Computations for Linear Models

J.H. Maindonald*

August 7, 2007

Abstract

These notes

review the theory of linear models, noting the utility of Householder reflections and the QR decomposition both for computation and for theory;
[c.f., Chapter 1 of Wood (2006)]

discuss regression splines, noting in passing the further extension into generalized additive models;
Maindonald & Braun (2007, Section 7.1, pp. 234–238).

demonstrate the use of resmampling and simulation methods for comparing models;

give brief summary details of the theory of Generalized Linear Models, with logistic regression and Poisson regression as special cases.
[Maindonald & Braun (2007, Sections 8.1 – 8.5); Wood (2006, Sections 2.1 – 2.3)]

*Centre for Mathematics & Its Applications, Australian National University, Canberra ACT 0200, Australia.
mailto:john.maindonald@anu.edu.au

Contents

1 Linear Models – Key Concepts

1.1 Terminology

Note the distinction between fixed and random effects. In

$$y_i = \alpha + \beta x_i + \epsilon_i, \quad i = 1, 2, \ldots n$$

α and β are fixed effects, while the ϵ_i are random effects. A common assumption is that the ϵ_i are distributed as N(0, σ^2), independently between observations. This is commonly known as the iid normal asssumption.

1.2 Basis Functions

We begin by defining what we mean by a "linear model". Define basis functions

$$\phi_1(x_1, x_2, \ldots x_k), \phi_2(x_1, x_2, \ldots x_k), \ldots, \phi_p(x_1, x_2, \ldots x_k)$$

In the simplest case $p = k$ and $\phi_1(x_1, x_2, \ldots x_p) = x_1$, $\phi_2(x_1, x_2, \ldots x_p) = x_2$, $\ldots$, $\phi_p(x_1, x_2, \ldots x_p) = x_p$.

Then any function with values on the real line such that

$$f(x_1, x_2, \ldots x_k) = \beta_1\phi_1(x_1, x_2, \ldots x_k) + \beta_2\phi_2(x_1, x_2, \ldots x_k) + \ldots + \beta_p\phi_p(x_1, x_2, \ldots x_k)$$

where the elements of $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots \beta_p)$ are the only unknowns, specifies a linear model. In words, any model in which E[y] is a linear combination of the ϕ_i is a linear model.

Note three important non-trivial special cases:

- Several ϕ_i's are defined that together account for a factor term. See below for the definition of "term".
- Several ϕ_i's are defined that together account for a spline term.
- Interaction terms may be created by multiplying together columns that relate to other terms in the model.

The model is linear in the values that the ϕ's take on the sample data. It is not, in general, linear in the x_i's.

1.3 Terms – groups of basis functions

A factor with k levels requires, assuming that the model already has a constant term, $k-1$ basis elements to represent it. These basis elements together constitute a "term", in this case a factor. The Wilkinson & Rogers notation makes it possible to specify the factor as a term in the model, leaving code that is called by R's `lm()` function to determine the basis functions that are needed.

Similarly a 4 degree of freedom spline term requires four basis elements. Again, code that is called by R's `bs()` (B-splines) or `ns()` (natural splines) function will determine the basis functions.

Quite generally, the basis functions $\phi_1, \phi_2, \ldots, \phi_p$ can be categorized into groups, with one group for each term the model, thus:

$$\underbrace{\phi_1, \ldots, \phi_{m_1}}_{\text{Term1}}, \underbrace{\phi_{m_1+1}, \ldots, \phi_{m_2}}_{\text{Term2}}, \cdots$$

1.4 Factor basis functions

A simple example will do for now. Consider the `sugar` data frame in the R package. There are three levels of `trt` – `Control`, `A`, `B` and `C`. Type into R:

```
> contrasts(sugar$trt)
        A B C
Control 0 0 0
A       1 0 0
B       0 1 0
C       0 0 1
```

The first factor level, ie `Control`, is taken as the baseline. The parameter associated with `A` is then its difference from the baseline, and similarly for `B` and `C`.

Now see how this works in practice:

```
> sugar.lm <- lm(weight ~ trt, data=sugar)
> model.matrix(sugar.lm)
   (Intercept) trtA trtB trtC
1            1    0    0    0
2            1    0    0    0
3            1    0    0    0
4            1    1    0    0
5            1    1    0    0
6            1    1    0    0
7            1    0    1    0
8            1    0    1    0
9            1    0    1    0
10           1    0    0    1
11           1    0    0    1
12           1    0    0    1
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$trt
[1] "contr.treatment"
```

In the above, `(Intercept)` is really `Control`. The columns `trtA`, `trtB` and `trtC` measure differences from `Control`

Another possibility is

```
> sugar.lm0 <- lm(weight ~ -1 + trt, data=sugar)
> model.matrix(sugar.lm0)
   trtControl trtA trtB trtC
1           1    0    0    0
2           1    0    0    0
3           1    0    0    0
4           0    1    0    0
5           0    1    0    0
6           0    1    0    0
7           0    0    1    0
8           0    0    1    0
9           0    0    1    0
10          0    0    0    1
11          0    0    0    1
12          0    0    0    1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$trt
[1] "contr.treatment"
```

Omission of the "constant term" from the formula forces the use of one parameter for each different factor level. With this parameterization no restriction on the parameters is needed.

1.5 Spline basis functions

We will come to these later. Spline basis functions are, in essence, a construction kit for constructing curves that have some predefined flexibility.

2 Solving Least Squares Systems

Least squares may be used because it seems intuitively sensible. Or it may be justified by maximum likelihood, assuming independently and identically distributed normal errors.

The QR decomposition, usually achieved by a sequence of Householder reflections, is widely used in the practical computer solution of linear least squares systems. As well as giving details of the computational steps, it will be shown that important theoretical results follow rather directly from the computational theory.

An understanding of the computational details can be important when efficient computation is important, e.g., a computation is repeated a large number of times. Certain special types of least squares problems are more efficiently handled by other methods, or by one or other adaptation of QR methods. Sparse linear systems are an important special case. For very large linear least squares systems, highly efficient computations and/or minimization of storage requirements can be crucial to completing calculations within reasonable time or even to getting calculations to run at all. See for example Bates (2006); Koenker and & Ng (2003).

Algorithms that are highly efficient on single processor systems may require substantial adaptation to get maximum advantage from multiple processor systems. Or different algorithms may be required.

2.1 Householder Reflections, and QR

Given an $n \times p$ matrix $\mathbf{X}$, the QR decomposition derives an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$ such that

$$\mathbf{Q}'\mathbf{X} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

Then

$$\mathbf{X} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

In the sequel, it will be shown that the decomposition can be achieved by constructing a sequence of reflections that successively reduce to zero below diagonal elements in columns 1, 2, $\ldots p$ of $\mathbf{X}$.

A Householder reflection of a vector is achieved by pre-multiplication by a Householder matrix, ie a matrix of the form:

$$\mathbf{H} = \mathbf{I} - \gamma \mathbf{u}\mathbf{u}', \quad \text{where} \quad \gamma = \frac{2}{||\mathbf{u}||^2}$$

In least squares applications, with a model matrix $\mathbf{X}$ and vector of observations $\mathbf{y}$, a sequence of Householder reflections is used to reduce $\mathbf{X}$ to upper triangular from. The same sequence of reflections is applied also to $\mathbf{y}$. The first reflection yields $\mathbf{H}_1\mathbf{X}$. The second reflection, which operates only on rows of $\mathbf{H}_1\mathbf{X}$ subsequent to the first, replaces below diagonal elements in the second column with zeros, yielding $\mathbf{H}_2\mathbf{H}_1\mathbf{X}$.

Figure 1 shows diagrammatically the sequence of Householder reflections, applied to a 9×4 model matrix $\mathbf{X}$, for reducing a 9×4 model matrix $\mathbf{X}$ to upper triangular form.

Before proceeding, note important properties of Householder reflections.

2.1.1 Properties of Householder reflections

1. A Householder matrix is orthogonal, ie, $\mathbf{H}'\mathbf{H} = \mathbf{I}$.
2. The product of two Householder matrices is an orthogonal matrix.

