Ridge, LASSO, and Elastic net regression

Emilija Perković [1]

Dept. of Statistics
University of Washington

[1] Based on lectures by Ryan Tibshirani

Bias-Variance trade-off

Suppose that the true model is:

$$y_i = f(x_i) + \epsilon_i,$$

with $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, $\mathrm{E}[\epsilon_i] = 0, \mathrm{Var}[\epsilon_i] = \sigma^2, \mathrm{Cov}[\epsilon_i, \epsilon_j] = 0$. In the linear case, $f(x_i) = x_i'\beta$, $\beta \in \mathbb{R}^{(p+1)}$.
Let $\widehat{f}(x_i)$ be an estimate of $f(x_i)$. In particular, think of OLS where we have $\hat{f}(x_i) = x_i'\hat{\beta}$. Suppose we observe a new data point x_0, $y_0 = f(x_0) + \epsilon_0$ and we predict y_0 with $\widehat{f}(x_0) = x_0'\hat{\beta}$.
Let us consider how the generalization (prediction) error relates to the bias and variance of our estimator

$$\begin{aligned}
\mathrm{E}[(y_0 - \widehat{f}(x_0))^2] &= \mathrm{E}[(y_0 - f(x_0) + f(x_0) - \widehat{f}(x_0))^2] \\
&= \mathrm{E}[(y_0 - f(x_0))^2] + \mathrm{E}[(f(x_0) - \widehat{f}(x_0))^2] + 2\mathrm{E}[(y - f(x_0))(f(x_0) - \widehat{f}(x_0))] \\
&= \sigma^2 + \mathrm{E}[(f(x_0) - \widehat{f}(x_0))^2] + 2\mathrm{E}[\epsilon_0(f(x_0) - \widehat{f}(x_0))] \\
&= \sigma^2 + \mathrm{E}[(f(x_0) - \widehat{f}(x_0))^2]
\end{aligned}$$

In the last line, we use that ϵ_0 is independent of $\hat{f}(x_0)$ and the linearity of expectation.

Bias-Variance trade-off

How to decompose the MSE of the estimator: $\mathrm{E}[(f(x_0)-\widehat{f}(x_0))^2]$?

$$\begin{aligned}&\mathrm{E}[(f(x_0)-\widehat{f}(x_0))^2] = \mathrm{E}[(f(x_0)-\mathrm{E}[\hat{f}(x_0)]+\mathrm{E}[\hat{f}(x_0)]-\widehat{f}(x_0))^2]\\&= (\mathrm{E}[\hat{f}(x_0)]-f(x_0))^2+\mathrm{E}[(\widehat{f}(x_0)-\mathrm{E}[\hat{f}(x_0)])^2]+2\mathrm{E}[(f(x_0)-\mathrm{E}[\hat{f}(x_0)])(\mathrm{E}[\hat{f}(x_0)]-\widehat{f}(x_0))]\\&= [Bias(\hat{f}(x_0))]^2+\mathrm{Var}[\hat{f}(x_0)]\end{aligned}$$

So the generalization error:

$$\mathrm{E}[(y_0-\widehat{f}(x_0))^2] = \sigma^2+[Bias(\hat{f}(x_0))]^2+\mathrm{Var}[\hat{f}(x_0)]$$

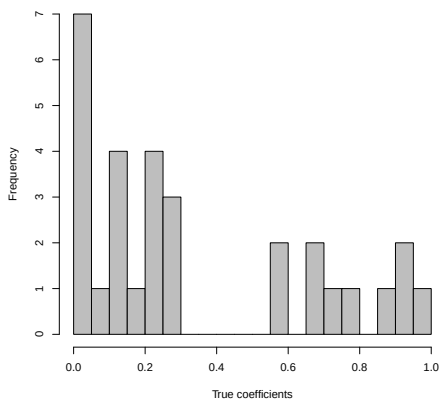
This is called the bias-variance decomposition.

Recall the Gauss-Markov theorem: OLS estimator is unbiased, that is $Bias(x_0'\hat{\beta}^{OLS}) = 0$.

What about the variance? It can be shown that $\mathrm{Var}[x_0'\hat{\beta}^{OLS}] \approx \sigma^2\frac{p+1}{n}$.
Can we obtain a lower variance (and lower gen. error) if we allow some bias?

Example: Subset of small coefficients

We have $n = 50, p = 30$,and $\sigma^2 = 1$ The true model is linear with 10 large coefficients (between 0.5 and 1) and 20 small ones (between 0 and 0.3).
Histogram:



Squared bias ≈ 0.006, Variance ≈ 0.627, Pred. error $\approx 1+0.006+0.627 \approx 1.633$.
Can we do better?

