Lecture 28: Penalized Regression

Pratheepa Jeganathan

12/02/2019

Recap

- What is a regression model?
- Descriptive statistics – graphical
- Descriptive statistics – numerical
- Inference about a population mean
- Difference between two population means
- Some tips on R
- Simple linear regression (covariance, correlation, estimation, geometry of least squares)
    - Inference on simple linear regression model
    - Goodness of fit of regression: analysis of variance.
    - F-statistics.
    - Residuals.
    - Diagnostic plots for simple linear regression (graphical methods).

Recap

- Multiple linear regression
  - Specifying the model.
  - Fitting the model: least squares.
  - Interpretation of the coefficients.
  - Matrix formulation of multiple linear regression
  - Inference for multiple linear regression
    - T-statistics revisited.
    - More F statistics.
    - Tests involving more than one β.
- Diagnostics – more on graphical methods and numerical methods
  - Different types of residuals
  - Influence
  - Outlier detection
  - Multiple comparison (Bonferroni correction)
  - Residual plots:
    - partial regression (added variable) plot,
    - partial residual (residual plus component) plot.

Recap

- Adding qualitative predictors
  - Qualitative variables as predictors to the regression model.
  - Adding interactions to the linear regression model.
  - Testing for equality of regression relationship in various subsets of a population
- ANOVA
  - All qualitative predictors.
  - One-way layout
  - Two-way layout
- Transformation
  - Achieving linearity
  - Stabilize variance
  - Weighted least squares
- Correlated Errors
  - Generalized least squares
- Bootstrapping linear regression
- Selection

Outline

- Collinearity
- Bias-variance trade-off
- Penalized Regression
  - Ridge
  - LASSO
  - Elastic net

Collinearity

Collinearity

- Existence of strong linear relationships among the predictor variables (collinear data, collinearity, or multicollinearity)
- Consequences
  - impossible to estimate the unique effects of individual variables in the regression equation.
  - regression coefficients have large sampling errors
- Not a modeling error
- Detecting collinearity (**CH** Chapter 9.4)
  - Large values of pairwise correlation coefficient, the regression results do not conform to prior expectations
  - Variance inflation factors, condition indices

Working with Collinear data

Standardization (**CH** Chapter 3.6)

- If collinearity is present due to different units of measurements of variables, we can use standardization to reduce the problem

Principal components regression (**CH** Chapter 10.2, 10.3)

- Use principal components method to transform $X_1, X_2, \cdots, X_p$ to a set of p orthogonal variables $C_1, C_2, \cdots, C_p$.
- Regressing Y on to $C_1, C_2, \cdots, C_p$.

Penalization

- Impose contstraints on the regression coefficients
  - For example, $\beta_1 + \beta_2 = 1$ and use ordinary least squares (OLS) method to estimate the regression coefficients.
- Compute biased estimators but tend to have more precision than the OLS estimators.
  - produce more precision in the estimated coefficients and smaller predicton error: sum of squares residuals is not small.
  - **predictions are generated using data other than those used for estimation**.

Bias-variance tradeoff

Bias-variance tradeoff

- One goal of a regression analysis is to "build" a model that predicts well: AIC or C_p or Cross-validation selection criteria based on this.
- This is slightly different than the goal of making inferences about β that we've focused on so far.
- What does "predict well" mean?

$$\begin{aligned} MSE_{pop}(\mathcal{M}) &= \mathbb{E}\left((Y_{new} - \widehat{Y}_{new,\mathcal{M}}(X_{new}))^2\right) \\ &= \mathrm{Var}(Y_{new}) + \mathrm{Var}(\widehat{Y}_{new,\mathcal{M}}) + \\ &\qquad \mathrm{Bias}(\widehat{Y}_{new,\mathcal{M}})^2. \end{aligned}$$

- Can we take an estimator for a model $\mathcal{M}$ and make it better in terms of MSE?

Shrinkage estimators: one sample problem

1. Generate $Y_{100\times 1} \sim N(\mu \cdot 1, 5^2 I_{100\times 100})$, with $\mu = 0.5$.
2. For $0 \leq \alpha \leq 1$, set $\hat{Y}(\alpha) = \alpha \bar{Y}$.
3. Compute $MSE(\hat{Y}(\alpha)) = \frac{1}{100}\sum_{i=1}^{100}(\hat{Y}_\alpha - 0.5)^2$
4. Repeat 1000 times, plot average of $MSE(\hat{Y}(\alpha))$.

For what value of α is $\hat{Y}(\alpha)$ unbiased?

Is this the best estimate of μ in terms of MSE?

