The Lasso

Ridge regression shrinks all the coefficients towards zero, but it does not actually set any of them to zero.

The **lasso** is conceptually similar, but it minimizes

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

Once again, λ is a tuning parameter that controls the amount of shrinkage. When $\lambda = 0$, lasso reduces to OLS.

This is an example of ℓ_1-regularization, since $\sum_{j=1}^{p}|\beta_j| = \|\boldsymbol{\beta}\|_1$, the ℓ_1 norm of $\boldsymbol{\beta}$.

Like ridge regression, lasso does not shrink the constant term.

The word “lasso” originally stood for **least absolute selection and shrinkage operator**; see Tibshirani (1996) with 62,887 cites.

Minimizing (1) is equivalent to solving the constrained minimization problem

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \Big(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \Big)^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t. \tag{2}$$

In (1), λ plays the role of a Lagrange multiplier on the constraint in (2).

For any value of t in (2), there is an associated value of λ, which might be 0 if t is sufficiently large.

Since it does not make sense to penalize all the β_j in the same way if the magnitudes of the x_j differ substantially, it is essential to rescale all the non-dummy regressors to have variance 1.

During the computation, they and the y_i are also rescaled to have mean 0, so that we first estimate $\boldsymbol{\beta}$ and then β_0. `glmnet` silently rescales all non-dummy regressors.

Binary (dummy) regressors are generally not rescaled.

We have to choose λ (or, equivalently, t) in some way. It is common to use k-fold cross-validation.

For t, we get OLS whenever the inequality in (2) is strict. So $t = \|\hat{\boldsymbol{\beta}}\|_1$ is the smallest value that gives the OLS estimates.

- Some books (not ISLR/ISLP) put a factor of $1/(2n)$ in front of the first term in (1). This makes the chosen value of λ more or less invariant to the sample size.
- Equivalently, we could multiply the second term by $2n$.
- With the usual formulation, λ needs to be proportional to n
- Making λ invariant to the sample size is useful for cross-validation.
- With the conventional formulation, the 10-fold-CV λ would need to be multiplied by $10/9$ for use with the full sample.
- With either of the invariant formulations, that step can be avoided.

Figure 6.7 from ISLR/ISLP explains why lasso selects as well as shrinks.

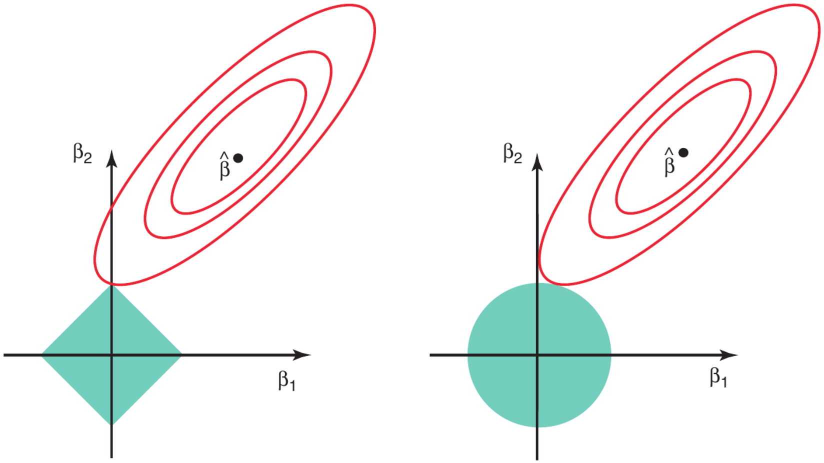


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$*, while the red ellipses are the contours of the RSS.*

Properties of Lasso Estimates

- Like ridge regression, lasso yields biased estimates.
- Increasing λ increases the bias but reduces the variance.
- Lasso rarely yields consistent estimates, although it can do so when $\boldsymbol{\beta}$ is sufficiently sparse.
- If there are actually d non-zero coefficients, d/p must be small for the estimates to be consistent.
- Lasso can handle problems with $p >> n$, but it yields at most n non-zero coefficients.
- More correlation among the columns of $\boldsymbol{X}$ makes it harder to pick the correct predictors.

Lasso can produce strange results when some of the regressors are highly correlated. It may set all but one, or all but a few, of the coefficients on the highly correlated regressors to zero.

In contrast, ridge regression is likely to shrink all of the coefficients on the highly correlated regressors, and they may well end up with similar coefficients.

We may care about both the **false discovery rate**, or **FDR**, and the **false exclusion rate**, or **FER**.

The FDR is the expected number of non-zero $\hat{\beta}_j$ for which $\beta_j = 0$, divided by the total number of non-zero $\hat{\beta}_j$.