Ridge regression

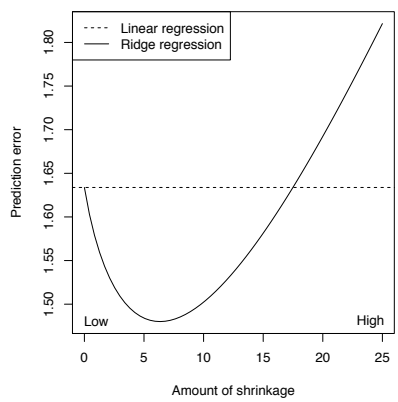
Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$, the ridge regression coefficients are defined as

$$\begin{aligned}\hat{\beta}^{\text{ridge}} &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}\end{aligned}$$

Here $\lambda \geq 0$ is a **tuning parameter**, which controls the strength of the penalty term. Note that:

- When $\lambda = 0$, we get the linear regression estimate.
- When $\lambda = \infty$, we get $\hat{\beta}^{ridge} = 0$.
- For λ i in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

Prediction error comparison

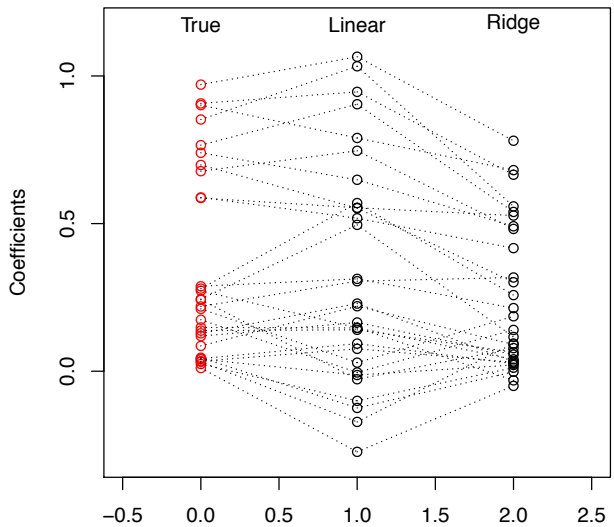


Linear regression:
Squared bias ≈ 0.006
Variance ≈ 0.627
Pred. error $\approx 1 + 0.006 + 0.627$
≈ 1.633

Ridge regression, at its best:
Squared bias ≈ 0.077
Variance ≈ 0.403
Pred. error $\approx 1 + 0.077 + 0.403$
≈ 1.48

Example: visual representation of ridge coefficients

Recall our last example ($n = 50, p = 30$, and $\sigma^2 = 1$; 10 large true coefficients, 20 small). Here is a visual representation of the ridge regression coefficients for $\lambda = 25$:



Estimation and Inference

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

- $\hat{\beta}^{ridge} = (X'X + \lambda I)^{-1} X'y$
- Inclusion of λ makes the problem non-singular even if $X'X$ is not invertible. (This was the original motivation for ridge regression (Hoerl and Kennard, 1970)).

What about inference?

- Since $\hat{\beta}^{ridge}$ is a biased estimate of β, there is a question of how to perform hypothesis tests and how and whether to construct confidence intervals for $\hat{\beta}^{ridge}$.
- Still an ongoing and active area of research.

How to evaluate the model?

- We can still use cross-validation to evaluate the performance of our model (e.g. out of sample MSE).

Important details

When including an intercept term in the regression, we usually leave this coefficient unpenalized. Otherwise we could add some constant amount c to the vector y, and this would not result in the same solution.

Hence ridge regression with intercept solves

$$\hat{\beta}_0, \hat{\beta}^{\text{ridge}} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \|y - \beta_0 \mathbb{1} - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

If we center the columns of X, then the intercept estimate ends up just being $\hat{\beta}_0 = \overline{y}$, so we usually just assume that y, X have been centered and don't include an intercept.

Also, the penalty term $||\beta||_2^2 = \sum_{j=1}^p \beta_j^2$ is unfair if the predictor variables are not on the same scale. (Why?)

Therefore, if we know that the variables are not measured in the same units, we typically scale the columns of X (to have sample variance 1), and then we perform ridge regression.

Bias and variance of ridge regression

The bias and variance are not quite as simple to write down for ridge regression, but closed-form expressions are still possible (we will not be discussing this in class).

Recall that

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

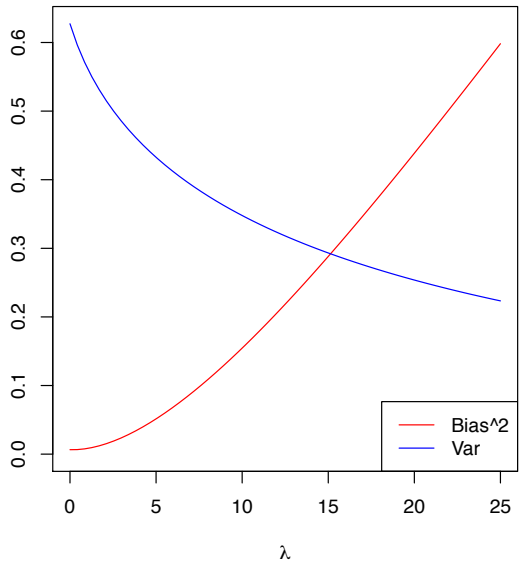
The general trend is:

- The bias increases as λ (amount of shrinkage) increases.
- The variance decreases as λ (amount of shrinkage) increases.

What is the bias at $\lambda = 0$? The variance at $\lambda = \infty$?

