Biostatistics 140.754
Advanced Methods in Biostatistics IV

Jeffrey Leek

Assistant Professor
Department of Biostatistics
jleek@jhsph.edu

Lecture 12

Tip + Paper

Tip: As a statistician the results of your analysis will often determine whether/where a co-investigators results can be published. There will often be a lot of pressure to make the results "significant", especially if if generating the data took a lot of time/$$. If you do the analysis and nothing is significant, be sure to stick to your guns (even as a young statistician). **But**, do be sympathetic and make an effort to really understand the data and whether you can say something interesting with it. It can be all too easy to say, "not significant, go collect more samples". Try to be sympathetic to your co-investigators and be a facilitator to the extent you can.

Paper of the Day: "The MPG Illusion"
http://www.sciencemag.org/content/320/5883/1593.summary

An email you may receive

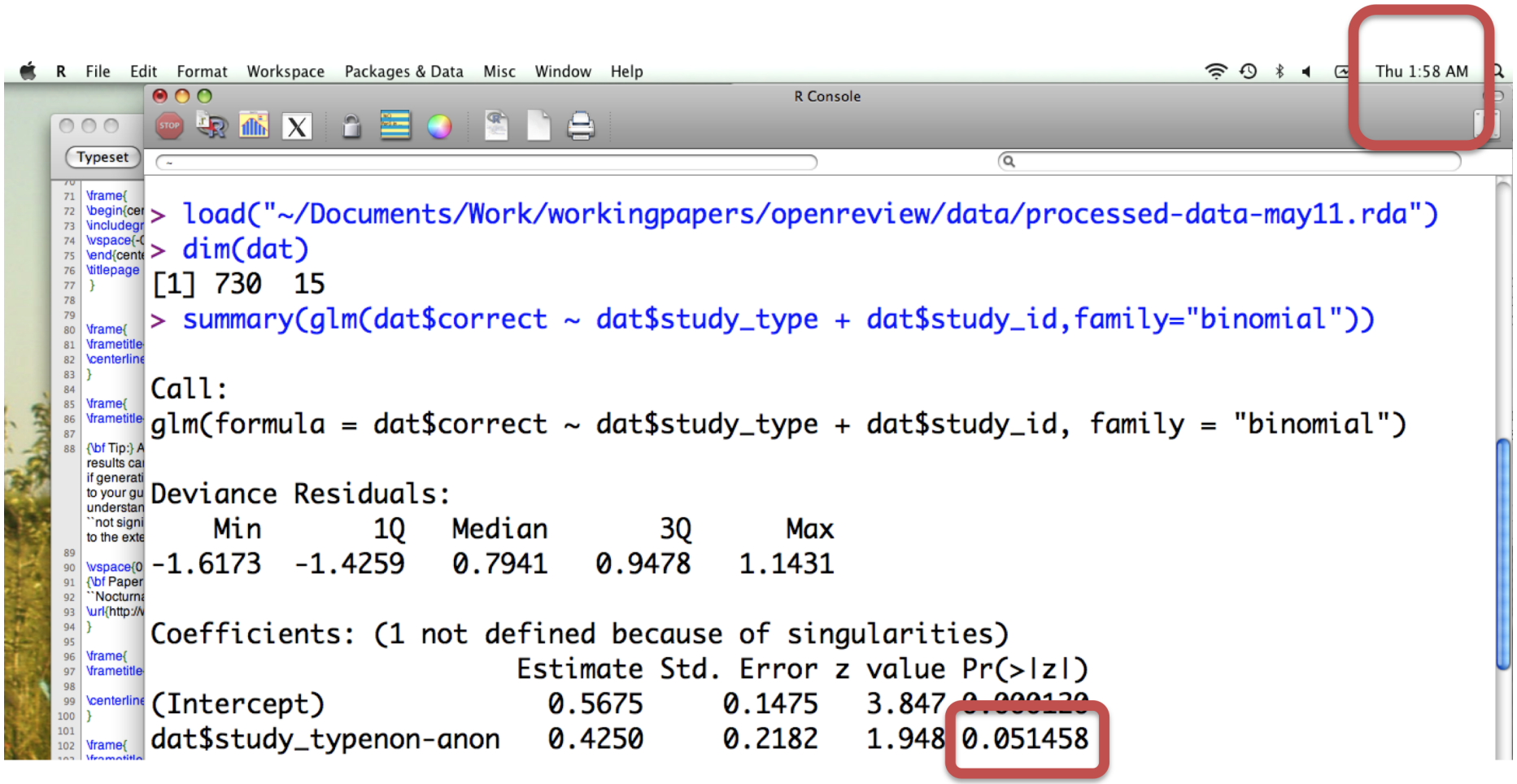
Subject: A curse on you and your progeny!!!

