Lecture 6: Penalty

Last time

We needed a couple facts from linear algebra -- The first was the Singular Value Decomposition

Let A be an n-by-p matrix -- Then there exists an n-by-p matrix U and a p-by-p matrix V, each with orthonormal columns, such that

$$A = U D V^t$$

where D is a p-by-p diagonal matrix with elements $s_1 \geq s_2 \geq \cdots \geq s_p \geq 0$ referred to as the singular values of A

Relation to regression

Now, if we let M be an n-by-p design matrix associated with a set of predictor variables, then if we can decompose $M = UDV^t$ we have that

$$M^tM = (UDV^t)^tUDV^t = VD^2V^t$$

which means that the columns of V are just the eignenvectors of M^tM

Relation to regression

In terms of solving a least squares system using the predictors in M, we also found that we could write

$$\begin{aligned}
\widehat{\mu} &= \mathsf{M}(\mathsf{M}^{\mathsf{t}}\mathsf{M})^{-1}\mathsf{M}^{\mathsf{t}}\mathsf{y} \\
&= \mathsf{U}\mathsf{D}\mathsf{V}^{\mathsf{t}}(\mathsf{V}\mathsf{D}\mathsf{U}^{\mathsf{t}}\mathsf{U}\mathsf{D}\mathsf{V}^{\mathsf{t}})^{-1}\mathsf{V}\mathsf{D}\mathsf{U}^{\mathsf{t}}\mathsf{y} \\
&\quad \text{(grinding noise of lots of things cancelling)} \\
&= \mathsf{U}\mathsf{U}^{\mathsf{t}}\mathsf{y}
\end{aligned}$$

where we see that our regression is just a projection into the (orthonormal) column space of U

Relation to regression

In the last lecture, we focused on a special kind of regression problem, one in which all the columns of M had been centered -- That is, the

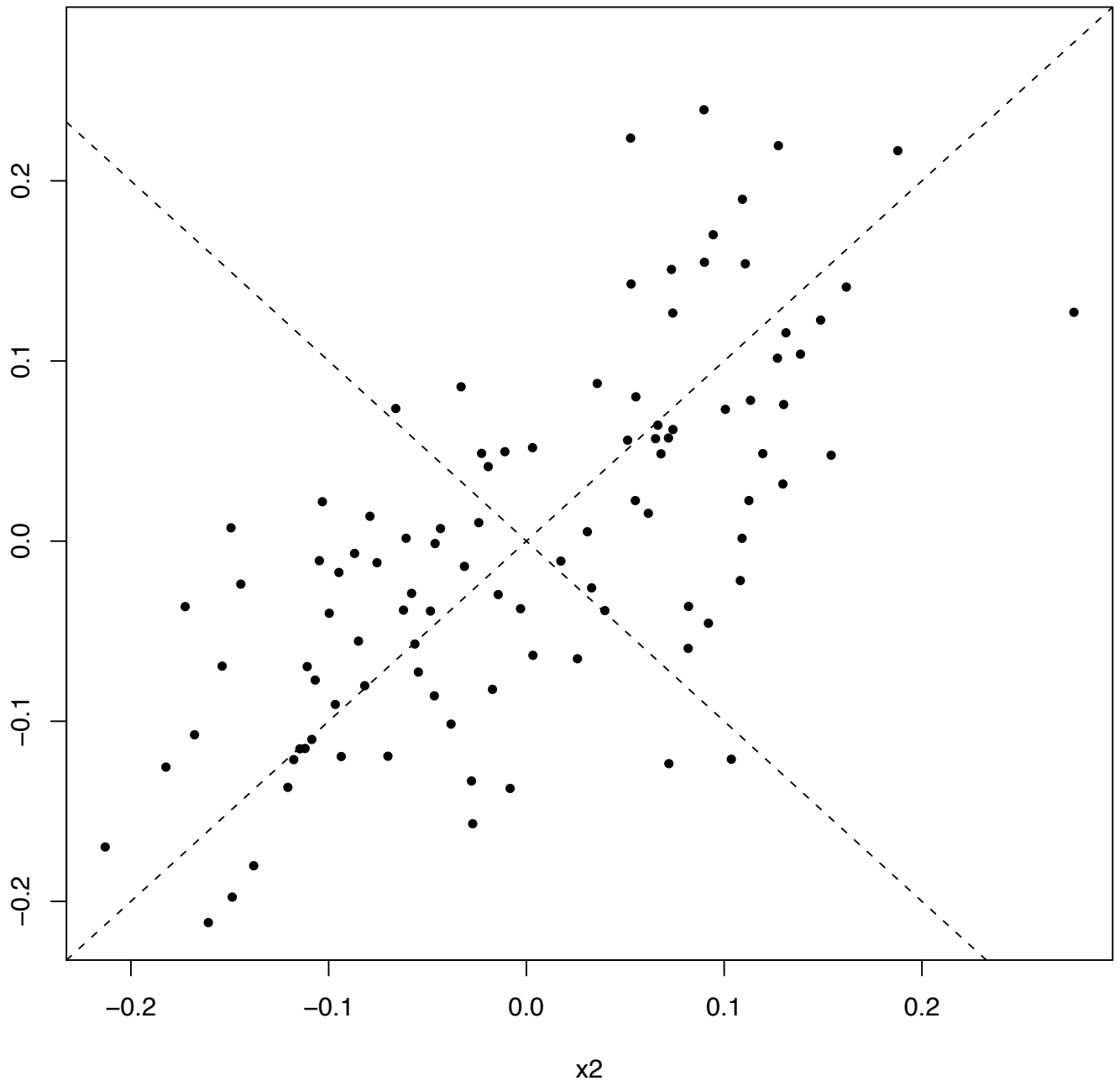
$$\widetilde{M} = [\, x_1 - \bar{x}_1 \,|\, x_2 - \bar{x}_2 \,|\, \cdots \,|\, x_p - \bar{x}_p \,]$$