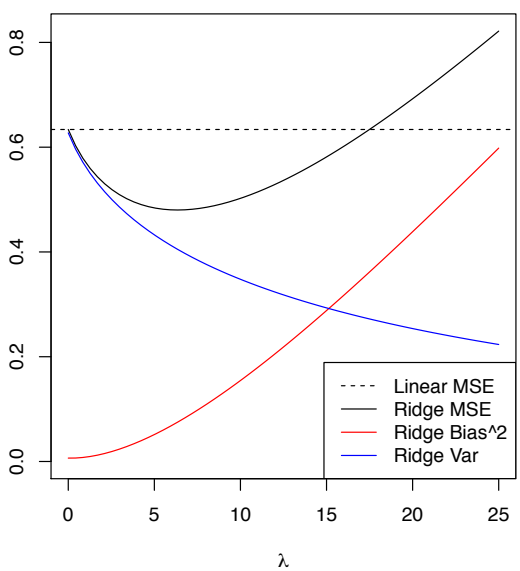
Example: bias and variance of ridge regression

Bias and variance for our last example ($n = 50, p = 30, \sigma^2 = 1$; 10 large true coefficients, 20 small):



Example: bias and variance of ridge regression

Bias and variance for our last example ($n = 50, p = 30, \sigma^2 = 1$; 10 large true coefficients, 20 small):



Questions

- This only works for some values of λ. So how would we choose λ in practice?

This is actually quite a difficult question, but the practical answer is that we will be doing this through cross-validation. (see `cv.glmnet() in R`)

- What happens when we none of the coefficients are small?

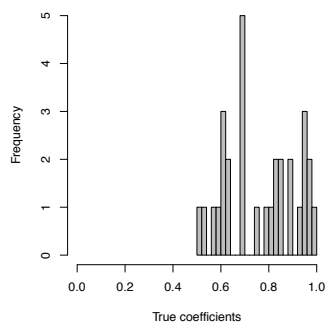
If all the true coefficients are moderately large, is it still helpful to shrink the coefficient estimates?

The answer is still “yes”.

But the advantage here is less dramatic, and the corresponding range for good values of λ is smaller.

Example: moderate regression coefficients

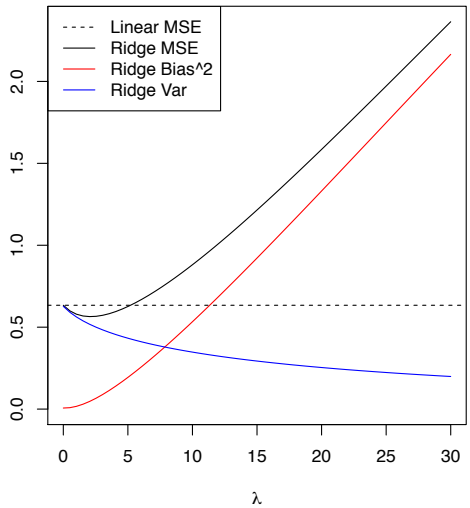
Same setup as our last example: $n = 50, p = 30, \sigma^2 = 1$. Except now the true coefficients are all moderately large (between 0.5 and 1).
Histogram:



The linear regression fit: Squared bias ≈ 0.006, Variance ≈ 0.628, Pred. error $\approx 1 + 0.006 + 0.628 \approx 1.634$.
Why are these numbers essentially the same as those from the last example, even though the true coefficients changed?

Example: moderate regression coefficients

Ridge regression can still outperform linear regression in terms of mean squared error:



Only works for λ less than ≈ 5, otherwise it is very biased.

Example: Ridge, ozone data

Dataset ozone in R package gss.
Daily measurements of ozone concentration and eight meteorological quantities in the Los Angeles basin for 330 days of 1976.

Variables:

- upo3 - Upland ozone concentration, in ppm.
- vdht - Vandenberg 500 millibar height, in meters.
- wdsp - Wind speed, in miles per hour.
- hmdt - Humidity.
- sbtp - Sandburg Air Base temperature, in Celsius.
- ibht - Inversion base height, in foot.
- dgpg - Dagget pressure gradient, in mmHg.
- ibtp - Inversion base temperature, in Fahrenheit.
- vsty - Visibility, in miles.
- day - Calendar day, between 1 and 366.

Example: Ridge, ozone data

After pre-processing fit a linear model:

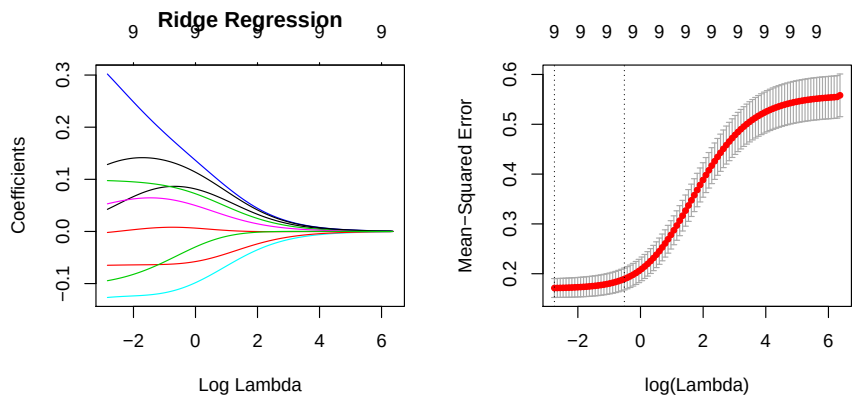
```
fit1 <- lm(scale(d.ozone$logupo3,scale=F)~scale(as.matrix(d.ozone[,-10]))+0,
data = d.ozone.e,x=TRUE,y=TRUE)
> coef(fit1)
(Intercept)  .
vdht         0.002376246
wdsp        -0.009374444
hmdt         0.099789034
sbtp         0.430436030
ibht         0.156800087
dgpg         0.018201447
ibtp         0.034109228
vsty        -0.067330809
day         -0.104772687
> cv.lm(fit1)
Mean squared error         :  0.170485
```

Let us compare this with ridge regression output.

