Ridge regression is thus rather subjective, without much "science" behind it. Perhaps the most scientific approach is to use cross validation. Thus let's try **ridge.cv()**:

```
> ridge.cv(as.matrix(curr1[,-5]),curr1[,5])
$intercept

224.9451

$coefficients
    XCanada        XMark       XFranc       XPound
 -5.786784    57.713978   -34.399449    -5.394038

$lambda.opt
[1] 0.3168208
```

The recomended λ value here is about 0.32, rather larger than what we might have chosen using the "knee" method. On the other hand, this larger value makes sense in light of our earlier observation concerning the *mark*.

Shrinkage did occur. Here are the OLS estimates:

```
> lm(Yen ~ .,data=curr1)
...
Coefficients:
(Intercept)       Canada         Mark         Franc
    224.945       -5.615       57.889       -34.703
      Pound
     -5.332
```

Ridge slightly reduced the absolute values of most of the coefficients, Canada being the exception. The fact that the reductions were only slight should not surprise us, given the rough guidelines in Section 8.11.1.3. The n/p ratio is pretty large, and even the multicollinearity was mild according to the generally used rule of thumb (Section 8.2.3.1).

8.4 The LASSO

Much of our material on the LASSO will appear in Chapters 9 and 12 but we introduce it in this chapter due to its status as a shrinkage estimator. To motivate this method, recall first that shrinkage estimators form another

example of the bias-variance tradeoff. With ridge regression, for instance, by shrinking $\widehat{\beta}$, we are reducing its variance (actually, its covariance matrix), at the expense of introducing some bias. If we can choose a good value of λ, we can find a "sweet spot" in that tradeoff, and hopefully improve predictive ability. This of course is the motivation for using cross-validation to choose λ.

The *Least Absolute Shrinkage and Selection Operator* — the LASSO — takes another approach to shrinking. As with ridge regression, the LASSO actually has two equivalent formulations, which in rough terms are:

- Penalize large values of $\widehat{\beta}$.
- Place an explicit limit to the size of $\widehat{\beta}$.

We will begin with the first of these.

8.4.1 Definition

As noted in Section 8.2.4.2 and in earlier chapters, a more traditional way than shrinkage to improve prediction error is *subset selection*, meaning to pare down the set of predictor variables into a smaller but representative set. As discussed earlier, this reduces variance, though again increasing bias. One advantage of this approach is that it is appealing to deal with just a small number of predictors, often termed a *parsimonious* model.

The LASSO was invented with the goal of combining the best aspects of ridge regression on the one hand, and subset selection on the other. It involves shrinkage, like ridge regression, but often results in a roundabout way of doing subset selection.

So, how does the LASSO accomplish all this? The answer is remarkably simple: In (8.11), simply replace $||b||_2^2$ by $||b||_1$ (see (A.2)). In other words, the LASSO estimator is defined to be the value of b that minimizes

$$\sum_{i=1}^{n}(Y_i - \widetilde{X}_i b)^2 + \lambda||b||_1 \tag{8.18}$$

Similar to the ridge case, one can show that an equivalent definition is that

the LASSO estimator is chosen to minimize

$$\sum_{i=1}^{n}(Y_i - \widetilde{X}_i b)^2 \tag{8.19}$$

subject to a constraint of the form

$$||b||_1 \leq \gamma \tag{8.20}$$

Using the argument in Section 8.11.2, we see that the LASSO does produce a shrinkage estimator. But it is designed so that typically many of the estimated coefficients turn out to be 0, thus effecting subset selection, which we will see in Section 9.7.7.1.

8.4.2 The lars Package

We'll use the R package **lars** [65]. It starts with no predictors in the model, then adds them (in some cases changing its mind and deleting some) one at a time. At each step, the action is taken that is deemed to best improve the model, as with *forward stepwise regression*, to be discussed in Chapter 9. At each step, the LASSO is applied, with λ determined by cross-validation.

The **lars** package is quite versatile. Only its basic capabilities will be shown here.

8.4.3 Example: Currency Data

As noted, the LASSO is commonly used as a method for variable selection, the topic of Chapter 9. Since we have only $p = 4$ predictors, and more than 700 observations, variable selection is not really an issue. But in this chapter's context of multicollinearity, it is of interest to see how much the software decides to shrink.

```
> lassout <- lars(as.matrix(curr1[,-5]),curr1[,5])
> lassout
...
R-squared: 0.892
Sequence of LASSO moves:
     Canada Mark Pound Franc
Var       1    2     4     3
```

```
Step      1     2     3     4
```

Note that **lars** requires the predictor values to be given as a matrix.