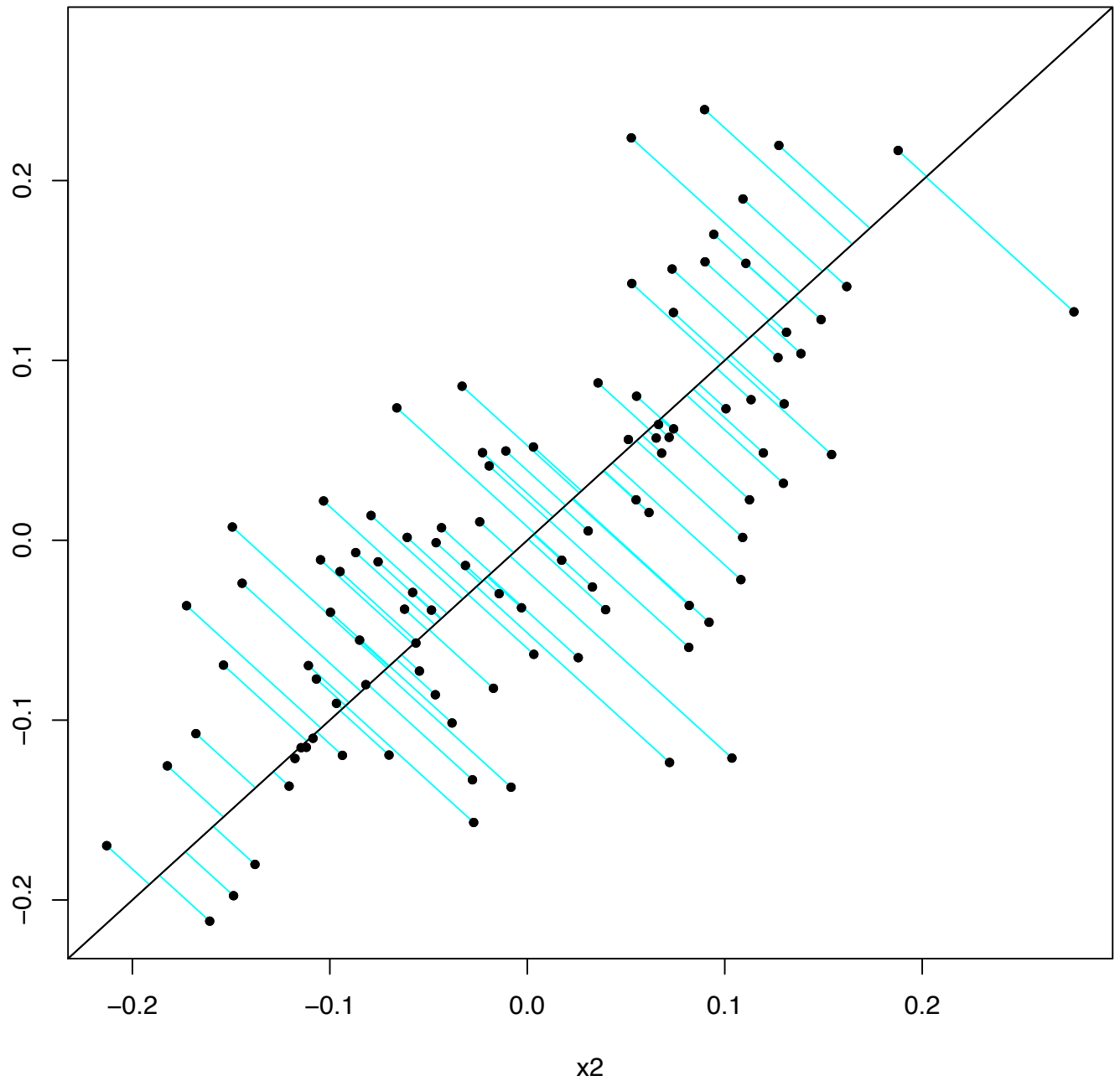
where $x_j = (x_{1j}, \cdots, x_{nj})^t$ is an n-dimensional column vector representing the jth input variable, j=1,...,p