Shrinkage estimators: one sample problem (simulation)

```r
mu = 0.5
sigma = 5
nsample = 100
ntrial = 1000

MSE = function(mu.hat, mu) {
  return(sum((mu.hat - mu)^2) / length(mu))
}
```

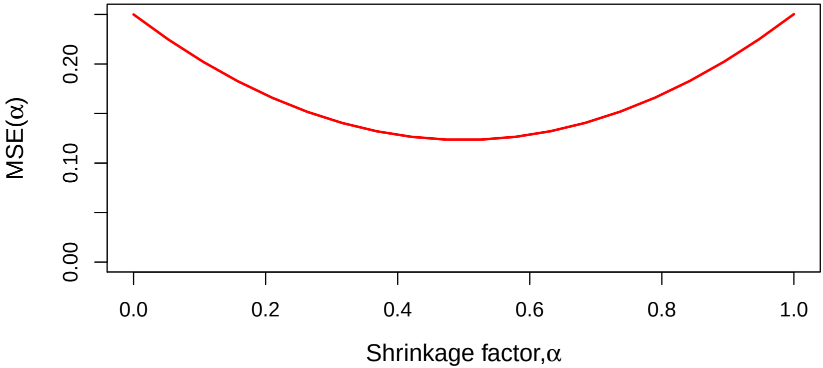
Shrinkage estimators: one sample problem (simulation)

```
alpha = seq(0, 1,length=20)
mse = numeric(length(alpha))
bias = (1 - alpha) * mu
variance = alpha^2 * 25 / 100

for (i in 1:ntrial) {
  Z = rnorm(nsample) * sigma + mu
  for (j in 1:length(alpha)) {
    mse[j] = mse[j] +
      MSE(alpha[j] * mean(Z) * rep(1, nsample),
          mu * rep(1, nsample))
  }
}
mse = mse / ntrial
```

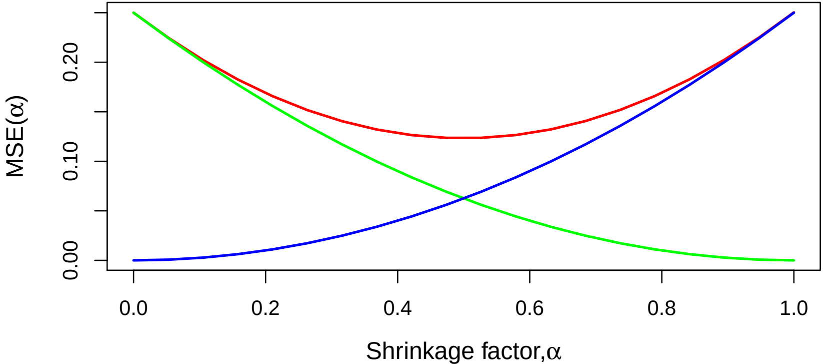
Shrinkage estimators: one sample problem (simulation)

```
plot(alpha, mse, type='l', lwd=2, col='red',
     ylim=c(0, max(mse)),
     xlab=expression(paste('Shrinkage factor,', alpha)),
     ylab=expression(paste('MSE(', alpha, ')')),
     cex.lab=1.2)
```



Shrinkage estimators: one sample problem (simulation)

```
plot(alpha, mse, type='l', lwd=2, col='red',
     ylim=c(0, max(mse)),
     xlab=expression(paste('Shrinkage factor,', alpha)),
     ylab=expression(paste('MSE(', alpha, ')')),
     cex.lab=1.2)
lines(alpha, bias^2, col='green', lwd=2)
lines(alpha, variance, col='blue', lwd=2)
```



Shrinkage & Penalties

- Shrinkage can be thought of as "constrained" or "penalized" minimization.
- Constrained form:

$$\text{minimize}_\mu \sum_{i=1}^n (Y_i - \mu)^2 \quad \text{subject to } \mu^2 \leq C$$

- Lagrange multiplier form: equivalent to

$$\widehat{\mu}_\lambda = \text{argmin}_\mu \sum_{i=1}^n (Y_i - \mu)^2 + \lambda \cdot \mu^2$$

  for some $\lambda = \lambda_C$.
- As we vary λ we solve all versions of the constrained form.

Solving for $\widehat{\mu}_\lambda$

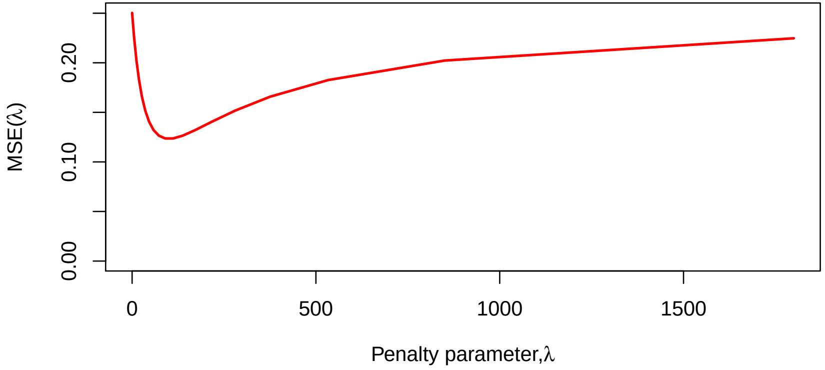
- Differentiating: $-2\sum_{i=1}^n (Y_i - \widehat{\mu}_\lambda) + 2\lambda\widehat{\mu}_\lambda = 0$
- Solving $\widehat{\mu}_\lambda = \frac{\sum_{i=1}^n Y_i}{n+\lambda} = \frac{n}{n+\lambda}\overline{Y}$.
- As $\lambda \to 0$, $\widehat{\mu}_\lambda \to \overline{Y}$.
- As $\lambda \to \infty$ $\widehat{\mu}_\lambda \to 0$.