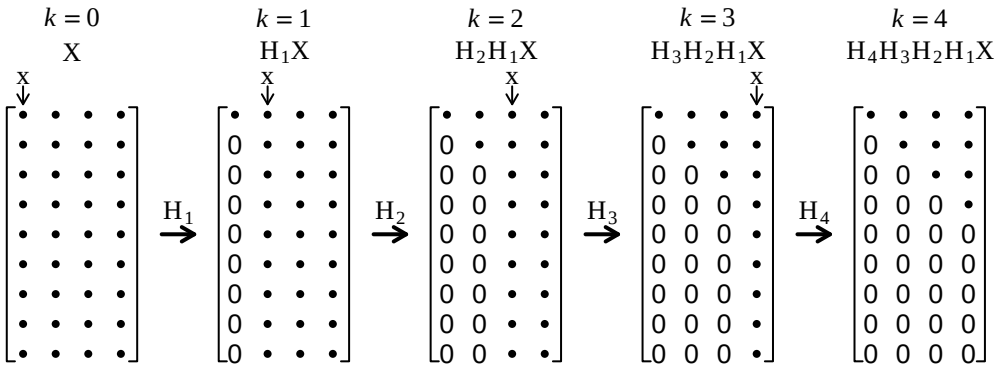

Figure 1: Diagrammatic representation of the sequence of steps that are applied to a 9×4 model matrix $\mathbf{X}$ when Householder reflections are used in least squares calculations. The vector $\mathbf{x}$ whose final $n - k$ elements are used in forming $\mathbf{u}_k$ and hence $\mathbf{H}_k$ is identified, for each of the successive steps, with an arrow ($\downarrow$). Thus $\mathbf{H}_1$ is formed using the first column of $\mathbf{X}$ ($k = 0$), $\mathbf{H}_2$ is formed using the second and later elements in the second column of $\mathbf{H}_1\mathbf{X}$ ($k = 1$), $\mathbf{H}_3$ is formed using the third and later elements in the third column of $\mathbf{H}_2\mathbf{H}_1\mathbf{X}$ ($k = 2$), and so on. The same sequence of Householder reflections is applied to $\mathbf{y}$.

3. For a suitable choice of $\mathbf{H}$

$$\mathbf{Hx} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{k-1} \\ \eta_k \\ 0 \\ . \\ . \\ 0 \end{bmatrix}$$

The proof of (1) is a straightforward use of matrix algebra.

For (2), observe that

$$\begin{aligned} (\mathbf{H}_2\mathbf{H}_1)'(\mathbf{H}_2\mathbf{H}_1) &= \mathbf{H}_1'\mathbf{H}_2')\mathbf{H}_2\mathbf{H}_1 \\ &= \mathbf{H}_1'(\mathbf{H}_2'\mathbf{H}_2)\mathbf{H}_1 \\ &= \mathbf{H}_1'\mathbf{I}\mathbf{H}_1 \\ &= \mathbf{H}_1'\mathbf{H}_1 \end{aligned}$$

For (3), observe that

$$(\mathbf{I} - \gamma\mathbf{u}\mathbf{u}')\mathbf{x} = \mathbf{x} - \mathbf{u}\gamma(\mathbf{u}'\mathbf{x})$$

Consider the case $k = 1$. We set $u_i = x_i \quad (i = 2, \ldots n)$, and require that $\gamma(\mathbf{u}'\mathbf{x}) = 1$. The elements $x_i, i = 2, \ldots n$ will then be replaced by zeros.

We ensure that $\gamma(\mathbf{u}'\mathbf{x}) = 1$ by a suitable choice of u_1; it is convenient to write $u_1 = x_1 + \alpha$.

Thus we have

$$\mathbf{u} = \begin{bmatrix} x_1 + \alpha \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$$

Then

$$\gamma\mathbf{u}'\mathbf{x} = \frac{2}{\|\mathbf{u}\|^2}(\|\mathbf{x}\|^2 + \alpha x_1)$$

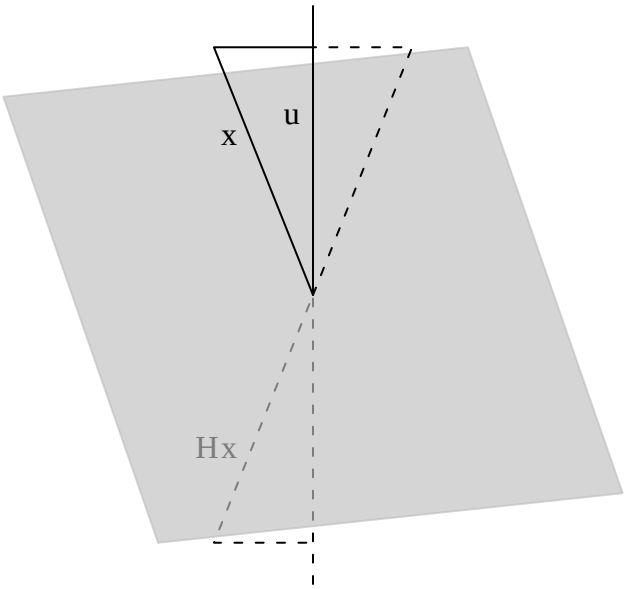

Figure 2: Geometrical representation of the reflection of $\mathbf{x}$ in the plane that is orthogonal to $\mathbf{u}$. Here $\mathbf{H} = I - \gamma \mathbf{u}\mathbf{u}'$, with $\mathbf{u} = x_1 + \text{sgn}(x_1)\|\mathbf{x}\|$.

For this to equal 1

$$\begin{aligned} 2(\|\mathbf{x}\|^2 + \alpha x_1) &= \|\mathbf{u}\|^2 \\ &= \|\mathbf{x}\|^2 + 2\alpha x_1 + \alpha^2 \end{aligned}$$

Thus

$$\alpha^2 = \|\mathbf{x}\|^2 \quad \alpha = \pm\|\mathbf{x}\|$$

Figure 2 gives a geometrical representation of the reflection of $\mathbf{x}$ in the plane that is orthogonal to $\mathbf{u}$. Note that $\mathbf{Hx}$ is a positive or negative multiple $(1, 0, \ldots 0)$.

For numerical stability, we require $\alpha = \text{sgn}(x_1)\|\mathbf{x}\|$, thus ensuring that x_1 and α have the same sign. It follows that

$$\|\mathbf{u}\|^2 = 2\alpha(x_1 + \alpha)$$

$$\mathbf{u}'\mathbf{x} = \alpha(x_1 + \alpha)$$

$$\begin{aligned} \gamma &= \frac{2}{\|\mathbf{u}\|^2} \\ &= \frac{1}{\alpha(x_1 + \alpha)} \end{aligned}$$

More generally, choose

$$\mathbf{u}_{(k)} = \begin{bmatrix} 0 \\ . \\ 0 \\ x_k + \alpha_k \\ x_{k+1} \\ . \\ . \\ x_n \end{bmatrix}$$

where

$$\alpha_k = \text{sgn}(x_k)\sqrt{\sum_{i=k}^{n} x_i^2}$$

Then

$$\mathbf{H}_k = \mathbf{I} - \gamma \mathbf{u}_{(k)}\mathbf{u}'_{(k)}, \quad \text{where} \quad \gamma = \frac{2}{\|\mathbf{u}_{(k)}\|^2} = \frac{1}{\alpha(x_k + \alpha)}$$

The sequence of reflections $\mathbf{H}_1$, $\mathbf{H}_2$, ..., with $\mathbf{x}$ chosen as indicated in Figure 1, then achieves the result that is required for the QR reduction.

2.2 Least Squares

Form

$$\mathbf{Q}'\mathbf{X} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{Q}'\mathbf{y} = \begin{bmatrix} \mathbf{f} \\ \mathbf{r} \end{bmatrix}$$

where $\mathbf{R}$ $(p \times p)$ is upper triangular, $\mathbf{0}$ is an $(n-p) \times p$ array of zeros, $\mathbf{f}$ is $p \times 1$ and $\mathbf{r}$ is $(n-p) \times 1$. Then

$$\begin{aligned} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 &= \|\mathbf{Q}'\mathbf{y} - \mathbf{Q}'\mathbf{X}\mathbf{b}\|^2 \\ &= \left\| \begin{matrix} \mathbf{f} - \mathbf{R}\mathbf{b} \\ \mathbf{r} \end{matrix} \right\|^2 \\ &= \|\mathbf{f} - \mathbf{R}\mathbf{b}\|^2 + \|\mathbf{r}\|^2 \end{aligned}$$

Hence $\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$ is minimized by choosing $\mathbf{b}$ such that

$$\mathbf{R}\mathbf{b} = \mathbf{f}$$

Solve first for b_p, then for b_{p-1}, ... b_p, in a procedure called *backward elimination.* The residual sum of squares is $\|\mathbf{r}\|^2$.

2.2.1 Normal equations

We have

$$\mathbf{R}\mathbf{b} = \mathbf{f}$$

Then

$$\mathbf{R}'\mathbf{R}\mathbf{b} = \mathbf{R}'\mathbf{f}$$

i.e.

$$\mathbf{R}'\mathbf{R}\mathbf{b} = \mathbf{R}'\mathbf{f}$$

Observe that

$$\mathbf{R}'\mathbf{f} = \begin{bmatrix} \mathbf{R}' & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{r} \end{bmatrix} = \mathbf{X}'\mathbf{Q}\mathbf{Q}'\mathbf{y} = \mathbf{X}'\mathbf{y}$$

and that

$$\mathbf{R}'\mathbf{R}\mathbf{b} == \mathbf{X}'\mathbf{Q}\mathbf{Q}'\mathbf{X} = \mathbf{X}'\mathbf{X}$$

Hence the so-called normal equations

$$\mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{y}$$

Writing $\mathbf{X}\mathbf{b} = \boldsymbol{\mu}$ (the fitted values), the normal equations can then be written

$$\mathbf{X}'\boldsymbol{\mu} = \mathbf{X}'\mathbf{y}$$

This form of the least squares equations carries over to generalized linear models, in the special case where the canonical link is specified.