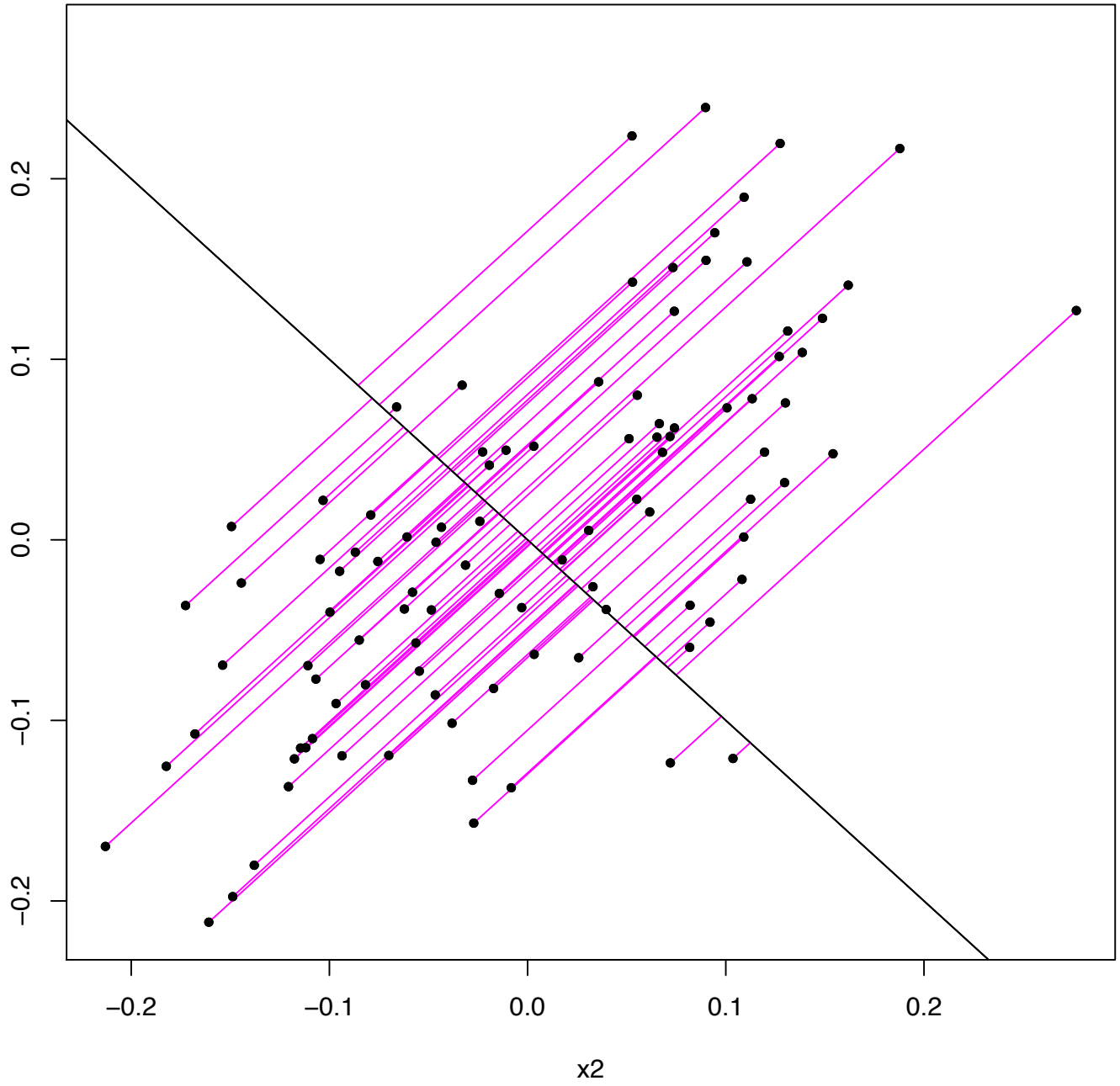
Relation to regression

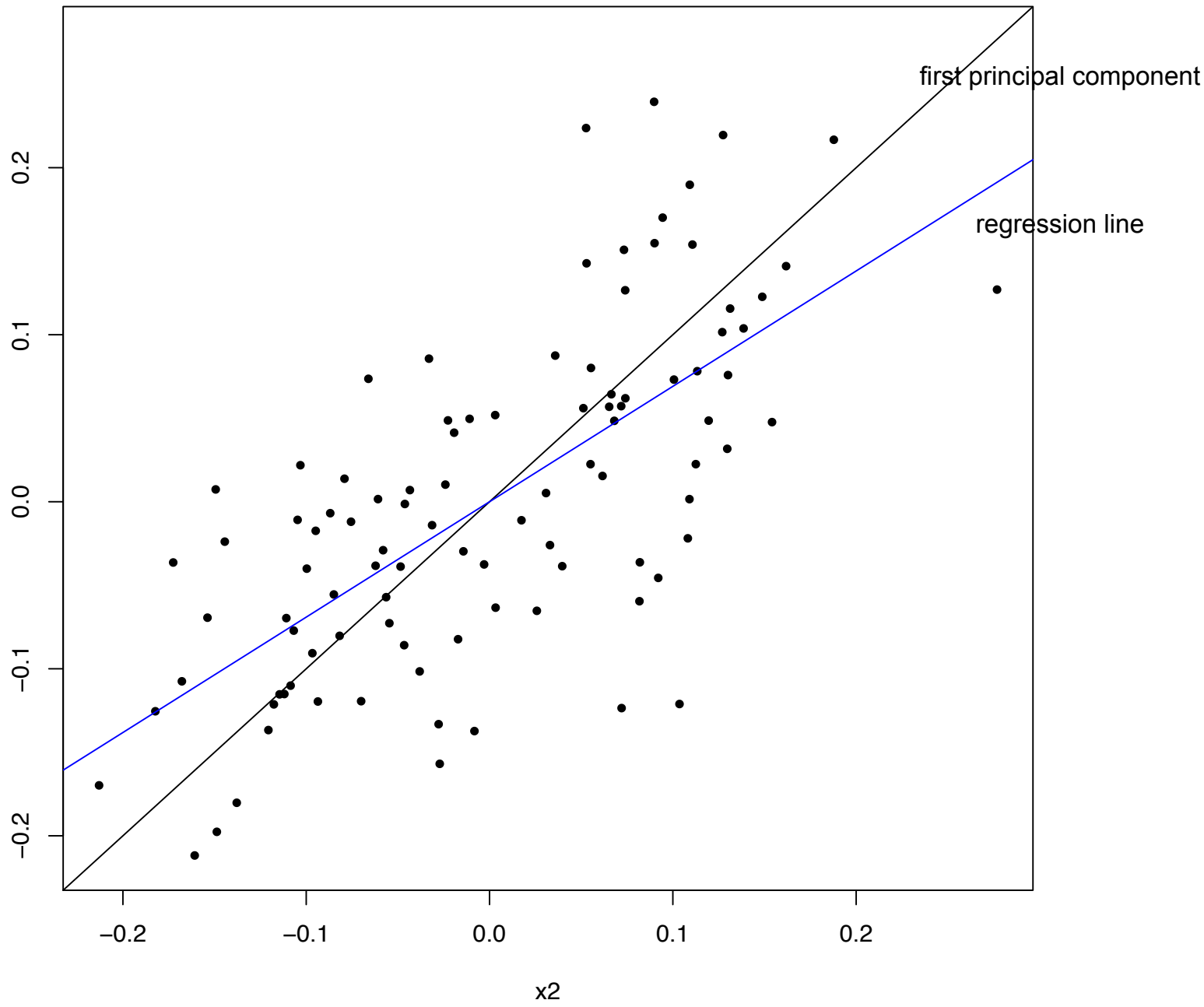
Notice that because of the centering, we can view $\widetilde{M}^t\widetilde{M}/(n-1)$ as (an estimate of?) the variance-covariance matrix of our input variables -- With this interpretation, the columns of U are the so-called principal components associated with this VC matrix