Example: Ridge, ozone data

After pre-processing fit a linear model:

```
ridge <- glmnet(x=scale(as.matrix(d.ozone[,-10])),
     y=scale(d.ozone$logupo3,scale=F), alpha=0,intercept=F)
ridge.cv <- cv.glmnet(x=scale(as.matrix(d.ozone[,-10])),
     y=scale(d.ozone$logupo3,scale=F), alpha=0,intercept=F)
```



How to choose λ?

Example: Ridge, ozone data

The output of `cv.glmnet` gives two suggestions for the λ

- `lambda.min` - which is the λ that yields the lowest CV error (mean squared k-fold error).
- `lamda.1se` - which is the largest λ that yields a CV error, within one standard error of the lowest CV error.

The authors of `glmnet` suggest using `lambda.1se` as it corresponds to the most regularized model that still has a low CV error.
In our case:

```
> ridge.cv$lambda.min
[1] 0.06402547
> ridge.cv$lambda.1se
[1] 0.5971037
```


Example: Ridge, ozone data

What about the coefficients and CV error?

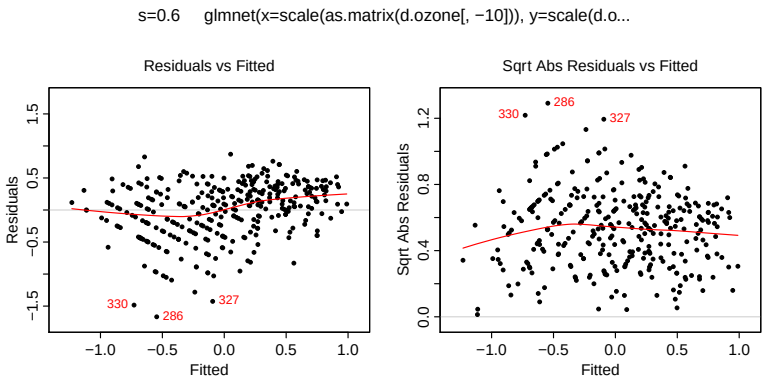
```
## with lamda.1se
(Intercept)  .
vdht         0.086328667
wdsp         0.007873711
hmdt         0.081846970
sbtp         0.163135018
ibht        -0.110259291
dgpg         0.057949139
ibtp         0.127608079
vsty        -0.061613873
day         -0.044268577
## cv.error
 0.1894518
```

```
## with lamda.min
(Intercept)  .
vdht         0.044591755
wdsp        -0.001354003
hmdt         0.097156037
sbtp         0.296427923
ibht        -0.126338983
dgpg         0.054272190
ibtp         0.130474136
vsty        -0.064828824
day         -0.093376710
## cv.error
 0.1712951
```

Model Diagnostics

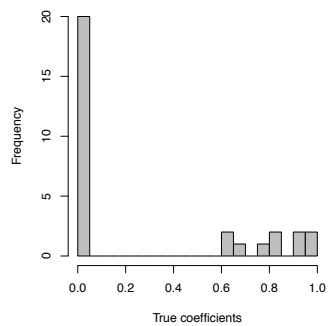
We can still perform some model diagnostics on our ridge regression model

- Can still plot residuals against fitted/predictors etc.
- Use plots to check for non-linearities, non-constant variance, correlated errors.
- See function `plotres` in R package `plotmo`.
- The QQ-plot is no longer relevant.
- Ridge automatically deals with issues of multicolinearities.



Example: subset of zero coefficients

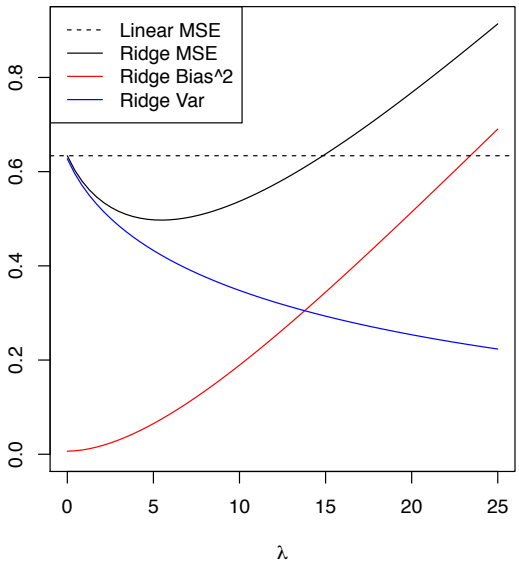
Suppose that a group of true coefficients is zero. This means that the mean response does not depend on these predictors.
Again: $n = 50, p = 30,$ and $\sigma^2 = 1$. Now, the true coefficients: 10 are large (between 0.5 and 1) and 20 are exactly 0.
Histogram:



The linear regression fit: Squared bias ≈ 0.006, Variance ≈ 0.627, Pred. error $\approx 1 + 0.006 + 0.627 \approx 1.633$.
Note again that these numbers haven't changed.

