigma^2 C).$$

Suppose we knew σ^2. We call the following an *exact* 95% *confidence interval for* β_0

$$\left(\hat{\beta}_0 - 1.96\sqrt{\mathrm{Var}\hat{\beta}_0}, \hat{\beta}_0 + 1.96\sqrt{\mathrm{Var}\hat{\beta}_0}\right)$$
$$= \left(\hat{\beta}_0 - 1.96\sigma\sqrt{C}, \hat{\beta}_0 + 1.96\sigma\sqrt{C}\right).$$

This would be the interval that contains β_0 95% of the time, or, in other words, if we repeated the experiment 1000 times, we would expected this interval to contain β_0 950 of those times. We don't know σ^2, however, so we must rely on a plug-in estimator $\hat{\sigma}^2$ when constructing such an interval.

Two options:

Option 1 Use

$$\hat{\beta}_0 \pm 1.96\hat{\sigma}\sqrt{C} \qquad \text{or} \qquad \hat{\beta}_0 \pm 2\hat{\sigma}\sqrt{C}$$

as an *approximate* 95% *confidence interval for* β_0.

Option 2 Be more careful.

Recall that a t random variable with r degrees of freedom can be represented as

$$T = \frac{X}{\sqrt{Y/r}}$$

where $X \sim N(0,1)$ and $Y \sim \chi^2_r$ are independent random variables and where we write $T \sim t_r$. [Picture of t_r distribution compared to normal] Under A1, $\hat{\beta}_0 \sim N(\beta_0, \sigma^2 C)$ is *exact*. Thus,

$$\begin{aligned}\frac{\hat{\beta}_0 - \beta_0}{\hat{\sigma}\sqrt{C}} &=_d \frac{N(0,\sigma^2)}{\hat{\sigma}} =_d \frac{N(0,\sigma^2)}{\sqrt{\hat{\sigma}^2}} =_d \frac{\sigma N(0,1)}{\sqrt{\hat{\sigma}^2}} =_d \frac{N(0,1)}{\sqrt{\frac{\hat{\sigma}^2}{\sigma^2}}} =_d \frac{N(0,1)}{\sqrt{\frac{(n-2)\hat{\sigma}^2}{(n-2)\sigma^2}}} \\ &=_d \frac{N(0,1)}{\sqrt{\chi^2_{n-2}/(n-2)}} =_d t_r\end{aligned}$$

The last step requires a little extra care since it's not obvious that $\hat{\beta}_0$ and $\hat{\sigma}^2$ are independent.

Option 2 An *exact* 95% *confidence interval for* β_0 is

$$\hat{\beta}_0 \pm t_{n-2}(0.025)\hat{\sigma}\sqrt{C}$$

Note that, in general if we wanted a $100(1-\alpha)\%$ confidence interval, we would use

$$\begin{aligned}&\hat{\beta}_0 \pm t_{n-2}(\alpha/2)\hat{\sigma}\sqrt{C} \\ &(\pm t_{n-2}(1-\alpha/2)\hat{\sigma}\sqrt{C})\end{aligned}$$

Table 1: Values of $t_{n-2}(0.975)$ for various n.

$n-2$	10	20	50	100	200
$t_{n-2}(0.975)$	2.23	2.09	2.01	1.98	1.97

where $t_{n-2}(\alpha/2)$ is the lower $\alpha/2\%$ quantile of the t_{n-2} distribution. [e.g., if $\alpha = 0.05$, then $100(1-\alpha)\% = 95\%$] Table 1 gives some typical values of $t_{n-2}(0.975)$. [Note that small n implies we end up using wider confidence intervals, i.e., we exhibit less confidence if we don't approximate by using normal cutoffs]

The α level controls the Type I error – that is, the frequency of which the CI does *not* contain the truth. E.g., $\alpha = 0.05$, then in 95% of experiments our CI will contain the true parameters.

In summary, 95% approximate CIs for our estimated parameters are

$$\hat{\beta}_0 \pm 1.96 SE(\hat{\beta}_0)$$
$$\hat{\beta}_1 \pm 1.96 SE(\hat{\beta}_1)$$

while exact 95% CIs are

$$\hat{\beta}_0 \pm t_{n-2}(0.025) SE(\hat{\beta}_0)$$
$$\hat{\beta}_1 \pm t_{n-2}(0.025) SE(\hat{\beta}_1)$$

where $SE = \sqrt{\widehat{\text{Var}}}$ are standard errors, that is, *estimated* standard deviations. In particular,

$$\widehat{\text{Var}}(\hat{\beta}_1) = \frac{\hat{\sigma}^2}{\sum_{i=1}^n (x_i - \overline{x})^2}$$
$$\widehat{\text{Var}}(\hat{\beta}_0) = \frac{\hat{\sigma}^2}{n} + \frac{\hat{\sigma}^2 \overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2}.$$

The standard error attempts to quantify an answer to the question *how certain are we about the value of the estimator*?

Hypothesis Testing

A hypothesis test usually looks like

$$H_0 : \beta_1 = 0$$
$$H_A : \beta_1 \neq 0$$

where H_0 is called the *null hypothesis* and H_A is the *alternative hypothesis*. That is, our *working hypothesis* is that the slope β_1 is zero, i.e., that X and Y do *not* actually share a (linear) relationship. The goal then, based on noisy data, is to assess whether H_0 is a reasonable hypothesis. The typical way this works is we compute a *test statistic* (in this case known as a *t-statistic*) of the form

$$\frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

and compare its observed value against some threshold. *Under* H_0 (and A1), this test statistic is *exactly* distributed as

$$\frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \sim t_{n-2}.$$

Thus, heuristically, if $\frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$ takes on a very unusual value compared to a t_{n-2} distribution, we might suspect that our null hypothesis is faulty. This is exactly how the test works, and we *reject* H_0 *in favor of* H_A *if*

$$\left| \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \right| > t_{n-2}(1 - \alpha/2).$$

[Draw picture] Note

$$\left| \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \right| > t_{n-2}(1 - \alpha/2) \iff 0 \notin (\hat{\beta}_1 - t_{n-2}(1 - \alpha/2)SE(\hat{\beta}_1), \hat{\beta}_1 + t_{n-2}(1 - \alpha/2)SE(\hat{\beta}_1))$$

so, equivalently, if our confidence interval for β_1 doesn't include 0 we would reject H_0 (in favor of H_A).

Note we could just as easily set up a test of the form

$$H_0 : \beta_1 = 5$$
$$H_A : \beta_1 \neq 5$$

(or β_1^* instead of 5), in which case our test would reject if

$$\left| \frac{\hat{\beta}_1 - 5}{SE(\hat{\beta}_1)} \right| > t_{n-2}(1 - \alpha/2).$$

The identical argument for

$$H_0 : \beta_0 = \beta_0^*$$
$$H_A : \beta_0 \neq \beta_0^*$$

works, in which case we reject if

$$\left| \frac{\hat{\beta}_0 - \beta_0^*}{SE(\hat{\beta}_0)} \right| > t_{n-2}(1 - \alpha/2).$$

Note that to perform a hypothesis test or create a confidence interval we have to choose the α-level (traditionally 5%). An alternative is to report the *p-value*, that is, *the probability of observing something more extreme than our test statistic under* H_0. In these cases this p-value is known as a *two sided p-value*. [Draw picture]

Some warnings:

- Interpret a p-value with care: it is the probability of seeing something as-or-more extreme than our observed test statistic *under* H_0. For example, if we set the α level at 5%, if H_0 is true we would see "significant" p-values 5% of the time.
- If we cannot reject H_0 based on our test statistic at a particular α level, *this is NOT evidence for the null*, this just means there *is insufficient evidence against* H_0.

Hypothesis testing, confidence intervals and p-values all give (almost) the same information

$\hat{\beta}_1 \pm t_{n-2}(1 - \alpha/2)SE(\hat{\beta}_1)$ does not contain zero

$\iff$

$H_0 : \beta_1 = 0$ is rejected at level α

$\iff$

p-value $p < \alpha$.

ANOVA

An analysis of variance (ANOVA) table is often used to compare competing linear models. For example, we might want to compare the two models

$$Y = \beta_0 + \varepsilon \tag{2}$$

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{3}$$

(of course we already know this could boil down to testing $H_0 : \beta_1 = 0$). If (2) were the correct model, then $\hat{\beta}_0 = \overline{y}$ would be the least squares estimate, and the residual sum of squares would be

$$SYY = \sum_{i=1}^{n} (y_i - \hat{\beta}_0)^2 = \sum_{i=1}^{n} (y_i - \overline{y})^2.$$

For the "full" model (3), we have the residual sum of squares as

$$RSS = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

using the OLS estimators for β_0 and β_1. SYY is an estimate of remaining variability under (2) while RSS is estimate of remaining variability under (3).

If $\hat{y}_i$ is the fitted value under (3), then

$$y_i - \overline{y} = (\hat{y}_i - \overline{y}) + (y_i - \hat{y}_i)$$

and it can be shown that

$$SYY = \sum_{i=1}^{n} (y_i - \overline{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2 + \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = SS_{reg} + RSS$$

where we have defined SS_{reg} as the *sum of squares due to regression.* Finally recall that

$$\hat{\sigma}^2 = \frac{RSS}{n-2}.$$

Then the *analysis of variance table* (ANOVA table) for simple regression is shown as Table 2.

A few notes:

Table 2: Analysis of variance table for simple regression.

Source	df	SS	MS	F	p-value
Regression	1	SS_{reg}	$SS_{reg}/1$	$MS_{reg}/\hat{\sigma}^2$	
Residual	$n-2$	RSS	$\hat{\sigma}^2 = RSS/(n-2)$		
Total	$n-1$	SYY			

- If SS_{reg} is large then (3) is a big improvement over (2). To quantify this improvement, we compare comparing mean square regression errors

$$F = \frac{SS_{reg}/1}{RSS/(n-2)} = \frac{MS_{reg}}{\hat{\sigma}^2}$$

  where $MS_{reg} = SS_{reg}/1$ is the mean square error of regression. If (3) is true, we expect $MS_{reg} \gg \hat{\sigma}^2$ and $F \gg 1$.

  Under A1 (or if the sample size is sufficiently large) and if (2) is assumed to be the true model, then F, the *F-statistic* is distributed as a $F(1, n-2)$ random variable, and thus if its p-value is large this is evidence that (3) is a better model.

  [Picture of F-distribution and p-value]

- The df are *degrees of freedom* that correspond to the number of free variables minus the number of estimated parameters (e.g. for RSS it is $n-2$ due to estimation of β_0 and β_1).

- The regular test of $H_0 : \beta_1 = 0$ vs. $H_A : \beta_1 \neq 0$ relies on the t-statistic where

$$t^2 = \left(\frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} \right)^2 = F$$

  after some manipulation of terms. Thus, the F-test in the ANOVA table is *equivalent* to the t-test in this simple case.

We will see ANOVA as a useful tool to compare more complicated mean functions in the next section.

Coefficient of Determination r^2

Define the *coefficient of determination* (r^2 or R^2) as

$$r^2 = 1 - \frac{RSS}{SYY} = \frac{SS_{reg}}{SYY}.$$

Note $r^2 \in [0, 1]$ is one minus the remaining unexplained variability. If RSS/SYY is close to 1 then we didn't improve much using our linear model, whereas if $RSS/SYY \approx 0$ then $r^2 \approx 1$ and we explained *most* of the variability in the data using our model. Indeed, r^2 can be interpreted as the *percentage of explained variability*. For simple regression we have r^2 is the squared *sample correlation* between $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^n (x_i - \overline{x})^2 \sum_{i=1}^n (y_i - \overline{y})^2}} = \frac{SXY}{\sqrt{SXX \cdot SYY}}$$

Prediction

Given a linear model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

there are two quantities we might want to predict with uncertainty at a *new* covariate value $X = x_*$:

- The mean value $\beta_0 + \beta_1 x_*$ (fit)
- A single new observation $y_* = \beta_0 + \beta_1 x_* + \varepsilon_*$ (prediction).

If we knew β_0 and β_1, our uncertainty for the fit would be zero and our uncertainty for the prediction would be σ^2.

[In the first case we want to quantify our uncertainty about the mean function, whereas in the second case we want to include our uncertainty about what a potential residual might be]

The *optimal point predictor* for *both cases* is

$$\hat{\beta}_0 + \hat{\beta}_1 x_*.$$

[Note predictor is linear in the observations!]

The uncertainty (standard error) depends on which case we're looking at. The predictive standard error is then

$$SE(\hat{\beta}_0 + \hat{\beta}_1 x_*) = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_* - \overline{x})^2}{SXX}} \qquad \text{(fit/confidence)}$$
$$SE(\hat{\beta}_0 + \hat{\beta}_1 x_* + \varepsilon_*) = \hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_* - \overline{x})^2}{SXX}} \qquad \text{(prediction)}$$

[Note where the minimum occurs]

95% predictive confidence intervals are then

$$\hat{\beta}_0 + \hat{\beta}_1 x_* \pm t_{n-2}(\alpha/2) SE(\hat{\beta}_0 + \hat{\beta}_1 x_*) \qquad \text{(fit/confidence interval)}$$
$$\hat{\beta}_0 + \hat{\beta}_1 x_* \pm t_{n-2}(\alpha/2) SE(\hat{\beta}_0 + \hat{\beta}_1 x_* + \varepsilon_*) \qquad \text{(prediction interval)}$$

if A1 holds (we could swap in 1.96 for the t-quantile if approximate intervals suffice). [Note the standard error of prediction is *greater* than just $\hat{\sigma}$, since there is uncertainty in estimating β_0 and β_1].

[R example (SimpleLinearRegression.R)]

Maximum Likelihood

If $A1$ holds and $\varepsilon_1, \ldots, \varepsilon_n$ are iid $N(0, \sigma^2)$, then since

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

we have

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

so that it has pdf

$$f(y_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y - (\beta_0 + \beta_1 x_i)}{\sigma}\right)^2}.$$

The joint pdf of $y_1, \ldots, y_n$ is then

$$f(y_1, \ldots, y_n) = \prod_{i=1}^{n} f(y_i) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y - (\beta_0 + \beta_1 x_i))^2\right).$$

by independence. Minimizing the negative log-likelihood $-\log f(y_1, \ldots, y_n)$ is equivalent to maximizing the likelihood, so the *maximum likelihood estimators* for β_0 and β_1 minimize

$$-\log f(y_1, \ldots, y_n) = C + \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y - (\beta_0 + \beta_1 x_i)\right)^2 .$$

In particular, under A1, the OLS estimators are also ML estimators, and enjoy additional theoretical properties. The MLE of σ^2 is different, and is biased, but $\hat{\sigma}^2_{MLE} \to \sigma^2$.

2.3 Diagnostics

Based on our estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we have *estimated residuals*

$$\hat{\varepsilon}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i).$$

If our assumptions are correct, then the estimated residuals $\{\hat{\varepsilon}_i\}_i$ should not have any structure. One important diagnostic plot is the residuals vs. fitted values scatterplot. If there are definite trends in the plot then the assumed linear relationship may be violated. [Picture] If the variability changes with fitted values, then homoskedasticity may be violated; this latter case is when the errors exhibit *heteroskedasticity*. [Picture] A few unusually large residuals may be evidence of *outliers*. Outliers won't (necessarily) change $\hat{\beta}_i$, but they do inflate $\hat{\sigma}^2$. [Picture] The strongest set of assumptions, A1, suggest the residual terms ε_i are normally distributed. To assess normality of the estimated residuals $\{\hat{\varepsilon}_i\}_i$, the *quantile-quantile plot* (Q-Q plot) is often used. The Q-Q plot plots the theoretical quantiles of a standard normal versus the estimated quantiles of the standardized observed residuals. If the plot falls along the identity line, it may be reasonable to assume the errors arose from a normal distribution. Q-Q plots can be used to assess heavy-tailedness and skewness compared to the normal distribution. [Picture of standard and heavy-tailed quantile-quantile plot]

Note that the OLS-estimated regression line is *forced* to go through the pair $(\overline{x}, \overline{y})$, that is,

$$\hat{\beta}_0 + \hat{\beta}_1 \overline{x} = \overline{y}.$$

Thus, you can imagine if a particular covariate x_k were unusually far from $\overline{x}$, it might exert a particularly large *leverage* on the estimates $\hat{\beta}_i$, given this constraint. The *leverage* of the

kth data point (x_k, y_k) is defined by

$$\frac{1}{n} + \frac{(x_k - \overline{x})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}.$$

Note that leverage is a quantification of how far x_k is from the average covariate. High leverage points can have a substantial effect on the ordinary least squares fit, but doesn't necessarily inflate $\hat{\sigma}^2$.