We see that $\widehat{\mu}_\lambda = \bar{Y} \cdot \left(\frac{n}{n+\lambda}\right)$.

In other words, considering all penalized estimators traces out the MSE curve above.

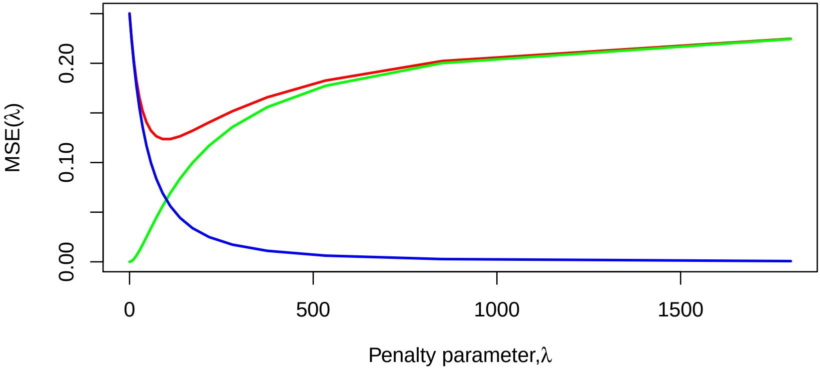
Solving for $\widehat{\mu}_\lambda$

```
lam = nsample / alpha - nsample
plot(lam, mse, type='l', lwd=2, col='red',
     ylim=c(0, max(mse)),
     xlab=expression(paste('Penalty parameter,',  lambda)),
     ylab=expression(paste('MSE(', lambda, ')')))
```



Solving for $\widehat{\mu}_\lambda$

```
plot(lam, mse, type='l', lwd=2, col='red',
     ylim=c(0, max(mse)),
     xlab=expression(paste('Penalty parameter,',  lambda)),
     ylab=expression(paste('MSE(', lambda, ')')))
lines(lam, bias^2, col='green', lwd=2)
lines(lam, variance, col='blue', lwd=2)
```



How much to shrink?

- In our one-sample example,
- $$\begin{aligned} MSE_{pop}(\alpha) &= \mathrm{Var}(\alpha\bar{Y}) + \mathrm{Bias}(\alpha\bar{Y})^2 + \mathrm{Var}(Y_{new}) \\ &= \frac{\alpha^2\sigma^2}{n} + \mu^2(1-\alpha)^2 + \mathrm{Var}(Y_{new}) \end{aligned}$$
- Differentiating and solving:

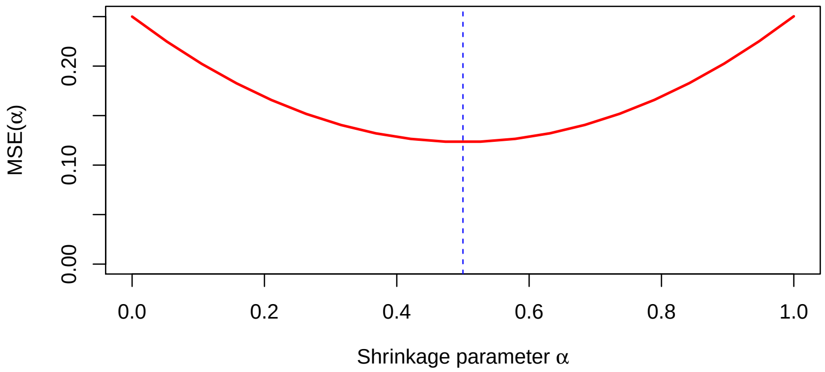
$$\begin{aligned} 0 &= -2\mu^2(1-\alpha^*) + 2\frac{\alpha^*\sigma^2}{n} \\ \alpha^* &= \frac{\mu^2}{\mu^2+\sigma^2/n} = \frac{(\mu/(\sigma/\sqrt{n}))^2}{(\mu/(\sigma/\sqrt{n}))^2+1} \\ &= \frac{0.5^2}{0.5^2+25/100} = 0.5 \end{aligned}$$

We see that the optimal α depends on the unknown $SNR = \mu/(\sigma/\sqrt{n})$. Value is 1/8.

In practice we might hope to estimate MSE with cross-validation.

- Let's see how our theoretical choice of α matches the MSE on our 100 sample.

```r
plot(alpha, mse, type='l', lwd=2, col='red',
     ylim=c(0, max(mse)),
     xlab=expression(paste('Shrinkage parameter ', alpha)),
     ylab=expression(paste('MSE(', alpha, ')')))
abline(v=mu^2/(mu^2+sigma^2/nsample), col='blue', lty=2)
```



Penalties & Priors

- Minimizing $\sum_{i=1}^{n}(Y_i - \mu)^2 + \lambda\mu^2$ is similar to computing "MLE" of μ if the likelihood was proportional to

$$\exp\left(-\frac{1}{2\sigma^2}\left(\|Y - \mu\|_2^2 + \lambda\mu^2\right)\right).$$

- If $\lambda = m$, an integer, then $\widehat{\mu}_\lambda$ is the sample mean of $(Y_1, \dots, Y_n, 0, \dots, 0) \in \mathbb{R}^{n+m}$.
- This is equivalent to adding some data with $Y = 0$.
- To a Bayesian, this extra data is a *prior distribution* and we are computing the so-called Maximum A Posteriori (*MAP*) or posterior mode.