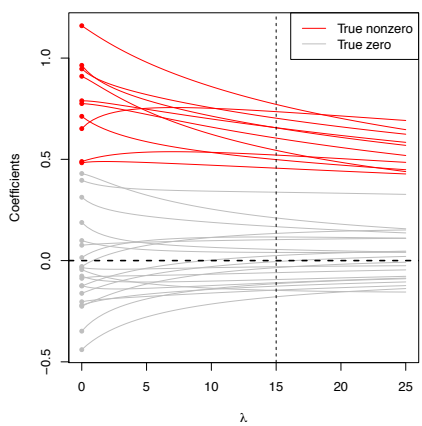
Example: subset of zero coefficients

Ridge regression performs well in terms of mean-squared error:



Example: subset of zero coefficients

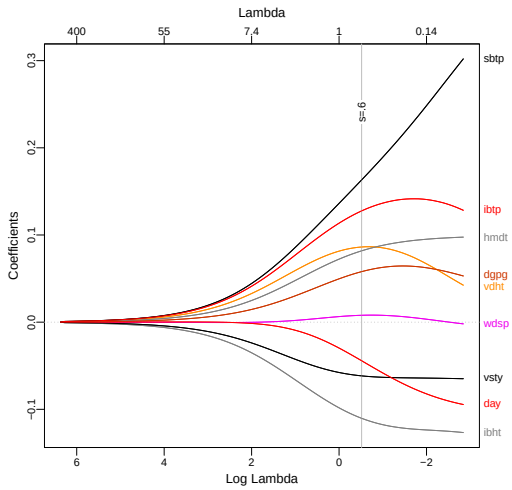
As we vary λ we get different ridge regression coefficients, the larger the λ the more shrunken. Here we plot them again



The vertical dashed line at $\lambda = 15$ marks the point above which ridge regression's MSE starts losing to that of linear regression.

An important thing to notice is that the gray coefficient paths are not exactly zero; they are shrunken, but still nonzero.

Example: Ozone



What if only a subset of the 9 predictor variables actually have non-zero coefficients?

LASSO, Tibshirani 1996

The LASSO estimate is defined as

$$\begin{aligned}\hat{\beta}^{\text{lasso}} &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \underbrace{\|y - X\beta|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}\end{aligned}$$

- The name “LASSO” is actually an acronym for: Least Absolute Selection and Shrinkage Operator.

The LASSO uses an $L1$ penalty $||\beta||_1$, whereas ridge regression uses a (squared) $L2$ penalty $||\beta||_2^2$. But even though these problems look similar, their solutions behave very differently.

LASSO

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

The tuning parameter λ controls the strength of the penalty, and (like ridge regression) we get $\beta^{LASSO} =$ the OLS regression estimate when $\lambda = 0$, and $\beta^{LASSO} = 0$, when $\lambda = \infty$.

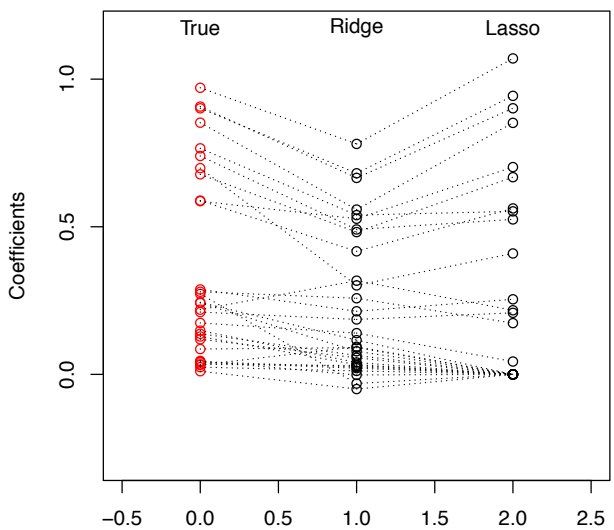
For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

The nature of the $L1$ penalty causes some coefficients to be shrunken to zero exactly.

This is what makes the LASSO substantially different from ridge regression: it is able to perform variable selection in the linear model. As λ increases, more coefficients are set to zero (less variables are selected), and among the nonzero coefficients, more shrinkage is employed.

Example: visual representation of LASSO coefficients

Same example: $n = 50, p = 30, \sigma^2 = 1$, 10 large true coefficients, 20 small. Here is a visual representation of LASSO vs. ridge coefficients (with the same degrees of freedom):



Estimation and Inference

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \underbrace{\|y - X\beta|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

- This problem does not have a closed form solution. It also does not have a unique solution in the general case.
- A solution can be found numerically using:
  - coordinate descent
  - ADMM (alternating direction method of multipliers)
  - LARS (least angle regression)
  - etc.
- LASSO models are generally not used for inference in practice (biased coefficients, non-unique solution problems, etc.)
- However, this method is very often used in the exploratory part of the regression analysis as it performs variable selection.

Important details

When including an intercept term in the model, we usually leave this coefficient unpenalized, just as with ridge regression.