Ingo:

Curse you, Ingo! Yet another disappearing act!

The association between flame broiled food consumption and breast cancer disappears in the imputed dataset (see below). I'm beginning to hate this imputation stuff! I much prefer biased data. The findings are more interesting (and more publishable).

Why your instructor is so sympathetic



Today's Outline

- Ridge regression
- Lasso
- Cross validation

Many slides borrowed from Art Owen, Jonathan Taylor

Shrinkage and Penalties

Possibly "the" classic shrinkage result is the James-Stein estimator. Suppose that we have vectors Y_i $Y_i \sim N(\theta, \sigma^2 I)$ where we would like to estimate $\hat{\theta}$ on the basis of the Y variables and σ^2 is known.

Then we know what estimator we would usually choose $\hat{\theta} = \bar{Y}$.

Stein showed that if the goal is to minimize the mean squared error $\mathbb{E}[(\theta - \hat{\theta})^2]$ and we have $n \geq 3$ observations, we can always do better than the mean with estimators like this:

$$\hat{\theta}_{JS} = \left(1 - \frac{(m-2)\sigma^2}{||\bar{Y}||^2}\right) \bar{Y}$$

Even more surprising, if we consider any direction ν

$$\hat{\theta}_{JS} = \left(1 - \frac{(m-2)\sigma^2}{||\bar{Y} - \nu||^2}\right) (\bar{Y} - \nu) + \nu$$

also has lower MSE than the mean!!

Shrinkage and Penalties

Shrinkage can be thought of as "constrained" minimization.

Minimize:

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

Differentiating:

$$-2\sum_{i=1}^{n}(Y_i - \hat{\mu}_c) + 2\lambda_c\hat{\mu}_c = 0$$

Finally

$$\hat{\mu}_c = \frac{\sum_{i=1}^{n} Y_i}{n+\lambda_c} = K_c\bar{Y} \text{ where } K_c < 1$$

The precise form of λ_c is unimportant: as $c \to 0$, $\hat{\mu} \to \bar{Y}$, as $c \to \infty$, $\hat{\mu}_c \to 0$.

Shrinkage and Penalties

- Not all biased models are better - we need to find "good" biased models
- Generalized one-sample problem: penalize large values of β this should lead to "multivariate" shrinkage
- Heuristically, "large β" is interpreted as "complex model". Goal is really to penalize "complex" models, i.e. Occam's razor.
- For many "good" penalties there is an equivalent Bayesian interpretation
- If the truth is really complex, this may not work! (But it will then be hard to build a good model anyway...)

Regularization for regression

If the β_j's are unconstrained:

- They can explode
- And hence are susceptible to very high variance

To control variance, we might regularize/shrink the coefficients. One example constraint is to minimize:

$$\sum_{j=1}^{n}(Y_j - \sum_{i=1}^{m}\beta_{1i}X_{ij})^2 \text{ subject to } \sum_{i=1}^{m}\beta_{1i}^2 \leq t$$

By convention (very important!):

- The X_i are assumed to be standardized (mean 0, unit variance)
- Y is assumed to be centered

Ridge regression: ℓ_2-penalty

We can write the ridge constraint as the following penalized residual sum of squares (PRSS):

$$\begin{aligned} PRSS(\beta)_{\ell_2} &= \sum_{j=1}^{n}(Y_j - \sum_{i=1}^{m}\beta_{1i}X_{ij})^2 + \lambda\sum_{i=1}^{m}\beta_{1i}^2 \\ &= (Y - X\beta)^T(Y - X\beta) + \lambda||\beta||_2^2 \end{aligned}$$

The solution may have smaller average population mean squared error (sometimes called prediction error) than $\hat{\beta}^{ls}$.

$PRSS(\beta)_{\ell_2}$ is convex, and hence has a unique solution.

Taking derivatives, we obtain:

$$\frac{\partial PRSS(\beta)_{\ell_2}}{\partial\beta} = -2X^T(Y - X\beta) + 2\lambda\beta$$

Ridge regression: ℓ_2-penalty

The solution to $PRSS(\hat{\beta})_{\ell_2}$ is now seen to be:

$$\hat{\beta}_{\lambda}^{ridge} = (X^T X + \lambda I_m)^{-1} X^T Y$$

Remember that X is standardized and Y is centered.

The solution is indexed by λ

Inclusion of λ makes the problem non-singular even if $X^T X$ is not invertible. (this was the original motivation for ridge regression - see Hoerl and Kennard 1970).

