

Linear Regression Models

P8111

Lecture 16

Jeff Goldsmith

March 24, 2016



THE DEPARTMENT OF
BIostatISTICS



Columbia University
**MAILMAN SCHOOL
OF PUBLIC HEALTH**

Today's Lecture

Model Selection

- Penalized
- Ridge regression ✓
 - Lasso regression

Variable selection

Suppose $Var(\epsilon) = \sigma^2 I$. In Lecture 15 we talked about model selection:

- Given a lot of variables, which should we include in a model?
- Several approaches, but variables were either in or out
- Difficult for large p
- Gives results that are 'unbiased' for the truth, but can be high variance

$$E(\hat{\beta}_{OLS}) = \beta$$

MS focused on Estimation / Testing

Gauss-Markov and MSE

Recall the Gauss-Markov theorem says OLS is BLUE. Maybe “unbiased” is more restrictive than we’re interested in.

- Alternatively, we could try to minimize the mean squared error:

$$\begin{aligned} \text{MSE}(\hat{\beta}) &= E \left[\left(\hat{\beta} - \beta \right)^2 \right] \\ &= E \left[\left(\hat{\beta} - E(\hat{\beta}) + E(\hat{\beta}) - \beta \right)^2 \right] \\ &= E \left[\left(\hat{\beta} - E(\hat{\beta}) \right)^2 \right] + \left(E(\hat{\beta}) - \beta \right)^2 \\ &= \underbrace{\text{variance}(\hat{\beta})}_{\downarrow} + \underbrace{\text{bias}^2(\hat{\beta})}_{\leftarrow} \end{aligned}$$

Penalized regression

- Could try a shrinkage / penalization approach to trade some bias for lower variance and overall MSE
- Rather than a variable selection approach, all parameters stay in the model, but we restrict their effect
- We penalize the size of the coefficients – unimportant variables will have their coefficients forced closer to zero

Ridge regression

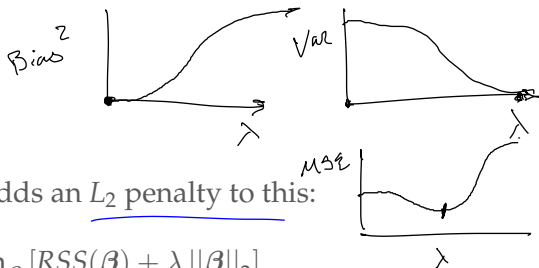
OLS is derived by minimizing the RSS:

$$\begin{aligned}\hat{\beta}_{OLS} &= \arg \min_{\beta} [\text{RSS}(\beta)] \\ &= \arg \min_{\beta} \left[\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right]\end{aligned}$$

$$\frac{\partial \text{RSS}(\beta)}{\partial \beta} = 0$$

$$(y - X\beta)^T (y - X\beta)$$

Ridge regression



Ridge regression adds an L_2 penalty to this:

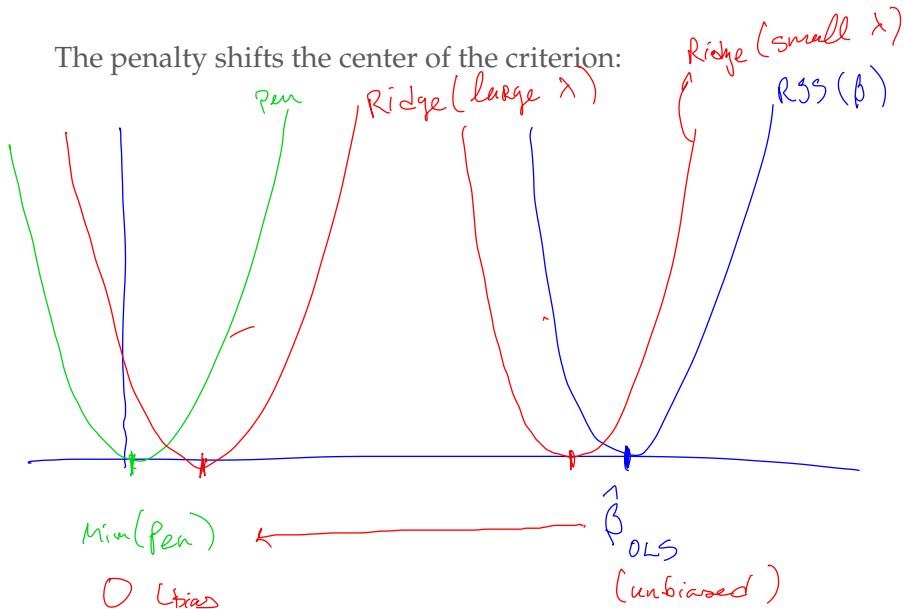
$$\hat{\beta}_R = \arg \min_{\beta} [\text{RSS}(\beta) + \lambda \|\beta\|_2]$$

$$\text{~~arg min~~}_{\beta} \left[\underbrace{\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2}_{\text{RSS}(\beta)} + \lambda \underbrace{\sum_{j=1}^p \beta_j^2}_{\text{Pen}_{\lambda}(\beta)} \right]$$

$$\left\{ \begin{array}{l} \lambda = 0 \Rightarrow \text{OLS} \\ \lambda \rightarrow \infty \Rightarrow \hat{\beta}_R = 0 \end{array} \right.$$