Hence, the LASSO problem with intercept is

$$\hat{\beta}_0, \hat{\beta}^{\text{lasso}} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - \beta_0 \mathbb{1} - X\beta\|_2^2 + \lambda \|\beta\|_1$$

As seen before, if we center the columns of X, then the intercept estimate is $\hat{\beta}_0 = \overline{y}$. Therefore we typically center y, X and don't include an intercept term.

As with ridge regression, the penalty term $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ is not fair is the predictor variables are not on the same scale.
Hence, if we know that the variables are not on the same scale to begin with, we scale the columns of X (to have sample variance 1), and then we solve the LASSO problem.

Bias and variance of the LASSO

Although we can't write down explicit formulas for the bias and variance of the LASSO estimate (e.g., when the true model is linear), we know the general trend.

Recall that

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

Generally speaking:

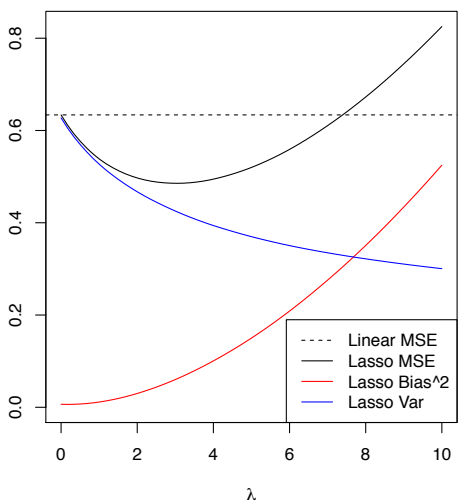
- The bias increases as λ (amount of shrinkage) increases.
- The variance decreases as λ (amount of shrinkage) increases.

What is the bias at $\lambda = 0$? The variance at $\lambda = \infty$?

In terms of prediction error (or mean squared error), the LASSO performs comparably to ridge regression.

Example: subset of small coefficients

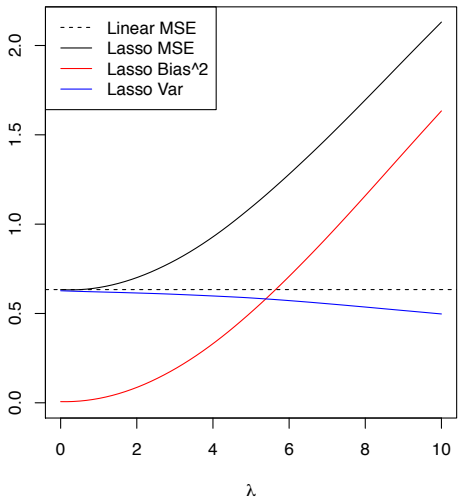
Example: $n = 50, p = 30, \sigma^2 = 1$; true coefficients: 10 large, 20 small.



The lasso: see the function `lars` in the package `lars`, or function `glmnet` in package `glmnet`.

Example: all moderate coefficients

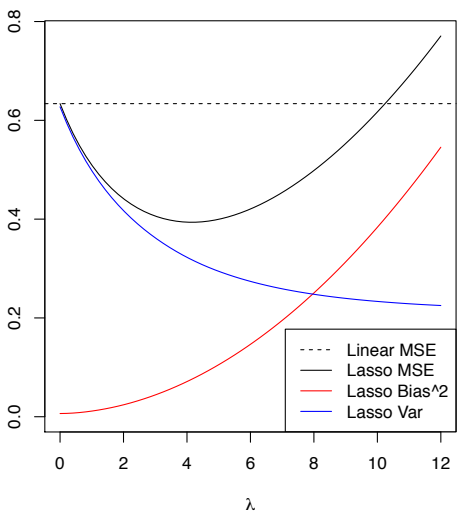
Example: $n = 50, p = 30, \sigma^2 = 1$; true coefficients: 30 moderately large.



Note that here, as opposed to ridge regression the variance doesn't decrease fast enough to make the LASSO favorable for small λ.

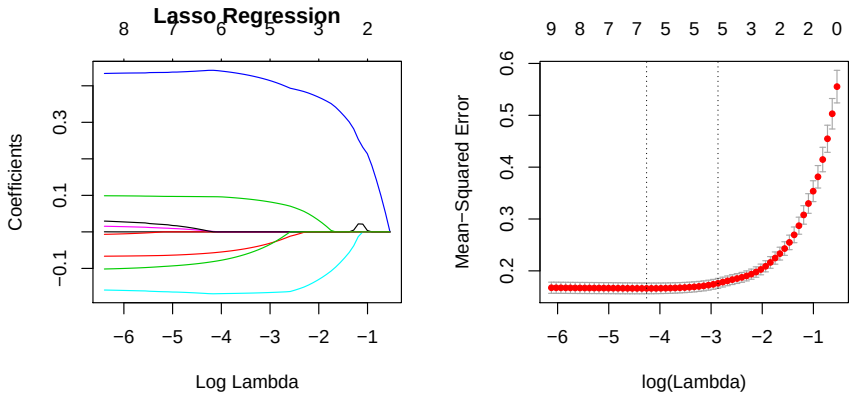
Example: subset of zero coefficients

Example: $n = 50, p = 30, \sigma^2 = 1$; true coefficients: 10 large, 20 zero.