AIC as penalized regression

- Model selection with C_p (or AIC with σ^2 assumed known) is a version of penalized regression.
- The best subsets version of AIC (which is not exactly equivalent to *step*)

$$\hat{\beta}_{AIC} = \text{argmin}_{\beta} \frac{1}{\sigma^2} \|Y - X\beta\|_2^2 + 2\|\beta\|_0$$

  where

$$\|\beta\|_0 = \# \{j : \beta_j \neq 0\}$$

  is called the ℓ_0 norm.
- The ℓ_0 penalty can be thought of as a measure of *complexity* of the model. Most penalties are similar versions of *complexity*.

Penalized regression in general

Penalized regression in general

- Not all biased models are better – we need a way to find "good" biased models.
- Inference (F, χ^2 tests, etc) is not quite exact for biased models. Though, there has been some recent work to address the issue of post-selection inference, at least for some penalized regression problems.
- Heuristically, "large β" (measured by some norm) is interpreted as "complex model".
  - Goal is really to penalize "complex" models, i.e. Occam's razor.
- If truth really is complex, this may not work! (But, it will then be hard to build a good model anyways ... (statistical lore))

Ridge regression

Ridge regression

- Assume that columns $(X_j)_{1 \leq j \leq p}$ have zero mean, and standard deviation (SD) 1 and Y has zero mean.
- This is called the *standardized model*.
- The ridge estimator is

$$\begin{aligned} \hat{\beta}_\lambda &= \text{argmin}_\beta \frac{1}{2n} \|Y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2 \\ &= \text{argmin}_\beta MSE_\lambda(\beta) \end{aligned}$$

- Corresponds (through Lagrange multiplier) to a quadratic constraint on β's.
- This is the natural generalization of the penalized version of our shrinkage estimator.

Solving the normal equations

- Normal equations

$$\frac{\partial}{\partial \beta_l} MSE_\lambda(\beta) = -\frac{1}{n}(Y - X\beta)^T X_l + \lambda \beta_l$$

-

$$-\frac{1}{n}(Y - X\widehat{\beta}_\lambda)^T X_l + \lambda \widehat{\beta}_{l,\lambda} = 0, \qquad 1 \leq l \leq p$$

- In matrix form

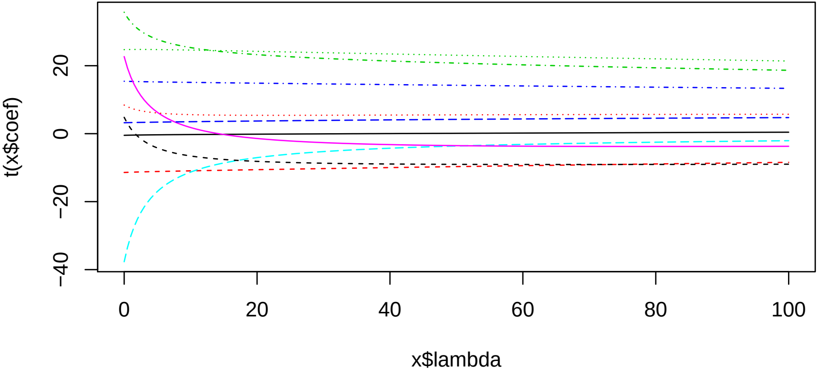
$$-\frac{X^T Y}{n} + \left(\frac{X^T X}{n} + \lambda I\right)\widehat{\beta}_\lambda = 0.$$

- Or

$$\widehat{\beta}_\lambda = \left(\frac{X^T X}{n} + \lambda I\right)^{-1} \frac{X^T Y}{n}.$$

Ridge regression

```
library(lars)
data(diabetes)
library(MASS)
diabetes.ridge = lm.ridge(diabetes$y ~ diabetes$x,
                          lambda=seq(0, 100, 0.5))
plot(diabetes.ridge, lwd=3)
```



Choosing λ

- If we knew $E[MSE_\lambda]$ as a function of λ then we would simply choose the λ that minimizes it.
- To do this, we need to estimate it.
- A popular method is cross-validation as a function of λ. Breaks the data up into smaller groups and uses part of the data to predict the rest.
- We saw this in diagnostics (Cook's distance measured the fit with and without each point in the data set) and model selection.

K-fold cross-validation for penalized model

- Fix a model (i.e. fix λ). Break data set into K approximately equal sized groups $(G_1, \ldots, G_K)$.
- for (i in 1:K)
  - Use all groups except G_i to fit model, predict outcome in group G_i based on this model $\widehat{Y}_{j(i),\lambda}, j \in G_i$.
- Estimate $CV(\lambda) = \frac{1}{n}\sum_{i=1}^{K}\sum_{j\in G_i}(Y_j - \widehat{Y}_{j(i),\lambda})^2$.

K-fold cross-validation for penalized model

- Here is a function to estimate the CV for our one parameter example.
- In practice, we only have one sample to form the CV curve.
- In this example below, we will compute the average CV error for 500 trials to show that it is roughly comparable in shape to the MSE curve.