Principal components find their use in so-called dimension reduction problems, either numerical or graphical -- The are a new, orthogonal coordinate system for the column space of $\widetilde{M}$ such that the projection having the greatest variability lies along the first coordinate, the projection with the next greatest variability along the second and so on


0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2


0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2


0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2


first principal component
regression line
0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
x2

Last time

We started by considering the expected squared error of our least squares coefficients

$$\mathsf{E}\|\widehat{\beta} - \beta\|^2 = \mathsf{E}(\widehat{\beta} - \beta)^t(\widehat{\beta} - \beta)$$

We saw that we could rewrite this as

$$\mathsf{E}(\widehat{\beta} - \beta)^t(\widehat{\beta} - \beta) = \sigma^2 \,\mathsf{trace}\,(\widetilde{M}^t\widetilde{M})^{-1} = \sigma^2 \sum_{j=1}^{p} \frac{1}{d_j}$$

where the d_j are the eigenvalues of $\widetilde{M}^t\widetilde{M}$

The upshot

We found that if our predictors can be well-represented in a lower-dimensional space (very little variance in higher coordinates), then we end up paying a price in terms of the overall squared error of our coefficients

Previously we used VIFs to get at this kind of instability -- It turns out that the understanding in terms of principal components will be a nice tool to talk about a range of solutions

Initially, we start with the idea of instability (that two predictors might be highly correlated) and move eventually to notions of parsimony (finding a “good” set of predictors)

Ridge regression

Starting with stability, Hoerl and Kennard (1970) describe a problem with OLS:

> *"The least squares estimate suffers from the deficiency of mathematical optimization techniques that give point estimates; the estimation procedure does not have built into it a method for portraying the sensitivity of the solution to the optimization criterion."*

To get around this, they propose an alternative to OLS...

Ridge regression

In their paper, they first present the form of their solution -- Rather than consider the usual least squares estimates

$$\widehat{\beta} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}\widetilde{\mathsf{M}}\mathsf{y}$$

They consider adding a "ridge" to $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}}$ to yield

$$\widehat{\beta}^{*} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathsf{I}_{\mathsf{p}\times\mathsf{p}})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y}$$

where $\lambda \geq 0$

Ridge regression

There are a number of ways to arrive at this solution -- The most popular approach involves adding a constraint to the original OLS criterion

That is, find the value of β that minimizes

$$\sum_{i=1}^{n}(\tilde{y} - \beta_1\tilde{x}_{i1} - \cdots - \beta_p\tilde{x}_{ip})^2$$

subject to the constraint that $\sum_{j=1}^{p}\beta_j^2 \leq s$

This constraint is meant to have the effect of preventing the “cancellation” we have seen in previous lectures -- A very large positive coefficient being “cancelled” by an equally large negative coefficient on another correlated variable

Ridge regression

To solve this, we could introduce a Lagrange multiplier and come up with the equivalent minimization problem

$$\sum_{i=1}^{n}(\widetilde{y} - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

There is a one-to-one correspondence between s on the previous slide and the "penalty parameter" λ here -- Each act to control the size (Euclidean norm) of the coefficient vector