[R example (Diagnostics.R)]

2.4 Pause: Crash Course in Matrix Algebra

A matrix $\mathbf{A}$ with n rows and m columns (i.e., an $n \times m$ matrix) is an element of $\mathbb{R}^n \times \mathbb{R}^m$. For example, a 3×2 matrix has real-valued elements and

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}$$

where, generally a_{ij} will refer to the (i, j)th element, that is, the row i column j element. Sometimes we shorthand this as

$$\mathbf{A} = (a_{ij})_{i=1,j=1}^{3,2} = (a_{ij})$$

A vector is just a matrix with one column, (known as a column vector), e.g.,

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

The *transpose* of a matrix is $\mathbf{A}^{\mathrm{T}} = (a_{ji})$, switching column and row places. E.g.,

$$\mathbf{A} = \begin{pmatrix} 2 & \pi \\ 0 & 10 \\ -3 & 3 \end{pmatrix} \qquad \text{has} \qquad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 2 & 0 & -3 \\ \pi & 10 & 3 \end{pmatrix}.$$

Note that $(\mathbf{A}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{A}$. A matrix is *square* if its number of rows is the number of columns. A square matrix is *symmetric* if $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$, e.g.,

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 3 & 10 \end{pmatrix}$$

The elements $\text{diag}(\mathbf{A}) = (a_{11}, a_{22}, \ldots, a_{nn})$ define the *diagonal* of a matrix. The n-dimensional *identity matrix* $\mathbf{I}$ is the square $n \times n$ matrix with 1s along the diagonal

$$\mathbf{I}_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

An *upper triangular matrix* is a matrix with zero entries below the main diagonal:

$$\begin{pmatrix} 2 & 8 & 3 & 0 \\ 0 & 0 & 10 & 7 \\ 0 & 0 & 4 & 7 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

and a *lower triangular matrix* has zeros above the diagonal. If we wanted to sum all elements of

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix},$$

we can write this as

$$\sum_{i=1}^{3} \sum_{j=1}^{2} a_{ij} = a_{11} + a_{12} + a_{21} + a_{22} + a_{31} + a_{32}$$

Any two $n \times p$ matrices $\mathbf{A}$ and $\mathbf{B}$ may be added by taking elementwise sums,

$$\begin{pmatrix} 2 & \pi \\ 0 & 10 \\ -3 & 3 \end{pmatrix} + \begin{pmatrix} 3 & 1 \\ 4 & 2 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 5 & \pi + 1 \\ 4 & 12 \\ 2 & 3 \end{pmatrix}.$$

Multiplication by a constant $b\mathbf{A} = b(a_{ij}) = (ba_{ij})$ multiplies elementwise. Some properties of matrix addition: (both $\mathbf{A}$ and $\mathbf{B}$ must be $n \times m$):

- $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$
- $(\mathbf{A} + \mathbf{B})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}$
- $(\mathbf{A} - \mathbf{B})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} - \mathbf{B}^{\mathrm{T}}$
- $(\mathbf{x} + \mathbf{y})^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}} + \mathbf{y}^{\mathrm{T}}$

- $(\mathbf{x}-\mathbf{y})^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}} - \mathbf{y}^{\mathrm{T}}$

Matrices $\mathbf{A}(n \times m)$ and $\mathbf{B}(m \times p)$ multiply to form a $n \times p$ matrix $\mathbf{C}$ with

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj},$$

that is, take the sum of elementwise products of the ith row of $\mathbf{A}$ with the jth column of $\mathbf{B}$. E.g.,

$$\begin{pmatrix} 2 & 0 \\ 0 & 10 \\ -3 & 3 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} 6 & 2 \\ 40 & 20 \\ 3 & 3 \end{pmatrix}.$$

Two matrices can multiply if the number of columns of the first equals the number of rows of the second matrix, e.g., $\mathbf{AB}$ makes sense, but then $\mathbf{BA}$ may not conform, and, even if they do,

$$\mathbf{AB} \neq \mathbf{BA}.$$

Some properties of matrix multiplication:

- $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
- $\mathbf{A}(\mathbf{B}-\mathbf{C}) = \mathbf{AB} - \mathbf{AC}$
- $(\mathbf{A}+\mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$
- $(\mathbf{A}-\mathbf{B})\mathbf{C} = \mathbf{AC} - \mathbf{BC}$
- $(\mathbf{A}+\mathbf{B})(\mathbf{C}+\mathbf{D}) = \mathbf{AC} + \mathbf{BC} + \mathbf{AD} + \mathbf{BD}$
- $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

Transposes work as

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}.$$

Note, then, that if $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$,

$$\mathbf{x}^{\mathrm{T}}\mathbf{y} = \sum_{i=1}^{n} x_i y_i \in \mathbb{R}.$$

and

$$\|\mathbf{x}\|_2^2 \equiv \mathbf{x}^{\mathrm{T}}\mathbf{x} = \sum_{i=1}^{n} x_i^2$$

where $\|\cdot\|_2^2$ denotes the squared L_2 length of the vector $\mathbf{x}$. More properties:

- $\sum_{i=1}^{n} \mathbf{a}^{\mathrm{T}}\mathbf{x}_i = \mathbf{a}^{\mathrm{T}} \sum_{i=1}^{n} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{A}\mathbf{x}_i = \mathbf{A} \sum_{i=1}^{n} \mathbf{x}_i$
- $\sum_{i=1}^{n} (\mathbf{A}\mathbf{x}_i)(\mathbf{A}\mathbf{x}_i)^{\mathrm{T}} = \mathbf{A} \left(\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} \right) \mathbf{A}^{\mathrm{T}}$

If $\mathbf{A}$ is square $(n \times n)$ and $\mathbf{x}$ and $\mathbf{y}$ are vectors,

$$\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{y} = \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j a_{ij}$$

is called a *quadratic form*. Note it is a scalar.

The *inverse* of a square matrix $\mathbf{A}$, denoted $\mathbf{A}^{-1}$, is such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I},$$

if it exists. We have

$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1},$$

if all inverses exist.

A square $n \times n$ matrix $\mathbf{A}$ is *nonnegative definite* if, for any vector $\mathbf{a}$,

$$\mathbf{a}^{\mathrm{T}}\mathbf{A}\mathbf{a} \geq 0.$$

Positive definite matrices (replacing $\geq$ with $>$) *always admit an inverse*. Additionally, $\mathbf{A}$ has a "square root" called the *Cholesky factor* or *Cholesky decomposition*

$$\mathbf{A} = \mathbf{T}^{\mathrm{T}}\mathbf{T}$$

where $\mathbf{T}$ is an invertible upper triangular matrix.

The *determinant* of a matrix is a number and is written $\det(\mathbf{A}) = |\mathbf{A}|$, and has the following properties:

- $\det\mathbf{I} = 1$
- $\det(\mathbf{A}^{\mathrm{T}}) = \det\mathbf{A}$
- $\det\mathbf{A}^{-1} = 1/(\det\mathbf{A})$
- $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ for square matrices
- $\det(c\mathbf{A}) = c^n\det\mathbf{A}$ where $\mathbf{A}$ is $n \times n$.
- The determinant of a triangular or diagonal matrix is the product of its diagonal.

The *trace* of a matrix is the sum of its diagonals, $\mathrm{tr}\mathbf{A} = \sum_{i=1}^{n} a_{ii}$. We have

- $\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}\mathbf{A} + \mathrm{tr}\mathbf{B}$
- $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$ (even if $\mathbf{AB} \neq \mathbf{BA}$)

Two vectors $\mathbf{a}$ and $\mathbf{b}$ *orthogonal* if

$$\mathbf{a}^{\mathrm{T}}\mathbf{b} = 0.$$

If $\|\mathbf{a}\|_2 = 1$ then $\mathbf{a}$ is said to be *normalized*. Any vector can be *normalized* by

$$\mathbf{b} = \frac{\mathbf{a}}{\|\mathbf{a}\|_2}.$$

A square matrix $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p)$ is said to be *orthogonal* if its columns and rows are normalized and mutually orthogonal, whence

$$\mathbf{C}^{\mathrm{T}}\mathbf{C} = \mathbf{C}\mathbf{C}^{\mathrm{T}} = \mathbf{I}.$$

Thus, $\mathbf{C}^{-1} = \mathbf{C}^{\mathrm{T}}$. Multiplication by an orthogonal matrix has the effect of *rotating the axes*, that is, if $\mathbf{z} = \mathbf{C}\mathbf{x}$ then note

$$\mathbf{z}^{\mathrm{T}}\mathbf{z} = \mathbf{x}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x} = \mathbf{x}^{\mathrm{T}}\mathbf{x}$$

maintains the same length.

An *eigenvalue* λ of a matrix $\mathbf{A}$ is a number where

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

and $\mathbf{x}$ is called the corresponding *eigenvector*. They can be calculated by finding solutions to $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. Note that, if $\mathbf{x}$ is an eigenvector, and k is a number, then

$$\mathbf{A}(k\mathbf{x}) = k\lambda(\mathbf{x}) = \lambda(k\mathbf{x})$$

so $k\mathbf{x}$ is also an eigenvector, and thus we usually scale eigenvectors to be unit length.

If λ is an eigenvalue of $\mathbf{A}$ and $\mathbf{x}$ is the corresponding eigenvector, then $1 \pm \lambda$ is an eigenvalue of $\mathbf{I} \pm \mathbf{A}$, and $\mathbf{x}$ is still the corresponding eigenvector.

If $\mathbf{A}$ is a square matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$,

- $\mathrm{tr}\mathbf{A} = \sum_{i=1}^{n} \lambda_i$
- $\det\mathbf{A} = \prod_{i=1}^{n} \lambda_i$.

Importantly,

- If $\mathbf{A}$ is positive definite, then its eigenvalues are *all positive*
- If $\mathbf{A}$ is nonnegative definite, then its eigenvalues are either positive or zero.
- If $\mathbf{A}$ is symmetric, its eigenvectors are all mutually orthogonal.

By the last property, if $\mathbf{A}$ has eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, form

$$\mathbf{C} = [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_n]$$

and note $\mathbf{C}$ is orthogonal. Then

$$\begin{aligned}
\mathbf{A} &= \mathbf{AI} \\
&= \mathbf{ACC}^{\mathrm{T}} \\
&= [\mathbf{Ax}_1 \mathbf{Ax}_2 \cdots \mathbf{Ax}_n]\mathbf{C}^{\mathrm{T}} \\
&= [\lambda_1 \mathbf{x}_1 \lambda_2 \mathbf{x}_2 \cdots \lambda_n \mathbf{x}_n]\mathbf{C}^{\mathrm{T}} \\
&= \mathbf{CDC}^{\mathrm{T}}
\end{aligned}$$

where

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

is diagonal with eigenvalues on the diagonal. This is known as the *spectral decomposition* of $\mathbf{A}$. We can also *diagonalize* $\mathbf{A}$ by

$$\mathbf{C}^{\mathrm{T}}\mathbf{A}\mathbf{C} = \mathbf{D}.$$

A *square root matrix* is $\mathbf{A}^{1/2} = \mathbf{C}\mathbf{D}^{1/2}\mathbf{C}^{\mathrm{T}}$, where $\mathbf{D}^{1/2} = \mathrm{diag}(\lambda_1^{1/2}, \ldots, \lambda_n^{1/2})$. The square of $\mathbf{A}$ can be calculated

$$\mathbf{A}^2 = \mathbf{C}\mathbf{D}^2\mathbf{C}^{\mathrm{T}}$$

and the inverse

$$\mathbf{A}^{-1} = \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{\mathrm{T}}.$$

The *rank* of a matrix is the number of linearly independent column vectors (or row vectors). If $\mathbf{A}$ is $n \times p$, then

- $\mathrm{rank}(\mathbf{A}) \leq \min(n, p)$
- $\mathrm{rank}(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = \mathrm{rank}(\mathbf{A}\mathbf{A}^{\mathrm{T}}) = \mathrm{rank}(\mathbf{A})$
- If $\mathbf{A}$ is $n \times n$ and has full rank (n) then $\mathbf{A}$ is invertible.

We will mostly be only dealing with full rank matrices.

Let $\mathbf{A}$ be an $n \times p$ matrix of rank k. The *singular value decomposition* of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{U}$ is $n \times k$, $\mathbf{D}$ is $k \times k$ and $\mathbf{V}$ is $p \times k$. $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$ contains the positive square roots of the $\lambda_1^2, \ldots, \lambda_k^2$ nonzero eigenvalues of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ or $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. The k columns of $\mathbf{U}$ are normalized eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ corresponding to eigenvalues $\lambda_1^2, \ldots, \lambda_k^2$. The k columns of $\mathbf{V}$ are the normalized eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ corresponding to eigenvalues $\lambda_1^2, \ldots, \lambda_k^2$. Then,

$$\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}.$$

If $\mathbf{A}$ is positive definite, then the singular value decomposition is the same as the spectral decomposition.

2.5 Multivariate normal

Recall the definition of covariance and correlation: if X and Y are random variables with mean and standard deviation μ_X, σ_X and μ_Y, σ_Y, respectively, then

$$\mathrm{Cov}(X,Y) := \mathbb{E}\big((X-\mu_X)(Y-\mu_Y)\big)$$

and

$$\mathrm{Cor}(X,Y) := \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \quad (= \rho).$$

In particular, $\mathrm{Cor}(X,Y) \in [-1,1]$ is unitless and is a measure of the *linear dependence* between X and Y, and $\mathrm{Cov}(X,Y)$ is in the units of X times the units of Y. Covariances enjoy the following properties:

- $\mathrm{Var}X = \mathrm{Cov}(X,X)$
- $\mathrm{Cov}(aX+b,Y) = a\mathrm{Cov}(X,Y)$ for any $a,b \in \mathbb{R}$ [prove]
- $\mathrm{Cov}(X,Y) = \mathrm{Cov}(Y,X)$
- $\mathrm{Cov}(X+Y,Z) = \mathrm{Cov}(X,Z) + \mathrm{Cov}(Y,Z)$.

How are covariances defined for vectors of random variables? Suppose $\mathbf{X} = (X_1,\ldots,X_n)'$ and $\mathbf{Y} = (Y_1,\ldots,Y_m)'$ are two random vectors, where $'$ denotes the transpose. Define $\mathrm{Cov}(\mathbf{X},\mathbf{Y})$ to be the $n \times m$ matrix with (i,j)th entry $\mathrm{Cov}(X_i,Y_j)$, that is,

$$\mathrm{Cov}(\mathbf{X},\mathbf{Y}) = \begin{pmatrix} \mathrm{Cov}(X_1,Y_1) & \mathrm{Cov}(X_1,Y_2) & \cdots & \mathrm{Cov}(X_1,Y_m) \\ \mathrm{Cov}(X_2,Y_1) & \mathrm{Cov}(X_2,Y_2) & \cdots & \mathrm{Cov}(X_2,Y_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n,Y_1) & \mathrm{Cov}(X_n,Y_2) & \cdots & \mathrm{Cov}(X_n,Y_m) \end{pmatrix}$$

The following properties also hold for covariances of random vectors:

- $\mathrm{Var}\mathbf{X} = \mathrm{Cov}(\mathbf{X},\mathbf{X})$ [This is how we define the variance of a random vector]
- $\mathrm{Cov}(A\mathbf{X}+\boldsymbol{\mu}, B\mathbf{Y}+\boldsymbol{\nu}) = A\mathrm{Cov}(\mathbf{X},\mathbf{Y})B'$ for any $k \times n$ matrix A, $j \times m$ matrix B, $\boldsymbol{\mu} \in \mathbb{R}^k$ and $\boldsymbol{\nu} \in \mathbb{R}^j$

- $\text{Cov}(\mathbf{X}, \mathbf{Y}) = \text{Cov}(\mathbf{Y}, \mathbf{X})'$.

If $\mathbf{X}$ is a random vector such that $\text{Var}X_i = \sigma_i^2$, then the covariance matrix $\text{Cov}(\mathbf{X}, \mathbf{X}) = (\text{Cov}(X_i, X_j))_{i,j=1}^n$ can be transformed into a *correlation matrix*

$$\text{Cor}(\mathbf{X}, \mathbf{X}) = \left(\frac{\text{Cov}(X_i, X_j)}{\sigma_i \sigma_j} \right)_{i,j=1}^n .$$

The correlation matrix is usually easier to interpret since it consists of all pairwise correlations between the component random variables. Note the diagonal of $\text{Cor}(\mathbf{X}, \mathbf{X})$ is 1s.

Definition 2. *We say the random vector* (X_1, X_2) *has a* bivariate normal *distribution if*

$$f(x_1, x_2) = \frac{1}{\sqrt{(2\pi)^2 \det \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}}} \exp\left(-\frac{1}{2} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}' \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix} \right)$$

for $(x_1, x_2) \in \mathbb{R}^2$.

Here, parameters have the following interpretations,

- $\mathbb{E}X_1 = \mu_1$
- $\mathbb{E}X_1 = \mu_2$
- $\text{Var}X_1 = \sigma_1^2$
- $\text{Var}X_2 = \sigma_2^2$
- $\text{Cov}(X_1, X_2) = \rho\sigma_1\sigma_2$
- $\text{Cor}(X_1, X_2) = \rho$.