This approach is essentially equivalent to putting a $N(0, cI)$ prior on the standardized coefficents.

Tuning parameter λ

$$\hat{\beta}_{\lambda}^{ridge} = (X^T X + \lambda I_m)^{-1} X^T Y$$

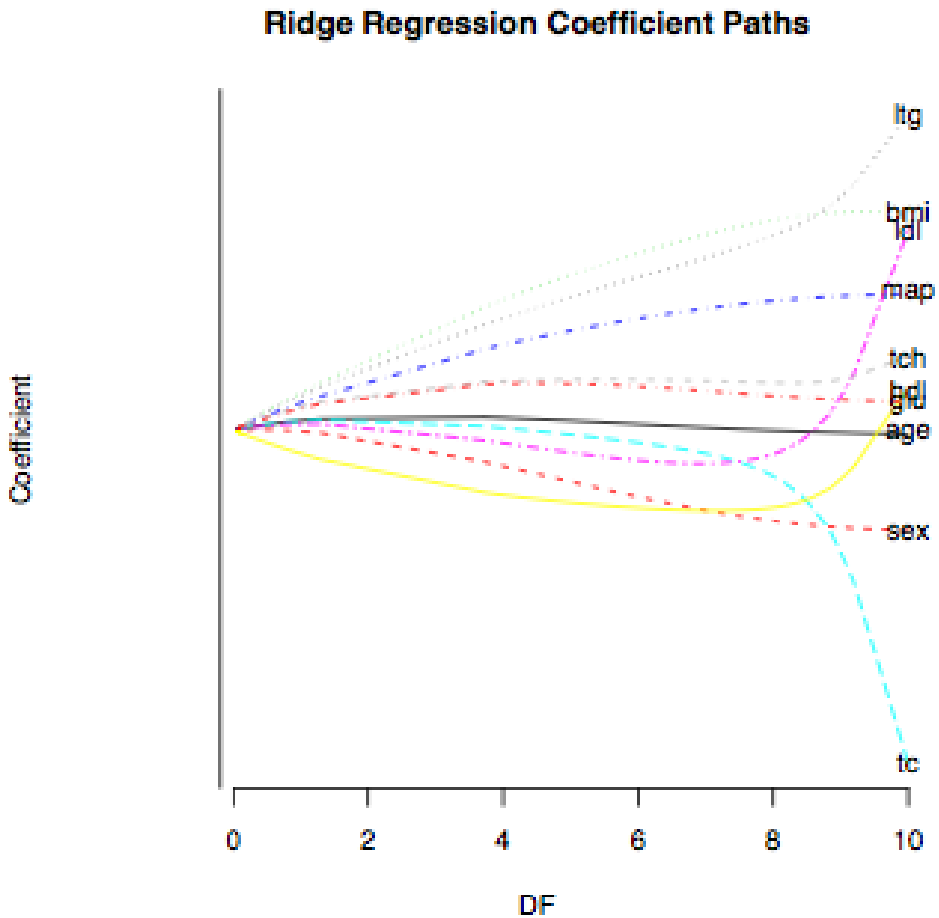
Notice the solution is indexed by the parameter λ, so for each λ we have a solution and the λ's trace out the path of the solutions.

λ is a shrinkage parameter

- λ controls the size of the coefficients
- λ controls the amount of **regularization**
- As $\lambda \longrightarrow 0$ we obtain the least square solution
- As $\lambda \longrightarrow \infty$ we have $\hat{\beta}_{\lambda=\infty}^{ridge} = 0$

Ridge coefficient paths

The λ's trace out a set of ridge solutions



Ridge coefficient path for the diabetes data set found in the `lars` library in `R`

Ridge coefficient paths

- We need a disciplined way of selecting λ
- That is we, need to "tune" the value of λ
- In their original paper, Hoerl and Kennard introduced ridge traces
  - Plot the components of $\hat{\beta}_{\lambda}^{ridge}$ against λ
  - Choose λ for which coefficients are not rapidly changing and have "sensible" signs
  - No objective basis, heavily criticized by many
- Standard practice now is to use cross-validation

Ridge coefficient paths

The ℓ_2 PRSS can be written as:

$$\begin{aligned} PRSS(\boldsymbol{\beta})_{\ell_2} &= \sum_{j=1}^{n}(Y_j - X_{\cdot j}^T\boldsymbol{\beta})^2 + \lambda\sum_{i=1}^{m}\beta_i^2 \\ &= \sum_{j=1}^{n}(Y_j - X_{\cdot j}^T\boldsymbol{\beta})^2 + \sum_{i=1}^{m}(0 - \sqrt{\lambda}\beta_i)^2 \end{aligned}$$

This means the ℓ_2 criterion can be recast as another least squares problem for an "augmented" data set.