Graphical representation

The penalty shifts the center of the criterion:



Ridge regression in matrix notation

In matrix notation, we want to minimize

$$\underbrace{RSS(\beta)} + \lambda \|\beta\|_2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^T \underbrace{P}_{\downarrow} \beta \quad /$$

where P is the penalty matrix. $RSS(\beta)$

$$\begin{aligned} & \lambda \sum \beta_j^2 \\ &= \lambda \beta^T \beta \\ &= \lambda \beta^T \mathbf{I} \beta \end{aligned}$$

Ridge regression in matrix notation

Finding solutions to

$$\min_{\beta} \left(\text{RSS}(\beta) + \lambda \|\beta\|_2 \right) = \left(\underbrace{(\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)} + \underbrace{\lambda \beta^T P \beta} \right)$$

Ridge regression estimates

$$\hat{\beta}_{OLS} = (X^T X)^{-1} X^T y$$

The ridge regression estimates are given by

$$\hat{\beta}_R = (X^T X + \lambda I)^{-1} X^T y$$

λ acts as a tuning parameter

- For “small” values of λ , $\hat{\beta}_R \approx \hat{\beta}_{OLS}$
- For “large” values of λ , $\hat{\beta}_R \approx 0$

Is there an MLE equivalent to this?

Sort of ...

- We'll worry more about this later
- If we assume the β_j 's are random (especially Normal) then there's a likelihood function that includes the penalty term

OLS

$$y = X\beta + \epsilon$$

$$\epsilon \sim N(0, \sigma^2 I)$$

$$\Rightarrow \hat{\beta}_{MLE} = (X^T X)^{-1} X^T y$$

Ridge

$$y = X\beta + \epsilon$$

$$\epsilon \sim N(0, \sigma^2 I)$$

$$\beta \sim N(0, \underbrace{\sigma^2 P^{-1}}_{\lambda I})$$

$$\Rightarrow \hat{\beta}_{MLE} = (X^T X + \lambda P) X^T y$$

Properties of ridge regression

- Ridge regression estimates are biased:

$$\begin{aligned} E(\hat{\beta}_R) &= E \left[\left(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{P} \right)^{-1} \mathbf{X}^T \mathbf{y} \right] \\ &= \left(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{P} \right)^{-1} \mathbf{X}^T \mathbf{X} \beta \end{aligned}$$

- Tend to have lower variance than OLS
- Often lead to lower MSE's
- Interesting note – penalized estimates may be identifiable even when $p > n$

$$\left(\mathbf{X}^T \mathbf{X} \right)^{-1} \text{ DNE}$$

$$\left(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{P} \right)^{-1} \text{ PE!}$$

MSE for predictions

MSE for β can be hard to discuss in practice

- MSE for predictions can be easier to focus on

$$MSE(\hat{y}) = E \left[(\hat{y} - y)^2 \right]$$

- Could evaluate this using cross-validation

Tuning parameter selection

The tuning parameter λ is important for overall model fit

- Depending on λ , we may be looking at OLS or $\hat{\beta} = 0$
- “Truth” is usually somewhere in the middle
- It turns out that we’ve avoided variable selection, but now have to focus on tuning parameter selection
- Cross-validation is a common way of choosing λ

Life expectancy example

- Response: life expectancy
- Predictors: population, capital income, illiteracy rate, murder rate, percentage of high-school graduates, number of days with minimum temperature < 32 , land area
- Data for 50 US states
- Time span: 1970-1975

Example

```
> data(state)
> statedata = data.frame(state.x77, row.names=state.abb)
> model.full = lm(Life.Exp ~., data=statedata)
> coef(model.full)
(Intercept) Population      Income Illiteracy      Murder      HS.Grad      Frost      Area
 7.094e+01  5.180e-05 -2.180e-05  3.382e-02 -3.011e-01  4.892e-02 -5.735e-03 -7.383e-08
```