2.2.2 Computational issues

1. Algebraically, one can write
$$\mathbf{Q}'\mathbf{X} = \mathbf{H}_p\mathbf{H}_{p-1}\ldots\mathbf{H}_1\mathbf{X}$$
Neither $\mathbf{Q}'$ nor any of the matrices $\mathbf{H}_i$ is ever formed explicitly. For large matrices, this would be an impossibly computationally expensive way to do the computations. In
$$\mathbf{HX} = (\mathbf{I} - \gamma\,\mathbf{uu}')\mathbf{X}$$
it is important to do the calculation as
$$\mathbf{X} - \gamma\,\mathbf{u}(\mathbf{u}'\mathbf{X})$$
Thus for each column $\mathbf{x}_j$ of $\mathbf{X}$, first form $\gamma\,\mathbf{u}'\mathbf{x}_j$, then subtract this multiple of $\mathbf{u}$ from $\mathbf{x}_j$.

2. For calculation of row k of the array resulting from pre-multiplication by $\mathbf{H}_k$, formulae are available that are simplified and numerically more accurate than from direct use of equation 1. A further simplification is to form the diagonal element as $|\alpha_k|$, multiplying remaining elements in the ith row by $\text{sgn}(\alpha_k)$. In the examples shown below, sequences of such modified Householder reflections have been used. We no longer have Householder reflections, but the $\mathbf{H}_i$ are, just as before, orthogonal matrices. The difference is that when α_k is negative, all elements in row k change sign relative to the Householder matrix $\mathbf{H}_k$.

3. If the calculated diagonal element is zero to within machine precision, the easiest recourse is to set all elements in the row to zero. The vector $\mathbf{b}$ is no longer uniquely defined. The residual sum of squares is unique, independent of the action that may be taken to resolve the indeterminacy in the parameters.

4. Let $\mathbf{X}_k$ be the matrix that consists of the first k columns of $\mathbf{X}$. Then the information needed to minimize
$$\|\mathbf{y} - \mathbf{X}_k\mathbf{b}\|^2$$
is available once the first k Householder steps are complete. It is found in the $k\times k$ submatrix of $\mathbf{R}$ and in the first k elements of $\mathbf{Q}'\mathbf{y}$.

2.3 Demonstrations of the computational steps

Example 1

We will start with
$$\mathbf{X} = \begin{bmatrix} 1 & -6 & 0 \\ 1 & -3 & 6 \\ 1 & 6 & 15 \\ 1 & 21 & 9 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -3 \\ 1 \\ 2 \\ 6 \end{bmatrix}$$
The aim is to minimize
$$[-3 - (b_0 - 6b_1)]^2 + [1 - (b_0 - 3b_1 + 6b_2))]^2 + [2 - (b_0 + 6b_1 + 15b_2)]^2 + [6 - (b_0 + 21b_1 + 9b_2)]^2$$
The sequence of sweeps is
$$\begin{bmatrix} 1 & -6 & 0 & -3 \\ 1 & -3 & 6 & 1 \\ 1 & 6 & 15 & 2 \\ 1 & 21 & 9 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{2} & \mathbf{9} & \mathbf{15} & \mathbf{3} \\ 1 & -4 & 1 & 1 \\ 1 & 5 & 10 & 2 \\ 1 & 20 & 4 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{2} & \mathbf{9} & \mathbf{15} & \mathbf{3} \\ 1 & \mathbf{21} & \mathbf{6} & \mathbf{6} \\ 1 & 5 & 9 & 1 \\ 1 & 20 & 0 & 2 \end{bmatrix}$$
The first sweep is equivalent to pre-multiplication by $\mathbf{H}_1$, and the second to to pre-multiplication by $\mathbf{H}_2$. At this point, the element in the (4,3) position is zero, and use of $\mathbf{H}_3$ is not needed. ($\mathbf{H}_3$, if it were formed, would be the identity matrix.)

Notice that, rather than recording below diagonal elements in the ith column as zero once calculations for row i are complete, the values that were there previously have been left in place, albeit printed in small type. It is straightforward to ensure that any computations with the upper triangular matrix treat those elements as zeros. The values that are left in place can be used, should they be required, to reconstruct the sequence of Householder steps.

Use of R

```
X1df <- data.frame(X1=c(-6,-3,6,21), X2=c(0,6,15,9), y=c(-3,1,2,6))
lm(y ~ X1+X2, data=X1df)
summary(lm(y ~ X1+X2, data=X1df))  # Gives more information
anova(lm(y ~ X1+X2, data=X1df))    # Gives different information
lm(y ~ X1+X2, data=X1df)$qr        # qr decomposition
```

Observe that R stores somewhat different information in below diagonal positions.

Alternatively, R has a suite of functions that can be used for the underlying computations of least squares calculations. These include `qr()`, `qr.coef()` and `qr.solve()`. Because they work with matrices rather than data frames, they may be much faster than `lm()` for the handling of large least squares problems. Approaches to the calculations that use these will be demonstrated for the second slightly more substantial example that will now be presented.

Example 2

Table 1 shows the application of a sequence of 4 modified Householder reflections to a 9×5 array

$$(\mathbf{X}, y) \quad = \quad \left[\begin{array}{cccc|r} \text{X0} & \text{X1} & \text{X2} & \text{X3} & \text{y} \\ 1 & 1 & -1 & -14 & -7.7 \\ 1 & -1 & 6 & -2 & 8.5 \\ 1 & 1 & 9 & 8 & 19.6 \\ 1 & -2 & 8 & -3 & 11.4 \\ 1 & 0 & 5 & 1 & 11.4 \\ 1 & 0 & 3 & -1 & 6.8 \\ 1 & 4 & 2 & 9 & 2.5 \\ 1 & 7 & 0 & 8 & -1.6 \\ 1 & 8 & -5 & 3 & -13 \end{array}\right]$$

The numbers in the X-matrix have been chosen so that the values in the R-matrix are integers. The least squares system is for a model in which there is a constant term X0 and explanatory variables X1, X2 and X3.

The least squares estimates are obtained by solving

$$\left[\begin{array}{cccc} \mathbf{3} & \mathbf{6} & \mathbf{9} & \mathbf{3} \\ 0 & \mathbf{10} & \mathbf{-10} & \mathbf{10} \\ 0 & 0 & \mathbf{8} & \mathbf{16} \\ 0 & 0 & 0 & \mathbf{8} \end{array}\right] \mathbf{b} \quad = \quad \left[\begin{array}{r} \mathbf{12.63} \\ \mathbf{-20.04} \\ \mathbf{19.95} \\ \mathbf{4} \end{array}\right]$$

Observe the subsystems for which information is available

$\left[\begin{array}{c} \mathbf{3} \end{array}\right] \quad \mathbf{12.63} \leftarrow$

Solve $3b_0 = 12.63 \quad \Rightarrow \quad b_0 = 4.21$.
SS reduces by 12.63^2 (to CSS[a])

[a] SS = sum of squares; RSS = residual SS; CSS = corrected SS, i.e., RSS about mean

$\left[\begin{array}{cc} \mathbf{3} & \mathbf{6} \\ 0 & \mathbf{10} \end{array}\right] \quad \begin{array}{r} \mathbf{12.63} \\ \mathbf{-20.04} \leftarrow \end{array}$

$10b_1 = -20.04 \quad \Rightarrow \quad b_1 = -2.004$; $3b_0 + 6b_1 = 12.63 \quad \Rightarrow \quad b_0 = 8.22$.
Additional reduction in RSS $= 20.04^2$.

$\left[\begin{array}{ccc} \mathbf{3} & \mathbf{6} & \mathbf{9} \\ 0 & \mathbf{10} & \mathbf{-10} \\ 0 & 0 & \mathbf{8} \end{array}\right] \quad \begin{array}{r} \mathbf{12.63} \\ \mathbf{-20.04} \\ \mathbf{19.95} \leftarrow \end{array}$

$8b_2 = 19.95$; $10b_1 - 10b_2 = -20.04$; $3b_0 + 6b_1 + 9b_2 = 12.63$.
Additional reduction in RSS $= 19.95^2$.