Orthonormal X in ridge regression

If X is orthonormal, then $X^T X = I_m$ and a couple of closed form properties exist

- Let $\hat{\beta}^{ls}$ denote the LS solution for our orthonormal X, then

$$\hat{\beta}_{\lambda}^{ridge} = \frac{1}{1+\lambda}\hat{\beta}^{ls}$$

- The optimal choice of λ minimizing the expected prediction error is:

$$\lambda^* = \frac{p\sigma^2}{\sum_{j=1}^{m} \beta_m^2}$$

where the β_i are the true coefficients.

Smoother matrices and effective degrees of freedom

A smoother matrix S is a linear operator satisfying:

$$\hat{Y} = Sy$$

- Smoothers put the "hats" on Y
- So the fits are a linear combination of the Y_i

In ordinary least squares the hat matrix is $H = X(X^T X)^{-1} X^T$. For $\mathrm{rank}(X) = m$, we know that $\mathrm{tr}(H) = m$, which is how many degrees of freedom are used in the model.

By analogy, define the effective degrees of freedom (the effective number of parameters) for a smoother to be

$$\mathrm{df}(S) = \mathrm{tr}(S)$$

Degrees of freedom for ridge regression

In ridge regression, the fits are given by:

$$\hat{Y} = X(X^T X + \lambda I_m)^{-1} X^T Y$$

So the smoother ("hat") matrix in ridge regression takes the form:

$$X(X^T X + \lambda I_m)^{-1} X^T$$

The effective degrees of freedom are given by:

$$\mathrm{df}(\lambda) = \mathrm{tr}(S_\lambda) = \mathrm{tr}[X(X^T X + \lambda I_m)^{-1} X^T] = \sum_{i=1}^{m} \frac{d_i^2}{\lambda + d_i^2}$$

where d_i is the ith singular value of X (we'll talk about singular values in a future lecture)

- Note that $\mathrm{df}(\lambda)$ is monotone decreasing in λ
- Question: What happens when $\lambda = 0$?

The Lasso: ℓ_1 penalty

- Tibshirani (JRSSB, 1996) introduced the Lasso: *least absolute shrinkage and selection operator*
- Lasso coefficients are the solutions to the ℓ_1 optimization problem:

  minimize $(Y - X\beta)^T(Y - X\beta)$ such that $\sum_{i=1}^{m} |\beta_i| \leq t$
- This is equivalent to the loss function:

$$\begin{aligned} PRSS(\beta)_{\ell_1} &= \sum_{j=1}^{n}(Y_j - X_{.j}^T\beta)^2 + \lambda\sum_{i=1}^{m}|\beta_j| \\ &= (Y - X\beta)^T(Y - X\beta) + \lambda||\beta||_1 \end{aligned}$$

λ or t as a tuning parameter

- Again, we have a tuning parameter λ that controls the amount of regularization
- One-to-one correspondence with the threshold t:

$$\sum_{i=1}^{m} |\beta_i| \leq t$$

  - We have a path of solutions indexed by λ or t
  - If $\lambda = 0$ (equivalently $t_0 = \sum_{i=1}^{m} |\hat{\beta}_i^{ls}|$) there is no shrinkage
  - Often, the path of solutions is indexed by a fraction of the shrinkage factor of t_0.
- Under an orthonormal design ($X^T X = I_m$) the Lasso solution is a "soft shrinkage":