Note that the marginals are also normally distributed, e.g., $X_1 \sim N(\mu_1, \sigma_1^2)$. [Draw bivariate density on board, also projected with contours]

Definition 3. *We say the random vector* $\mathbf{X} = (X_1, X_2, \ldots, X_n)'$ *has a* multivariate normal *distribution if*

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right)$$

for $\mathbf{x} \in \mathbb{R}^n$. *We often write* $\mathbf{X} \sim MVN(\boldsymbol{\mu}, \Sigma), \mathbf{X} \sim N_n(\boldsymbol{\mu}, \Sigma)$ *or* $\mathbf{X} \sim N(\boldsymbol{\mu}, \Sigma)$.

Here, $\mathbb{E}\mathbf{X} = \boldsymbol{\mu}$ is the vector of means (i.e., $\mathbb{E}X_i = \mu_i$), and $\text{Cov}(\mathbf{X}, \mathbf{X}) = \Sigma$ is the covariance matrix (sometimes called variance-covariance matrix). The (i, j)th entry of Σ is $\text{Cov}(X_i, X_j)$. [Imagine this as two parameters, $\boldsymbol{\mu}$ tell us the average behavior of the vector, and Σ tells us how all elements relate to each other]

The covariance matrix is *symmetric* since $\Sigma_{ij} = \text{Cov}(X_i, X_j) = \text{Cov}(X_j, X_i) = \Sigma_{ji}$. The diagonal of Σ contains the variances of each component. Additionally, we have

- Each X_i is marginally normally distributed: $X_i \sim N(\mu_i, \sigma_i^2)$
- If A is a $k \times n$ real matrix and $\mathbf{b} \in \mathbb{R}^k$, then $A\mathbf{X} + \mathbf{b} \sim MVN(A\boldsymbol{\mu} + \mathbf{b}, A\Sigma A')$
- If Σ is a covariance matrix then it is nonnegative definite
- If Σ is nonnegative definite, then it is a covariance matrix.

[R example (MultivariateNormal.R)]

2.6 Multiple Regression

[R example (IntroMultipleRegression.R)]

Sometimes it makes sense to regress Y on multiple covariates $X_1, X_2, \ldots, X_p$. In this case the *multiple linear regression* model assumes

$$Y = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p + \varepsilon$$

with the only difference here being the introduction of extra covariates. If we have n observations, $y_1, \ldots, y_n$, where the ith has p corresponding covariates $x_{i1}, \ldots, x_{ip}$, then

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots \beta_p x_{ip} + \varepsilon_i$$

where we might make any of the analogous assumptions A1-A4 for $\{\varepsilon_i\}$. [Write example model for temperature on elevation and latitude on board, e.g., since we expect temperature to change with elev but also latitude, and neither can explain the other]

Note the interpretation of β_i *is different* here: for a unit increase in X_i, β_i is the average increase in Y *with all other covariates held fixed.*

The model for $y_1, \ldots, y_n$ can be written in matrix notation. Define

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix},$$

Note $\boldsymbol{\beta}$ is $(p+1) \times 1$ and $\mathbf{X}$ is $n \times (p+1)$. The first column of 1s in $\mathbf{X}$ is for the β_0 term. We can then write the model for all observations $\mathbf{y}$ succinctly as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Note under A1-A3,

$$\mathbb{E}\boldsymbol{\varepsilon} = \mathbf{0} \quad \text{and} \quad \text{Var}\boldsymbol{\varepsilon} = \text{Cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}_n.$$

Sometimes it's convenient to write

$$\mathbf{x}_i = \begin{pmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix}, \quad \text{so that} \quad y_i = \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \vdots \\ \mathbf{x}_n' \end{pmatrix}.$$

OLS

The OLS estimator for $\boldsymbol{\beta}$ is found by minimizing

$$\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}))^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),$$

which results in

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

[Check dimensions on board] Note this is *not* the best way to compute $\hat{\boldsymbol{\beta}}$; in practice most programs use a QR decomposition that is more robust against rounding errors.

The OLS estimators are *unbiased*:

$$\begin{aligned} \mathbb{E}\hat{\boldsymbol{\beta}} &= \mathbb{E}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbb{E}\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \quad \text{[why?]} \\ &= \boldsymbol{\beta}. \end{aligned}$$

Moreover,

$$\begin{aligned}
\mathrm{Var}\hat{\boldsymbol{\beta}} &= \mathrm{Var}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{Var}(\mathbf{Y},\mathbf{Y})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \quad \text{[why?]} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.
\end{aligned}$$

[note this is a variance-covariance matrix]

Standard errors for the OLS estimator $\hat{\boldsymbol{\beta}}_{OLS}$ are found by taking the square root of the diagonal of

$$\widehat{\mathrm{Var}}(\hat{\boldsymbol{\beta}}_{OLS}) = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$$

where the residual variance is estimated by

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{n - (p+1)}.$$

This is an unbiased estimator for σ^2 (but is *not* the MLE). We can also write the residual sum of squares in a few formats:

$$\begin{aligned}
RSS &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\
&= \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'(\mathbf{X}'\mathbf{X})\hat{\boldsymbol{\beta}} \\
&= \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y}.
\end{aligned}$$

As previously, under A1,

$$\frac{n - (p+1)}{\sigma^2}\hat{\sigma}^2 \sim \chi^2_{n-(p+1)}.$$

Hypothesis Testing and Confidence Intervals

The most basic hypothesis test we would want to check is

$$\begin{aligned}
H_0 &: \beta_1 = \cdots = \beta_p = 0 \\
H_A &: \text{At least one } \beta_j \neq 0,
\end{aligned}$$

that is, are *any* of the covariates useful as linear predictors? The test statistic is the *F-statistic*,

$$F = \frac{(TSS - RSS)/p}{RSS/(n-(p+1))}$$

where

$$SYY = TSS = \sum_{i=1}^{n}(y_i - \overline{y})^2.$$

Under H_0,

$$\mathbb{E}(RSS/(n-(p+1))) = \sigma^2$$

and

$$\mathbb{E}((TSS - RSS)/p) = \sigma^2,$$

so we would expect $F \approx 1$. Thus, our test would reject H_0 if F were too large. In fact, under H_0,

$$F \sim F_{p,n-(p+1)},$$

so we would reject in favor of H_A if $F > F_{p,n-(p+1)}(1-\alpha)$ (this is a one-sided test). Equivalently, we can calculate the p-value of F,

$$1 - \text{quantile}_{F_{p,n-(p+1)}}(F),$$

and just report this. [picture of how the quantile is calculated]

A test for an *individual* parameter β_j

$$H_0 : \beta_j = 0$$
$$H_A : \beta_j \neq 0$$

follows from the test statistic

$$\frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \sim t_{n-(p+1)}$$

(and from which individual confidence intervals can be derived).

Prediction

Suppose we want to predict Y_* for new set of covariates $\mathbf{x}_* = (1, x_{*1}, x_{*2}, \ldots, x_{*p})'$. There are two we might want to predict with uncertainty at $\mathbf{x}_*$:

- The mean value $\mathbf{x}_*'\boldsymbol{\beta}$ (fit)
- A single new observation $y_* = \mathbf{x}_*'\boldsymbol{\beta} + \varepsilon_*$ (prediction).

[In the first case we want to quantify our uncertainty about the mean function, whereas in the second case we want to include our uncertainty about what the residual will be]

The natural point predictor for *both cases* is

$$\hat{y}_* = \mathbf{x}_*'\hat{\boldsymbol{\beta}} = \mathbf{x}_*'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y}.$$

[Note predictor is linear in the observations $\mathbf{y}$!]

Our uncertainty depends on which case we're looking at. Recall we define $SE(\cdot) = \widehat{\mathrm{Var}}(\cdot)$. Note

$$\mathrm{Var}(\mathbf{x}^{\mathrm{T}}\hat{\boldsymbol{\beta}}) = \mathbf{x}^{\mathrm{T}}\mathrm{Var}(\hat{\boldsymbol{\beta}})\mathbf{x}.$$

The standard error of prediction is then

$$\begin{aligned} \widehat{\mathrm{Var}}(\mathbf{x}_*'\hat{\boldsymbol{\beta}}) &= \hat{\sigma}^2\mathbf{x}_*'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_* && \text{(fit/confidence)} \\ \widehat{\mathrm{Var}}(\mathbf{x}_*'\hat{\boldsymbol{\beta}} + \varepsilon_*) &= \hat{\sigma}^2(1 + \mathbf{x}_*'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_*) && \text{(prediction)} \end{aligned}$$

and 95% predictive confidence intervals are then

$$\begin{aligned} &\hat{y}_* \pm t_{n-(p+1)}(\alpha/2)SE(\mathbf{x}_*'\hat{\boldsymbol{\beta}}) && \text{(fit/confidence)} \\ &\hat{y}_* \pm t_{n-(p+1)}(\alpha/2)SE(\mathbf{x}_*'\hat{\boldsymbol{\beta}} + \varepsilon_*) && \text{(prediction)} \end{aligned}$$

under A1. [Note the standard error of prediction is *greater* than just $\hat{\sigma}$, since there is uncertainty in estimating $\boldsymbol{\beta}$ – follows since $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is nonnegative definite]. If the residuals $\boldsymbol{\varepsilon}$ are correlated, then we can improve the point and interval estimates by taking account of this extra structure, stay tuned.

Maximum Likelihood

Under A1, the OLS estimators $\hat{\boldsymbol{\beta}}$ are the same as the MLEs, since

$$-\log f(y_1, \ldots, y_n) = -\log f(\mathbf{y}) = C + \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),$$

and minimizing the negative log-likelihood is equivalent to performing OLS.

Diagnostics

Similar diagnostics can be used for simple and multiple regression. For instance, r^2 still is a measure of *fraction of variance explained*, and values close to 1 indicate a good model fit. Beware, though, that R^2 will *always* increase as more covariates are added, whether they are useful or not. Root mean squared error (RMSE or RSE) is sometimes a useful summary for model comparisons

$$RMSE = \sqrt{\frac{RSS}{n-(p+1)}} \quad (= \hat{\sigma}),$$

and is in the units of the response Y. Note that the dependence on p implies that RMSE *can* increase with the addition of covariates.

[R example (MultipleRegression.R)]

2.7 Model Selection (Variable Selection)

A fundamental problem in multiple regression is in choosing a *set of relevant* covariates. This problem is known as *variable selection*. To start, if we reject

$$H_0 : \beta_1 = \cdots = \beta_p = 0$$

then we know *at least* one covariate is useful, but this test doesn't indicate which one.

Why Choose Variables?

Why not just throw all covariates into the dataset? If

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

but we have available covariates $X_1, \ldots, X_p, X_{p+1}, \ldots, X_{p+k}$ and use *all* of them, then

- At risk of *overfitting* (that is, "finding" trends that do not exist)
- $\hat{\boldsymbol{\beta}}$ is still unbiased
- $\text{Var}(\hat{\boldsymbol{\beta}})$ is inflated.

What if we don't use enough covariates? If $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$ contains the true set of covariates for a dataset, but we only use $\mathbf{X}_1$, then

$$\begin{aligned}
\mathbb{E}\hat{\boldsymbol{\beta}} &= \mathbb{E}(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{Y} \\
&= (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbb{E}\mathbf{Y} \\
&= (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2) \\
&= \boldsymbol{\beta}_1 + (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2\boldsymbol{\beta}_2 \\
&\neq \boldsymbol{\beta}_1.
\end{aligned}$$

So,

- $\hat{\beta}$ is *biased*.
- At risk of *underfitting*.

The Wrong Way

What about the proposed approach: for $i = 1, \ldots, p$, perform the hypothesis test

$$\begin{aligned}
H_0 &: \beta_i = 0 \\
H_A &: \beta_i > 0
\end{aligned}$$

and keep all X_i where the test resulted in a rejection?

Pause: suppose we know our test statistic T has a cdf F. If t is the observed statistic, then the p-value is

$$\begin{aligned}
P(T \geq t) &= 1 - P(T < t) \\
&= 1 - F(t) \\
&\sim 1 - U(0,1) \\
&\sim U(0,1)
\end{aligned}$$

[R example (PValues.R)]

Assessing Quality of Model Fit

Given two competing models (e.g., one with one without a particular covariate), how do we decide which is better? That is, how do we quantify the *goodness-of-fit*? Recall that

$$RSS = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}))^2$$

and

$$r^2 = 1 - \frac{RSS}{SYY}$$

We could use the model that has a better (lower RSS or higher r^2) value, except that these will *always* decrease/increase respectively with the addition of new covariates. Thus, we need an approach that *rewards* model fit while *penalizing* complexity.

If we entertain a model with d covariates $X_1, \ldots, X_d$ fit to n observations, then we define *Mallow's* C_p as

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2).$$

The second term typically increases as d increases, and we favor models with the smallest C_p value. [Note the book gives a different version of C_p which some people use in practice].

If a model is fit by maximum likelihood, then *Akaike's information criterion* (AIC) or the *Bayesian information criterion* (BIC) may be used. [Recall that OLS is equivalent to ML if errors are normal]. Up to a constant, AIC is

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

so, for least squares models, C_p and AIC are proportional (i.e., they agree on the best model). Up to a constant, BIC is

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2).$$

Note that since $\log n > 2$ for any $n > 7$, BIC tends to place more penalty on models with many variables d. Under BIC, the best model is that with the smallest BIC value.

Finally, the *adjusted* r^2 statistic is defined by

$$\text{Adjusted } r^2 = 1 - \frac{RSS/(n-(d+1))}{TSS/(n-1)}$$

The best model is that that *maximizes* the adjusted r^2. Intuition: once all relevant variables are in the model, RSS is relatively stable, and increasing d should be penalized (note that if RSS constant then the adjusted r^2 is decreasing in d).

Variable Selection Methods

Warning: Our model selection technique may depend on our objective. We may care about:

- *Prediction accuracy*: in which case we don't (really) care about which variables are used, so long as we can predict new Ys with good skill.
- *Model interpretability*: Including numerous variables with lots of interactions usually results in a model that is *difficult to interpret*.

Ideally, we would entertain *all* possible models. To do this we would fit the model

$$Y = \beta_0 + \varepsilon$$

and the models

$$Y = \beta_0 + \beta_1 X_i + \varepsilon$$

for $i = 1, \ldots, p$ and the models

$$Y = \beta_0 + \beta_1 X_i + \beta_2 X_j + \varepsilon$$

for all $i \neq j = 1, \ldots, p$, etc., up to the full model

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \varepsilon.$$

For each of these competing models, we calculate any goodness-of-fit criterion (AIC, BIC, C_p, adjusted r^2), and choose the single model with the best value (e.g., lowest BIC). Although appealing, note that for p covariates there are 2^p possible models, so this method is often computationally prohibited. This method is called *best subset selection*.

The *forward stepwise selection* algorithm works as follows:

- Fit the *null model* $\mathcal{M}_0$, that is, the model with no covariates

- For $k = 1, \ldots, p$
  - Consider all models that add one covariate to $\mathcal{M}_{k-1}$
  - Pick the "best" of these and call it $\mathcal{M}_k$
- Pick the "best" model amongst $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$.

Here, picking the "best" model amounts to choosing one of the criterions AIC, BIC, C_p, adjusted r^2 to quantify quality. Forward selection tends to computationally *much* easier than best subset selection, as it is a *guided* search over model space.

Alternatively, we could start with all covariates and then remove them in a stepwise fashion. This is known as *backward elimination* or *backward stepwise selection*. The algorithm is:

- Fit the *full model* $\mathcal{M}_p$ with all covariates
- For $k = p, \ldots, 1$
  - Consider all models that delete one covariate from $\mathcal{M}_k$
  - Pick the "best" model and call it $\mathcal{M}_{k-1}$
- Pick the "best" model amongst $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$.

Neither forward or backward selection is guaranteed to find the best model that best subset selection does, and indeed these techniques often disagree in practice. Also note that backward elimination will *not* work for *high-dimensional problems*, $p > n$. One could also consider a *hybrid* algorithm that adds and removes variables simultaneously, since sometimes adding a variable makes another's coefficient insignificant.

[R example (VariableSelection.R)]

2.8 Potential Issues

In multiple regression there are numerous potential issues to consider.

Categorical (Qualitative) Predictors

So far we've dealt with *quantitative* predictors, e.g., $X \in \mathcal{D} \subseteq \mathbb{R}$. *Categorical* or *qualitative* covariates only take on *finitely* many values (also sometimes called *factors*).

For example, if x_i is gender, then it has only two *levels*. We need to choose a convention to code it numerically using an indicator or *dummy variable* that takes on two possible values, e.g.,

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

What happens to a simple model in this case?