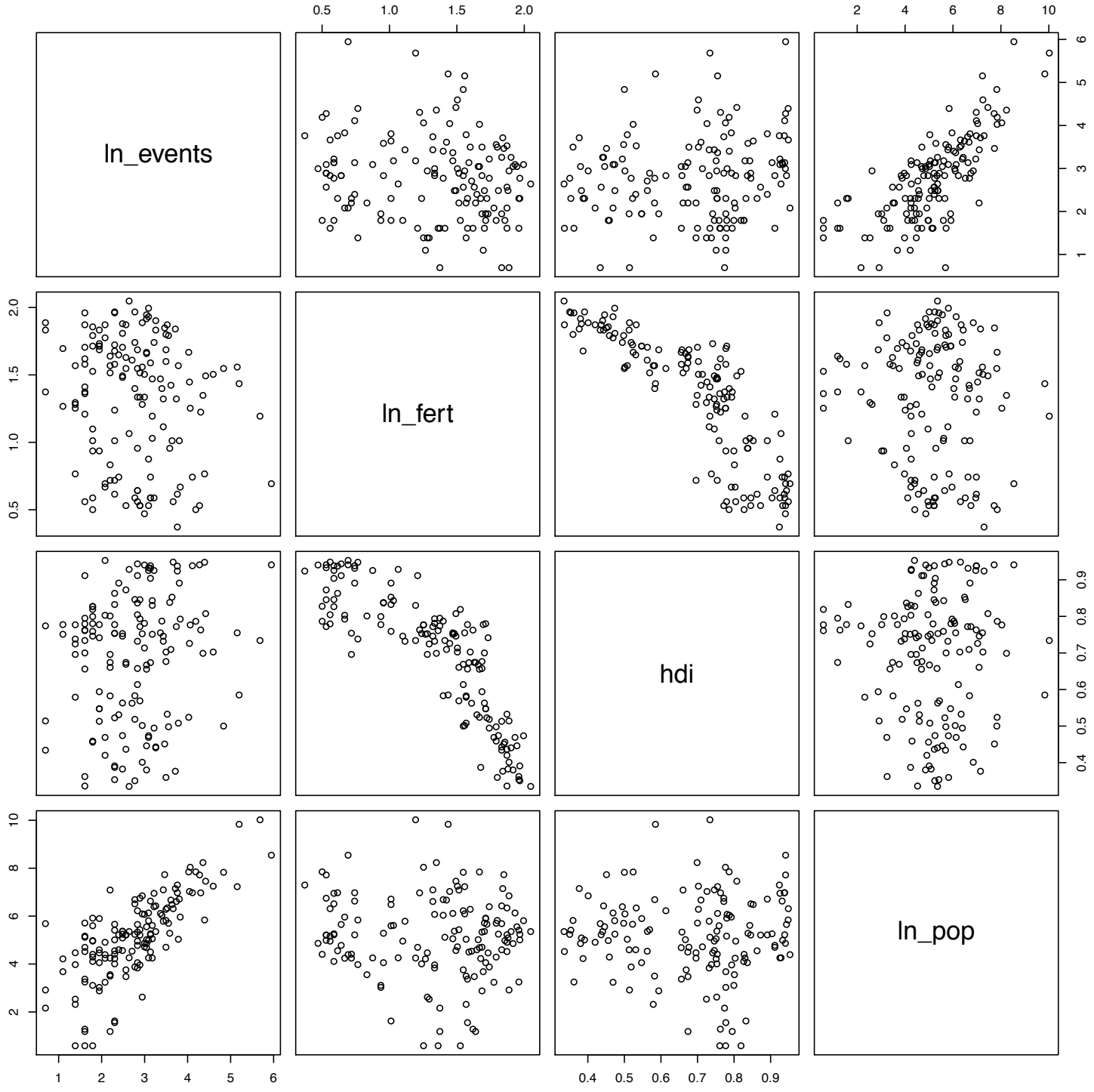
Initially, Hoerl and Kennard (1970) introduced ridge regression purely from a stability standpoint and start with the definition given two slides back -- More modern approaches to the subject start with this penalized expression, primarily because recent work examines different penalties on the coefficients