At Step 0, there are no predictors; it is a regression model with just a constant term, so we are just predicting Y from its unconditional mean. We see that at Step 1, **lars()** brought in the Canada predictor, then the *mark*, then the *pound* and lastly, the *franc*.

Let's take a closer look:

```
> summary(lassout)
LARS/LASSO
Call: lars(x = as.matrix(curr1[, -5]), y = curr1[, 5])
  Df      Rss      Cp
0  1 2052191 6263.50
1  2 2041869 6230.18
2  3  392264  587.31
3  4  377574  539.04
4  5  220927    5.00
```

The C_p criterion is similar to adjusted-R^2, and will be discussed in full in Chapter 9. The user may choose to use the C_p value as a guide as to which model to use. In this case, that approach would choose the full model, with all predictors, not surprising in this context of $p << n$.

Since the LASSO is mainly used for subset selection, the actual values of the estimated coefficients are rather secondary, and not presented in the output of **summary()**. But they are indeed accessible:

```
> lassout$beta
      Canada      Mark      Franc      Pound
0  0.0000000   0.00000   0.00000   0.000000
1 -0.2042481   0.00000   0.00000   0.000000
2 -28.6567963 28.45255   0.00000   0.000000
3 -28.1081479 29.61350   0.00000 -1.393401
4 -5.6151436 57.88856 -34.70273 -5.331583
...
```

Again, this is presented in terms of the values at each step, and the 0s show which predictors were *not* in the model as of that step. In our multicollinearity context in this chapter, we are interested in the final values, at Step 4. They are seen to provide shrinkage similar to the mild amount we saw in Section 8.3.3.

increase prediction error.

In other words, yes, dimension reduction is just as much an issue in the nonparametric setting as in the parametric one.

9.7.7 The LASSO

One of the reasons for the popularity of the LASSO is that it does automatic variable selection. We will take a closer look at LASSO methods in this section.

9.7.7.1 Why the LASSO Often Performs Subsetting

First, similar to the ridge case, minimizing (8.18) is equivalent to minimizing[4]

$$q(b) = \sum_{i=1}^{n} (Y_i - \widetilde{X}_i b)^2 \tag{9.21}$$

subject to the constraint

$$||b||_1 \leq \lambda \tag{9.22}$$

This motivates Figure 9.1.

The figure is for the case of $p = 2$ predictors (for simplicity, we assume there is no constant term β_0). Writing $b = (b_1, b_2)'$, then the horizontal and vertical axes are for b_1 and b_2, as shown. The corners of the diamond are at $(\lambda, 0)$, $(0, \lambda)$ and so on. Due to the constraint (9.22), our LASSO estimator $\widehat{\beta}_l$ must be somewhere within the diamond.

What about the ellipses? They are *contours* of q: For a given value of q, say c, then the locus of points b for which $q(b) = c$ takes the form of an ellipse. Each value of c gives us a different ellipse; two of them, out of infinitely many, are shown in the figure, with the smaller one corresponding to a smaller value of c.

But remember, we are trying to minimize q, so we want c to be as small as possible, i.e., we want the countour curve to be small — but our constraint

[4]The computational details of the minimization process are beyond the scope of this book.

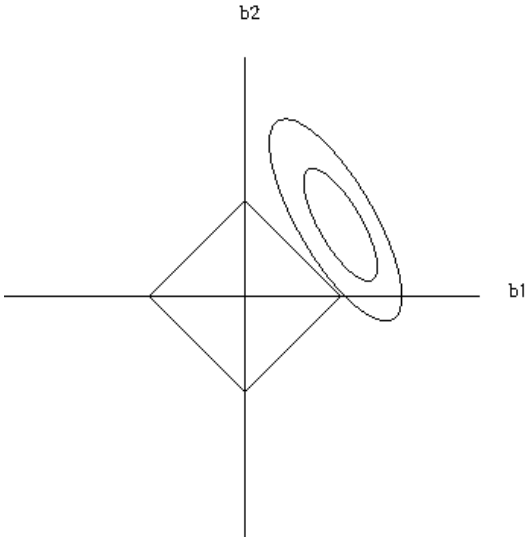

Figure 9.1: Subsetting nature of the LASSO

requires that the curve must include at least one point within the diamond. In our figure here, this implies that we must choose c so that the ellipse is barely touching the diamond, as the larger ellipse does.