In the full model, the RSS reduces by a further 4^2. The residual sum of squares from the full

$$\begin{bmatrix} 1 & 1 & -1 & -14 \\ 1 & -1 & 6 & -2 \\ 1 & 1 & 9 & 8 \\ 1 & -2 & 8 & -3 \\ 1 & 0 & 5 & 1 \\ 1 & 0 & 3 & -1 \\ 1 & 4 & 2 & 9 \\ 1 & 7 & 0 & 8 \\ 1 & 8 & -5 & 3 \end{bmatrix} \begin{matrix} 7.7 \\ 8.5 \\ 19.6 \\ 11.4 \\ 11.4 \\ 6.8 \\ 2.5 \\ -1.6 \\ -13.0 \end{matrix} \rightarrow \begin{bmatrix} \mathbf{3} & \mathbf{6} & \mathbf{9} & \mathbf{3} \\ 1 & -2.75 & 4 & 0.75 \\ 1 & -0.75 & 7 & 10.75 \\ 1 & -3.75 & 6 & -0.25 \\ 1 & -1.75 & 3 & 3.75 \\ 1 & -1.75 & 1 & 1.75 \\ 1 & 2.25 & 0 & 11.75 \\ 1 & 5.25 & -2 & 10.75 \\ 1 & 6.25 & -7 & 5.75 \end{bmatrix} \begin{matrix} \mathbf{12.63} \\ 7.27 \\ 18.37 \\ 10.17 \\ 10.17 \\ 5.57 \\ 1.27 \\ -2.83 \\ -14.23 \end{matrix}$$

$$\rightarrow \begin{bmatrix} \mathbf{3} & \mathbf{6} & \mathbf{9} & \mathbf{3} \\ 1 & \mathbf{10} & \mathbf{-10} & \mathbf{10} \\ 1 & -0.75 & 6.18 & 11.29 \\ 1 & -3.75 & 1.88 & 2.47 \\ 1 & -1.75 & 1.08 & 5.02 \\ 1 & -1.75 & -0.92 & 3.02 \\ 1 & 2.25 & 2.47 & 10.12 \\ 1 & 5.25 & 3.76 & 6.94 \\ 1 & 6.25 & -0.14 & 1.22 \end{bmatrix} \begin{matrix} \mathbf{12.63} \\ \mathbf{-20.04} \\ 16.76 \\ 2.14 \\ 6.42 \\ 1.82 \\ 6.09 \\ 8.41 \\ -0.85 \end{matrix} \rightarrow \begin{bmatrix} \mathbf{3} & \mathbf{6} & \mathbf{9} & \mathbf{3} \\ 1 & \mathbf{10} & \mathbf{-10} & \mathbf{10} \\ 1 & -0.75 & \mathbf{8} & \mathbf{16} \\ 1 & -3.75 & 1.88 & -1.15 \\ 1 & -1.75 & 1.08 & 2.94 \\ 1 & -1.75 & -0.92 & 4.79 \\ 1 & 2.25 & 2.47 & 5.36 \\ 1 & 5.25 & 3.76 & -0.31 \\ 1 & 6.25 & -0.14 & 1.48 \end{bmatrix} \begin{matrix} \mathbf{12.63} \\ \mathbf{-20.04} \\ \mathbf{19.95} \\ -2.74 \\ 3.63 \\ 4.21 \\ -0.31 \\ -1.34 \\ -0.49 \end{matrix}$$

$$\rightarrow \begin{bmatrix} \mathbf{3} & \mathbf{6} & \mathbf{9} & \mathbf{3} \\ 1 & \mathbf{10} & \mathbf{-10} & \mathbf{10} \\ 1 & -0.75 & \mathbf{8} & \mathbf{16} \\ 1 & -3.75 & 1.88 & \mathbf{8} \\ 1 & -1.75 & 1.08 & 2.94 \\ 1 & -1.75 & -0.92 & 4.79 \\ 1 & 2.25 & 2.47 & 5.36 \\ 1 & 5.25 & 3.76 & -0.31 \\ 1 & 6.25 & -0.14 & 1.48 \end{bmatrix} \begin{matrix} \mathbf{12.63} \\ \mathbf{-20.04} \\ \mathbf{19.95} \\ \mathbf{4} \\ 1.46 \\ 0.68 \\ -4.26 \\ -1.11 \\ -1.58 \end{matrix}$$

Table 1: Reduction of a 9×4 array to upper triangular form, with the same operations applied to the fifth column $\mathbf{y}$. Four modified Householder reflections are applied to the 9×5 array $(\mathbf{X}, \mathbf{y})$.

model is

$$1.46^2 + 0.68^2 + 4.26^2 + 1.11^2 + 1.58^2 = 24.46$$

(Note that 24.46 is the result of rounding the RSS when it is calculated to full machine accuracy.)

Use of R

```
X2df <- data.frame(X1=c(1, -1, 1, -2, 0, 0, 4, 7, 8),
           X2=c(-1, 6, 9, 8, 5, 3, 2, 0, -5),
           X3=c(-14, -2, 8, -3, 1, -1, 9, 8, 3),
           y=c(-7.7, 8.5, 19.6, 11.4, 11.4, 6.8, 2.5, -1.6, -13))
lm(y ~ X1+X2+X3, data=X2df)
summary(lm(y ~ X1+X2+X3, data=X2df))  # Gives more information
anova(lm(y ~ X1+X2+X3, data=X2df))    # Gives different information
lm(y ~ X1+X2+X3, data=X2df)$qr        # qr decomposition
```

Alternatives to `lm()`

For computationally intensive least squares calculations, consider the use of `qr()` and associated functions. For use of this approach, the user must first extract or construct the design matrix. The following demonstrates the use of `qr()` and allied functions:

```
X <- with(X2df, model.matrix(~ X1+X2+X3))
## X <- cbind(rep(1,9), X2df[, 1:3])  # More efficient alternative
QR <- qr(X)
qr.coef(QR, X2df$y)
qr.resid(QR, X2df$y)
## etc, etc
```

Calculations may be computationally intensive because a major component of the calculation is repeated a large number of times, and/or because they involve one or more large design matrices.

2.4 The Analysis of Variance Table

We show the sequential analysis of variance table that is relevant to Example 2 in Subsection 2.3. From Table 1, we see that

$$\mathbf{Q}'\mathbf{y} = \begin{bmatrix} \mathbf{12.63} \\ \mathbf{-20.04} \\ \mathbf{19.95} \\ \mathbf{4} \\ 1.46 \\ 0.68 \\ -4.26 \\ -1.11 \\ -1.58 \end{bmatrix}$$

The usual form of sequential analysis of variance table, which partitions the sum of squares about the mean, is obtained by picking off the elements of $\mathbf{Q}'\mathbf{y}$ in turn:

Term	Sum of squares	DF
X1	20.04^2	1
X2	19.95^2	1
X3	4^2	1
Residual	24.46	$n-p$

Compare this with

```
X2df.lm <- lm(y ~ X1+X2+X3, data=X2df)
anova(X2df.lm)
```

Now observe how this table can be obtained from the output of `qr()` and friends.

```
## Now extract the table from the successive components of Qy
QR <- qr(cbind(rep(1,9), as.matrix(X2df[,1:3])))
Qty <- qr.qty(QR, X2df$y)
duetoSS <- Qty[2:4]^2   # Why is Qty[1] omitted?
rss <- sum(Qty[-(1:4)]^2)
aovtab <- data.frame(row.names=c("X1", "X2", "X3", "Residual"),
  Df=c(rep(1,3), length(Qty)-4), SS=c(duetoSS, rss))
aovtab
```

It is important to note that this is a sequential analysis of variance table. In general the contribution of each term depends on which (if any) term(s) precede it in the model.

2.5 Weighted Least Squares

If var[$\mathbf{y}$] = $\mathbf{V}$ is known to within the scale factor σ^2, then a linear transformation of the vector $\mathbf{y}$ can be determined such that elements of the linearly transformed vector are iid.

A suitable linear transformation is conveniently obtained from the Cholesky decomposition of $\mathbf{V}$, by which

$$\mathbf{V} = \mathbf{L}\mathbf{L}'$$

Then

$$\begin{aligned} \text{var}[\mathbf{L}^{-1}\mathbf{y}] &= \mathbf{L}^{-1}\text{var}[\mathbf{y}]\mathbf{L}'^{-1} \\ &= \mathbf{L}^{-1}\mathbf{V}\mathbf{L}'^{-1} \\ &= \mathbf{I}_n\sigma^2 \end{aligned}$$

Then we minimize

$$\|\mathbf{L}^{-1}(\mathbf{y} - \mathbf{X}\mathbf{b})\|^2$$

For this, replace $\mathbf{y}$ by $\mathbf{y}^* = \mathbf{L}^{-1}\mathbf{y}$, and $\mathbf{X}$ by $\mathbf{X}^* = \mathbf{L}^{-1}\mathbf{X}$, and proceed as before.

The normal equations become

$$\begin{aligned} \mathbf{X}'\mathbf{W}\mathbf{X}\mathbf{b} &= \mathbf{X}'\mathbf{W}\mathbf{y} \\ \text{i.e.,}\mathbf{X}^{*\prime}\mathbf{X}^*\mathbf{b} &= \mathbf{X}^{*\prime}\mathbf{y}^* \end{aligned}$$

where $\mathbf{W} = \mathbf{V}^{-1}$

Then setting $\mathbf{X}\mathbf{b} = \boldsymbol{\mu}$ (the fitted values), this becomes

$$\mathbf{X}'\mathbf{W}\boldsymbol{\mu} = \mathbf{X}'\mathbf{W}\mathbf{y}$$

This is form of the equations that carries across to Generalized Linear Models.