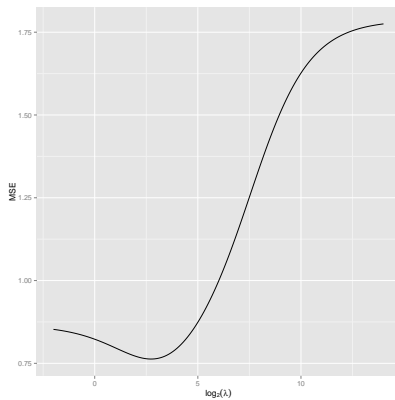
Example

```
> model.ridge1 = lm.ridge(Life.Exp ~., data=statedata, lambda = 1000000)
> coef(model.ridge1)
(Intercept) Population      Income Illiteracy      Murder      HS.Grad      Frost      Area
 7.087e+01 -1.022e-09  3.716e-08 -6.479e-05 -1.419e-05  4.837e-06  3.383e-07 -8.442e-11
```

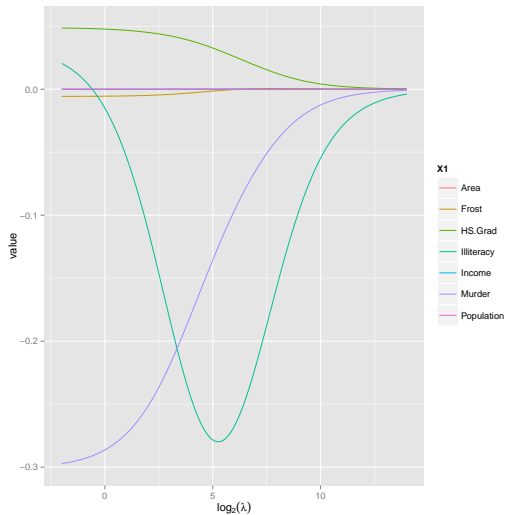
Example

```
> model.ridge2 = lm.ridge(Life.Exp ~ ., data=statedata, lambda = .0000001)
> coef(model.ridge2)
(Intercept) Population      Income Illiteracy      Murder      HS.Grad      Frost      Area
 7.094e+01  5.180e-05 -2.180e-05  3.382e-02 -3.011e-01  4.892e-02 -5.735e-03 -7.383e-08
```

CV Plot



Coef Plot



Example

```
> Lam.Final = lam[which(apply(MSE, 2, mean) == min(apply(MSE, 2, mean)))]
> model.ridge2 = lm.ridge(Life.Exp ~., data=statedata, lambda = Lam.Final)
> round(coef(model.ridge3), 5)
```

	Population	Income	Illiteracy	Murder	HS.Grad	Frost	Area
70.55067	0.00003	0.00006	-0.16047	-0.22998	0.04334	-0.00438	0.00000

Lasso penalization

- Lasso (least absolute shrinkage and selection operator) is a more recent penalized regression estimator
- Basic form is similar to that of ridge regression, but penalty function is different:

$$\begin{aligned}\hat{\beta}_L &= \arg \min_{\beta} [RSS(\beta) + \lambda \|\beta\|_1] \\ &= \arg \min_{\beta} \left[\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right]\end{aligned}$$

- Quite popular at the moment – broadly used, many adaptations

Lasso penalization

Some properties of Lasso penalties

- No closed form solution (although there are some computationally useful tricks)
- The different penalty form means Lasso has a tendency to shrink coefficients *all the way* to zero
- Can be useful as an automated variable selection approach
- Still have to choose λ ; cross validation is a popular tool for this

Example: Mortality Rate

```
lm(formula = Life.Exp ~ ., data = statedata)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	7.094e+01	1.748e+00	40.586	< 2e-16	***
Population	5.180e-05	2.919e-05	1.775	0.0832	.
Income	-2.180e-05	2.444e-04	-0.089	0.9293	
Illiteracy	3.382e-02	3.663e-01	0.092	0.9269	
Murder	-3.011e-01	4.662e-02	-6.459	8.68e-08	***
HS.Grad	4.893e-02	2.332e-02	2.098	0.0420	*
Frost	-5.735e-03	3.143e-03	-1.825	0.0752	.
Area	-7.383e-08	1.668e-06	-0.044	0.9649	

Example: Mortality Rate