K-fold cross-validation for penalized model

```
CV = function(Z, alpha, K=5) {
    cve = numeric(K)
    n = length(Z)
    for (i in 1:K) {
        g = seq(as.integer((i-1)*n/K)+1,
                as.integer((i*n/K)))
        mu.hat = mean(Z[-g]) * alpha
        cve[i] = sum((Z[g]-mu.hat)^2)
    }
    return(c(sum(cve)/n, sd(cve)/sqrt(n)))
}
```

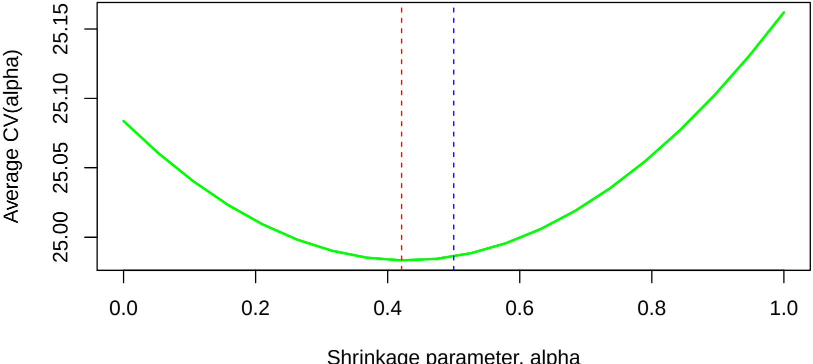
K-fold cross-validation for penalized model

- Let's see how the parameter chosen by 5-fold CV compares to our theoretical choice.

```
alpha = seq(0.0,1,length=20)
mse = numeric(length(alpha))
avg.cv = numeric(length(alpha))
for (i in 1:ntrial) {
     Z = rnorm(nsample) * sigma + mu
     for (j in 1:length(alpha)) {
         current_cv = CV(Z, alpha[j])
         avg.cv[j] = avg.cv[j] + current_cv[1]
     }
}
avg.cv = avg.cv/ntrial
```

K-fold cross-validation for penalized model

```
plot(alpha, avg.cv, type='l', lwd=2, col='green',
     xlab='Shrinkage parameter, alpha',
     ylab='Average CV(alpha)')
abline(v=mu^2/(mu^2+sigma^2/nsample),
       col='blue', lty=2)
abline(v=min(alpha[avg.cv == min(avg.cv)]),
       col='red', lty=2)
```



K-fold cross-validation for penalized model

- The curve above shows what would happen if we could repeat this and average over many samples. In reality, we only get one sample.

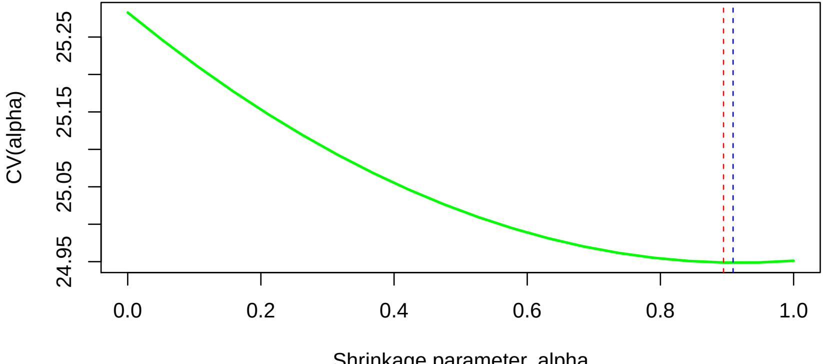
K-fold cross-validation for penalized model

- Let's see what one curve looks like on our sample.
- This is the result we might get in practice on a given data set.

```
cv = numeric(length(alpha))
cv.sd = numeric(length(alpha))
nsample = 1000
Z = rnorm(nsample) * sigma + mu
for (j in 1:length(alpha)) {
    current_cv = CV(Z, alpha[j])
    cv[j] = current_cv[1]
    cv.sd[j] = current_cv[2]
}
```

K-fold cross-validation for penalized model

```
plot(alpha, cv, type='l', lwd=2, col='green',
     xlab='Shrinkage parameter, alpha',
     ylab='CV(alpha)', xlim=c(0,1))
 abline(v=mu^2/(mu^2+sigma^2/nsample),
        col='blue', lty=2)
 abline(v=min(alpha[cv == min(cv)]),
        col='red', lty=2)
```

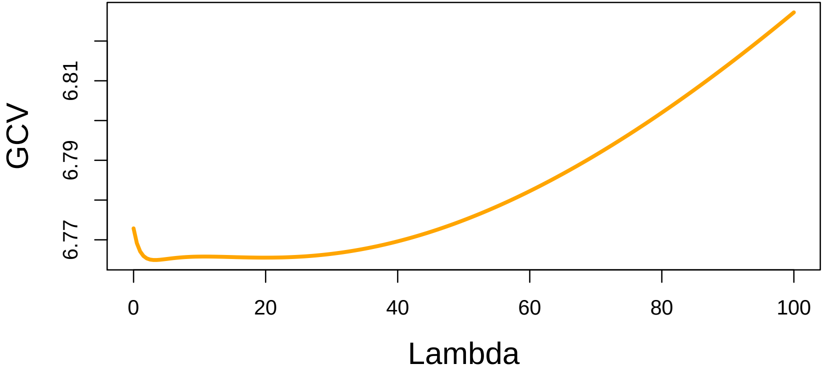


Generalized Cross Validation

- A computational shortcut for *n*-fold cross-validation (also known as leave-one out cross-validation).
- Let $S_\lambda = X(X^TX + n\lambda I)^{-1}X^T$ be the matrix in ridge regression that computes $\hat{Y}_\lambda$
- Then $GCV(\lambda) = \frac{\|Y - S_\lambda Y\|^2}{n - \mathrm{Tr}(S_\lambda)}$.
- The quantity $\mathrm{Tr}(S_\lambda)$ can be thought of as the *effective degrees of freedom* for this choice of λ.