2.6 Unbalance – implications for addition or deletion of model terms

In data with suitable balance coefficients do not change when terms are added to, or dropped from, the model. This is reflected in the **R**-matrix. It is insightful to observe how the **R**-matrix (and regression coefficients) change when balanced data are modified, by use of unequal weights or by dropping observations.

```
> pain <- data.frame(vasScore=c(0.67, 0.05, 3.67, 3.13),
+                    trt=factor(rep(c("placebo", "baclofen"), 2),
                               levels=c("placebo", "baclofen")),
+                    gender=factor(rep(c("male", "female"), c(2,2)),
                                  levels=c("male", "female")),
+                    number=c(3, 9, 15, 7))
> pain
   vasScore      trt gender number
1      0.67  placebo   male      9
2      0.05 baclofen   male      3
3      3.67  placebo female      7
4      3.13 baclofen female     15

> pain.lm <- lm(vasScore ~ gender + trt, data=pain)
> round(coef(pain.lm), 2)
 (Intercept) genderfemale  trtbaclofen
        0.65         3.04        -0.58
```

Now omit consideration of the Gender effect:

```
> pain1.lm <- lm(vasScore ~ trt, data=pain)
> round(coef(pain1.lm), 2)
(Intercept) trtbaclofen
       2.17       -0.58
```

Observe that the estimate of the treatment effect is unchanged.

Weighted analysis

```
> pain.wlm <- lm(vasScore ~ gender + trt, weight=number, data=pain)
> round(coef(pain.wlm), 2)
 (Intercept) genderfemale  trtbaclofen
        0.66         3.03        -0.57
```

Observe that the estimate of the treatment effect has hardly changed. In general, the change may not be so small, but will not reverse an effect that goes in the same direction for the genders separarately.

Now omit consideration of the gender effect:

```
> pain.wlm1 <- lm(vasScore ~ trt, weight=number, data=pain)
> round(coef(pain.wlm1), 2)
(Intercept) trtbaclofen
       1.98        0.63
```

Observe that the treatment effect now goes in the other direction. The combination of unequal weights and omission of a relevant factor generates this misleading result.

Implications for the R-matrix

First, form the matrix on which the calculations will be performed.

```
> Xy <- cbind(model.matrix(~ gender + trt, data=pain),
              vasScore = pain$vasScore)
> Xy
  (Intercept) genderfemale trtbaclofen vasScore
1           1            0           0     0.67
2           1            0           1     0.05
3           1            1           0     3.67
4           1            1           1     3.13
```

Now use my function `house()` for the calculation. This modifies the Householder reflections so that diagonal elements are positive. Relative to the result from use of Householder reflections all elements in rows with a negative diagonal element are multiplied by -1.

```
> house(Xy, showzeros=3)
  (Intercept) genderfemale trtbaclofen vasScore
1           2            1           1     3.76
2           0            1           0     3.04
3           0            0           1    -0.58
4           0            0           0     0.04
```

To obtain estimates of the model parameters, solve:

$$\mathbf{X} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{b} = \begin{bmatrix} 3.76 \\ 3.04 \\ -0.58 \end{bmatrix}$$

Here are the calculations for the weighted analysis:

```
> house(Xy*sqrt(pain$number), showzeros=3)
  (Intercept) genderfemale trtbaclofen    vasScore
1    5.830952     3.772969    2.743977 13.62041763
2    0.000000     2.786522   -1.203271  9.17652419
3    0.000000     0.000000    2.650043 -1.49894659
4    0.000000     0.000000    0.000000  0.09892627
```

Now solve (values are rounded to three decimal places):

$$\begin{bmatrix} 5.831 & 3.773 & 2.744 \\ 0 & 2.787 & -1.203 \\ 0 & 0 & 2.650 \end{bmatrix} \mathbf{b} = \begin{bmatrix} 13.620 \\ 9.177 \\ -1.499 \end{bmatrix}$$

A key difference is that the R matrix no longer has a zero in the (2, 3) position. The consequence is that the estimate of the treatment effect changes (and changes in sign) if there is no allowance for Innings.

Further examples

An interesting experiment is to observe how dropping a few rows from a balanced design affects the R matrix and hence the parameter estimates. For example, try the following:

```
house(model.matrix(lm(headwt ~ Cult + Date, data=cabbages))
xtabs(~ Cult + Date, data=cabbages[11:14, ])
house(model.matrix(lm(HeadWt ~ Cult + Date, data=cabbages[-(11:14),])),
      showzeros=4)[1:4,]
  ## The (2,3) and (2,4) positions are not now occupied by zeros.
## Drop out row 11
house(model.matrix(lm(HeadWt ~ Cult + Date, data=cabbages[-(11),])),
      showzeros=4)[1:4,]
```

3 Model Assumptions and Model Choice

This section will discuss the the classical theory, and note implications for model choice. Finally, recourses will be noted that can be used when the classical theory fails or is in doubt.

The classical theory that will now be described assumes that conditional on $\mathbf{X}$

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_n\sigma^2) \tag{1}$$

3.1 Limitations of the classical theory

According to the strict requirements of the theory, the model must be known in advance. Neither data based choice of transformation(s) nor variable selection are permitted. It is however permissible to divide the data set into two parts – a training set that is used to develop the model, and a test set that is used to derive the eventual model fit. Or three parts may be required – a training set, a holdout set on which which performance is optimized in order to tune the model, and a test set that is used for the eventual model fit. An objection is that this makes poor use of the data, which may be a serious issue for smallish data sets. Matters can be improved somewhat by repeating the process, with the roles of training, test and any holdout set interchanged.

With respect to Equation 1, note that:

- Depending on how the fitted model will be used, the normality assumption is commonly not of critical importance; approximate normality is enough. Independence is more crucial, and less open to checking. If the form of departure from independence is known or can be guessed, the dependence can be modelled. This leads into areas (time series modeling, multi-level models, repeated measures, etc.) whose details are beyond the scope of the present course.

- In practice limited use of plots or other mechanisms to determine appropriate transformations may be essential, to get a model fit that does not do violence to the data. Depending on the details of the specific model and associated data, this may not not seriously invalidate assumptions. Limited variable selection, e.g., choose two explanatory variables out of four, may be similarly acceptable. Again, much will depend on the details of the model and associated data. Selection effects are in general a more serious problem than effects from limited use of data-based transformations.
- Where one model emerges as very substantially superior to other models considered, selection effects are not an issue. Why?

When the model is fitted to the data used to select the model from a set of possible models, the effect is anti-conservative. Thus, standard errors will be smaller than indicated by the theory, and coefficients and t-statistics larger. Such anti-conservative estimates of standard errors and other statistics may, unless the bias is huge, nevertheless provide the useful guidance.

Almost inevitably, none of the models on offer will be strictly correct. Mis-specification of the fixed effects, and to a lesser extent of the random effects, is likely to bias model estimates, at the same time inflating the error variance or variances, i.e., it may to some extent work in the opposite direction to selection effects.

Recourses that are available when the classical theory fails will be discussed below.

3.1.1 Distributional Results – the classical theory

Observe that

$$\text{var}[\mathbf{Qy}] = \mathbf{Q}'\mathbf{I}_n\mathbf{Q}\sigma^2 = \mathbf{I}_n\sigma^2$$

Then

$$\text{E}\begin{bmatrix}\mathbf{f}\\ \mathbf{r}\end{bmatrix} = \text{E}[\mathbf{Q}'\mathbf{y}] = \mathbf{Q}'\mathbf{X}\boldsymbol{\beta} = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\boldsymbol{\beta}$$

Thus,

$$\mathbf{f} \sim N(\mathbf{R}\boldsymbol{\beta}, \mathbf{I}_p\sigma^2),\ \mathbf{r} \sim N(\mathbf{0}, \mathbf{I}_{n-p}\sigma^2)$$

Moreover $\|\mathbf{f}\|^2$ and $\|\mathbf{r}\|^2$ are each sums of independent $\sigma^2\chi^2_1$ terms and therefore independent.

Finally $\mathbf{b} = \mathbf{R}^{-1}\mathbf{f}$ is distributed as

$$N(\boldsymbol{\beta}, \mathbf{R}^{-1}\mathbf{R}'^{-1}\sigma^2), \quad \text{i.e., as } N(\boldsymbol{\beta}, (\mathbf{X}'\mathbf{X})^{-1}\sigma^2)$$

As we will use an estimate of σ^2 that has $n-p$ degrees of freedom, individual elements of $\mathbf{b}$ are distributed as t_{n-p}. Elements b_i of $\mathbf{b}$ are independent if and only if $\mathbf{X}'\mathbf{X}$ is diagonal.