Example

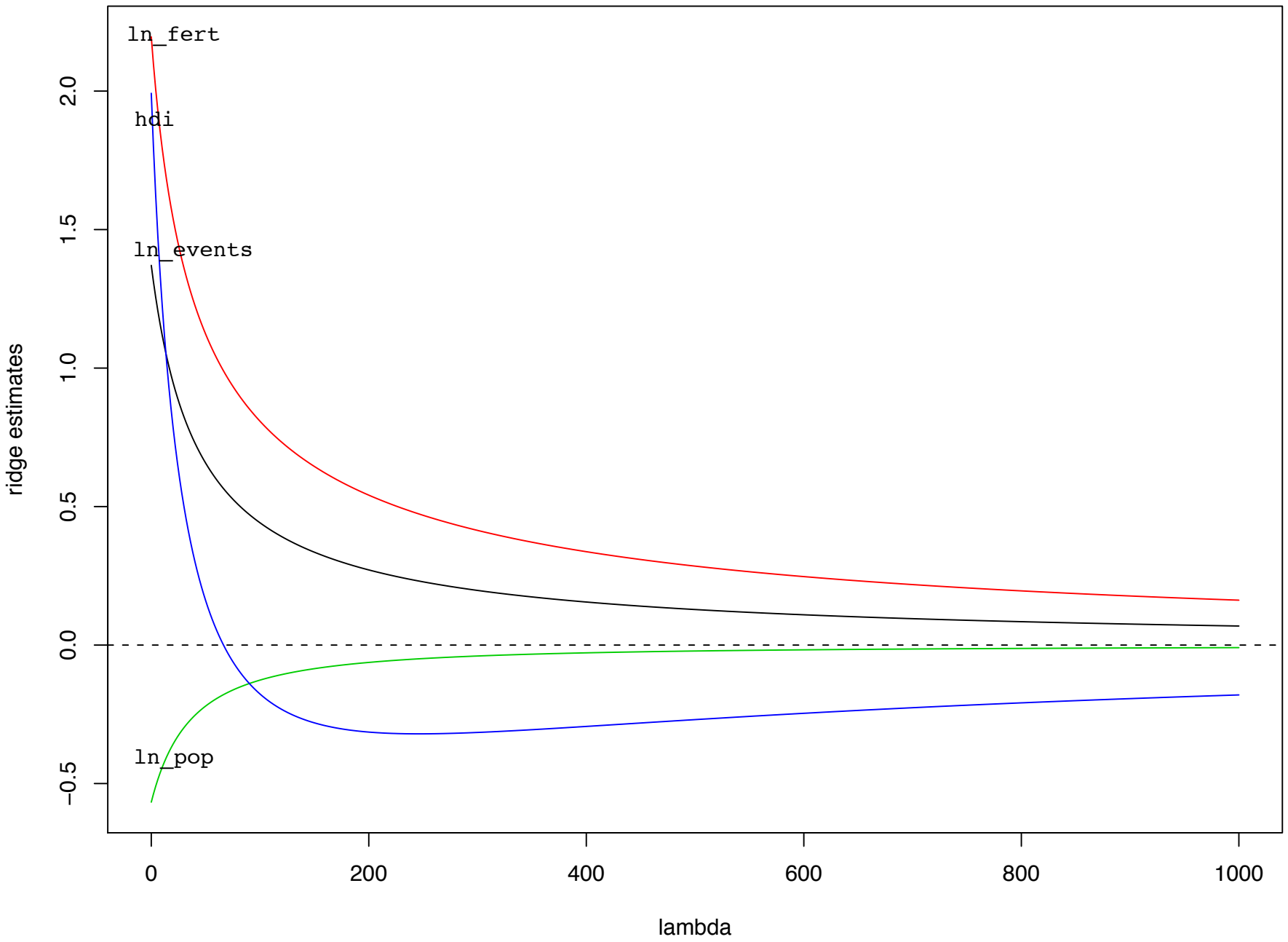
Consider again our vulnerability data -- The MASS library contains a specialty function for fitting ridge regression at one or more values of the penalty parameter