GCV for Ridge regression

```r
par(cex.lab=1.5)
plot(diabetes.ridge$lambda, diabetes.ridge$GCV,
     xlab='Lambda', ylab='GCV', type='l',
     lwd=3, col='orange')
```



GCV for Ridge regression

- Find λ

```
select(diabetes.ridge)
```

```
## modified HKB estimator is 5.462251
## modified L-W estimator is 7.641667
## smallest value of GCV  at 3
```

Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO

- Another popular penalized regression technique.
- Use the standardized model.
- The LASSO estimate is

$$\hat{\beta}_\lambda = \text{argmin}_\beta \frac{1}{2n} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

where

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$$

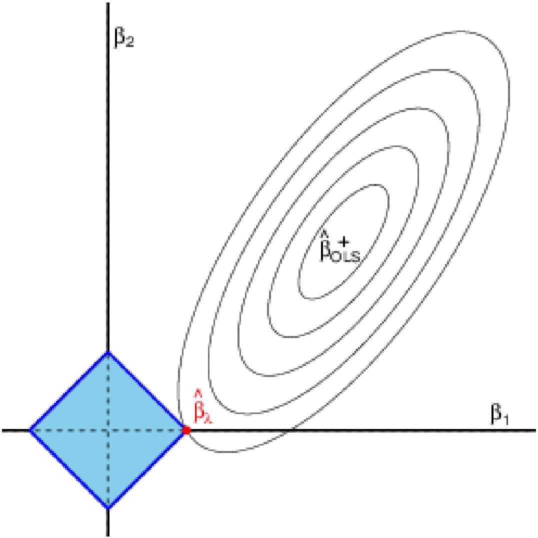
is the ℓ_1 norm.

- Corresponds (through Lagrange multiplier) to an ℓ_1 constraint on β's.

LASSO

- In theory and practice, it works well when many β_j's are 0 and gives "sparse" solutions unlike ridge.
- It is a (computable) approximation to the best subsets AIC model.
- It is computable because the minimization problem is a convex problem.

Why do we get sparse solutions with the LASSO?



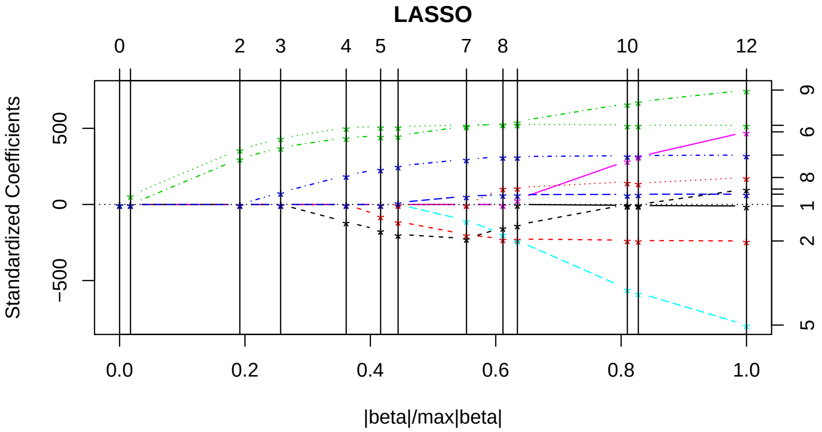
Example (LASSO)

`diabetes` data frame has Y a numeric response and $\mathbf{X}$ has 10 predictor variables.

```r
library(lars)
data(diabetes)
diabetes.lasso = lars(diabetes$x, diabetes$y,
                      type='lasso')
```

Example (LASSO)

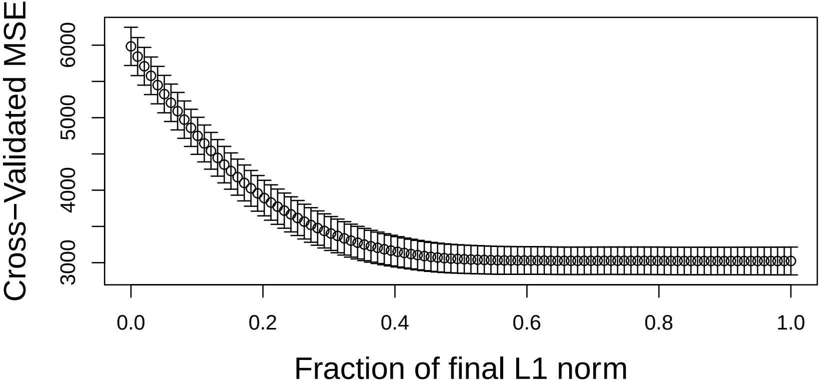
```r
plot(diabetes.lasso, xvar = "norm")
```



Cross-validation for the LASSO

- The `lars` package has a built in function to estimate CV.