3.2 Summary of the classical theory

Models with iid errors

- $\mathbf{y}$ (n by 1) is a vector of observed values, $\mathbf{X}$ (n by p) is model matrix, and $\boldsymbol{\beta}$ (p by 1) is a vector of coefficients.
- The model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, i.e. $\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i$ where the vector $\boldsymbol{\epsilon}$ of residuals is n by 1
- Least squares *normal* equations are

  $$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}$$

  (assuming $\boldsymbol{\epsilon}_i$ are iid normal, these are the maximum likelihood estimates)
- If variances are unequal, modify the *normal* equations to

  $$\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{y}$$

  where $\mathbf{W}$ is a diagonal matrix with elements equal to the inverses of the variances (justification is from maximum likelihood, or argue that leverage should be independent of variance)
- Assume $[\text{E}[\mathbf{y}] = \boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$, i.e. $\text{E}[\boldsymbol{\epsilon}] = \mathbf{0}$.

General known variance-covariance matrix

More generally, if $\boldsymbol{\epsilon}$ is multivariate normal with known variance-covariance matrix $\boldsymbol{\Sigma}$, then ML theory gives the equation as above with $\mathbf{W} = \boldsymbol{\Sigma}^{-1}$.

Two values with a high positive correlation contain, jointly, less information than two independent values. Consider an extreme case; if the correlation is 1, they duplicate the same information.

If the variance-covariance matrix $\boldsymbol{\Sigma}$ is not known, many different special methods are available for various special cases that occur in pracitce. Models that may be relevant include time-series models, spatial analysis and multi-level models.

Limitations

Standard errors, t- and F-statristics may be optimistic, because of model selection. Almost inevitably, none of the models on offer will be strictly correct. Mis-specification of the fixed effects, and to a lesser extent of the random effects, is likely to bias model estimates, at the same time inflating the error variance or variances.

4 Regression splines

See Maindonald & Braun (2007, Section 7.5, pp. 234–238). See Ruppert et al (2003) for a more complete account of regression splines.

We will use cubic splines, i.e., cubic polynomial curves are joined at internal *knots*, with the requirement that the slopes and the second derivative must be continuous over the whole ranges of values. Specifically, the slope and the second derivative must agree at the knots. Cubic splines are the most commonly used form of spline, and have "nice" theoretical properties. Natural splines have the (often sensible) further constraint that the second derivative is zero at the boundary points, or *boundary knots*. Boundary knots are commonly taken to coincide with the extremes of the data, but this is not necessary. It turns out that such curves can be expressed as a linear combination of basis functions.

For our purposes, spline basis terms are a kitset for constructing a flexible variety of curves. As the number of degrees of freedom increases, it becomes possible to accommodate an ever increasing variety of curves. We will use natural splines, generated by the function `ns()` in R's `spline()` package.

A good way to get a feel for spline functions is to plot the basis functions. These depend on the x-values that are chosen. We will examine (Figure 3) the spline basis functions when x-values range from 1 to 50 in increments of one.

```
## Generate the matrix of values of the basis functions
x <- 1:50
Xns <- ns(x, df=5)
```

Observe the attributes `knots` and `Boundary.knots`

```
> attributes(Xns)$knots
 20%  40%  60%  80%
10.8 20.6 30.4 40.2
> attributes(Xns)$Boundary.knots
[1]  1 50
```

Code that will create the plots is

```
## Create the plots
plot(1:50, Xns[,1], ylim=range(Xns), pch="1")
points(1:50, Xns[,2], pch="2", col="red")
points(1:50, Xns[,3], pch="3", col="green")
points(1:50, Xns[,4], pch="4", col="blue")
points(1:50, Xns[,5], pch="5", col="magenta")
```

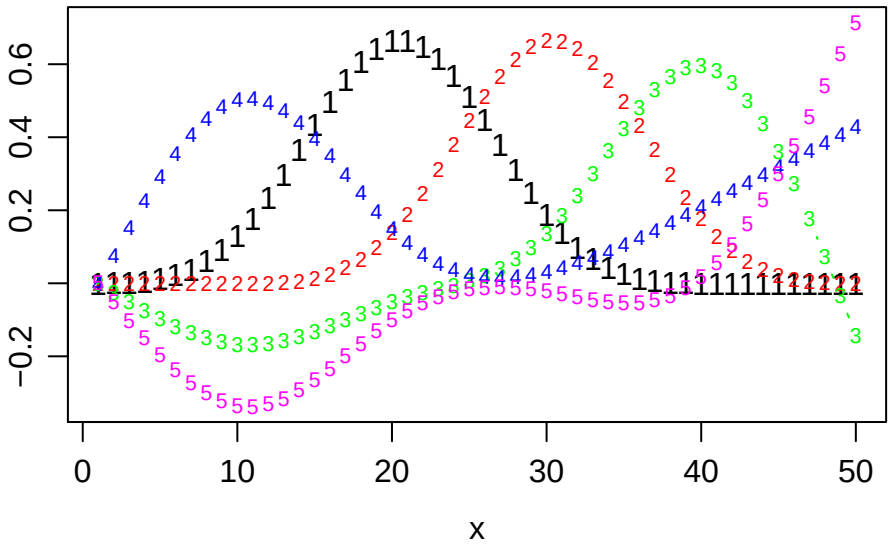

Figure 3: Spline basis functions, with `x = 1:50`. Basis function 1 is plotted with 1's, and so on.

The figure shows the first five curves in the kitset (all that are available with `df=5`), when x ranges from 1 to 50 in increments of 1.

4.0.1 How many degrees of freedom?

A quadratic (hill or valley-shaped curve) has one degree of freedom for the intercept, plus one degree of freedom for the slope, plus one degree of freedom for the curvature. Every additional point where the second derivative is zero (the slope instantaneously ceases changing) accounts for one additional degree of freedom.

Consider the following sine curve

```
x <- seq(from=0, to = 6*pi, length=50)
y <- sin(x)
plot(x,y)
```

The first "hill" accounts for two degrees of freedom. There are five additional valleys/hills, accounting in total for seven degrees of freedom, additional to the intercept. Hence we try

```
lines(x, fitted(lm(y~ns(x,7))), col="red")
```

Any fewer degrees of freedom is grossly inadequate. The choice `df=6` shows how bad the spline fit can be when there are not enough degrees of freedom to capture the major features of the data. The choice `df=8` is an obviously worthwhile improvement on `df=7`. Figure 4 provides a comparison.

Note If B-splines are used (function `bs()`), two extra degrees of freedom might seem required to capture the broad shape of the curve. There is however a choice in whether the available degrees of freedom are used to give freedom to change shape near the boundary knots, or whether to fit the necessary number of hills and valleys. The least squares algorithm uses that discretion to good effect.

Regression splines – an example

The `covsample` dataset that is in the *DAAGxtras* package gives forest cover type (one of eight types) as a function of various environmental attributes. There is no information on geographical coordinates. However, it might be expected that there would be an ordering in

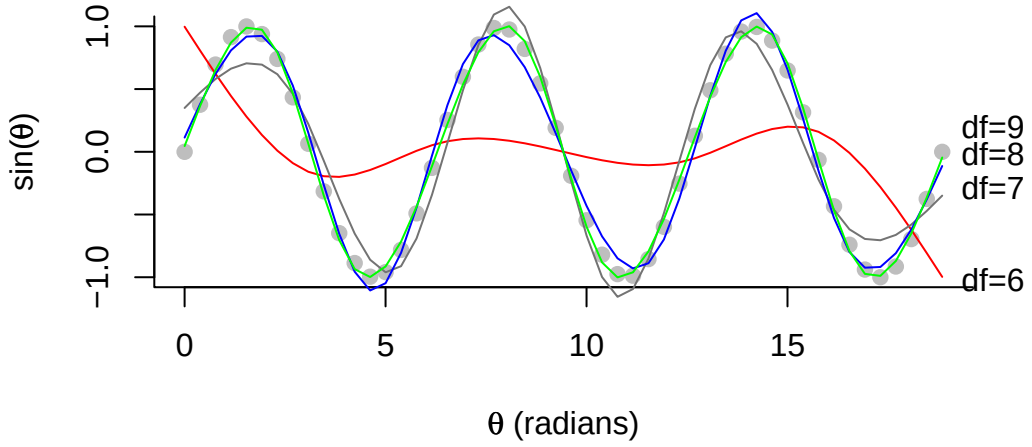

Figure 4: Regression spline fit to points that are determined by a sine curve, over the range 0 to 6π. A 6 d.f. natural spline curve is clearly grossly inadequate. A 7 d.f. curve gets the broad shape more or less correct. An 8 d.f. curve is clearly preferable. The improvement with a 9 d.f. curve is more marginal.

the data, reflecting some kind of geographical ordering. If so, this is likely to be reflected in a systematic variation in the environmental attributes. The first 10 of there are continuous variables. These are: Elevation, Aspect, Slope, Horizontal_Distance_To_Hydrology, Vertical_Distance_To_Hydrology, Horizontal_Distance_To_Roadways, Hillshade_9am, Hillshade_Noon, Hillshade_3pm, Horizontal_Distance_To_Fire_Points.