Example: LASSO, ozone data

```
lasso <- glmnet(x=scale(as.matrix(d.ozone[,-10])),
   y=scale(d.ozone$logupo3,scale=F), alpha=1,intercept=F)
lasso.cv <- cv.glmnet(x=scale(as.matrix(d.ozone[,-10])),
   y=scale(d.ozone$logupo3,scale=F), alpha=1,intercept=F)
```



Example: LASSO, ozone data

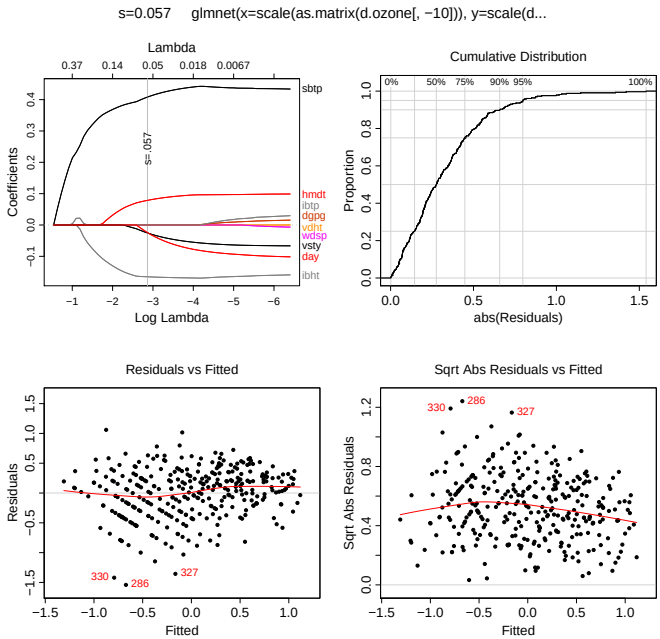
What about the coefficients and CV error?

```
## with lamda.1se
(Intercept)  .
vdht         .
wdsp         .
hmdt         0.07864410
sbtp         0.40825271
ibht        -0.16527328
dgpg         .
ibtp         .
vsty        -0.02552882
day         -0.02432932
## cv.error
 0.1760033
```

```
## with lamda.min
(Intercept)  .
vdht         .
wdsp         .
hmdt         0.096385636
sbtp         0.438918761
ibht        -0.167089677
dgpg         0.003413171
ibtp         0.006747144
vsty        -0.058278956
day         -0.083273006
## cv.error
 0.1661991
```

Model diagnostics

Same idea as in ridge regression.



Constrained form

It can be helpful to think of our two problems in constrained form:

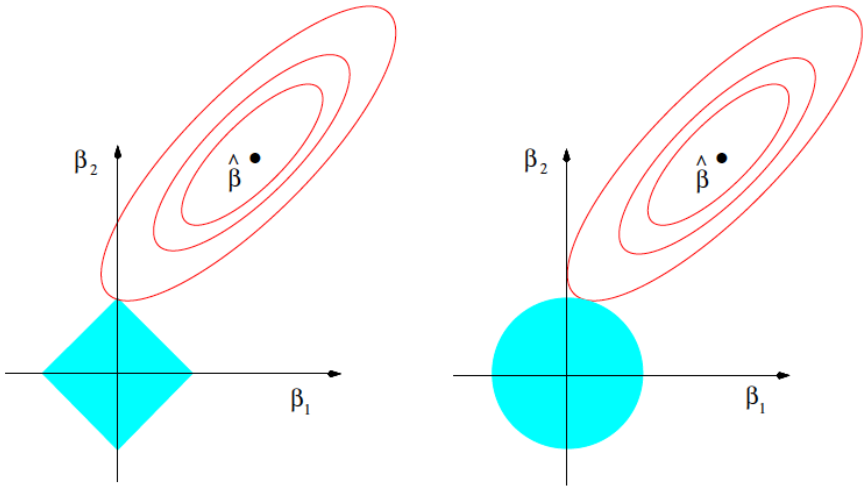
$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_2^2 \leq t$$

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_1 \leq t$$

Now t is the tuning parameter (instead λ). For any λ and corresponding solution in the previous formulation (sometimes called penalized form), there is a value of t such that the above constrained form has this same solution.

The usual linear regression estimate solves the unconstrained least squares problem and the LASSO and ridge estimates constrain the coefficient vector to lie in some geometric shape centered around the origin. This generally reduces the variance because it keeps the estimate close to zero. But the grometric shape of the penalty term really matters!

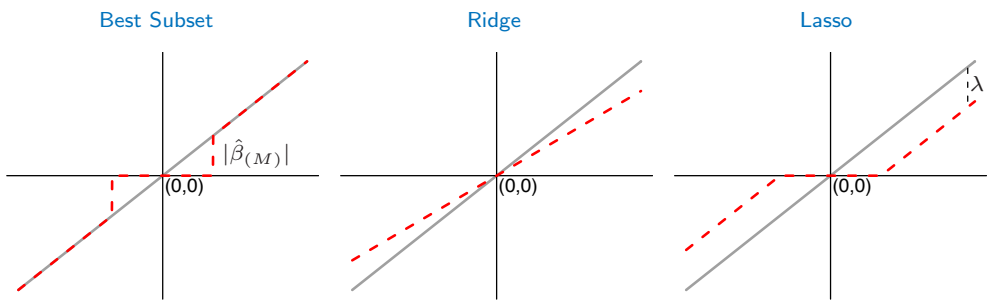
Why does the LASSO give zero coefficients?