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \varepsilon_i & \text{if } i\text{th person is male} \end{cases}$$

Now β_0 is *the average outcome for males*, while $\beta_0 + \beta_1$ is the average outcome for females. Another interpretation is that β_1 is the average difference between females and males. The coefficient can be interpreted as a *shift in the mean* for $x_i = 1$. If we coded this differently, say

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ -1 & \text{if } i\text{th person is male} \end{cases}$$

then

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{th person is female} \\ \beta_0 - \beta_1 + \varepsilon_i & \text{if } i\text{th person is male} \end{cases}$$

Now β_0 is the *overall average* (ignoring gender), and β_1 is the amount females are above (and males are below) the average. Note that predictions will always be the same, regardless of the coding of x_i, but the *interpretation* will change.

If there are more than two levels, then we use additional dummy variables. For example, if Y is life expectancy measured in each country of the world, and countries are grouped into *Africa*, *OECD* and *Other* [OECD is the Organization for Economic Cooperation and Development, an international think tank charged with promoting policies that will improve global social and economic well-being] then to regress on group we might set

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th country is in OECD} \\ 0 & \text{if } i\text{th country is not in OECD} \end{cases}$$

and

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th country is Other} \\ 0 & \text{if } i\text{th country is not Other} \end{cases}$$

Then

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i = \begin{cases} \beta_0 + \varepsilon_i & \text{if } i\text{th country is in Africa} \\ \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{th country is in OECD} \\ \beta_0 + \beta_2 + \varepsilon_i & \text{if } i\text{th country is neither in Africa or OECD} \end{cases}$$

Thus, β_0 is the average life expectancy in Africa, β_1 is the additional expectancy for a person in an OECD country and β_2 is the additional expectancy for a person in a different (non-African or OECD) country. For instance, we found

$$\hat{\beta}_0 = 59.8 \qquad \hat{\beta}_1 = 22.7 \qquad \hat{\beta}_2 = 15.6$$

so that life expectancy in Africa is 59.8, life expectancy in OECD countries is $59.8 + 22.7 = 82.5$ and everywhere else is $59.8 + 15.6 = 75.4$ years. In this case, β_0 is known as the *baseline*. Switching coding will keep predictions the same, but will change interpretations of coefficients.

Bad idea: set

$$x_i = \begin{cases} 0 & \text{if } i\text{th country is Africa} \\ 1 & \text{if } i\text{th country is OECD (switch to Other)} \\ 0 & \text{if } i\text{th country is Other (switch to OECD)} \end{cases}$$

[Picture of how linear relationship turns sour if switch Other/OECD]

Beyond Additivity and Linearity

Two restrictive assumptions we have made so far are *additivity* and *linearity*. The additive assumption implies the effect of one predictor on Y is independent of the values of the other predictors. The linearity assumption implies that the change in Y for a one-unit change in X does *not* depend on the value of X.

The usual generalization to remove the additive assumption is to consider *interactions*. A simple model with an interaction is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon.$$

For instance, if Y is salary, X_1 is years since degree and X_2 is 1 for male (0 for female), then β_1 is the average raise per year for females, while $\beta_1 + \beta_3$ is the average raise per year for males, and β_0 is the average starting salary for females while $\beta_0 + \beta_2$ is the average starting salary for males. The *hierarchical principle* states that if an interaction is to be included, then the *main effects* of X_1 and X_2 alone should also be included, even if their coefficients are not significantly different than zero.

[R example (Salary.R)]

To overcome linearity, *polynomial regression* is usually the first step. For example, to generalize

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

we could consider

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

which fits a *quadratic* function to the data, but note the model is still a *linear model* in that it is *linear in the coefficients*. This is equivalent to using two predictors $X_1 = X$ and $X_2 = X^2$. We will spend a lot of time later on more methods for nonlinear modeling. Variable selection techniques can still be used for each

[R example (LifeExpectancy.R)]

Collinearity

Collinearity refers to when two (or more) covariates are closely related to one another. If two covariates are nearly linearly related, then it can be difficult to tease out the effect of one versus the other. Collinearity results in a less stable $(\mathbf{X}'\mathbf{X})^{-1}$ and causes standard errors to grow. How to detect collinearity:

- Pairwise scatterplot of covariates
- Correlation matrix of covariates

It is possible for three or more covariates to be linearly related, in which case we can examine a *variance inflation factor*. In particular,

$$VIF(\hat{\beta}_j) = \frac{1}{1 - r^2_{X_j|X_{-j}}}$$

where $r^2_{X_j|X_{-j}}$ is the r^2 from regressing X_j on the remaining predictors. Rule of thumb: VIFs greater than $5 - 10$ are problematic.

To overcome collinearity:

- Use only one covariate from the set of collinear ones
- Orthogonalize:

  $$\mathbf{x}_2^* = \mathbf{x}_2 - \mathbf{x}_1(\mathbf{x}_1'\mathbf{x}_1)^{-1}\mathbf{x}_1'\mathbf{x}_2$$

  where now

  $$\mathbf{x}_1'\mathbf{x}_2^* = 0.$$

  (Problem: $\mathbf{x}_2^*$ may be difficult to interpret)
- Combine via PCA or other methods

[R example (MultipleRegression2.R)]

Transformations

Recall assumptions A1:

1. $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$
2. Var$\varepsilon_i = \sigma^2$ for all i.
3. ε_i are normally distributed
4. ε_i are independent (uncorrelated)

If these are not met, then we can either transform the response, predictors or both. Transformation of Y or X can help ensure 1, whereas only transformations of Y can help ensure 2-3. Data transformation cannot help meeting assumption 4.

We've seen polynomial regression as a possible way to transform covariates, and will focus on response transformations. If $Y > 0$ is a strictly positive then a log transformation $\log Y$ is common (and often helps ensure assumption 2). Other common possibilities are square-root or reciprocal ($\sqrt{Y}$ or $1/Y$). Note that if we use

$$\log Y = \beta_0 + \beta_1 X + \varepsilon$$

then

$$Y = e^{\beta_0} e^{\beta_1 X} e^{\varepsilon},$$

or, in other words, the errors are *multiplicative* on the original scale.

Another common family of transformation is the *Box-Cox* family,

$$g_\lambda(Y) = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log Y & \lambda = 0. \end{cases}$$

where we would then model, e.g.,

$$g_\lambda(Y) = \beta_0 + \beta_1 X + \varepsilon.$$

The best value of λ is usually chosen by maximizing the profile-likelihood.

[R example (Transformations.R)]

Diagnostics Revisited

A fundamental quantity in regression (and beyond) is the *hat matrix*,

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'.$$

First note

$$\mathbf{H}^2 = \mathbf{H}\mathbf{H} = \mathbf{H}$$

so $\mathbf{H}$ is idempotent. The *fitted values* are

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y},$$

so the hat matrix converts observations into fitted values. The estimated residuals are

$$\hat{\varepsilon}_i = y_i - \hat{y}_i$$

which we can write as

$$\begin{aligned}\hat{\boldsymbol{\varepsilon}} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - \mathbf{H}\mathbf{y} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{y}.\end{aligned}$$

Thus,

$$\begin{aligned}\mathrm{Var}\hat{\boldsymbol{\varepsilon}} &= \mathrm{Cov}(\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\varepsilon}}) \\ &= (\mathbf{I} - \mathbf{H})\mathrm{Cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})(\mathbf{I} - \mathbf{H})' \\ &= \sigma^2(\mathbf{I} - \mathbf{H}).\end{aligned}$$

[We need to use $\mathbf{H}^2 = \mathbf{H}$ for this] Thus, the standard error for $\hat{\varepsilon}_i$ is

$$SE(\hat{\varepsilon}_i) = \hat{\sigma}\sqrt{1 - H_{ii}}$$

where H_{ii} is the ith diagonal entry of $\mathbf{H}$. Under A1, the *studentized residuals*

$$\hat{\varepsilon}_i^* = \frac{\hat{\varepsilon}_i - 0}{SE(\hat{\varepsilon}_i)} \approx N(0, 1).$$

This gives us a method to look for *outliers*, where we might suspect a particular y_i is an outlier if $\hat{\varepsilon}_i^*$ is smaller than -2 or larger than 2.

Now note

$$\hat{y}_i = H_{i1}y_1 + \cdots + H_{ii}y_i + \cdots + H_{in}y_n,$$

so H_{ii} is the influence of the observation y_i on its own fitted value. We define H_{ii} to be the *leverage* of y_i, a measure of its *potential* for being influential on the final fit, but note that H_{ii} only depends on $\mathbf{X}$, and not the actual value of y_i. To measure the the actual influence of a data point y_i, we use *Cook's distance (Cook's D)* as

$$D_i = \frac{H_{ii}}{p(1 - H_{ii})}(\hat{\varepsilon}_i^*)^2.$$

High values of leverage imply *potential influence* on the model fit while high values of Cook's D imply *actual influence* on the model fit.

[R example (Diagnostics2.R)]

Degrees of Freedom

If we fit the model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

we say that we have used $(p+1)$ *degrees of freedom* due to the estimation of the $p+1$ parameters. However, in the future it will be more difficult to define an analogous quantity. If $\mathbf{H}$ is the hat matrix, then note

$$\begin{aligned}\sum_{i=1}^{n} H_{ii} &= \operatorname{tr}(\mathbf{H}) \\ &= \operatorname{tr}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \\ &= \operatorname{tr}(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) \\ &= \operatorname{tr}(\mathbf{I}_p) = p.\end{aligned}$$

Recalling that H_{ii} is the leverage of the ith data point, then $p = \sum_{i=1}^{n} H_{ii}$ is the total influence of all observations, so the greater the number of parameters the greater the aggregate influence of the observations. We will see later that semiparametric regression methods also fall into a class of predictors like $\hat{\mathbf{y}} = \mathbf{M}\mathbf{y}$, where we can define $\operatorname{tr}(\mathbf{M})$ as the *effective degrees of freedom*.

3 Classification

Classification refers to the case where the outcome of interest, Y, is *categorical* or *qualitative*, rather than quantitative. For instance, we might want to predict whether a

- New e-mail is spam or not
- Person will default on home mortgage
- Person is likely to have heart disease as they age
- Person will vote for Bernie Sanders

based on a set of possibly useful covariates. Classification can be for either *binary* outcomes (spam or not) or *multivariate/multinomial* outcomes (mutations in a stretch of DNA are associated with different phenotypes). Common methods are

- Logistic regression (and probit regression)
- Discriminant analysis
- K-nearest neighbors
- Support vector machines

Note that if $Y \in \{0, 1\}$ we don't want to use regression since typically $\mathbf{x}'\hat{\boldsymbol{\beta}} \neq 0$ or 1. An alternative is to model the probabilities $P(Y = 0)$ and $P(Y = 1)$. If we can model this probability, then the *classifier* (also known as the *classification rule*) is just a rule that assigns some probabilities to 1 and others to 0 (e.g., predict $Y = 1$ if $P(Y = 1) > 0.5$).

3.1 Logistic Regression

Let $p(X) = P(Y = 1|X)$, where X is a covariate. Why wouldn't we attempt a model like

$$p(X) = \beta_0 + \beta_1 X?$$

Instead, logistic regression is based on the *logistic function*

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

[Picture of $e^x/(1+e^x)$]

Note that $p(X) \in (0, 1)$ for all X. There are other functions that transform a regression to $(0, 1)$, such as the *probit model*

$$p(X) = \Phi(\beta_0 + \beta_1 X)$$

where Φ is the cdf of a $N(0, 1)$.

Inverting the logistic function yields the *logit transform* of the *odds ratio* is

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

In other words, for every unit increase in X, the *log odds-ratio* increases by β_1. The odds $p/(1-p)$ take on values in $[0, \infty)$, so, for instance, if $p = 0.5$, then the odds are even, whereas if $p = 0.2$, the odds are $1/4$. Note that the *sign* of β_1 tells us the direction of influence of X on $P(Y = 1|X)$ – $\beta_1 > 0$ implies probability grows with X and $\beta_1 < 0$ implies probabilities is inversely related to X.

In this setup, $Y \sim \text{Bernoulli}(p(X))$. If we assume $Y_1, \ldots, Y_n$ are independent, then given training data $(x_1, y_1), \ldots, (x_n, y_n)$, the likelihood function for $\mathbf{y}$ is

$$f(\mathbf{y}) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}.$$

Estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are found by maximizing $f(\mathbf{y})$, resulting in MLEs. Closed form solutions for $\hat{\beta}_0$ and $\hat{\beta}_1$ don't exist, but they can be approximated numerically.

Sanity check: suppose no covariate, so $p(x) = p = \exp(\beta_0)/(1 + \exp(\beta_0))$. Then

$$f(\mathbf{y}) = \prod_{i=1}^{n} p^{y_i}(1 - p)^{1-y_i} = p^{n\overline{y}}(1 - p)^{n-n\overline{y}}.$$

The derivative of $\log f(\mathbf{y})$ with respect to p is

$$\frac{\mathrm{d}}{\mathrm{d}p} \log f(\mathbf{y}) = \frac{n\overline{y}}{p} - \frac{n - n\overline{y}}{1 - p} = 0$$

implies

$$\hat{p} = \overline{y}.$$

In the absence of covariates, logistic regression just uses the proportion of positives as the estimate of p.

Multiple logistic regression follows analogously,

$$p(X_1, \ldots, X_p) = \frac{e^{\beta_0 + \sum_{i=1}^p \beta_i X_i}}{1 + e^{\beta_0 + \sum_{i=1}^p \beta_i X_i}},$$

or equivalently

$$\log\left(\frac{p(\mathbf{X})}{1 - p(\mathbf{X})}\right) = \beta_0 + \sum_{i=1}^p \beta_i X_i.$$

Similar concepts from linear regression are appropriate for logistic regression as well, e.g., the z-statistic is based off of $\hat{\beta}_1 / SE(\hat{\beta}_1)$, and can be used to generate p-values or test $H_0 : \beta_1 = 0$.

For a set of new covariates $\mathbf{x} = (x_1, \ldots, x_p)'$, our prediction for $p(\mathbf{x})$ is

$$\hat{p}(\mathbf{x}) = \frac{e^{\hat{\beta}_0 + \sum_{i=1}^p \hat{\beta}_i x_i}}{1 + e^{\hat{\beta}_0 + \sum_{i=1}^p \hat{\beta}_i x_i}}.$$

[but this gives a probability, not a decision to forecast as 0/1]

The usual classification rule is

$$\hat{y} = \begin{cases} 1 & \hat{p}(\mathbf{x}) > 0.5 \\ 0 & \hat{p}(\mathbf{x}) < 0.5 \end{cases}$$

(with randomization for $\hat{p}(\mathbf{x}) = 0.5$). Note, then

$$\frac{1}{2} < \hat{p} \iff \frac{1}{2} < \frac{e^x}{1 + e^x} \iff \frac{1}{2} < \frac{1}{2} e^x \iff 0 < x.$$

Thus, this decision rule is equivalent to

$$\hat{y} = \begin{cases} 1 & \hat{\beta}_0 + \mathbf{x}' \hat{\boldsymbol{\beta}} > 0 \\ 0 & \hat{\beta}_0 + \mathbf{x}' \hat{\boldsymbol{\beta}} < 0. \end{cases}$$

This is also known as the *Bayes classifier*. The line $\hat{\beta}_0 + \mathbf{x}' \hat{\boldsymbol{\beta}} = 0$ is the *decision boundary*, and note that it is *linear* in the covariates. Thus, for prediction/classification, we don't need to calculate probabilities, only the sign of the regression line.

Assessing Quality of Model Fit

Given a set of predictions, $\hat{y}_1, \ldots, \hat{y}_n$, how do we assess quality of fit? We can calculate the *error rate*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[y_i \neq \hat{y}_i]} = \text{percent misspecified}$$

and look at a *confusion matrix*

	True # of 1s	True # of 0s
Predicted # of 1s	10	4
Predicted # of 0s	2	12

From here, we can read off the

- *Sensitivity*: percent of true positives (10/12)
- *Specificity*: percent of true negatives (12/16)
- *Positive predictive value*: % of 1s correctly identified (10/14)
- *Negative predictive value*: % of 0s correctly identified (12/14)
- *Error rate*: $(4 + 2)/28$.

We need specificity, e.g., since if we only cared about predicting positives, we would always use 1 as the classifier, regardless of covariate. Additionally define

- *False positive rate (FPR)* = 4/16 = 1 - specificity
- *True positive rate (TPR)* = 10/12 = sensitivity

Goal: try to *minimize* FPR and simultaneously *maximize* TPR.

These should be compared against random guessing. For example: if the data had 100 positives and 1000 negatives, what might the confusion matrix look like for guessing 1 with 90% probability every time? [Go to table] Thus, we would get $TPR = 0.9$ and $FPR = 0.9$. To improve on this, for *any fixed FPR*, we want to have *higher TPR*.

	True # of 1s	True # of 0s
Predicted # of 1s	90	900
Predicted # of 0s	10	100

A *ROC (receiver operating characteristic) graph*, plots FPR (1-specificity) on the x-axis and TPR (sensitivity) on the y-axis. The point at $(0, 1)$ indicates perfect predictions, and the diagonal line indicates random guessing (thus curves closer to the upper left corner indicate better models).