Now, here is the key point: The point at which the ellipse barely touches the diamond will typically be one of the four corners of the diamond. And at each of those corners, either b_1 or b_2 is 0 — i.e., $\widehat{\beta}_l$ has selected a *subset* of the predictors, in this case a subset of size 1.

The same geometric argument works in higher dimensions, and this is then the appeal of the LASSO for many analysts:

> The LASSO often does automatic subset selection. The analyst need only use the predictors $X^{(i)}$ for which $\widehat{\beta}_i \neq 0$.

We say that the LASSO tends to produce a *sparse* estimator of β. Needless to say, though this is indeed an appealing property, there is no guarantee that this produces a "good" set of predictors.

Suppose in the figure, the inner ellipse corresponds to the ordinary estima-

tor $b = \widehat{\beta}_{OLS}$, i.e.,

$$c = q(\widehat{\beta}_{OLS}) \tag{9.23}$$

In order to satisfy the LASSO constraint, we needed to accept a larger value of q, corresponding to the outer ellipse, and thus a smaller $\widehat{\beta}$. This illustrates the shrinkage nature of the LASSO.

On the other hand, the ellipse corresponding to OLS might already dip into the diamond. In this case,

$$\widehat{\beta}_l = \widehat{\beta}_{OLS} \tag{9.24}$$

So, it is not guaranteed that the LASSO will choose a sparse $\widehat{\beta}$. As was noted earlier for shrinkage estimators in general, for fixed p, the larger n is, the less need for shrinkage, and the above situation may occur.

There is of course the matter of choosing the value of λ. Our old friend, cross-validation, is an obvious approach to this, and others have been proposed as well. The **lars** package includes a function **cv.lars()** to do k-fold cross-validation.

9.7.7.2 Example: Bodyfat Data

Let's continue the example of Section 9.7.4. Let's see what **lars** finds here.

```
> library(lars)
> larsout <- lars(as.matrix(bodyfat[,-1]),bodyfat[,1])
> larsout
Call:
lars(x = as.matrix(bodyfat[, -1]), y = bodyfat[, 1])
R-squared: 0.749
Sequence of LASSO moves:
     abdomen height age wrist neck forearm hip
Var        6      3   1    13    4      12   7
Step       1      2   3     4    5       6   7
     weight biceps thigh ankle chest knee
Var       2     11     8    10     5    9
Step      8      9    10    11    12   13
```

So, at Step 1, the abdomen predictor was brought in, then height at Step 2, and so on. Now look further:

```
> summary(larsout)
...
    Df      Rss      Cp
0    1 15079.0 698.131
1    2  5423.4  93.012
2    3  5230.7  82.893
3    4  4914.9  65.038
4    5  4333.6  30.484
5    6  4313.5  31.225
6    7  4101.8  19.910
7    8  4090.5  21.202
8    9  4006.5  17.919
9   10  3980.0  18.252
10  11  3859.5  12.679
11  12  3793.0  10.495
12  13  3786.0  12.057
13  14  3785.1  14.000
```

Based on the C_p value, we might stop after Step 11, right after the ankle variable is brought in. The resulting model would consist of predictors abdomen, height, age, wrist, neck, forearm, hip, weight, biceps, thigh and ankle.

By contrast, if one takes the traditional approach and selects the variables on the basis of p-values, as discussed in Section 9.5, only 4 predictors would be chosen (see output in Section 9.7.4), rather than 9 as above.

We can also determine what λ values were used:

```
> larsout$lambda
 [1] 99.9203960 18.1246879 15.5110550 10.7746865
 [5]  4.8247693  4.5923026  2.6282871  2.5472757
 [9]  1.9518718  1.7731184  1.0385786  0.3162681
[13]  0.1051796
```

9.8 Post-Selection Inference

Stepwise predictor selection is an *adaptive* technique. This refers to any statistical method that works in stages, with the outcome of any stage determining what action is taken at the next stage. The problem with this is that a proper statistical analysis would be based on the *conditional* distribution in the later stage, given the earlier stage, rather than the unconditional