We will investigate whether the first of these (Elevation) seems to vary systematically through the data. We create a variable `dist` that measures distance through the data.

```
covdf <- cbind(covsample[, 1:10], dist=(1:11318-0.5)/11318)
## First, investigate use of lowess()
with(covdf, plot(lowess(dist, V1)))  ## too smooth
with(covdf, plot(lowess(dist, V1, f=0.1)))  ## less smooth
with(covdf, plot(lowess(dist, V1, f=0.01))) ## reasonable?
```

Now, use regression splines. The final plot from the `lowess` has 11 or 12 features that might be characterised as hills or valleys or plateaus. We will try 15 d.f.

```
hat <- fitted(lm(V1 ~ ns(dist,15), data=covdf))
## Overplot on the earlier lowess smooth
lines(hat ~ dist, data=covdf, col="red")
```

This seems not quite adequate. It does rather poorly at approximating one of the major features, as identified by the `lowess` curve. Use of a much higher number of degrees of freedom makes little difference, however. The difference is probably that `lowess()` gives a robust smooth. This can be checked by using `rlm()` (a robust version of `lm()`) for the fit. We will be generous in the number of degrees of freedom that are allowed.

```
with(covdf, plot(lowess(dist, V1, f=0.01)))
hat <- fitted(rlm(V1 ~ bs(dist,40), data=covdf))
lines(hat ~ dist, data=covdf, col="red")
## Compare with use of lm() with 40 d.f.
```

```
hat <- fitted(lm(V1 ~ bs(dist,40), data=covdf))
lines(hat ~ dist, data=covdf, col="blue")
```

5 Comparing Models – Resampling Approaches

Here our interest will be in comparing residual sums of squares for models that are not necessarily nested. The methodology will be demonstrated using the `fruitohms` data in the *DAAG* package.

Resampling methods can be a useful recourse when theoretical results are unavailable, or when the accuracy of asymptotic results cannot be guaranteed. They allow a removal or weakening of normality assumptions. While a recourse that is often useful, they are not the answer to every problem of failure of assumptions.

Three methods will be noted:

1. Permutation methods locate a fitted model statistic (e.g., the residual mean square) within the distribution that is obtained when observed values are randomly permuted. If the residual mean square for the fitted model is in the upper tail of this distribution, this is taken as evidence that the model has some predictive power. Another possibility is to work with permuted residuals, thus estimating a permutation distribution for model coefficients.

2. In cross-validation, observations are split into k parts. For each of $i = 1, 2, \ldots, k$, the ith part is left aside for use in testing, the model is fitted to the remaining $k - 1$ parts, and predictions made for the ith part. This yields predictions for all observations that have been derived independently of those observations. An accuracy measure can then be computed.

3. The idea of the bootstrap is to regard the sample as a microcosm of the population. Repeated with replacement resamples are taken, and the model fitted to each resample, yielding a sampling distribution of parameter estimates. There are many variations on this simple scheme.

5.1 Notation & Strategy

For this initial discussion, denote the models as M1 and M2. We first fit the model M1 to the original data, then obtaining fitted values and residuals. Also, we fit M2, and note the change as measured by a statistic $\tilde{F}_{12}$ that is calculated just as for the usual F-statistic.

If M2 is not giving an improvement, then $\tilde{F}_{12}$ will differ only by statistical error from the F_{12} observed when a random set of residuals are added. It will, in effect, be a random sample from the distribution obtained when repeated random sets of residuals are added on to the fitted values, and F_{12} is calculated for each such new set of "observations". We can then locate $\tilde{F}_{12}$ within this distribution. If $\tilde{F}_{12}$ falls above its 95th percentile then we will judge that, at the 5% significance level, M2 improves on M1. The change when the residuals are correctly attached is, on this criterion, more than statistical variation.

With slight modification, the argument applies if the resdiduals are obtained by permuting the original residuals. If model M2 is not improving on M1, then the change when new "observations" are derived by adding randomly permuted residuals back onto the fitted values will be attributable to statistical variation.

It also applies, again in slightly modified from, if repeated boostrap samples of the original residuals are used. Note that we are not, here, deriving bootstrap estimates of some parameter. While there are theoretical issues even with this use of the bootstrap, we do not have the issues of bias that commonly arise when the bootstrap is used to estimate variance-like statistics.

Note that the three distributions contemplated above are different. There will, accordingly, be differences of power between the three approaches.

The methodology can be varied or extended in various ways:

- We can choose M2 if it appears, on average, to improve on M1, i.e., choose M2 if the p-value is less than 0.5.

- This same style of approach can be used if M2 is selected from a wider class of models. The model selection step must then be repeated for each new bootstrap or permutation sample.

Code

Here are functions that can handle the calculations:

```
‘funF‘ <-
  function(mmat1, mmat2, y,
           showstats=FALSE, divide=10^6){
    ## mmat1 & mmat2 must be model matrices
    M1 <- mmat1
    M2 <- mmat2
    n <- dim(M1)[1]
    qrM1 <- qr(M1)
    qrM2 <- qr(M2)
    ss <- c(sum(qr.resid(qrM1, y)^2),
            sum(qr.resid(qrM2, y)^2))
    df <- dim(mmat1)[1] - c(qrM1$rank, qrM2$rank)
    ssd <- ss[1] - ss[2]
    F12 <- ssd/ss[2]*df[2]    # F-like statistic
    if(showstats){
      print(paste("Estimates of sigma^2 (Xply by ", divide, ")", sep=""))
      names(ss) <- paste(df, "df", sep="")
      print(ss/df/divide)
      print(paste("'F-statistic' = ", round(F12,4),
                  " (df=", df[1]-df[2], " & ", df[2],")", sep=""))
    }
    invisible(F12)
  }
```

First try running `funF()` with the initial data:

```
> M1.lm <- lm(ohms ~ poly(juice, 3), data=fruitohms)
> M2.lm <- lm(ohms ~ ns(juice, 4), data=fruitohms)
> M1 <- model.matrix(M1.lm)
> M2 <- model.matrix(M2.lm)
> funF(M1, M2, y=fruitohms$ohms,showstats=TRUE)
[1] "Estimates of sigma^2 (Xply by 1e+06)"
    124df     123df
0.9723468 0.9343716
[1] "'F-statistic' = 6.0397 (df=1 & 123)"
```

The function `bootF` now shown can be used for the calculations. The default is to use bootstrap samples of the residuals (`type="ordinary"`). Other options are `type="permutation"` and `type="parametric"`.

```
‘bootF‘ <-
  function(data=fruitohms, statistic=funF, R=999,
           form1= ohms ~ poly(juice, 3), form2= ohms ~ ns(juice,4),
           type=c("ordinary", "parametric", "permutation")){
    ## By default, type[1]="ordinary" is used.
    ## Alternatives are type="parametric" or type="permutation"
    M1mod <- lm(form1, data=data)
    sigma <- summary(M1mod)$sigma
    n <- dim(data)[1]
    M1 <- model.matrix(M1mod)
    M2 <- model.matrix(form2, data=data)
    M1fit <- fitted(M1mod)
    M1resid <- resid(M1mod)
    R2 <- R+1
```

```
    y <- M1fit+M1resid
    boot.out <- numeric(R2)
    boot.out[1] <- funF(mmat1=M1, mmat2=M2, y=y, showstats=TRUE)
    for(i in 2:R2){
      if(type[1] =="permutation")
        index <- resid <- M1resid[sample(1:n)] else
      if(type[1] ==
         "ordinary") resid <- M1resid[sample(1:n, replace=TRUE)] else
      if(type[1] =="parametric") resid <- rnorm(n, sd=sigma)
      y <- M1fit+resid
      boot.out[i] <- statistic(mmat1=M1, mmat2=M2, y=y)
    }
    testval <- boot.out[1]
    pval <- sum(boot.out >= testval)/R2
    print(c("p-value" = round(pval,5)))
    invisible(boot.out)
  }
```

5.2 Examples – Bootstrap, parametric bootstrap and permutation approaches

Bootstrap samples of the residuals

We will now test the use of ordinary bootstrap samples in a situation where we pretty much know the answer. For this purpose, we take the models to be `ohms$poly(juice,2)` and `ohms$poly(juice,3)`, so that the models are nested.

```
> poly45stats <- bootF(form1=ohms ~ poly(juice,4), form2 = ohms ~ poly(juice,5))
[1] "Estimates of sigma^2 (Xply by 1e+06)"
    123df     122df
0.8878564 0.8859704
[1] "'F-statistic' = 1.2618 (df=1 & 122)"
p-value
  0.263
```

Here, this statistic is an analysis of variance F-statistic. Thus, we may, provided iid normality assumptions are acceptable, refer it to the relevant theoretical F-distribution. This gives a p-value that is essentially the same as the bootstrap distribution p-value.

```
> 1-pf(1.2618, 1,122)
[1] 0.2635162
```

Saving the values in poly45stats allows us, if we wish, to examine other percentiles of the bootstrap distribution.