The FER is the expected number of zero $\hat{\beta}_j$ for which $\beta_j \neq 0$, divided by the total number of zero $\hat{\beta}_j$.

Ideally, the FDR and the FER would both be zero. In practice, they can be far from zero.

This is especially so when p is large relative to n, and when there is collinearity.

ISLR/ISLP considers the very special case in which $n = p$ and $\boldsymbol{X}$ is the identity matrix. Thus the OLS estimates are simply $\hat{\beta}_j = y_j$.

In this special case, the ridge estimates are

$$\hat{\beta}_j^R = y_j/(1+\lambda). \tag{3}$$

So the OLS estimates are all shrunk by the same proportion.

In contrast, the lasso estimates for this special case are

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 \;\text{ if }\; y_j > \lambda/2; \\ 0 \;\text{ if }\; |y_j| \le \lambda/2; \\ y_j + \lambda/2 \;\text{ if }\; y_j < -\lambda/2. \end{cases} \tag{4}$$

So when y_j is smallish, $\beta_j^L = 0$. When it is not so small, β_j^L is shrunk towards 0. See Figure 6.10 from ISLR/ISLP.

Notice that the shrinkage is not proportional to $\hat{\beta}_j$. If the lasso estimate is shrunk, it is always shrunk by $\lambda/2$.

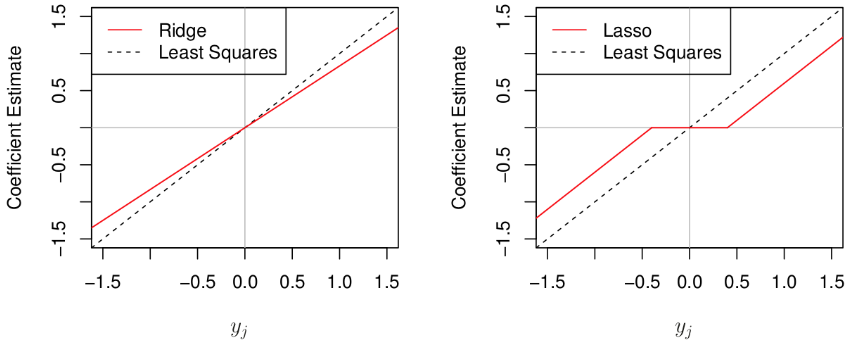


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

The case of just one observation for each input is extremely special.

Suppose instead that there are multiple observations for each input.

- If the inputs are binary and disjoint, then $\hat{\beta}_j$ is just the average of the y_{ij} associated with the j^{th} input.
- In this case, the amount of shrinkage and selection would depend on the sum of the y_{ij} for each input.
- An estimate of 0 would be more likely if there were not many non-zero y_{ij} for a feature or if their average were small.

The penalty on every coefficient is the same. So coefficients that do not affect the fit much are more likely to be set to zero.

- This is the case for coefficients that affect few observations because the y_{ij} are mostly equal to 0.
- It is also the case whenever $\beta_j y_{ij}$ is small for all observations where $y_{ij} \neq 0$.

An Example of the Lasso

We return to the heart-disease data previously used to illustrate ridge regression.

Figure 10.1 shows the lasso coefficients as a function of $\log(\lambda)$.

Here there are 50 values of λ running from $1/1000$ to 1.

These differ quite a bit from the ones used previously. For ridge, the coefficients kept changing as λ moved from 1 to 1000. For lasso, all coefficients are zero once $\lambda > 0.18$.

Evidently, the magnitude of λ should differ between lasso and ridge.

From Figure 10.1, we see that several coefficients get set to 0 quite early, while others survive until λ is fairly large.

Figure 10.2 shows the same estimates as Figure 10.1, but now $\|\hat{\boldsymbol{\beta}}^L\|_1$ is on the horizontal axis.