(From page 71 of ESL)

Comparison of the bias for $X'X = I$

Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_{(M)}\rvert)$
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$



(From page 71 of ESL)

Elastic net

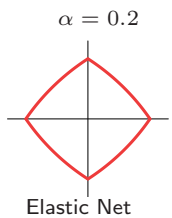
LASSO struggles when covariates are correlated and tends to pick only one of them even if both are related to the outcome.
There is also an issue of using LASSO with categorical variables. Remember: you want to either select all dummy variables representing the levels of your categorical predictor, or none of them.
Solutions:

- We can form groups of correlated variables and run group-lasso (will not be discussed in the course).
- We can let "the data decide for us" by altering the penalty as follows:

$$\min_{\beta} \frac{1}{2}||y - X\beta||^2 + (1-\alpha)\lambda\frac{1}{2}||\beta||_2^2 + \alpha\lambda||\beta||_1$$

As you can see, this uses both an $L1$ and an $L2$ penalty on β. This penalty strategy is called the **elastic net**.

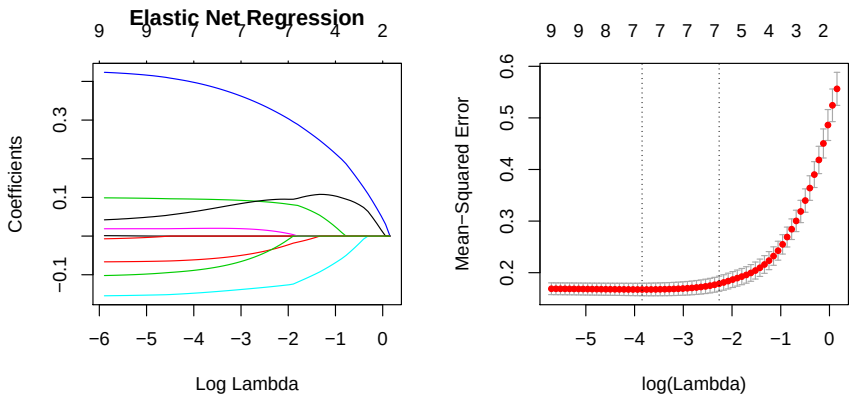
Elastic net



- The parameter α determines the mix of the penalties, and is often pre-chosen on qualitative grounds.
- What tends to happen is that the bigger you make the $L2$ penalty (small α) the more elastic net will add groups of variables together into the model.
- Note that we can also choose α (and λ) by performing a 2-dimensional cross-validation.

Example: Elastic net, $\alpha = 0.5$, ozone data

```
elastic.net <- glmnet(x=scale(as.matrix(d.ozone[,-10])),
   y=scale(d.ozone$logupo3,scale=F), alpha=.5,intercept=F)
elastic.net.cv <- cv.glmnet(x=scale(as.matrix(d.ozone[,-10])),
   y=scale(d.ozone$logupo3,scale=F), alpha=.5,intercept=F)
```



Example: Elastic net, $\alpha = 0.5$, ozone data

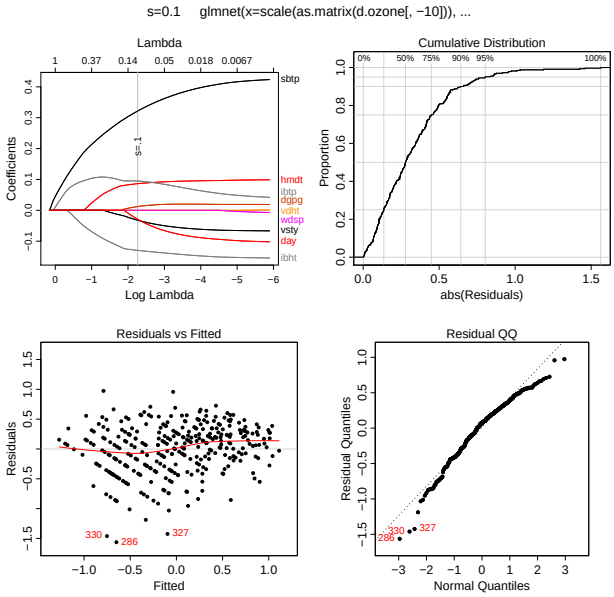
What about the coefficients and CV error?

```
## with lamda.1se
(Intercept)  .
vdht         .
wdsp         .
hmdt         0.08625966
sbtp         0.32159003
ibht        -0.12967390
dgpg         0.01209648
ibtp         0.09484161
vsty        -0.03210367
day         -0.02896496
## cv.error
 0.1786288
```

```
## with lamda.min
(Intercept)  .
vdht         .
wdsp         .
hmdt         0.09547872
sbtp         0.39352728
ibht        -0.14627825
dgpg         0.01987310
ibtp         0.06623956
vsty        -0.06037078
day         -0.08649907
## cv.error
 0.1674589
```

Model diagnostics

Same idea as in ridge and LASSO regression.



We can also select both α and λ using a 2-dim. CV search. See R code.