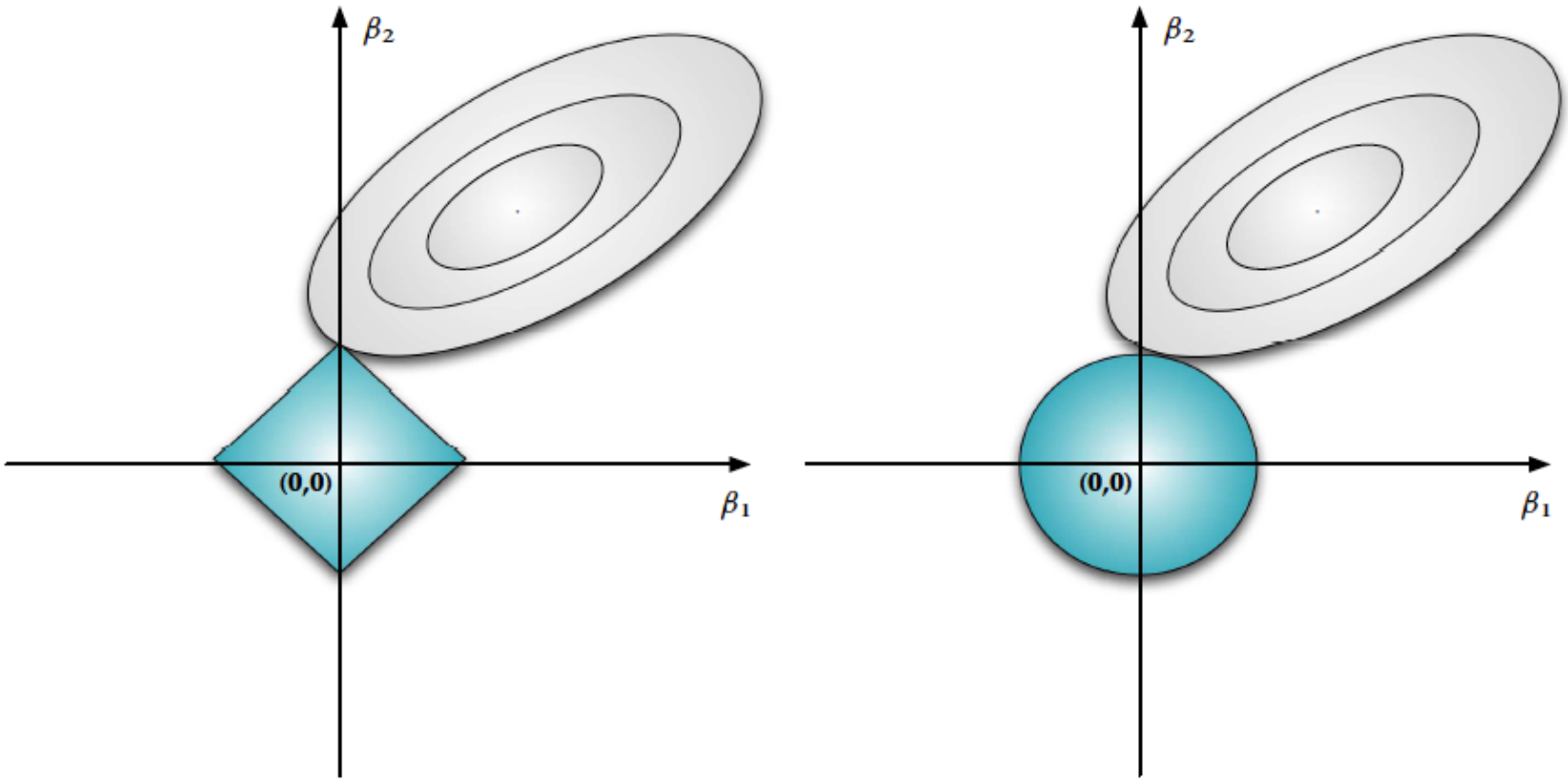
$$\hat{\beta}_i = \text{sign}(\hat{\beta}_i^{ls})(|\hat{\beta}_i^{ls}| - \gamma)^+$$

Sparsity and exact zeros

- Often, we believe that many of the β_j's should be 0
- Hence, we seek a set of sparse solutions
- Large enough λ (or small enough t) will set some of the coefficients exactly equal to 0!
- So the Lasso will perform model selection for us

Why does Lasso produce zeros?

In two-dimensions we can plot the contours of the squared loss $\sum_{j=1}^{n}(Y_j - \beta_1 X_{1j} + \beta_2 X_{2j})^2$ (grey) and the constrains for the lasso (left) and ridge regression (right)



Pretty picture courtesy Han Liu (who knows a ton about these ideas!)

Computing the Lasso solution

- Unlike ridge regression $\hat{\beta}_\lambda^{lasso}$ has no closed form
- Original implementation involves quadratic programming techniques from convex optimization
- The `lars` package in `R` implements the Lasso
- Efron et al. (Annals of Statistics, 2004) proposed LARS (least angle regression), which computes the Lasso path efficiently
  - An interesting modification is called forward stagewise
  - In many cases it is the same as the Lasso solution (for example when $X^T X = I_m$)
  - It is easy to implement!