[Picture]

For any given model and decision rule, we get a single *point* in ROC space. But what if we took our classification threshold ($p(X) > 0.5$ implies classify as 1), and tried to vary the threshold 0.5? For each different threshold we consider, we would get a *different* confusion matrix. By considering *many* thresholds from $[0, 1]$ we can create a *ROC curve*, which can be compared against the identity line, and also against competing classification procedures.

Thought experiment: What would the TPR and FPR be for a threshold of 0 (that is, always predict positive)? What would the TPR/FDR be for a threshold of 1 (always predictive negative)? What would the best point on a ROC plot be?

[Upper right-hand corner classifiers are more liberal – they make positive classifications with weak evidence, but their FPR is high. Lower left-hand corner classifiers are conservative – fewer positive classifications leads to fewer false positives]

A numerical summary of *area under ROC (AUC)* is often used, and is compared to 0.5, which is the expected AUC under chance guessing.

In linear regression, RSS is used as a measure of model fit. The analogous quantity for logistic regression is the *deviance*, denoted G^2 where

$$G^2 = 2\sum_{i=1}^{n}\left[y_i \log\left(\frac{y_i}{\hat{y}_i}\right) + (1-y_i)\log\left(\frac{1-y_i}{1-\hat{y}_i}\right)\right]$$

where $\hat{y}_i = \hat{p}(\mathbf{x})$. G^2 can be used to test for differences between models by comparing against an appropriate χ^2 distribution, with smaller values indicating better fit. We define the *deviance residual* as

$$dev_i = \pm\left(-2\big[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\big]\right)^{1/2}$$

where $\pm$ is positive if $y_i \geq \hat{y}_i$. dev_i is the (signed square root) contribution of the ith data point to the total deviance.

Without the decision rule, logistic regression predictions are *probabilities*. Require some way of *quantifying quality of a probabilistic forecast*. These are called *scoring rules*. Given a set of forecast probabilities $f_1, \ldots, f_n$ and observations $y_1, \ldots, y_n \in \{0, 1\}$, the *Brier score* is

$$BS = \frac{1}{n} \sum_{i=1}^{n} (f_i - y_i)^2.$$

Lower Brier scores indicate better performance. [Note where minimum occurs]

For classification problems, it is *crucial* to split the data into *training* and *testing* data. It turns out that the *in-sample* error (that is, the error on the data used to train the model) is almost always better than the *out-of-sample* error (testing data).

[R example (LogisticRegression.R)]

3.2 Discriminant Analysis

Now suppose $Y \in \{1, \ldots, K\}$ falls into one of K *classes*. Discriminant analysis takes a slightly different approach by assuming a distribution on the predictor $[X|Y]$ and inverting via Bayes' formula to predict $[Y|X]$. Some notation:

- $\pi_k = P(Y = k)$ is the *prior probability* that Y falls in the kth class
- $f_k(x) = P(X = x|Y = k)$ is the conditional density function for X given that Y is in class k

Then we have

$$\begin{aligned} p_k(x) = P(Y = k|X = x) &= \frac{P(X = x|Y = k)P(Y = k)}{P(X = x)} \\ &= \frac{f_k(x)\pi_k}{\sum_{i=1}^{K} f_i(x)\pi_i} \end{aligned}$$

defines the *posterior probability* that Y will be in the kth class given $X = x$. For a fixed x, the denominator is constant across $p_k(x)$, so finding the *most probable class* is equivalent to finding which of $\pi_1 f_1(x), \ldots, \pi_K f_K(x)$ is greatest.

Linear Discriminant Analysis for $p = 1$

Linear discriminant analysis (LDA) *assumes* $[X|Y = k]$ are approximately normally distributed with all conditionals having the same variance, so

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)}.$$

Classification of $[Y|X = x]$ boils down to finding the maximal (over k)

$$\begin{aligned}
\log(\pi_k f_k(x)) &= C_1 + \log \pi_k - \frac{1}{2\sigma^2}(x^2 - 2x\mu_k + \mu_k^2) \\
&= C_1 + \log \pi_k + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + C_2(x) \\
&= \log \pi_k + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \text{constants that don't depend on } k.
\end{aligned}$$

Thus, finding the k for which the *discriminant function*

$$\delta_k(x) = \log \pi_k + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2}$$

is maximized is the classification rule for linear discriminant analysis (note that it is linear in x, hence the name). In the special case of two classes ($K = 2$) and $\pi_1 = \pi_2$, we assign $[Y|X = x]$ to class 1 if

$$\begin{gathered}
\log \pi_1 + x\frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} > \log \pi_2 + x\frac{\mu_2}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} \\
\Longleftrightarrow \\
x\mu_1 - \mu_1^2 > x\mu_2 - \mu_2^2 \\
2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2,
\end{gathered}$$

and to class 2 otherwise. The *decision boundary* is

$$2X(\mu_1 - \mu_2) = \mu_1^2 - \mu_2^2 \iff X = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}.$$

[Picture]

Estimation of μ_k, π_k and σ^2 follow by

- $\hat{\pi}_k = \frac{n_k}{n}$

- $\hat{\mu}_k = \frac{1}{n_k} \sum_{i|y_i=k} x_i$
- $\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i|y_k=k} (x_i - \hat{\mu}_k)^2$

where n_k is the total number of observations in class k and $n = \sum_{i=1}^{K} n_k$ is the total number of observations.

Linear Discriminant Analysis for $p > 1$

If there are $\mathbf{X} = (X_1, \ldots, X_p)'$ covariates per observation, then the extension of LDA is that

$$[\mathbf{X}|Y = k] \sim N_p(\boldsymbol{\mu}_k, \Sigma)$$

where $\boldsymbol{\mu}_k = (\mathbb{E}X_1, \ldots, \mathbb{E}X_p)'$ and $\Sigma = \{\mathrm{Cov}(X_i, X_j)\}_{i,j=1}^{p}$. The Bayes classifier in this case for an observation $\mathbf{X} = \mathbf{x}$ is the class k for which

$$\delta_k(x) = \mathbf{x}\Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k'\Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

is maximized. Note it is still linear in $\mathbf{x}$. Estimation is analogous to $p = 1$, for observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, let

- $\hat{\pi}_k = \frac{n_k}{n}$
- $\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i|y_i=k} \mathbf{x}_i$
- $\hat{\Sigma} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i|y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)'$.

Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) relaxes the assumption that Σ is constant across classes $k = 1, \ldots, K$. That is, in QDA we assume $[\mathbf{X}|Y = k] \sim N_p(\boldsymbol{\mu}_k, \Sigma_k)$. The Bayes classifier for $\mathbf{X} = \mathbf{x}$ is the class k that maximizes

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\mathbf{x}'\Sigma_k^{-1}\mathbf{x} + \mathbf{x}'\Sigma_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k'\Sigma_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\Sigma_k| + \log \pi_k.$$

Note $\delta_k(\mathbf{x})$ is *quadratic* in $\mathbf{x}$.

QDA seems more general than LDA, when would we opt for one or the other?

LDA: Large p or n not much bigger than p

QDA: Small p or $n \gg p$.

If we have lots of covariates, then QDA will attempt to estimate p covariance matrices Σ_k where the number of parameters scales like $p(p+1)/2$ – we will incur much variability for lower bias. In these cases, LDA will reduce variability at the cost of some bias, which may be desirable.

Warning: LDA and QDA are *not* appropriate for categorical covariates. [Thought experiment: can you think of a generalization to allow for categorical covariates?]

[R example (DiscriminantAnalysis.R)]

3.3 K-nearest Neighbors

K-nearest neighbors (KNN) is a simple, nonparametric method for classification which estimates probabilities $P(Y = j|X = x)$ by empirically averaging nearby observations to x. In particular, the KNN estimate is

$$\hat{P}(Y = j|X = x) = \frac{1}{K} \sum_{i \in \mathcal{N}_x} \mathbb{1}_{[y_i = j]}$$

where K is some pre-specified integer and $\mathcal{N}_x$ is the set of K nearest training points to x.

[Picture]

Note that if $K = 1$ or K is small, the decision boundary will typically be rather erratic (high variance, but low bias), whereas if K is large then there will be reduced variance (but high bias).

It is particularly important to split the data into *training* and *testing* sets. In general a chosen method will perform better on a training dataset than the testing data. For instance, if we use the whole dataset to train, then the estimated error rate for KNN with $K = 1$ will be zero, but will surely suffer with the introduction of new points.

[R example (KNN.R)]

4 Classification 2

In this section we'll introduce the *support vector machine*, which generalizes the *support vector classifier* which generalizes the *maximal margin classifier*.

4.1 Maximal Margin Classifier

In p-dimensional space, a *hyperplane* is a flat subspace of dimension $p-1$ – so in 2 dimensions a hyperplane is a line. In 3 dimensions, a hyperplane is just a plane.

Definition 4. *In p-dimensions, a* hyperplane *is all* $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ *satisfying*

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0. \tag{4}$$

Note in $p = 2$ this reduces to the equation for a line

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0.$$

[Picture]

If $\mathbf{x}$ does not satisfy (4), then necessarily

$$\begin{aligned} \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p > 0 \quad \text{or} \\ \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p < 0, \end{aligned}$$

so the hyperplane divides p-dimensional space into halves.

Suppose we have a set of training data $y_1, \ldots, y_n$ and corresponding covariates $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where

$$y_i \in \{-1, 1\} \quad \mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$$

where -1 is one class and 1 is the other.

The goal of a *separating hyperplane* is to find a hyperplane (defined by $\beta_0, \ldots, \beta_p$) such that

$$\begin{aligned} \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} > 0 \qquad \text{if} \quad y_i = 1 \\ \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} < 0 \qquad \text{if} \quad y_i = -1 \end{aligned}$$

This is equivalent to

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) > 0$$

for all $i = 1, \ldots, n$.

[Picture]

Given a test covariate $\mathbf{x}_* = (x_{*1}, \ldots, x_{*p})'$, the classifier depends on

$$f(\mathbf{x}_*) = \beta_0 + \beta_1 x_{*1} + \cdots + \beta_p x_{*p}$$

$$\hat{y} = \begin{cases} 1 & f(\mathbf{x}_*) > 0 \\ -1 & f(\mathbf{x}_*) < 0 \end{cases}$$

with randomization on the boundary. The magnitude $f(\mathbf{x}_*)$ can be interpreted as a measure of confidence in the decision rule. Note this results in a linear decision boundary.

Problem: there are *many* possible separating hyperplanes, and the choice of any two valid ones will lead to different classification rules.

[Picture]

Separable Case

Suppose the data are *linearly separable* [picture]. The *maximum margin hyperplane* (*optimal separating hyperplane*) is that which is furthest from the training data. This hyperplane defines the *maximal margin classifier* as being above or below the line.

The maximum margin hyperplane is calculated via

maximize$_{\beta_0, \beta_1, \ldots, \beta_p} M$

subject to $\sum_{j=1}^{p} \beta_j^2 = 1$

and $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq M$ for all $i = 1, \ldots, n$.

Note that restricting $\boldsymbol{\beta}$ to be length one is not restrictive since

$$\beta_0 + \boldsymbol{\beta}'\mathbf{x} = 0$$

and

$$k(\beta_0 + \boldsymbol{\beta}'\mathbf{x}) = 0$$

define equivalent hyperplanes for any $k \neq 0$. Additionally,

$$y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i)$$

is the distance from the ith observation to the hyperplane, so M is the *margin of the hyperplane*, i.e., the least distance from the data to the hyperplane. To see this, note that $\boldsymbol{\beta}$ is the vector that is normal to the hyperplane. Any vector pointing in the direction of the hyperplane is defined by the difference between two points on the plane – if $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_2$ are on the hyperplane then

$$\boldsymbol{\beta}'(\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2) = \boldsymbol{\beta}'\boldsymbol{\ell}_1 - \boldsymbol{\beta}'\boldsymbol{\ell}_2 = -\beta_0 + \beta_0 = 0.$$

Due to our algorithm, also note $\|\boldsymbol{\beta}\| = 1$, so $\boldsymbol{\beta}$ is a unit vector. Thus the vector from a candidate point $\mathbf{x}$ to the line can be written

$$\mathbf{x} - \boldsymbol{\ell} = d\boldsymbol{\beta}$$

where d is the signed distance from $\mathbf{x}_i$ to the line. Multiplying by $\boldsymbol{\beta}$ gives

$$\boldsymbol{\beta}'\mathbf{x}_i - \boldsymbol{\beta}'\boldsymbol{\ell} = d\boldsymbol{\beta}'\boldsymbol{\beta}$$
$$(\boldsymbol{\beta}'\mathbf{x}_i + \beta_0) = d.$$

Thus, multiplying by y_i gives the unsigned distance.

Defining the Support Vectors

The vectors that are of length M from the hyperplane are known as the *support vectors*, and *only these vectors define the hyperplane* (that is, all other data points are not needed to define the hyperplane).

[Picture]

Recall the original optimization problem

$$\text{maximize}_{\beta_0,\boldsymbol{\beta}} M$$

subject to $\|\boldsymbol{\beta}\|_2 = 1$

and $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq M$ for all $i = 1, \ldots, n$.

We can remove the $\|\boldsymbol{\beta}\|_2 = 1$ criterion by replacing the conditions with

$$\frac{1}{\|\boldsymbol{\beta}\|_2} y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq M$$

(which redefines β_0). This is the same as

$$y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq M\|\boldsymbol{\beta}\|_2$$

and if $\beta_0, \boldsymbol{\beta}$ satisfies this equation then so does $k\beta_0, k\boldsymbol{\beta}$ where $k > 0$, so setting $M = 1/\|\boldsymbol{\beta}\|_2$, we have

$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2$

subject to $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq 1$.

This can be converted to a Lagrangian (using some advanced optimization theory involving the Karush-Kuhn-Tucker (KKT) conditions),

$$\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 - \sum_{i=1}^{n} \alpha_i (y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) - 1)$$

which we want to minimize over $\beta_0, \boldsymbol{\beta}$. Take derivatives wrt β and β_0:

- $\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$
- $\sum_{i=1}^{n} \alpha_i y_i = 0$.

Substituting back in to the Lagrangian results in the Wolfe dual objective function

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i' \mathbf{x}_j$$

which we want to maximize subject to $\alpha_i \geq 0$ (it gives a lower bound on the objective for any feasible point). At the solution, either

- $\alpha_i = 0$ which happens if $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) > 1$ or
- $\alpha_i > 0$ which happens if $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) = 1$.

If $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) = 1$, then $\mathbf{x}_i$ is *on* the boundary of the classification margin.

It turns out the classification function is

$$f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}'\mathbf{x} = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i'\mathbf{x}$$

and the classifications are just

$$\hat{y} = \begin{cases} 1 & f(\mathbf{x}_*) > 0 \\ - & f(\mathbf{x}_*) < 0. \end{cases}$$

Thus, the points for which $\alpha_i > 0$ are the *support vectors* while the points $(\mathbf{x}_i, y_i)$ for which $\alpha_i = 0$ are irrelevant for classification. This will become important for the generalization to support vector machines.

Nonseparable Case

[Picture]

In the *nonseparable case*, a separating hyperplane does not exist. We seek a hyperplane that *almost* separates the data, using a *soft margin*.

4.2 Support Vector Classifier

Maximal margin classifiers can be very sensitive to the addition of a single new data point.

[Picture (Fig 9.5 ISLR)]

We want a method that is more robust against individual observations and can successfully classify *most* of the training data.

The *support vector classifier* (SVC) or *soft margin classifier* allows some of the data to fall on the wrong side of the classifier (soft implies that some data violate the classification rule). The optimization problem resulting in the SVC is

$$\text{maximize}_{\beta_0,\beta_1,\ldots,\beta_p,\varepsilon_1,\ldots,\varepsilon_n} M$$

subject to $\sum_{j=1}^{p} \beta_j^2 = 1$

and $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) \geq M(1 - \varepsilon_i)$ for all $i = 1, \ldots, n$

and $\sum_{i=1}^{n} \varepsilon_i \leq C$ where $\varepsilon_i \geq 0$ for all $i = 1, \ldots, n$.

Note that if $\varepsilon_1 = \cdots = \varepsilon_n = 0$ this reduces to the maximal margin classifier (which doesn't exist in the nonseparable case).

As before, M is the width of the margin. Now, $\varepsilon_1, \ldots, \varepsilon_n$ are *slack variables*. If $\varepsilon_i = 0$ then $(\mathbf{x}_i, y_i)$ is correctly classified and is on the correct side of the margin. If $0 < \varepsilon_i < 1$ then the ith data point is correctly classified, but is on the wrong side of the margin. If $\varepsilon_i > 1$ then the ith point is misclassified. The constant C is a *tuning parameter* that controls the total amount of slack allowed. If C is close to zero, then our SVC will be close to the maximal margin classifier, whereas if $C \gg 0$ we tolerate many violations of the boundary and will also lead to large margins. C is typically chosen by cross-validation, and controls the bias-variance tradeoff (small C implies large bias, large C implies large variance).