```
> model.lassol = glmnet(X, y, lambda = 0.00001)
> coef(model.lassol)
8 x 1 sparse Matrix of class "dgCMatrix"
      s0
(Intercept) 7.094187e+01
Population  5.185263e-05
Income      -2.191147e-05
Illiteracy  3.467775e-02
Murder      -3.012157e-01
HS.Grad     4.894538e-02
Frost       -5.730853e-03
Area        -7.370497e-08
```

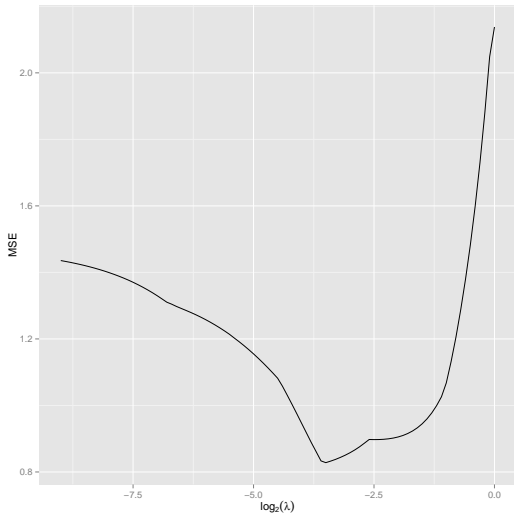
Example: Mortality Rate

```
> model.lasso2 = glmnet(X, y, lambda = 0.01)
> coef(model.lasso2)
8 x 1 sparse Matrix of class "dgCMatrix"
              s0
(Intercept)  7.101048e+01
Population   4.762476e-05
Income       .
Illiteracy   .
Murder       -2.944565e-01
HS.Grad      4.551701e-02
Frost        -5.542157e-03
Area         .
```

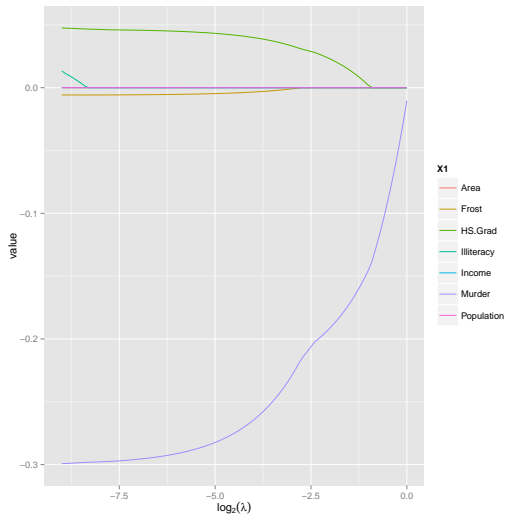
Example: Mortality Rate

```
> model.lasso3 = glmnet(X, y, lambda = 10)
> coef(model.lasso3)
8 x 1 sparse Matrix of class "dgCMatrix"
      s0
(Intercept) 70.8786
Population   0.0000
Income       .
Illiteracy   .
Murder       .
HS.Grad      .
Frost        .
Area         .
```

CV plot



Coef plot



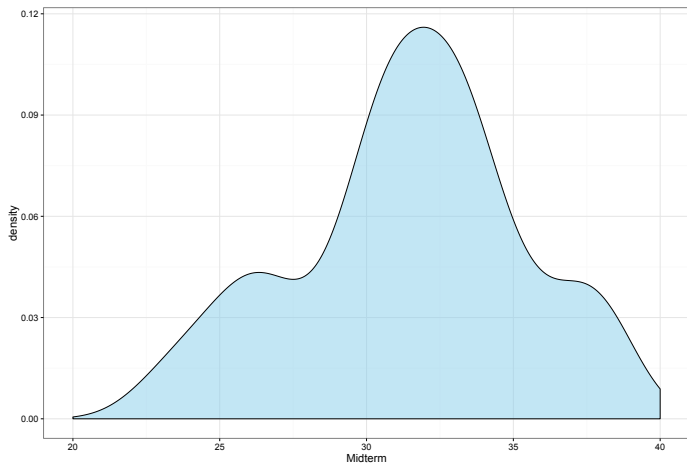
Example: Mortality Rate

```
> Lam.Final = lam[which(apply(MSE, 2, mean) == min(apply(MSE, 2, mean)))]
> model.lasso4 = glmnet(X, y, lambda = Lam.Final)
> coef(model.lasso4)
8 x 1 sparse Matrix of class "dgCMatrix"
              s0
(Intercept)  7.088048e+01
Population   2.786869e-05
Income       .
Illiteracy   .
Murder       -2.498481e-01
HS.Grad      3.716564e-02
Frost        -2.399294e-03
Area         .
```

Practical note

- ▶ In most cases, it's best to standardize predictors prior to penalizing
- ▶ Doing so ensures that the coefficients to be penalized have comparable effects on the outcome
- ▶ Not always obvious – see, e.g. categorical and binary predictors – but useful nonetheless

Midterm grades



Today's big ideas

- Ridge regression

-
- Suggested reading: Faraway Ch. 9.5, ISLR Ch 6.2