The Lasso

Patrick Breheny

September 6

Introduction

- As we have seen, ridge regression is capable of reducing the variability and improving the accuracy of linear regression models, and that these gains are largest in the presence of multicollinearity
- What ridge regression doesn't do is variable selection, and it fails to provide a parsimonious model with few parameters

The lasso

- Consider instead a different estimator, which minimizes

$$\frac{1}{2}\sum_i (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| ,$$

  the only difference from ridge regression being that absolute values, instead of squares, are used in the penalty function
- The change to the penalty function is subtle, but has a dramatic impact on the resulting estimator

The lasso (cont'd)

- Like ridge regression, penalizing the absolute values of the coefficients introduces shrinkage towards zero
- However, unlike ridge regression, some of the coefficients are shrunken all the way to zero; such solutions, with multiple values that are identically zero, are said to be *sparse*
- The penalty thereby performs a sort of continuous variable selection
- The resulting estimator was thus named the *lasso*, for "Least Absolute Shrinkage and Selection Operator"

Geometry of ridge vs. lasso

A geometrical illustration of why lasso results in sparsity, but ridge does not, is given by the constraint interpretation of their penalties:

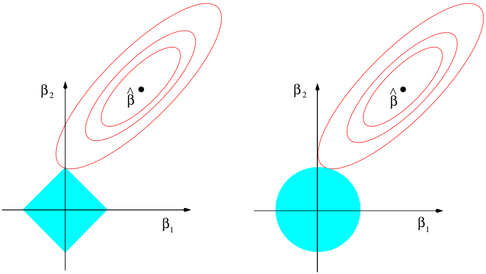
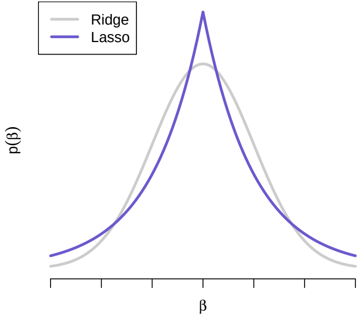


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Bayesian perspective

- Another way of seeing how the lasso produces sparsity is to view it from a Bayesian perspective, where the lasso penalty produces a double exponential prior:



- Note that the lasso prior is "pointy" at 0, so there is a chance that the posterior mode will be identically zero

Orthonormal Solutions

- Because the lasso penalty has the absolute value operation in it, the objective function is not differentiable and as a result, lacks a closed form in general
- However, in the special case of an orthonormal design matrix, it is possible to obtain closed form solutions for the lasso: $\hat{\beta}_J^{\text{lasso}} = S(\hat{\beta}_J^{\text{OLS}}, \lambda)$, where S, the *soft-thresholding operator*, is defined as

$$S(z, \lambda) = \begin{cases} z - \lambda & \text{if } \; z > \lambda \\ 0 & \text{if } \; |z| \leq \lambda \\ z + \lambda & \text{if } \; z < -\lambda \end{cases}$$

Hard vs. soft thresholding

- The function on the previous slide is referred to as "soft" thresholding to distinguish it from *hard thresholding*:

$$H(z,\lambda) = \begin{cases} z & \text{if } |z| > \lambda \\ 0 & \text{if } |z| \le \lambda \end{cases}$$

- In the orthonormal case, best subset selection is equivalent to hard thresholding
- Note that soft thresholding is continuous, while hard thresholding is not

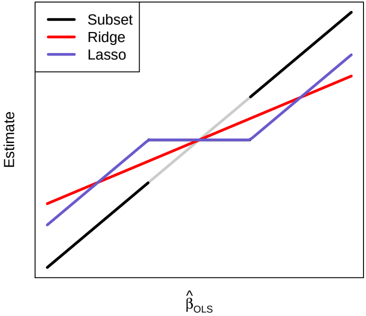
Ridge, lasso, and subset selection in the orthonormal case

Thus, in the orthonormal case, each of the methods we have discussed are simple functions of the least squares solutions:

$$\text{Subset selection: } \hat{\beta}_j = H(\hat{\beta}_j^{OLS}, \lambda)$$

$$\text{Ridge: } \hat{\beta}_j = \hat{\beta}_j^{OLS}/(1+\lambda)$$

$$\text{Lasso: } \hat{\beta}_j = S(\hat{\beta}_j^{OLS}, \lambda)$$



A brief history of lasso algorithms

- As we mentioned earlier, the lasso penalty lacks a closed form solution in general
- As a result, optimization algorithms must be employed to find the minimizing solution
- The historical efficiency of algorithms to fit lasso models can be summarized as follows:

Year	Algorithm	Operations	Practical limit
1996	Quadratic programming	$O(n2^p)$	~ 100
2003	LARS	$O(np^2)$	$\sim 10,000$
2008	Coordinate descent	$O(np)$	$\sim 1,000,000$

Selection of λ

- Unlike ridge regression, the lasso is not a linear estimator – there is no matrix $\mathbf{H}$ such that $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$
- Defining the degrees of freedom of the lasso is therefore somewhat messy
- However, a number of arguments can be made that the number of nonzero coefficients in the model is a reasonable quantification of the model's degrees of freedom, and this quantity can be used in AIC/BIC/GCV to select λ
- Other statisticians, however, feel these approximations to be untrustworthy, and prefer to select λ via cross-validation instead

Fitting lasso models in SAS

- SAS provides the GLMSELECT procedure to fit lasso-penalized linear models:

```
PROC GLMSELECT DATA=prostate PLOTS=ALL;
  MODEL lpsa = pgg45 gleason lcp svi lbph age lweight
               lcavol / SELECTION=LASSO(STOP=NONE) STATS=SBC;
RUN;
```

- GLMSELECT allows for many other selection criteria, include cross-validation
- Note that despite its name, GLMSELECT only fits linear models, not GLMs

Fitting lasso models in R

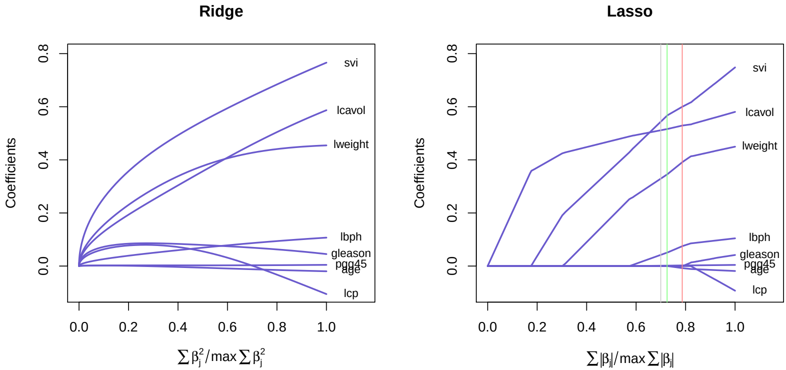
- In R, the `glmnet` package can fit a wide variety of models (linear models, generalized linear models, multinomial models, proportional hazards models) with lasso penalties
- The syntax is fairly straightforward, though it differs from `lm` in that it requires you to form your own design matrix:

```
fit <- glmnet(X,y)
```

- The package also allows you to conveniently carry out cross-validation:

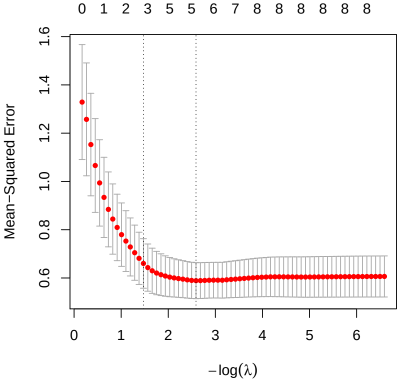
```
cvfit <- cv.glmnet(X,y)
plot(cvfit)
```

Ridge vs. lasso coefficient paths



Gray=CV, Red=AIC/GCV, Green=BIC

Cross-validation results



The line on the right is drawn at the minimum CV error; the other is drawn at the maximum value of λ within 1 SE of the minimum

OLS vs. Ridge vs. Lasso

Coefficient estimates:

	OLS	Ridge	Lasso
lcavol	0.587	0.516	0.511
lweight	0.454	0.443	0.329
age	-0.020	-0.015	0.000
lbph	0.107	0.096	0.042
svi	0.766	0.695	0.544
lcp	-0.105	-0.042	0.000
gleason	0.045	0.061	0.000
pgg45	0.005	0.004	0.001

CV used to select λ for lasso; GCV used to select λ for ridge