[Picture]

Defining the Support Vectors

The optimization program can be rewritten in Lagrangian form

$$\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \alpha_i(y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i) - (1 - \varepsilon_i)) - \sum_{i=1}^{n} \mu_i\varepsilon_i$$

which we minimize wrt $\beta_0, \boldsymbol{\beta}$ and $\varepsilon_1, \ldots, \varepsilon_n$. Setting derivatives equal to zero we get

$$\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i = C - \mu_i, i = 1, \ldots, n.$$

Its now clear that the solution has the form

$$f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}'\mathbf{x} = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i'\mathbf{x},$$

and is a linear classifier. The $(\mathbf{x}_i, y_i)$ such that $\alpha_i \neq 0$ are the *support vectors*. That only a few observations define the decision rule implies that the SVM is robust against outlying observations (whereas, e.g., LDA relies on the mean of *all* observations within a class and thus is non-robust).

[R example (SupportVectorClassifier.R)]

4.3 Support Vector Machines

Both the maximal margin classifier and the support vector classifier result in linear decision boundaries. A *support vector machine* (SVM) relaxes this and allows for nonlinear decision boundaries.

[Picture (ala Figure 9.8 ISLR)]

Transformations Revisited

In linear regression we saw that considering transformations of the covariates could lead to superior model fits. Rather than using

$$x_1, \ldots, x_p$$

we could use

$$\mathbf{x} = (x_1, \ldots, x_p, x_1^2, \ldots, x_p^2)'.$$

Then a potential support vector classifier would be calculated via

$\min_{\beta_0, \beta_1, \ldots, \beta_{2p}} M$

subject to $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{x}) \geq M(1 - \varepsilon_i)$

and $\sum_{i=1}^n \varepsilon_i \leq C$, $\|\boldsymbol{\beta}\|_2^2 = 1$.

Or we could include interactions $x_i x_j$, etc... In any case, the classifier can be written

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \alpha_i \mathbf{x}_i'\mathbf{x} \tag{5}$$

which depends on $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *only through* the inner product

$$\langle \mathbf{x}_i, \mathbf{x} \rangle = \mathbf{x}_i'\mathbf{x}.$$

The Support Vector Machine

The generalization of the SVC requires the use of a *positive definite kernel*.

Definition 5. *We call a function* $k(\mathbf{x}, \mathbf{y}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *a* positive definite function *if, for any* $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ *and* $a_1, \ldots, a_n \in \mathbb{R}$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

The *support vector machine* replaces (5) with

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}).$$

As previously, the *support vectors* are those for which $\alpha_i \neq 0$.

There are many options for kernels, some of the most popular are:

- Polynomial: $k(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}'\mathbf{x}_i)^d$
- Radial: $k(\mathbf{x}, \mathbf{x}_i) = e^{-a\|\mathbf{x}-\mathbf{x}_i\|^2}$
- Neural network: $k(\mathbf{x}, \mathbf{x}_i) = \tanh(a\mathbf{x}'\mathbf{x}_i + b)$.

The decision rule is still

$$\hat{y} = \text{sgn}\left(\beta_0 + \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})\right).$$

[R example (SupportVectorMachine.R)]

Which Classifier Should I Choose?

Depends on your goal:

- Inference: logistic or probit regression, discriminant analysis
- Prediction: discriminant analysis, KNN, SVM

SVMs With More Than Two Classes

It is difficult to extend the methodology of SVMs to more than two classes. Suppose there are $K > 2$ classes; the two most popular approaches are:

- *One vs. One Classification*: build $\binom{K}{2}$ SVMs for comparing all pairwise classes, and, to predict for a new set of covariates, tally the class outcomes for each of the $\binom{K}{2}$ SVMs and favor the one with the highest tally count.
- *One vs. All Classification*: fit K different SVMs for comparing the kth class against everything else (where everything else is coded the same, e.g., -1), $k = 1, \ldots, K$. Call the estimated parameters $\beta_{0k}, \beta_{1k}, \ldots, \beta_{pk}$ for the kth class. Then to predict for a new covariate $\mathbf{x}^*$, use the class for which $\beta_{0k} + \beta_{1k}x_1^* + \cdots + \beta_{pk}x_p^*$ is the greatest.

Support Vector Machines via Penalization

Consider minimizing the following

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} (1 - y_i f(\mathbf{x}_i))_+ + \lambda \|\boldsymbol{\beta}\|_2^2$$

where

$$(x)_+ = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

is known as the *hinge loss* function. This is equivalent to minimizing

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} \varepsilon_i + \lambda \|\boldsymbol{\beta}\|_2^2$$

subject to $y_i f(\mathbf{x}_i) \geq 1 - \varepsilon_i$ and $\varepsilon_i \geq 0$ for $i = 1, \ldots, n$

where $f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i k(\mathbf{x}_i, \mathbf{x})$. Since this problem results in the SVM solution, we see that the SVM falls into the form

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} L(y_i, \mathbf{x}_i, \beta_0, \boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})$$

where L is a *loss function* and P is a *penalty*.

It turns out that logistic regression has a loss function that looks like

$$L(y_i, \mathbf{x}_i, \beta_0, \boldsymbol{\beta}) = \frac{1}{\log 2} \log(1 + \exp(-y_i f(\mathbf{x}_i))),$$

which is closely approximated by the hinge loss. While the SVM coefficients will often be zeroed, logistic regression coefficients will almost always be positive.

[Picture]

5 Regularization

Recall the basic multiple linear regression problem. We model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon = \mathbf{x}'\boldsymbol{\beta} + \varepsilon.$$

If we have many predictors, $p \gg 0$, then our usual OLS estimator

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

will exhibit high variances and will be unstable. Worse, if we have $p > n$ predictors, then the OLS estimators are unidentifiable, since there are infinitely many solutions to

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})\boldsymbol{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}.$$

Rather than resorting to a subset selection method, the goal of this section is to include *all* variables, but to *penalize* or *regularize* their coefficients so as to encourage them to be closer to zero. We will see that this provides a reduction in the *effective degrees of freedom*.

Given samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ where $\mathbf{x} = (1, x_1, \ldots, x_p)$, the usual OLS estimator minimizes

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Regularization follows by adding a *penalty* term to this,

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \to (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})$$

where $P(\boldsymbol{\beta})$ is a penalty that increases as $\|\boldsymbol{\beta}\| \to \infty$, and λ is known as a *complexity parameter*, *smoothing parameter* or *shrinkage parameter* that controls the amount of weight put on the penalty. Thus, to minimize this over $\boldsymbol{\beta}$, we must *balance* between model fit and "model size."

5.1 Ridge Regression

The parameter β_0 only measure the average value of Y, and should not be penalized.

Note the following useful fact when using *centered* covariates:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\beta_0}\sum_{i=1}^{n}(y_i-\beta_0-\beta_1(x_i-\overline{x}))^2 &= -2\sum_i(y_i-\beta_0)+2\beta_1\sum_i(x_i-\overline{x})\\ &= -2n\overline{y}+2n\beta_0\end{aligned}$$

which implies the OLS estimator for β_0 is $\hat{\beta}_0=\overline{y}$.

The basic idea behind *ridge regression* is to penalize the *squared length* of the coefficients $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_p)^{\mathrm{T}}$. That is, using a penalty of the form $P(\boldsymbol{\beta})=\sum_{i=1}^n\beta_i^2=\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}=\|\boldsymbol{\beta}\|_2^2$.

Problem: what if $p=2$ and X_1 is budget of movie and X_2 is rating. Then $X_1\sim$ 10000s and $X_2\sim$ 1s. What units is $\|\boldsymbol{\beta}\|_2^2$ in? Does the length of the vector make sense in this case? We need to equalize the *sizes* of the β_is!

Thus, for the remainder of this section we will assume

- Observations y_i have been centered by $\overline{y}$, that is, $y_i\to y_i-\overline{y}$.
- Covariates have been centered and scaled, that is,

$$x_i\to\frac{x_i-\overline{x}}{\sqrt{\frac{1}{n}\sum_{i=1}^n(x_i-\overline{x})^2}}$$

Thus, Y is mean zero (at $(X_1,\ldots,X_p)=(0,\ldots,0)$) and X_i are *unitless*. Additionally, (with a bit of abuse of notation)

$$\begin{aligned}\sum_{i=1}^n x_i^2 &= \sum_{i=1}^n\left(\frac{x_i-\overline{x}}{\sqrt{\frac{1}{n}\sum_{i=1}^n(x_i-\overline{x})^2}}\right)^2\\ &= \frac{\sum_{i=1}^n(x_i-\overline{x})^2}{\frac{1}{n}\sum_{i=1}^n(x_i-\overline{x})^2}\\ &= n.\end{aligned}$$

The *ridge regression* solution minimizes

$$\begin{aligned}\sum_{i=1}^n(y_i-\sum_{j=1}^p x_{ij}\beta_j)^2+\lambda\sum_{i=1}^j\beta_j^2 &= \sum_{i=1}^n(y_i-\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)^2+\lambda\boldsymbol{\beta}'\boldsymbol{\beta}\\ &= (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\|\boldsymbol{\beta}\|_2^2.\end{aligned}$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ and $\mathbf{X}$ is the usual design matrix but *without the first column of* $1s$.

Example 6. *Suppose* $p = 1$. *Then*

$$\min_{\beta_1} \sum_{i=1}^{n} (y_i - \beta_1 x_i)^2$$

is

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{\sum_{i=1}^n x_i y_i}{n}.$$

by Homework 1. The ridge solution would set equal to zero (by taking a derivative wrt β_1*)*

$$-2\sum_{i=1}^{n} x_i(y_i - \beta_1 x_i) + 2\lambda\beta_1 = -2\sum_{i=1}^{n} x_i y_i + \beta_1(n + \lambda)$$

implying

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i}{n + \lambda}.$$

Thus, λ shrinks β_1 *toward zero!*

If $\lambda = 0$, then ridge regression reduces to OLS. If $\lambda \to \infty$, then the minimization pushes $\boldsymbol{\beta} \to \mathbf{0}$, and the model reduces to $Y = \beta_0 + \varepsilon$.

Let's derive the ridge solution:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} &= -2\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} \\ &= 2\left((\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} - \mathbf{X}^{\mathrm{T}}\mathbf{y}\right)\end{aligned}$$

which implies

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}.$$

If $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$ is not of rank p, it has some zero eigenvalues and is not invertible. The effect of $+\lambda\mathbf{I}$ is to bump these zero eigenvalues to λ, and $\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}$ is invertible. Thus, ridge regression allows us to fit models with $p > n$ covariates.

What is ridge regression doing? Note that $\hat{\boldsymbol{\beta}}_{ridge}$ is biased,

$$\mathbb{E}\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} \neq \boldsymbol{\beta}.$$

On the other hand,

$$\text{Var}\hat{\boldsymbol{\beta}} = \sigma^2(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1},$$

which should be compared against $\text{Var}\hat{\boldsymbol{\beta}}_{OLS} = \sigma^2(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$. Thus ridge regression is *sacrificing some bias to reduce the variability* in the estimates.

[XXX: R example (RidgeRegression.R)]

5.2 Lasso

One drawback to ridge regression is that the estimated β_is almost always take on positive values, even if negligibly small. To overcome this, the *lasso* (least absolute shrinkage and selection operator) is an alternative that minimizes

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\sum_{i=1}^{p}|\beta_i|$$

where the $\|\boldsymbol{\beta}\|_2^2$ has been replaced by the L_1 norm.

Lasso has the same effect of *shrinking* the coefficients closer to zero, but, amazingly, the lasso solution tends to threshold some covariates to *exactly* zero, and thus also acts as a variable selection tool. In other words, the lasso generates *sparse* models, that is, models with only a subset of variables.

5.3 Another View of Ridge and Lasso: Lasso as a Variable Selector

Note an equivalent way to write the ridge problem is as

$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)^2$

subject to $\sum_{j=1}^{p}\beta_j^2 \leq s$.

while lasso is equivalent to

$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)^2$

subject to $\sum_{j=1}^{p} |\beta_j| \leq s$.

We are minimizing the squared error (as in OLS), *subject to* living in either a L_2 (ridge) or L_1 (lasso) sphere of radius s. Note that if $s = 0$, our model reduces to $Y = \beta_0 + \varepsilon$, while as $s \to \infty$, we return the OLS estimators.

[Picture: try to find contours closest to OLS estimator that intersect the L_1/L_2 spheres, Figure 6.7 ISLR]

[XXX: R example (Lasso.R)]

5.4 A Bayesian View of Ridge and Lasso

In Bayesian statistics, we represent initial beliefs about a parameter β using a *prior* distribution and, given data, update these beliefs by formulating a *posterior* distribution.

If $\mathbf{y}$ are observations from $Y = \beta x + \varepsilon$ under A1, then $\mathbf{y}$'s pdf is

$$f(\mathbf{y}|\beta) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}(\mathbf{y}-\beta\mathbf{x})^{\mathrm{T}}(\mathbf{y}-\beta\mathbf{x})}.$$

Note that $f(\mathbf{y})$ is bad notation, since it depends on the *unknown* β. Then, given a *prior distribution* $\pi(\beta)$, the goal is to find the *posterior distribution*,

$$f(\beta|\mathbf{y}) = \frac{f(\mathbf{y}|\beta)\pi(\beta)}{f(\mathbf{y})}.$$

$\pi(\beta)$ represents our beliefs about β before any data is available, while $f(\beta|\mathbf{y})$ represents our *updated beliefs about* β *given the observation* $\mathbf{y}$.

[Picture]

The β with *highest probability* given the data y, is the *posterior mode*, i.e., it is the value of β that maximizes $f(\beta|\mathbf{y})$. Maximizing $f(\beta|\mathbf{y})$ is equivalent to maximizing $\log f(\beta|\mathbf{y})$ is equivalent to minimizing $-\log f(\beta|\mathbf{y})$. Thus, the posterior mode *minimizes*

$$\begin{aligned} -\log(f(\beta|\mathbf{y})) &= -\log(f(\mathbf{y}|\beta)) - \log(\pi(\beta)) + C \\ &= \frac{1}{2\sigma^2}(\mathbf{y} - \beta\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \beta\mathbf{x}) - \log(\pi(\beta)) + C \end{aligned}$$

where C does not depend on $\boldsymbol{\beta}$. The ridge solution is the minimizer of

$$(\mathbf{y} - \beta\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \beta\mathbf{x}) + \lambda\beta^2,$$

so set

$$-\log(\pi(\beta)) = \lambda\beta^2$$

and note

$$\pi(\beta) \propto e^{-\lambda\beta^2}.$$

In other words, if we use a normal, mean zero prior for β then *the ridge solution is the same as the posterior mode.*

The lasso solution is the minimizer of

$$(\mathbf{y} - \beta\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \beta\mathbf{x}) + \lambda|\beta|,$$

so set

$$-\log(\pi(\beta)) = \lambda|\beta|$$

and note

$$\pi(\beta) \propto e^{-\lambda|\beta|}.$$

In other words, if we use a double exponential prior (also known as a Laplace distribution) for β then *the lasso solution is the same as the posterior mode.*

6 Nonparametric Regression: Splines, Smoothing and Kernels

When the covariate X and response Y are *not* linearly related, we previously saw that we could try polynomial regression

$$Y = \beta_0 + \beta_1 X + \cdots + \beta_p X^p + \varepsilon$$

to capture nonlinear relationships. Some problems are:

- Often need p large to achieve reasonable model fits
- Extrapolated values behave extremely badly (fit is good locally, but not globally)

The basic idea of this chapter is to consider models of the form

$$Y = \sum_{j=0}^{p} \beta_j K_j(X) + \varepsilon$$

for some choices of functions $K_1, \ldots, K_p$. If

$$K_i(x) = X^i$$

this just reduces to polynomial regression. Choices include

- Step functions (breaking up the domain into chunks and fitting separate models within chunks)
- Regression and penalized splines (forcing some natural behavior at the boundary between regions)
- Smoothing splines (adding in a functional penalty)
- Local regression (locally varying linear models)
- Generalized additive models (extending these methods for multiple predictors)

6.1 Regression Splines

The *piecewise constant* model uses a step function,

$$K(x) = \mathbb{1}_{[a,b]}(x) = \begin{cases} 1 & x \in [a,b] \\ 0 & x \notin [a,b]. \end{cases}$$

If the data are defined on $X \in (0,1)$, define a set of N *knots* $0 < a_1 < a_2 < \cdots < a_N \leq 1$ with N corresponding functions

$$\begin{aligned} K_1(x) &= \mathbb{1}_{[a_1,a_2)}(x) \\ K_2(x) &= \mathbb{1}_{[a_2,a_3)}(x) \\ &\vdots \\ K_{N-1}(x) &= \mathbb{1}_{[a_{N-1},a_N)}(x) \\ K_N(x) &= \mathbb{1}_{[a_N,\infty)}(x). \end{aligned}$$

The corresponding model is

$$Y = f(X) + \varepsilon = \beta_0 + \sum_{j=1}^{N} \beta_j K_j(X) + \varepsilon.$$

Note that $f(X)$ *only* takes on values $\beta_0, \beta_1, \ldots, \beta_N$ since the K_i's have disjoint supports.