Forward stagewise algorithm

As usual assume X is standardized and Y is centered

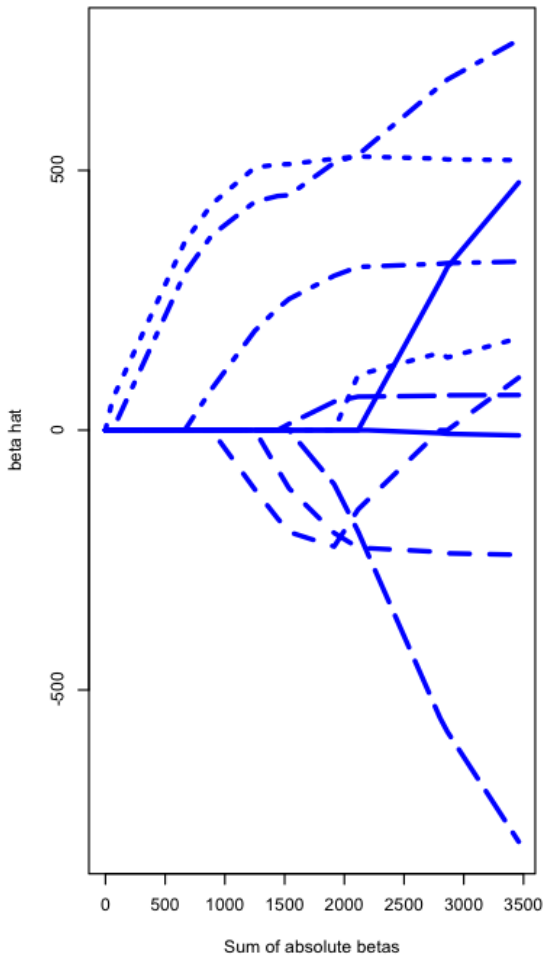
Choose a small ϵ

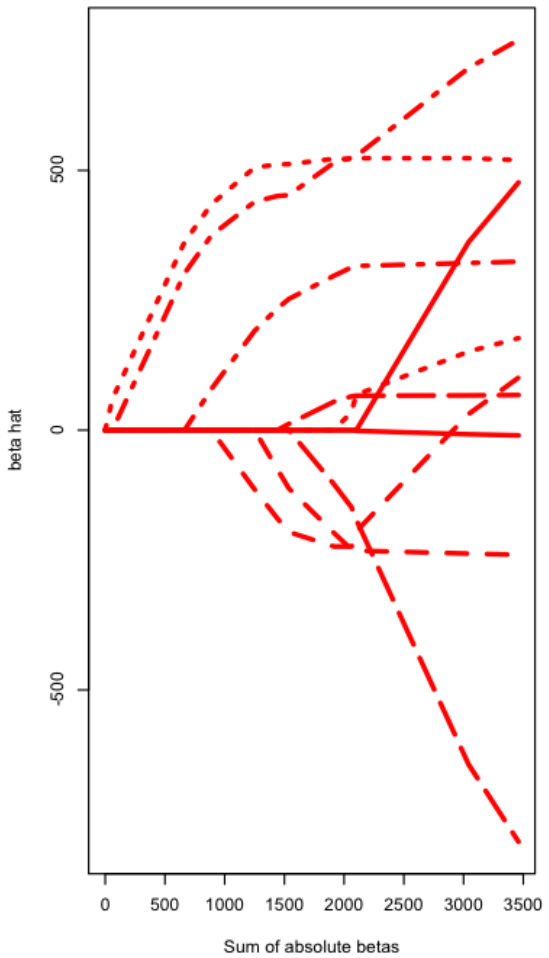
- Start with initial residual $R = Y$ and $\beta_1 = \beta_2 = \cdots = \beta_m = 0$
- Find the predictor X_i (i=1,. . . ,m) most correlated with R
- Updated $\beta_i \leftarrow \beta_i + \delta$ where $\delta_i = \epsilon \cdot \text{sign}(X_i^T R)$
- Set $R \leftarrow R - \delta_i X_i$.