Figure 10.1

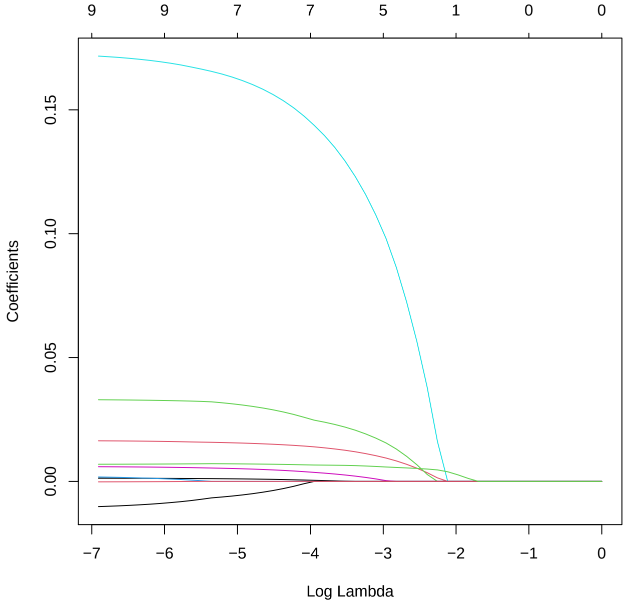


Figure 10.2

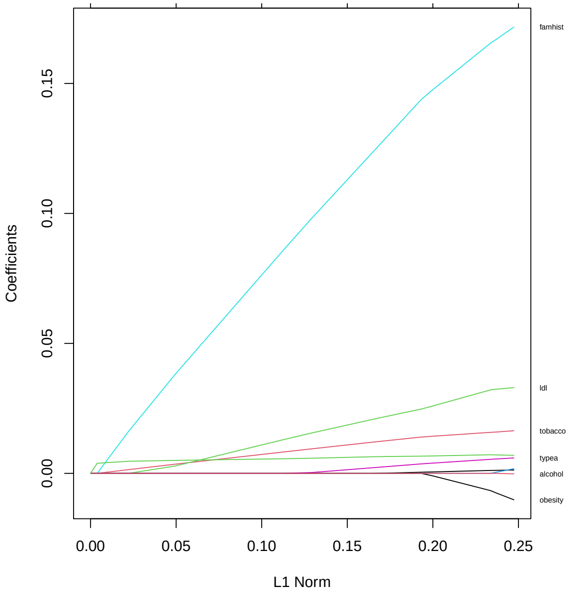

famhist
ldl
tobacco
typea
alcohol
obesity
Coefficients
0.00
0.05
0.10
0.15
0.20
0.25
L1 Norm

Figure 10.2 tried to show all variable names, but the `axis` command refused to print them on top of each other.

In particular, it shows `typea`, which goes to 0 fairly early, but does not show `age`, which survives longer than any other predictor.

The other predictors that survive when λ is fairly large are `famhist`, `tobacco`, and `ldl`.

Figure 10.3 shows the MSE based on 10-fold cross-validation.

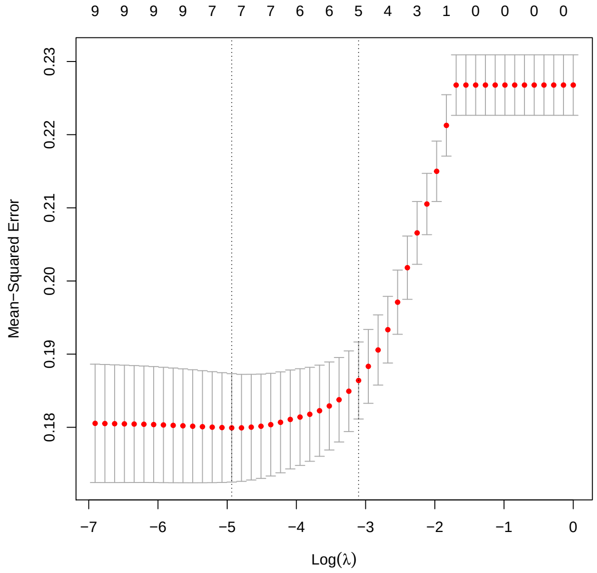
The minimum is at $\log(\lambda) = -4.93$, and the one-standard-error value is at $\log(\lambda) = -3.10$.

Setting $\log(\lambda) = -7$, which implies no selection and almost no shrinkage, works almost as well as setting $\log(\lambda) = -4.93$.

At $\log(\lambda) = -4.93$, the coefficients on `adiposity` and `alcohol` are 0, but the others are non-zero.

At $\log(\lambda) = -3.10$, the coefficients on `sbp` and `obesity` are also 0.

Figure 10.3



9 9 9 9 7 7 7 6 6 5 4 3 1 0 0 0 0
Mean−Squared Error
0.18
0.19
0.20
0.21
0.22
0.23
−7
−6
−5
−4
−3
−2
−1
0
Log(λ)

Other Lasso Variants

One popular lasso variant uses the lasso to identify the set of predictors with non-zero coefficients. Then we run another regression on just that set of predictors.

The lasso selects the non-zero coefficients, but the final estimates are not shrunk at all.

This is called **post-lasso estimation** or **post-lasso least squares**. Its properties were studied in various papers by Chernozhukov et al.

Post-lasso estimation should work well if many coefficients are zero or near zero, but a few of them are large.

This is the precise opposite of the case in which ridge regression should work well.

A closely related idea is to use the lasso twice. This is called the **relaxed lasso**.