If data $(y_1, x_1), \ldots, (y_n, x_n)$ are available, then we can use OLS technology with the design matrix

$$\mathbf{X} = \begin{pmatrix} 1 & K_1(x_1) & K_2(x_1) & \cdots & K_N(x_1) \\ 1 & K_1(x_2) & K_2(x_2) & \cdots & K_N(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & K_1(x_n) & K_2(x_n) & \cdots & K_N(x_n) \end{pmatrix},$$

to estimate $\beta_0, \ldots, \beta_N$ (and we get all of the usual properties of OLS estimators, confidence intervals, predictions, hypothesis testing, etc.). This results in a *piecewise constant* fit, but there are (always) breaks in $\hat{f}(x)$ at the knot boundaries a_i.

A *piecewise linear* function is *linear* between knots. For example, if $N = 1$ and $a_1 = 1/2$ with

$$\begin{aligned} K_1(x) &= x\mathbb{1}_{(-\infty,1/2)}(x) \\ K_2(x) &= x\mathbb{1}_{[1/2,\infty)}(x) \end{aligned}$$

then

$$f(x) = \beta_0 + \beta_1 K_1(x) + \beta_2 K_2(x) = \begin{cases} \beta_0 + \beta_1 x & x < 1/2 \\ \beta_0 + \beta_2 x & x \geq 1/2 \end{cases}$$

is linear on $[0, 1/2]$ and on $[1/2, 1]$. However,

$$\lim_{x \to 1/2^-} f(x) = \beta_0 + \beta_1/2 \neq \beta_0 + \beta_2/2 = \lim_{x \to 1/2^+} f(x)$$

is disjoint at the knot.

A *truncated linear spline* overcomes this by forcing continuity at the knots, for example using basis functions of the form

$$(x - a)_+ = \begin{cases} x - a & x - a > 0 \\ 0 & x - a \leq 0 \end{cases}$$

Define

$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - a)_+,$$

and note

$$f(x) = \begin{cases} \beta_0 + \beta_1 x & x < a \\ (\beta_0 - \beta_2 a) + (\beta_1 + \beta_2) x & x \geq a. \end{cases}$$

is linear on $x < a$ and $x \geq a$, and moreover $\lim_{x \to a^+} f(x) = \beta_0 + \beta_1 a = \lim_{x \to a^-} f(x)$ is continuous at the knot.

The general truncated linear spline model uses

$$K_i(x) = (x - a_i)_+$$

for knots $0 < a_1 < a_2 < \cdots < a_p < 1$, where

$$f(x) = \beta_0 + \beta_1 x + \sum_{j=1}^{p} \beta_{1i} K_i(x)$$

and is continuous, everywhere, and linear between knots.

The issue with linear splines is the obvious kinks at the knots. To overcome this, a *quadratic spline* replaces $(x - a)_+$ with $(x - a)_+^2$, so that

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \sum_{i=1}^{p} \beta_{2i} (x - a_i)_+^2$$

is *continuous* and has *continuous first derivatives everywhere.*

Typically the *cubic* spline is most common.

Definition 7. *A* cubic spline *with knots* $0 < a_1 < a_2 < \cdots < a_p < 1$ *is a cubic polynomial on each of* $(0, a_1), (a_1, a_2), \ldots, (a_p, 1)$ *and with* f, f' *and* f'' *being continuous at each* a_i*.*

A cubic spline can be represented via

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{i=1}^{p} \beta_{3i}(x - a_i)_+^3$$

and has *continuous first and second derivatives* on $[0, 1]$. The third derivative is constant between knots, but is disjoint at knot boundaries. A reason cubic splines are preferred is that it uses the lowest order of polynomial such that the eye cannot detect the knot locations. However, the truncated cubic spline can exhibit high variance at the outer range of predictors and the design matrix can behave badly (with potentially extremely large values and collinearity).

B-splines (Basis-splines) provide a well-conditioned basis for spline representations (that spans the same space as our truncated splines). For knots $0 = a_0 < a_1 < \cdots < a_p < a_{p+1} = 1$, the *B-spline of order 1* consists of constant functions

$$B_{i,1}(x) = \mathbb{1}_{[a_i, a_{i+1})}(x)$$

and note that these form a *partition of unity* in that

$$\sum_i B_{i,1}(x) = 1.$$

Higher order B-splines are obtained by recurrence, so

$$B_{i,k} = \frac{x - a_i}{a_{i+k-1} - a_i} B_{i,k-1} + \left(1 - \frac{x - a_{i+1}}{a_{i+k} - a_{i+1}}\right) B_{i+1,k-1}$$

so for $k = 2$,

$$B_{i,2}(x) = \frac{x - a_i}{a_{i+1} - a_i} \mathbb{1}_{[a_i, a_{i+1})}(x) + \frac{a_{i+2} - x}{a_{i+2} - a_{i+1}} \mathbb{1}_{[a_{i+1}, a_{i+2})}(x).$$

and note that the coefficient weights are just *linear* functions, thus the linear B-spline consists of piecewise linear functions, but moreover are compactly supported. For example, if $k = 4$ we have a cubic B-spline where second derivatives are continuous at the knots, just like cubic splines. [Note some conventions use k to denote the degree of the polynomial, rather than $k + 1$ as here].

Standard errors for cubic splines near the boundary of $[0,1]$ can behave poorly. A natural spline is a regularized version of a cubic spline.

Definition 8. *A* natural cubic spline *on* $[0,1]$ *is a cubic spline such that* $f'(0) = f''(0) = f'(1) = f''(1) = 0$.

In other words, a natural cubic spline is linear on $(0, a_1)$ and $(a_p, 1)$. How should a natural cubic spline be specified? We could use

$$f(x) = e_i(x - a_i)^3 + d_i(x - a_i)^2 + c_i(x - a_i) + b_i \quad \text{for} \quad a_i \le x \le a_{i+1},$$

that is, use a cubic polynomial between each knot. Continuity conditions of f, f' and f'' then put conditions on the coefficients $\{e_i, d_i, c_i, b_i\}$. There is a more useful representation, requiring some notation.

Define

$$\mathbf{f} = (f(a_1), \ldots, f(a_p))^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{\gamma} = (f''(a_2), \ldots, f''(a_{p-1}))^{\mathrm{T}},$$

noting that $f''(a_1) = f''(a_p) = 0$ by the boundary conditions. It turns out that $f(x)$ can be calculated *explicitly* at *any* $x \in [0,1]$, and only depends on $\mathbf{f}$ and $\boldsymbol{\gamma}$. More precisely, $\mathbf{f}$ and $\boldsymbol{\gamma}$ *define* the natural cubic spline (but not all choices of $\mathbf{f}$ and $\boldsymbol{\gamma}$ will result in such a spline).

Define

$$h_i = a_{i+1} - a_i, \quad i = 1, \ldots, p-1$$

and let $\mathbf{Q}$ be the $p \times (p-2)$ matrix

$$\mathbf{Q} = \begin{pmatrix} \frac{1}{h_1} & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{h_1} - \frac{1}{h_2} & \frac{1}{h_2} & 0 & \cdots & 0 & 0 \\ \frac{1}{h_2} & -\frac{1}{h_2} - \frac{1}{h_3} & \frac{1}{h_3} & \cdots & 0 & 0 \\ 0 & \frac{1}{h_3} & -\frac{1}{h_3} - \frac{1}{h_4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{1}{h_{p-3}} - \frac{1}{h_{p-2}} & \frac{1}{h_{p-2}} \\ 0 & 0 & 0 & \cdots & \frac{1}{h_{p-2}} & -\frac{1}{h_{n-2}} - \frac{1}{h_{p-1}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{h_{p-1}} \end{pmatrix}.$$

Let $\mathbf{R}$ be the symmetric $(p-2)\times(p-2)$ matrix

$$\mathbf{R} = \begin{pmatrix} \frac{1}{3}(h_1+h_2) & \frac{1}{6}h_2 & 0 & \cdots & 0 \\ \frac{1}{6}h_2 & \frac{1}{3}(h_2+h_3) & \frac{1}{6}h_3 & \cdots & 0 \\ 0 & \frac{1}{6}h_3 & \frac{1}{3}(h_3+h_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{3}(h_{p-2}+h_{p-1}) \end{pmatrix},$$

and note that $\mathbf{R}$ is positive definite. Finally define

$$\mathbf{K} = \mathbf{Q}\mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}.$$

Theorem 9. *The vectors* $\mathbf{f}$ *and* $\boldsymbol{\gamma}$ *define a natural cubic spline if and only if*

$$\mathbf{Q}^{\mathrm{T}} = \mathbf{R}\boldsymbol{\gamma},$$

and moreover

$$\int_0^1 f''(x)^2 \mathrm{d}x = \boldsymbol{\gamma}^{\mathrm{T}}\mathbf{R}\boldsymbol{\gamma} = \mathbf{f}^{\mathrm{T}}\mathbf{K}\mathbf{f}.$$

This theorem will have crucial implications in the next two sections.

The last remaining issue is *knot placement*.

- Place many knots near locations where you expect the function to vary a lot
- Place knots uniformly through $[0,1]$
- Place knots at uniform quantiles of the x covariates
- Use cross-validation (e.g., remove 10% of the data) and allow the knot numbers/locations to vary, favor model with best predictive power
- Use a forward/backward selection algorithm, say starting from a dense grid of possible knots

[XXX: R example (Splines1.R)]

6.2 Penalized Regression Splines

Regression splines are only as flexible as the number of knots used, however for $p \gg 0$ the regression model has many parameters and high variability of OLS estimators can be a problem. One route is to perform *automatic knot selection*, but another is to use relatively many knots, but to *regularize* the coefficients, as in lasso/ridge regression.

For example, using a truncated power basis of degree q, with p knots, the ith set of covariates is

$$\mathbf{x}_i = \left(1, x_1, x_1^2, \ldots, x_1^q, (x_1 - a_1)_+^q, (x_1 - a_2)_+^q, \ldots, (x_1 - a_p)_+^q\right)$$

yielding design matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix}.$$

A *penalized spline* then is the minimizer of

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^{\mathrm{T}}\mathbf{D}\boldsymbol{\beta}$$

for a penalty matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{0}_{(q+1)\times(q+1)} & \mathbf{0}_{(q+1)\times p} \\ \mathbf{0}_{p\times(q+1)} & \mathbf{I}_{p\times p} \end{pmatrix}.$$

Note that the penalty can be written

$$\boldsymbol{\beta}^{\mathrm{T}}\mathbf{D}\boldsymbol{\beta} = \sum_{i=q+2}^{q+1+p} \beta_i^2$$

and thus *does not* penalize the first q-degree polynomial. This is similar to ridge regression, but with a more complicated mean function.

If $\lambda = 0$, the solution reduces to OLS, whereas if $\lambda \to \infty$, the fitted curve nears

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_q X^q + \varepsilon,$$

that is, just a q-degree polynomial regression.

For observations $\mathbf{y}$, the minimizer is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

and the fitted values are then

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}.$$

The truncated power basis behaves poor, numerically, so we should recast this solution in terms of the B-spline basis of order q. In particular, there is a square invertible matrix $\mathbf{L}_q$ such that

$$\mathbf{X}_B = \mathbf{X}\mathbf{L}_q$$

where $\mathbf{X}_B$ is the corresponding B-spline basis. Then the fitted values become

$$\begin{aligned}(\mathbf{X}_B\mathbf{L}_q^{-1})\big((\mathbf{X}_B\mathbf{L}_q^{-1})^{\mathrm{T}}(\mathbf{X}_B\mathbf{L}_q^{-1}) + \lambda\mathbf{D}\big)^{-1}(\mathbf{X}_B\mathbf{L}_q^{-1})^{\mathrm{T}}\mathbf{y} &= \mathbf{X}_B\mathbf{L}_q^{-1}\big(\mathbf{L}_q^{-\mathrm{T}}\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B\mathbf{L}_q^{-1} + \lambda\mathbf{D}\big)^{-1}\mathbf{L}_q^{-\mathrm{T}}\mathbf{X}_B^{\mathrm{T}}\mathbf{y} \\ &= \mathbf{X}_B(\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B + \lambda\mathbf{L}_q^{\mathrm{T}}\mathbf{D}\mathbf{L}_q)^{-1}\mathbf{X}_B^{\mathrm{T}}\mathbf{y}\end{aligned}$$

so that the fitted values are *equivalent* to the B-spline solution under the penalty matrix $\mathbf{L}_q^{\mathrm{T}}\mathbf{D}\mathbf{L}_q$.

Other penalties may be used. For instance, Eilers and Marx (1996) suggested a fitting procedure based on n observations and p B-splines $B_1, \ldots, B_p$ of order k

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} a_j B_j(x_i)\right)^2 + \lambda \sum_{j=k+1}^{p} (\Delta^k a_j)^2$$

where $\Delta a_j = a_j - a_{j-1}$. Note this penalty *encourages adjacent a_j's to be similar.* [There is a connection to a MRF prior for this penalty].

[XXX: R example (Splines2.R)]

6.3 Smoothing Splines

A smoothing spline approaches the fitting procedure in a different way. Recall the model can be written

$$Y = f(X) + \varepsilon,$$

so we would desire to minimize

$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

over some class of functions $\{f\}$ (up to now we have seen parametric or 'nonparametric' classes of functions for f). What if we wanted to allow f to be (almost) *any* function with two continuous derivatives? The issue is that even very smooth classes can interpolate the data by producing extremely wiggly functions, i.e., we can get $RSS = 0$.

Instead, consider

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int_0^1 f''(x)^2 \mathrm{d}x, \tag{6}$$

where the penalty term now penalizes the (integrated squared) curvature of the fitted function f. The function that minimizes (6) is called a *cubic smoothing spline*. In fact, it turns out to be a natural cubic spline with knots at $x_1, \ldots, x_n$, but whose coefficients are *shrunken* by a relationship with λ. Conversely, a natural cubic spline with knots at every x_i yields a cubic smoothing spline.

We show this in three steps. The first is to note that, given a set of target points $z_1, \ldots, z_n$ at points $0 < x_1 < \cdots < x_n < 1$, there is a *unique* natural cubic spline f such that $f(x_i) = z_i$ for all i. To see this, if $\mathbf{z} = (z_1, \ldots, z_n)^{\mathrm{T}}$ and $\mathbf{f} = (f(x_1), \ldots, f(x_n))^{\mathrm{T}}$, then recall that $\mathbf{f}$ is a natural cubic spline if and only if there is a vector $\boldsymbol{\gamma}$ so that

$$\mathbf{Q}^{\mathrm{T}}\mathbf{f} = \mathbf{R}\boldsymbol{\gamma}.$$

Setting $\boldsymbol{\gamma} = \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{z}$ gives the unique solution.

Theorem 10. *Suppose $n \geq 2$ and f is the (unique) natural cubic spline such that $f(x_i) = z_i$ for $0 < x_1 < \cdots < x_n < 1$. Let $\tilde{f}$ be* any *other function that is continuously twice differentiable on $[0, 1]$ also satisfying $\tilde{f}(x_i) = z_i$. Then*

$$\int_0^1 f''(x)^2 \mathrm{d}x \leq \int_0^1 \tilde{f}''(x)^2 \mathrm{d}x.$$

Proof.

$$\begin{aligned}\int_0^1 \tilde{f}''^2 &= \int (f'' + (\tilde{f}'' - f''))^2 \\ &= \int f''^2 + 2\int f''(\tilde{f}'' - f'') + \int (\tilde{f}'' - f'')^2 \\ &= \int f''^2 + \int (\tilde{f}'' - f'')^2 \\ &\geq \int f''^2\end{aligned}$$

with equality only if $(\tilde{f}'' - f'')$ is linear on $[0,1]$, but $(\tilde{f}'' - f'')(x_i) = 0$ for $n \geq 2$ points, so $\tilde{f} \equiv f$. The cross product is zero using integration by parts and that f is a natural cubic spline and $f(x_i) - \tilde{f}(x_i) = 0$. □

Finally, we are ready for the main theorem.

Theorem 11. *There is a unique natural spline f that minimizes (6) which is given by*

$$\mathbf{f} = (f(x_1), \ldots, f(x_n))^{\mathrm{T}} = (\mathbf{I}_n + \lambda \mathbf{K})^{-1}\mathbf{y}.$$

Proof. Let $\tilde{f}$ be any other function (with $\int \tilde{f}'' < \infty$). Then let f be the unique natural spline interpolator for $f(x_i)$, i.e., $f(x_i) = \tilde{f}(x_i)$. Trivially,

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2$$

and moreover, by the previous theorem, $\int f''^2 \leq \int \tilde{f}''^2$. □

In other words, for any candidate function, we can always find a natural cubic spline with a smaller (or equal) value of (6).