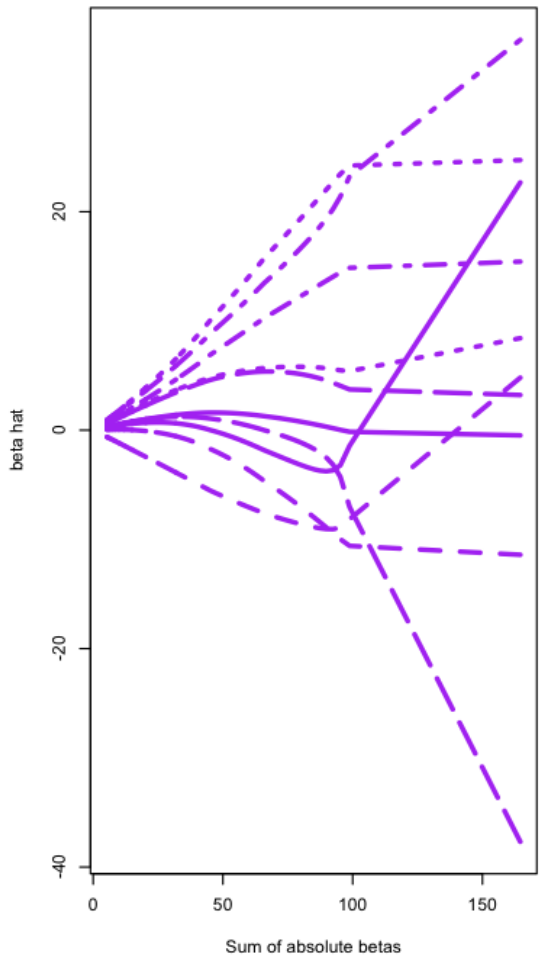
Ridge + Lasso + Forward Stagewise in R

```
>library(lars)
>data(diabetes)
>library(MASS)
>dim(lars1$x)
[1] 442 10
> length(diabetes$y)
[1] 442
>mod1 = lars(diabetes$x,diabetes$y,"lasso")
>mod2 = lars(diabetes$x,diabetes$y,"forward.stagewise")
> mod3 = lm.ridge(diabetes$y   diabetes$x,lambda=seq(0,2e4,by=20))
> par(mfrow=c(1,3))
>matplot(cbind(apply(abs(mod1$beta),1,sum)),mod1$beta,col="blue",type="l",lwd=3,xlab="Sum of
absolute betas", ylab="beta hat")
> matplot(cbind(apply(abs(mod2$beta),1,sum)),mod2$beta,col="red",type="l",lwd=3,xlab="Sum of
absolute betas", ylab="beta hat")

> matplot(cbind(apply(abs(mod3$coef),2,sum)),t(mod3$coef),type="l",col="purple",lwd=3,xlab="Sum

of absolute betas", ylab="beta hat")
```

The coefficient paths







How do we choose λ?

- We need a disciplined way of choosing λ
- We want to choose λ that minimizes the mean squared error for a new observation.
- This issue is part of the bigger problem of *model selection*

Training sets versus test sets

- If we have a good model, it should predict well when we have new data
- In machine learning terms, we compute our model $\hat{f}(X)$ from the training set
- A good estimator of $\hat{f}(X)$ should then perform well on a new, independent set of data
- We "test" or assess how well $\hat{f}(X)$ performs on the new data, which we call the test set

Training sets versus test sets

- Ideally, we would separate our available data into both training and test sets
- Of course this isn't always possible, especially if n is small
- We hope to come up with the best-trained algorithm that will stand up to the test
- How can we try to find the best-trained algorithm?

K-fold cross validation

The most common approach is K-fold cross-validation:

- Partition the training data T into K separate sets of equal size (common K's are 5 and 10)
- For each $k = 1, \ldots, K$ fit the model to the training set, excluding the kth fold T_k to get $\hat{f}^{\lambda}_{-k}(X)$.
- Compute the fitted values for the observations you held out T_k, based on the training data that excluded this fold
- Compute the cross-validation (CV) error for the k-th fold

$$(\text{CV error})^{\lambda}_{k} = |T_k|^{-1} \sum_{(X,Y)\in T_k} (Y - \hat{f}^{\lambda}_{-k})^2$$