- In the second step of the relaxed lasso, we include only the variables with non-zero coefficients in the first step.
- If cross-validation is used at each step, λ in the first step is likely to be larger than in the second step, because there are more noise variables to eliminate in the first step.
- Since bias increases with λ, there should be less bias after the second step.
- Conceptually, the relaxed lasso lies between post-lasso estimation and the ordinary lasso.
- It should work well when many coefficients are zero or very small.

A method that seems to be of great interest to theorists is the **adaptive lasso** (Zou, 2006).

It is not implemented directly in `glmnet`, but that package can be tricked into doing it by using the `penalty.factor` vector. Stata 16+ also implements it.

The adaptive lasso replaces the usual lasso penalty with

$$\lambda \sum_{j=1}^{p} w_j |\beta_j|, \quad w_j = 1/|\tilde{\beta}_j|^{\nu}, \quad \nu > 0, \tag{5}$$

where ν is an additional tuning parameter, perhaps $1/2$ or 1.

Here the $\tilde{\beta}_j$ are **pilot estimates**, perhaps OLS estimates (if $p << n$), univariate regression coefficients (if not), or ridge estimates.

- The penalty on small coefficients in (5) is larger than the penalty on large ones. Thus the former tend to get driven to zero, and the latter get shrunk less than ordinary lasso would do.
- The adaptive lasso has **oracle properties**. Under precisely stated conditions, it asymptotically selects truly non-zero coefficients.
- The proof requires that the $\tilde{\beta}_j$ be unbiased, which rules out ridge regression. However, Zou recommends using ridge when the regressors are highly correlated.

The Elastic Net

The package name `glmnet` comes from the fact that it implements the **elastic net** for **generalized linear models**. However, ISLR/ISLP does not discuss the elastic net!

The elastic-net penalty (Zou and Hastie, 2005) has the form

$$\mathrm{ENP}(\alpha) = \sum_{j=1}^{p} \left(\alpha |\beta_j| + (1-\alpha) \frac{1}{2} \beta_j^2 \right). \tag{6}$$

As usual, $\mathrm{ENP}(\alpha)$ is either multiplied by a tuning parameter λ and added to the sum of squares, or it is constrained not to exceed an upper limit t.

The penalty (6) reduces to the lasso penalty when $\alpha = 1$ and to the ridge penalty when $\alpha = 0$.

The `glmnet` user specifies α in order to obtain lasso, ridge, or elastic net estimates.

`glmnet` does not provide cross-validation for α. The user just has to specify α, or write a loop.

- The elastic net tends to yield more non-zero coefficients than lasso, but they are shrunk more towards zero.
- Unlike lasso, the elastic net can yield more than n non-zero coefficients when $p > n$.
- The elastic net is often recommended when there is a lot of correlation among the regressors.
- In this case, lasso will tend to set many coefficients to zero, often in a seemingly random fashion. Instead, the elastic net will yield more non-zero coefficients, but smaller ones.

I played with the elastic net for the heart-disease data, but the results were pretty similar to lasso for $\alpha = 0.25, 0.50$, and 0.75. The non-zero coefficients were noticeably smaller when λ was too large.

Classification

It is easy to modify lasso, ridge, and elastic net to work with logistic regression and several other models.

In fact, `glmnet` also estimates logit, poisson, multinomial, and Cox proportional hazards models. Hence the **GLM** in its name.

The penalty term is the same as for the regression case, but instead of minimizing the SSR, we minimize the deviance, which is -2 times the loglikelihood function. Use the option: `family="binomial"`.

The logit version of lasso (or ℓ_1-regularized logistic regression, to be formal) minimizes

$$\frac{1}{n}\sum_{i=1}^{n}\left(2\log\left(1+\exp(\beta_0+\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)-2y_i(\beta_0+\boldsymbol{x}_i^\top\boldsymbol{\beta})\right)+\lambda\|\boldsymbol{\beta}\|_1. \qquad (7)$$

For the heart-disease data, this yields results very similar to those for lasso on the linear probability model.

Similarly, the logit version of ridge (or ℓ_2-regularized logistic regression, to be formal) minimizes

$$\frac{1}{n}\sum_{i=1}^{n}\Big(2\log\big(1+\exp(\beta_0+\boldsymbol{x}_i^\top\boldsymbol{\beta}\big)-2y_i(\beta_0+\boldsymbol{x}_i^\top\boldsymbol{\beta})\Big)+\lambda\|\boldsymbol{\beta}\|^2. \qquad (8)$$

Methods like ridge, lasso, and elastic net that are designed for high-dimensional data necessarily make strong assumptions about functional form.

Methods for low-dimensional data that depend on local information, such as KNN and kernel regression, do not.

We can make high-dimensional methods more flexible by including squares, cross-products, or other nonlinear functions of some predictors. With lasso and elastic net, this can yield odd results.

Making inferences about parameter values is non-trivial, and `glmnet` does not attempt to do it.