For the comparison between `poly(juice,3)` and `ns(juice, 4)` we have:

```
> poly3ns4ord <- bootF(form1=ohms ~ poly(juice,3), form2 = ohms ~ ns(juice, 4))
[1] "Estimates of sigma^2 (Xply by 1e+06)"
    124df     123df
0.9723468 0.9343716
[1] "'F-statistic' = 6.0397 (df=1 & 123)"
p-value
  0.015
```

The parametric bootstrap

This is not, strictly, a bootstrap. Rather it is a simulation that is based on a theoretical distribution. Here, the residuals are sampled from the theoretical normal, with the standard deviation taken to be the square root of the mean residual sum of squares from fitting the model M1.

```
> poly3ns4sim <- bootF(form1=ohms ~ poly(juice,3),
+                      form2 = ohms ~ ns(juice, 4), type="parametric")
[1] "Estimates of sigma^2 (Xply by 1e+06)"
    124df     123df
0.9723468 0.9343716
[1] "'F-statistic' = 6.0397 (df=1 & 123)"
p-value
  0.011
```

The permutation distribution

```
> poly3ns4perm <- bootF(form1=ohms ~ poly(juice,3),
+                       form2 = ohms ~ ns(juice, 4), type="permutation")
[1] "Estimates of sigma^2 (Xply by 1e+06)"
    124df     123df
0.9723468 0.9343716
[1] "'F-statistic' = 6.0397 (df=1 & 123)"
p-value
  0.021
```

The permutation distribution is widely useful in contexts where the asymptotic or other theory is in doubt, and where the null hypothesis implies that permuting the y-values, or permuting the values of an explanatory variable, should on average not affect the model's fitted values. It can be useful in the fitting of logistic regression models.

5.3 References to worked examples

For examples of the use of resampling methods, see Maindonald & Braun (2007) thus:

p.90: Brief general comments;

pp.129–134 (Section 4.7): Permutation methods & the bootstrap;

pp.159–164 (Section 5.5): Cross-validation & the bootstrap, in straight line regression;

pp.257–258 (Subsection 8.2.3): Cross-validation, in logistic regression;

pp.381–383 (Subsections 12.1.2): Use of the bootstrap to check the stability of a principal components plot;

pp.386–388 (Subsections 12.2.2 & 12.2.3): Cross-validation, in discriminant analysis.

pp.400–403 (Subsection 12.3.3): Use of cross-validation in variable selection for discriminant analysis.

6 Generalized Linear Models (GLMs)

The main case of interest will be logistic regression models, where the outcome is binary, i.e. 0 or 1. There will be some discussion of the general theory, but mainly for its relevance to this special case.

For examples of logistic regression models, see Maindonald & Braun (2007, Sections 8.1 & 8.2). Suitable texts for further reading and reference, in increasing order of technical demands, are Faraway (2006); Wood (2006); McCullagh and Nelder (1989).

In principle, models with a 0/1 outcome can be fitted using least squares. If unweighted least squares is used, and the variance really is that for a Binomial(n, p) distribution with $n = 1$, then estimates will be inefficient, though the effect will be of little consequence if fitted proportions lie between perhaps 0.25 and 0.75. Some fitted values may be less than 0 or greater than 1. Where

a continuous explanatory variable has a non-zero coefficient, extrapolation to suitably small or suitably large values will always yield predicted proportions that are outside the range (0,1).

The efficiency of least squares estimates can be improved by taking the fitted proportions from an initial least squares fit, and using these to determine weights for a second fit. This is a step on the way to using a maximum likelihood fit for a generalized linear model. The remedy for preventing silly predicted values is to fit a linear model on the scale of a suitably transformed predicted value, which is exactly what GLMs are designed to do.

It will be interesting to compare logistic regression fits with least squares fits, for roughly equivalent models, to see what difference it makes.

6.1 Brief summary of theory

- As before, we have $\boldsymbol{\mu} = \mathrm{E}[\mathbf{y}]$ (n by 1), $\mathbf{X}$ (n by p), $\boldsymbol{\beta}$ (p by 1), and $\boldsymbol{\epsilon}$ (n by 1).
- The model is now
$$f(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta}, \quad \text{where } \mathrm{E}[\mathbf{y}] = \boldsymbol{\mu}$$
Here, $f()$, which must be monotonic, has the name *link* function. For example,
$$f(\boldsymbol{\mu}_i) = \log(\frac{\boldsymbol{\mu}_i}{N_i - \boldsymbol{\mu}_i})$$
- The distribution of y_i is a function of the predicted value μ_i, independently for different observations. The distributions thus cannot be described as identical.
- An extension is to the quasi-exponential family, where the variance is a constant multiple of an exponential family variance. The multiplying constant is estimated as part of the analysis.
- Commonly used distributions are the normal, binomial and Poisson. Applications for models with quasibinomial and quasipoisson errors may be more extensive than for their exponential family counterparts.
- GLMs with binomial errors are formally equivalant to discriminant models where there are two categories. The GLM framework has advantages for some problems.
- Output is in much the same form as for the `lm` models. There are additional subtleties of interpretation – a `z value` is not a t-statistic, though for some GLMs that yield `z value`s there are specific circumstances where it is reasonable to treat them `z value`s as t-statistics. [More technically, they are Wald statistics.]

GLMs – commentary on the theory

Maximum likelihood parameter estimates

- Recall that the equation is
$$f(\boldsymbol{\mu}) = E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$$
where $\boldsymbol{\mu} = E[\mathbf{y}]$
- Assuming a distribution from the exponential family, the maximum likelihood estimates of the parameters are given by
$$\mathbf{X}'\mathbf{W}\boldsymbol{\mu} = \mathbf{X}'\mathbf{W}\mathbf{y}$$
where $f(\boldsymbol{\mu}_i) = \mathbf{X}_i\boldsymbol{\beta}$, and $\mathbf{W}$ is a diagonal matrix.
- Note that the (diagonal) elements $\mathbf{W}_{ii}$ of $\mathbf{W}$ are functions both of $\mathrm{var}[\mathbf{y}_i]$ and of $f(\boldsymbol{\mu}_i)$

Adequacy of theoretical approximations

- Except in special cases, the statistical properties of parameters rely on asymptotic results. Standards errors and t-statistics rely on first-order Taylor series approximations that, in the worst case, can fail badly. This applies, especially, to binary logistic regression.
- For logistic regression models, and Poisson models with small expected values, assessments of predictive accuracy should be derived using a resampling approach, perhaps cross-validation.

6.2 Computation for GLMs

Above, it was noted that

$$\mathbf{X}'\mathbf{W}\boldsymbol{\mu} = \mathbf{X}'\mathbf{W}\mathbf{y}$$

where

$$f(\boldsymbol{\mu}) = E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta},$$

for a suitable monotonic "link" function $f()$. Here, $\boldsymbol{\mu} = g(\mathbf{X}\boldsymbol{\beta})$ where $g()$ is the inverse function to $f()$. For all the common link functions $f()$, other than the identity, $\boldsymbol{\mu}$ is a non-linear function of the regression parameters. Also, the matrix $\mathbf{W}$ is in general a function of the parameters that are to be estimated. Iteratively reweighted least squares is used, i.e. Newton-Raphson. Each step involves a standard least squares calculation with a diagonal matrix of weights. [To be expanded ...]

7 References

References

Bates, D, 2007. Comparing least squares calculations. *Vignette "Comparisons" accompanying the package "Matrix" for R.*

Bishop, C. M, 2006. *Pattern Recognition and Machine Learning.* Springer.
[Chapters 3 and 4 offer an interesting and somewhat novel perspective on regression and discriminant methods. The theoretical framework is that of Bayesian Decision theory. This is a demanding text. There is helpful comparative commentary on the methods that are described.]
Cook, R D and Weisberg, S., 1999. *Applied Regression Including Computing and Graphics.* Wiley.
[This emphasizes geometric insights, linear predictors (transformation of predictors, if possible, so that pairwise regression relationships are linear), and dimension reduction.]

Faraway, J. J. 2006. *Extending the Linear Model with R.* Chapman & Hall/CRC.

Koenker, R and Ng, P, 2003. SparseM: A sparse matrix package for R. *Journal of Statistical Software* 8(6).

Maindonald, J. H. and Braun, W.J. 2007. *Data Analysis and Graphics Using R – An Example-Based Approach.* 2^{nd} edition, Cambridge University Press.
URL:http://wwwmaths.anu.edu.au/~johnm/r-book.html

McCullagh, P. and Nelder, J. A., 1989. *Generalized Linear Models.* Chapman and Hall, 2^{nd} edition.

Ruppert, D., Wand, M., and Carroll, R. 2003. *Semiparametric Regression.* Cambridge University Press.

Wood, S. N., 2006. *Generalized Additive Models.* An Introduction with R. Chapman & Hall/CRC.
[This has an elegant treatment of linear models and generalized linear models, as a lead-in to generalized additive models.]