6.4 The Smoother Matrix

Recall that the predicted values for a penalized spline are of the form

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{D})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{W}_\lambda \mathbf{y}$$

for a *weight matrix* $\mathbf{W}_\lambda$. In particular, $\hat{\mathbf{y}}$ is a *linear smoother* in that it is in *linear in the observations*. Sometimes $\mathbf{W}_\lambda$ is called the *smoother matrix*.

Suppose the smoother matrix $\mathbf{W}_\lambda$ is $n \times n$ and nonnegative definite. Then let $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ be the ordered eigenvalues with associated eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Since the eigenvectors form a basis of $\mathbb{R}^n$, we have

$$\mathbf{y} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i$$

for some coefficients $\alpha_1, \ldots, \alpha_n$, and

$$\hat{\mathbf{y}} = \mathbf{W}_\lambda \mathbf{y} = \mathbf{W}_\lambda \sum_{i=1}^{n} \alpha_i \mathbf{v}_i = \sum_{i=1}^{n} \alpha_i W_\lambda \mathbf{v}_i = \sum_{i=1}^{n} \alpha_i \lambda \mathbf{v}_i$$

so when $\lambda_i \approx 0$, we are discarding the effects of all eigenvectors $j \geq i$.

Recall that the trace of the hat matrix defined the degrees of freedom of a parametric model. In the same way, we define $\mathrm{tr}(\mathbf{W}_\lambda)$ as the *effective number of parameters* for a spline. A smoother with ν degrees of freedom summarizes the data about to the same extent that a $(\nu - 1)$-degree polynomial does.

For a penalized spline with N knots and degree p, we have

$$\mathrm{tr}(\mathbf{W}_0) = p + 1 + N$$

and at the other extreme, as $\lambda \to \infty$,

$$\mathrm{tr}(\mathbf{W}_\lambda) \to p + 1,$$

so values of $0 < \lambda < \infty$ imply

$$p + 1 < \text{degrees of freedom} < p + 1 + N.$$

The degrees of freedom can be used to compare competing models as a measure of their complexity.

How do we estimate the remaining variability σ^2? Our estimator from multiple regression was $RSS/(n - p + 1)$, but we need the *residual degrees of freedom* to do this. Consider

smoothed estimates $\hat{\mathbf{y}}$ of observations from the model $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$, then (setting $\mathbf{W}_\lambda = \mathbf{W}$ for ease)

$$\begin{aligned}
\mathbb{E}(RSS) &= \mathbb{E}\big((\hat{\mathbf{y}} - \mathbf{y})^{\mathrm{T}}(\hat{\mathbf{y}} - \mathbf{y})\big) \\
&= \mathbb{E}\big(\mathbf{y}^{\mathrm{T}}(\mathbf{W} - \mathbf{I})^{\mathrm{T}}(\mathbf{W} - \mathbf{I})\mathbf{y}\big) \\
&= \mathbf{f}^{\mathrm{T}}(\mathbf{W} - \mathbf{I})^{\mathrm{T}}(\mathbf{W} - \mathbf{I})\mathbf{f} + \sigma^2 \mathrm{tr}\big((\mathbf{W} - \mathbf{I})^{\mathrm{T}}(\mathbf{W} - \mathbf{I})\big) \\
&= \|(\mathbf{W} - \mathbf{I})\mathbf{f}\|^2 + \sigma^2\big(\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathrm{T}}) - 2\mathrm{tr}(\mathbf{W}) + n\big).
\end{aligned}$$

[We used the fact $\mathbb{E}(\mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v}) = \mathbb{E}(\mathbf{v})^{\mathrm{T}}\mathbb{E}(\mathbf{v}) + \mathrm{tr}(\mathbf{A}\mathrm{Cov}(\mathbf{v}))$]. The first term is the squared bias, and, assuming it is small then a natural definition for residual degrees of freedom is

$$\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathrm{T}}) - 2\mathrm{tr}(\mathbf{W}) + n.$$

To choose the smoothing parameter λ, we rely on cross validation. In particular, we seek to minimize

$$RSS_{cv}(\lambda) = \sum_{i=1}^{n} (y_i - \hat{f}_\lambda^{-(i)}(x_i))^2$$

where $\hat{f}_\lambda^{-(i)}(x_i)$ is the fitted value at x_i using all data *except* (x_i, y_i). There is a convenient formula for this RSS,

$$RSS_{cv}(\lambda) = \sum_{i=1}^{n} \left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - (\mathbf{W}_\lambda)_{ii}} \right)$$

where $\hat{f}_\lambda(x_i)$ is the spline smoothed version using *all* of the data. This corresponds to *leave-one-out* cross-validation, but there are other versions leaving out more than one at a time.

[XXX: R example (Splines3.R)]

6.5 Local Regression

Local polynomial fitting follows polynomial regression, but where the residual weights vary with x. For some positive symmetric kernel function K the local polynomial fit at x minimizes

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1(x_i - x) - \cdots - \beta_p(x_i - x)^p) K\left(\frac{x_i - x}{b} \right)$$

where b is the *bandwidth*. The estimated curve is the value $\hat{\beta}_0$ where we use weighted least squares estimates

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y}$$

where

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 - x & \cdots & (x_1 - x)^p \\ 1 & x_2 - x & \cdots & (x_2 - x)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x & \cdots & (x_n - x)^p \end{pmatrix},$$

and

$$\mathbf{W} = \operatorname{diag}\left(K\left(\frac{x_1 - x}{b}\right), \cdots, K\left(\frac{x_n - x}{b}\right)\right).$$

Note that the estimated coefficients *vary with* x, so the locally estimated curve is

$$\hat{f}(x; p, b) = \mathbf{e}_1^{\mathrm{T}}\hat{\boldsymbol{\beta}}.$$

where $\mathbf{e}_1 = (1\,0\,0\cdots 0)^{\mathrm{T}}$. Usually $p = 2$ is sufficiently flexible, and b is estimated by cross-validation or visual inspection. If we set $p = 0$, we get the *Nadaraya-Watson* estimator

$$\hat{f}(x; 0, b) = \frac{\sum_{i=1}^{n} K\left(\frac{x_i - x}{b}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x_i - x}{b}\right)}.$$

This is a *locally constant* estimator that has been studied since the 60s.

6.6 Generalized Additive Models

A *generalized additive model* (GAM) is the generalization from a regression (classification) problem using linear predictors to non-linear transformations of those predictors. In particular, for covariates $X_1, \ldots, X_p$, the GAM model is

$$Y = \beta_0 + \sum_{i=1}^{p} f_i(X_i) + \varepsilon$$

where f_i are smooth non-linear functions (e.g., splines).

Fitting a GAM is nontrivial, and uses a method known as *backfitting*, repeatedly updating the fit for each predictor holding all others fixed. GAMs allow non-linear relationships and convenient interpretations (e.g., we can examine the smoothness for each fitted f_j by its degrees of freedom), but they are additive and miss out on interactions.

[XXX R example: (GAMs.R)]

6.7 Smoothing Splines Revisited

Some argue that penalized splines are more flexible than smoothing splines due to the ability to select knots as well as a sensible penalty. Indeed, sometimes penalized splines will be referred to as *low rank* since they reduce the number of knots below n, whereas smoothing splines are *full rank* since they place a knot at every x_i coordinate. However, there is another interpretation of the smoothing spline solution as *lying in a particular class of functions*.

Recall the cubic smoothing spline problem, if $\mathcal{H} = \{f | \int_0^1 f''(x)^2 \mathrm{d}x < \infty\}$, then the cubic smoothing spline solution is the minimizer of

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 f''(x)^2 \mathrm{d}x,$$

over $f \in \mathcal{H}$. We will show later that this solution can be written

$$\hat{f}(x) = \alpha_0 + \alpha_1 x + \sum_{i=1}^{n} \beta_i k(x, x_i)$$

where

$$k(u, v) = \begin{cases} \frac{1}{2}u^2 v - \frac{1}{6}u^3, & u < v \\ \frac{1}{2}v^2 u - \frac{1}{6}v^3, & u \geq v \end{cases}$$

and, moreover, $\mathcal{H}$ is the space of functions that is the completion of all linear combination of k, $\sum_{i=1}^{\infty} a_i k(x, z_i)$.

7 Tree-Based Methods

Tree-based methods are for both regression and classification. They work by *stratifying* or *segmenting* the predictor space into regions and use the mean of those regions as predictors. These are known as *decision tree* methods. Trees are useful for interpretation, but are not necessarily as good at prediction as the methods we have already discussed.

- Trees are easy to explain (and interpret)
- Trees have easy graphical depictions
- Trees can handle qualitative predictors without the need for dummy variables
- Trees will typically perform very poorly in terms of prediction compared to a linear model

7.1 Regression Trees

Regression trees just split the predictor space into regions, and uses the average response within each region as the predictor. For example, if Y is rating of a movie (out of 10) and X is the budget (in dollars), then a simple regression tree would forecast

$$\hat{y} = \begin{cases} 7.03, & x < 10250 \\ 6.07, & x > 10250. \end{cases}$$

The line $x = 10250$ defines the two regions in this case.

The basic steps for any general predictors $X_1, \ldots, X_p$ are to

(1) Divide predictor space into J disjoint regions $R_1, \ldots, R_J$.

(2) To forecast a new response in R_j use the the mean of all observations in R_j.

The regions R_j will be hyperrectangles, and if $\hat{y}_{R_j}$ is the average response in region R_j then we seek to minimize

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2.$$

To find regions we use *recursive binary splitting*, a *top-down greedy* approach.

[Top-down = begins at the top of the tree, lopping all observations together]

[Greedy = at each step of building, use the best split without looking into the future]

The first step is to select the predictor X_j and the cutoff s such that $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ provide the greatest reduction in RSS (over all possible js and ss). Then within *each* of the new regions, do binary splitting. This is continued until some stopping criterion is reached, e.g., no region contains more than 5 observations.

[Picture (figure 8.3 of ISLR)]

This procedure will overfit the data (the bias may be low, but at the cost of high variance). Thus, we will grow a very large tree T_0 and *prune* back to obtain a *subtree*. We do this by *cost complexity pruning*, also known as *weakest link pruning*. Rather than considering every possible subtree, consider a sequence of trees indexed by a tuning parameter $\alpha > 0$. In particular, for each α there is a subtree $T \subset T_0$ that minimizes

$$\sum_{m=1}^{|T|} \sum_{i|x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T|.$$

Here, $|T|$ is the number of terminal nodes in the tree, and R_m is the rectangle corresponding to the mth terminal node. Note this is just another loss + penalty idea, where the penalty is now on the number of nodes of the tree.

[XXX: R example (RegressionTrees.R)]

HW idea: do algorithm 8.1

7.2 Classification Trees

A *classification tree* is the same thing as a regression tree, but our prediction within a node is the *most commonly occurring class*. Rather than using RSS, we use the *classification error rate*, that is, the fraction of training observations that do *not* belong to the most common class.

Suppose we have K classes. It turns out that classification error is not sufficiently sensitive for growing trees, so we use two other measures. If we define $\hat{p}_{mk}$ as the proportion of training

observations in the mth region from the kth class, then the *Gini index* is

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}).$$

Note that G is small if all $\hat{p}_{mk}$s are close to zero or one, and thus the Gini index is a measure of node *purity*. The *cross-entropy* index is

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

and is also close to zero if $\hat{p}_{mk}$ are always close to zero or one.

[XXX: R example (ClassificationTrees.R)]

7.3 Bagging

8 Gaussian Process Regression

Our basic model is

$$Y = f(X) + \varepsilon$$

where, until now, we have assumed some parametric or nonparametric form for f based on X. In this section we will assume f is a *random function* from some particular *class of functions*.

[XXX: R example (GPIntro.R) – simulate some example functions and some example data from different covariance models]

8.1 RKHS

We need a few elementary definitions to proceed.

Definition 12. *We say* $\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ *is an* inner product *over the vector space* $\mathcal{H}$ *if*

- $\langle f, g \rangle = \langle g, f \rangle$
- $\langle af, g \rangle = a\langle f, g \rangle$ *for any* $a \in \mathbb{R}$ *and* $\langle f + h, g \rangle = \langle f, g \rangle + \langle h, g \rangle$ *and*
- $\langle f, f \rangle \geq 0$ *and if* $\langle f, f \rangle = 0$ *then* $f \equiv 0$.

Definition 13. *A vector space* $\mathcal{H}$ *is a* Hilbert space *if it is a complete metric space with respect to the distance*

$$\|f - g\| = \sqrt{\langle f - g, f - g \rangle}.$$

We will be interested in Hilbert spaces of functions, that is, spaces where each element is a function.

Definition 14. *A function* $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *is* reproducing kernel *of the Hilbert space* $\mathcal{H}$ *if and only if*

- $K(\cdot, \mathbf{s}) \in \mathcal{H}$ *for all* $\mathbf{s}$
- $\langle f, K(\cdot, \mathbf{s}) \rangle = f(\mathbf{s})$ *for all* $f \in \mathcal{H}$ *and all* $\mathbf{s}$.

Note that $K(\cdot, \mathbf{s})$ *reproduces* f at a location $\mathbf{s}$. Note that

$$\langle K(\cdot, \mathbf{s}), K(\cdot, \mathbf{t}) \rangle = K(\mathbf{s}, \mathbf{t})$$

and

$$K(\mathbf{s}, \mathbf{t}) = \langle K(\cdot, \mathbf{s}), K(\cdot, \mathbf{t}) \rangle = \langle K(\cdot, \mathbf{t}), K(\cdot, \mathbf{s}) \rangle = K(\mathbf{t}, \mathbf{s})$$

so that K is a symmetric function. A Hilbert space admitting a reproducing kernel is known as a *reproducing kernel Hilbert space* (RKHS).

Example 15. *Suppose* $(e_1, \ldots, e_n)$ *is an orthonormal basis for a Hilbert space* $\mathcal{H}$*. Then*

$$K(x, y) = \sum_{i=1}^{n} e_i(x) e_i(y)$$

is a reproducing kernel for $\mathcal{H}$*. To check, first note*

$$K(x, \cdot) = \sum_{i=1}^{n} e_i(x) e_i(\cdot) = \sum_{i=1}^{n} \lambda_i e_i(\cdot)$$

is an element of $\mathcal{H}$ *by the definition of a basis. Additionally, any* $\phi \in \mathcal{H}$ *can be written* $\phi(x) = \sum_{i=1}^{n} c_i e_i(x)$*, so*

$$\begin{aligned} \langle \phi(\cdot), K(\cdot, x) \rangle &= \left\langle \sum_{i=1}^{n} c_i e_i(\cdot), K(\cdot, x) \right\rangle \\ &= \sum_{i=1}^{n} c_i \langle e_i(\cdot), K(\cdot, x) \rangle \\ &= \sum_{i=1}^{n} c_i e_i(x) = \phi(x). \end{aligned}$$

Thus, any finite-dimensional Hilbert space is a RKHS.

The connection to spatial statistics is hidden in the following two theorems.

Theorem 16. *If* $\mathcal{H}$ *is a RKHS with reproducing kernel* $K(\cdot, \cdot)$*, then* K *is a positive definite function.*

Proof. For any $a_1, \ldots, a_n \in \mathbb{R}$ and any $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathbb{R}^d$, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j K(\mathbf{s}_i, \mathbf{s}_j) = \left\| \sum_{i=1}^{n} a_i K(\cdot, \mathbf{s}_i) \right\|^2 \geq 0.$$

□

Theorem 17 (Moore-Aronszajn)**.** *If $K(\cdot,\cdot)$ is a positive definite function on $\mathbb{R}^d \times \mathbb{R}^d$ then there exists a* unique *RKHS $\mathcal{H}$ having K as its reproducing kernel. Moreover, the functions $f(\mathbf{s}) = \sum_{i=1}^n a_i K(\mathbf{s}, \mathbf{s}_i)$ are dense in $\mathcal{H}$.*

Theorem 18. *If $\mathcal{H}$ is a Hilbert space with reproducing kernel K, then the solution $\hat{f}$ minimizing*

$$\mathcal{L}(f) = \sum_{i=1}^{n} (y(\mathbf{s}_i) - f(\mathbf{s}_i))^2 + \lambda \langle f, f \rangle \tag{7}$$

is

$$\hat{f}(\mathbf{s}) = \sum_{i=1}^{n} m_i K(\mathbf{s}, \mathbf{s}_i)$$

where $\mathbf{m}$ is the minimizing solution of

$$\|\mathbf{y} - \Sigma\mathbf{m}\|^2 + \lambda \mathbf{m}^{\mathrm{T}} \Sigma \mathbf{m}$$

with $\mathbf{m} = (m_1, \ldots, m_n)^{\mathrm{T}}$ and $\Sigma = (K(\mathbf{s}_i, \mathbf{s}_j))_{i,j=1}^n$.

In particular, the minimizing solution is

$$\mathbf{m} = (\Sigma + \lambda I)^{-1} \mathbf{y}$$

and thus if $K = C$ and $\lambda = \tau^2$, we have that the minimizing solution to (7) *is* the kriging smoother.