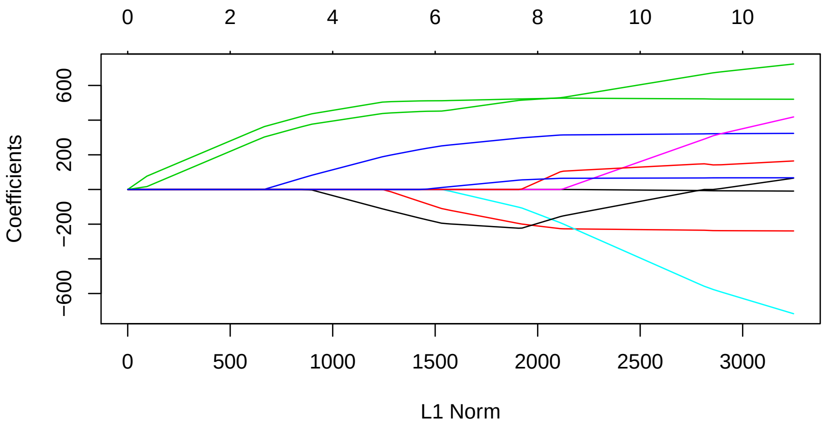
```
par(cex.lab=1.5)
cv.lars(diabetes$x, diabetes$y, K=10, type='lasso')
```



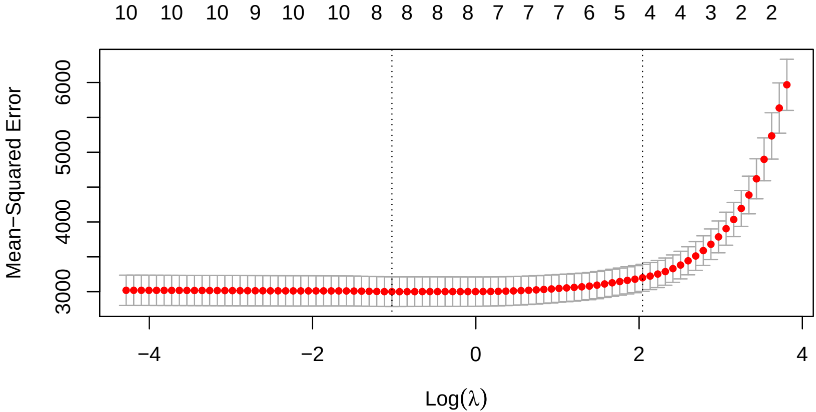
glmnet

```r
library(glmnet)
G = glmnet(diabetes$x, diabetes$y)
```

```r
plot(G)
```


0
2
4
6
8
10
10
600
200
−200
−600
Coefficients
0
500
1000
1500
2000
2500
3000
L1 Norm

```r
plot(cv.glmnet(diabetes$x, diabetes$y))
```




```r
cv.glmnet(diabetes$x, diabetes$y)$lambda.1se
```

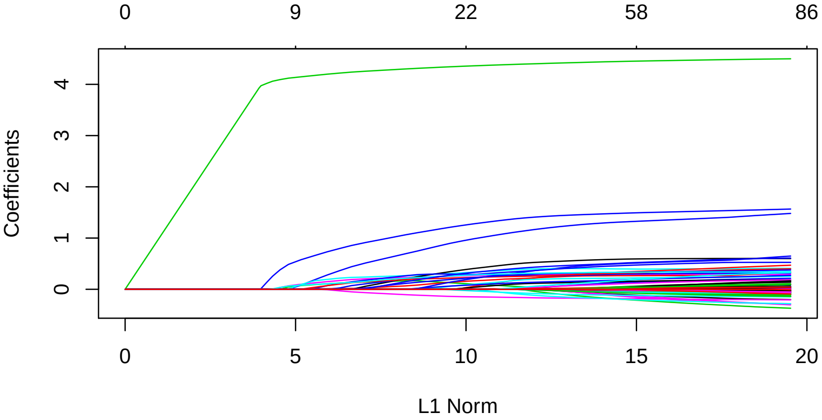
```
## [1] 5.314486
```

HIV example

```
X_HIV = read.table('http://stats191.stanford.edu/data/NRTI_
                   header=FALSE, sep=',')
Y_HIV = read.table('http://stats191.stanford.edu/data/NRTI_
                   header=FALSE, sep=',')
set.seed(0)
Y_HIV = as.matrix(Y_HIV)[,1]
X_HIV = as.matrix(X_HIV)
```