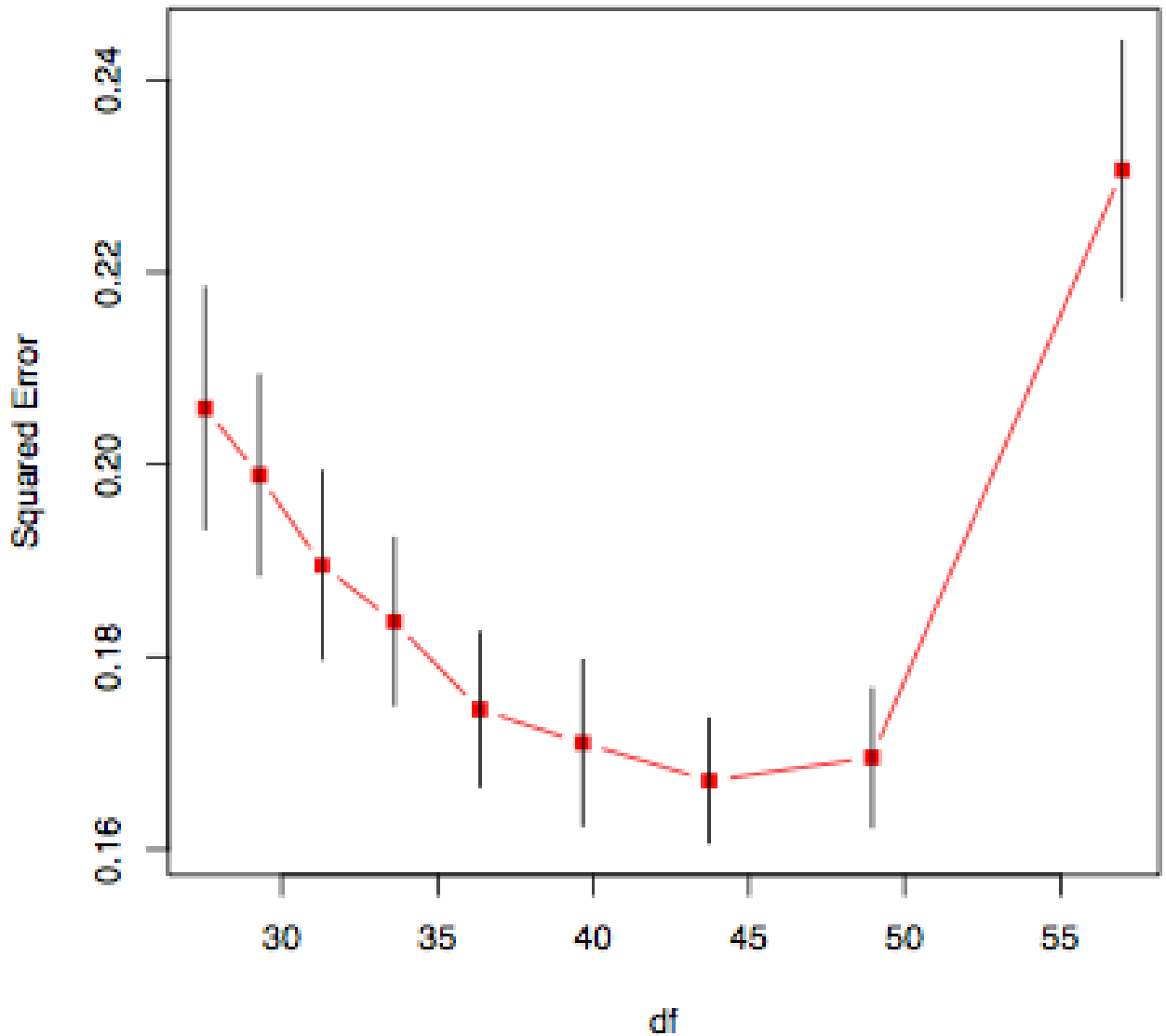
K-fold cross validation

The model then has overall cross-validation error:

$$(\mathrm{CVError})^{\lambda} = K^{-1} \sum_{k=1}^{K} (\mathrm{CV\ error})_k^{\lambda}$$

- Select λ^* as the one with minimum $(\mathrm{CVError})^{\lambda}$
- Compute the chosen model $\hat{f}^{\lambda^*}(X)$ on the entire training set
- Apply the function $\hat{f}^{\lambda^*}(X)$ to the test set to assess test-error

Example K-fold cross-validation output



Cross validation with few observations

Our data set might be small, so we might not have enough observations to put aside a test set:

- In this case, let all of the available data be our training set
- Still apply K fold cross validation
- Still choose λ^* as the minimizer of CV error
- Then refit the model with λ^* on the entire training set

Generalized CV for smoother matrices

Recall that our smoother (or "hat") matrix S satisfies:

$$\hat{Y} = SY$$

In many linear fitting methods (as in LS) we have:

$$CV(1) = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{f}_{-j}(X_j))^2 = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{Y_j - \hat{f}(X_j)}{1 - S_{jj}}\right)^2$$

A convenient approximation to $CV(1)$ is called the generalized cross validation, or GCV error:

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_j - \hat{f}(X_j)}{1 - \mathrm{tr}(S)/n}\right)^2$$

Summary of Ideas

- When you have a large number of covariates you have different options: (1) marginal tests and multiple testing correction, (2) empirical Bayes, or (2) penalized regression
- Multiple testing is generally concerned with controlling an error rate, but fits the covariates one-by-one
- Penalized regression may be less interpretable, but fits all the coefficients at once
- Often I have found that taking the top 10 marginal associations and including them in the model is nearly as effective as the Lasso (the Leekasso!). This is mostly a good straw man.
- There is a lot of literature on the Lasso, it was/is a hot topic in statistics, hence the Glasso, the Relaxo, the Grouped Lasso, etc.
- These ideas are being developed for estimating equation approaches as well: Wolfson "EEBoost: A general method for prediction and variable selection using estimating equations." JASA to appear.
- For the new era, prediction error is often the measure of choice, not unbiasedness or minimum variance.