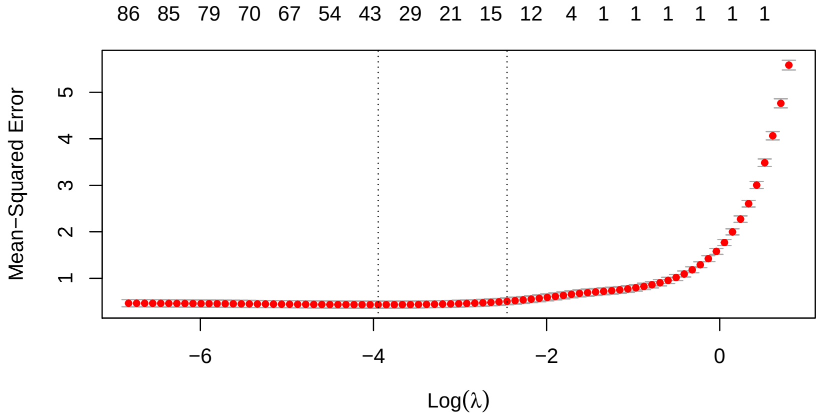
HIV example

```
library(glmnet)
G = glmnet(X_HIV, Y_HIV)
plot(G)
```



HIV example

```r
CV = cv.glmnet(X_HIV, Y_HIV)
plot(CV)
```



Extracting coefficients from glmnet

```
beta.hat = coef(G, s=CV$lambda.1se)
beta.hat # might want to use as.numeric(beta.hat) instead o
```

```
## 92 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)  0.670844817
## V1           .
## V2           .
## V3           .
## V4           .
## V5           .
## V6           .
## V7           .
## V8           0.004062047
## V9           .
## V10          .
## V11          .
## V12          .
```

Extracting coefficients from glmnet

- Number of non-zero coefficients

```r
sum(abs(beta.hat[,1]) >0)
```

```
## [1] 17
```

Elastic Net

Elastic Net

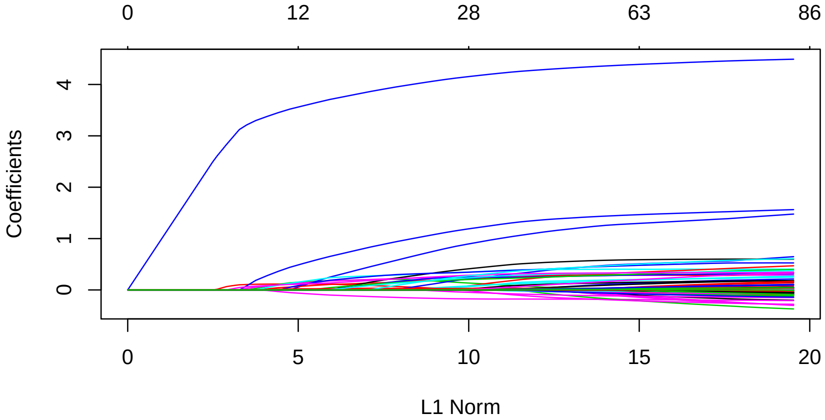
- Mix between LASSO and ridge regression.
- Sometimes a more stable estimator than LASSO.
- The ENET estimator is

$$\hat{\beta}_{\lambda,\alpha} = \mathrm{argmin}_{\beta} \frac{1}{2n} \|Y - X\beta\|_2^2 + \lambda \left(\alpha \|\beta\|_1 + (1-\alpha)\|\beta\|_2^2 \right).$$

Elastic Net (Example)

- Coefficient path for $\alpha = 0.25$

```
Enet = glmnet(X_HIV, Y_HIV, alpha=0.25)
plot(Enet)
```



Elastic Net (Example)

```r
CV = cv.glmnet(X_HIV, Y_HIV, alpha=0.25)
beta.hat = coef(Enet, s=CV$lambda.1se)
beta.hat
```

```
## 92 x 1 sparse Matrix of class "dgCMatrix"
##                           1
## (Intercept)  0.780910769
## V1           .
## V2           .
## V3           .
## V4           .
## V5           .
## V6           .
## V7           .
## V8           0.066467999
## V9           .
## V10          .
## V11          .
```

Extracting coefficients from `glmnet`

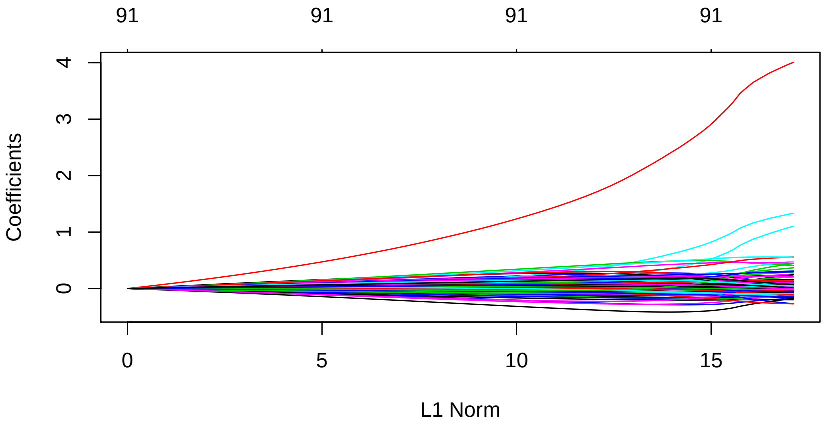
- Number of non-zero coefficients

```
sum(abs(beta.hat[,1]) > 0)
```

```
## [1] 21
```

Ridge regression (glmnet)

```
plot(glmnet(X_HIV, Y_HIV, alpha=0))
```



Reference

- More on the penalized regression: An Introduction to Statistical Learning
- **CH** Chapter 9 and 10.
- Lecture notes of Jonathan Taylor .