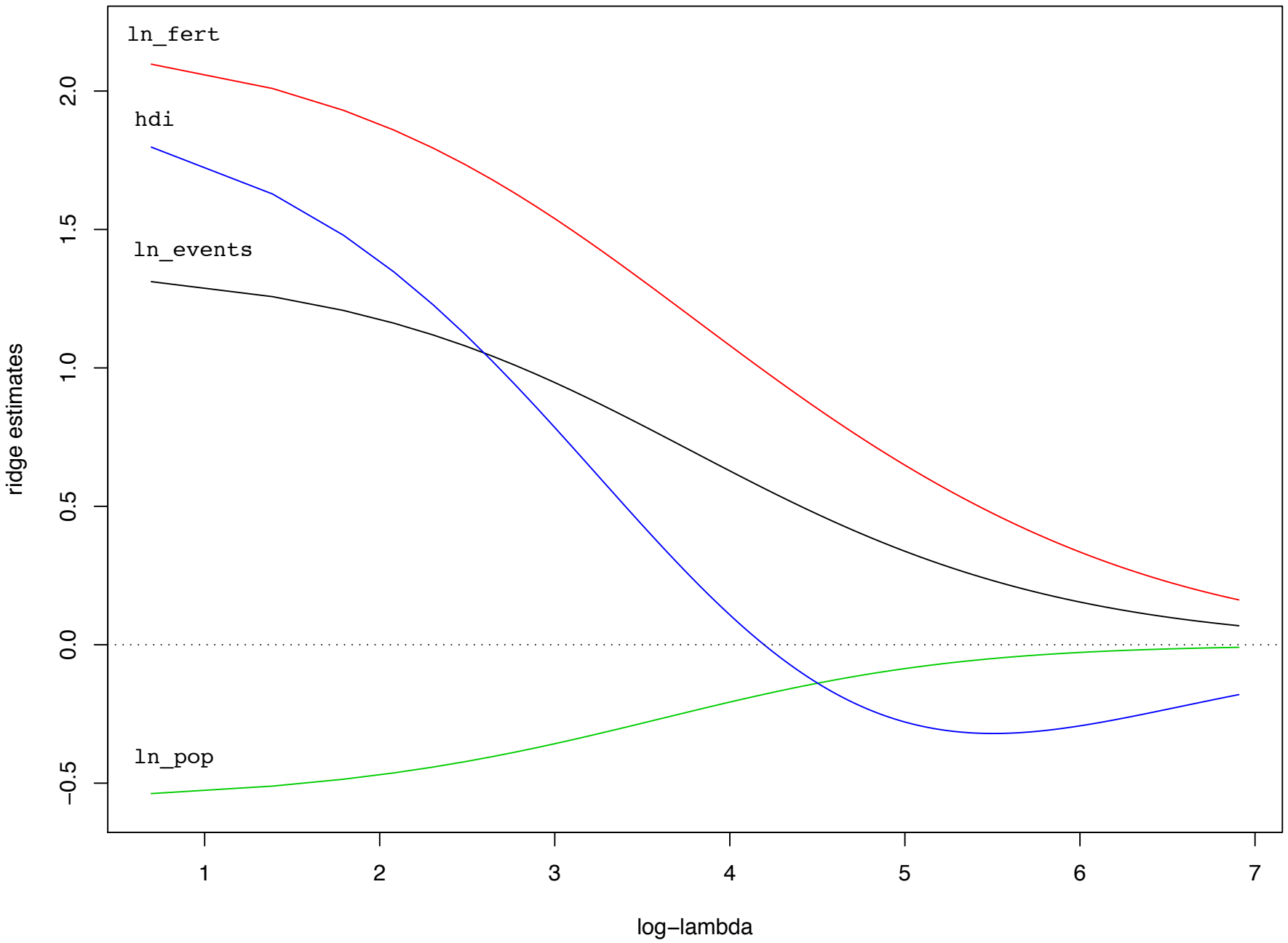
Plots of the ridge regression coefficients as a function of the penalty parameter are often referred to as the “ridge trace” and can tell us about the nature of the dependence between the variables

The function `lm.ridge` takes care of the centering of the variables -- On the next page, we illustrate how to use the function and illustrate the ridge traces...


ln_events
ln_fert
hdi
ln_pop
0.5
1.0
1.5
2.0
2
4
6
8
10
1
2
3
4
5
6
0.5
1.0
1.5
2.0
0.4
0.5
0.6
0.7
0.8
0.9
2
4
6
8
10
1
2
3
4
5
6
0.4
0.5
0.6
0.7
0.8
0.9

```
# specialty function in the MASS (Modern Applied Statistics with S by Venables and Ripley)
library(MASS)

# create values for the penalty parameter
lam <- seq(0,1000,len=500)

# and fit a series of ridge regressions (it's worth looking at the code
# to see how they are doing them all in one go)

fits <- lm.ridge(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul,lambda=lam)

# exhibit the coefficient vectors

matplot(log(lam),coefficients(fits)[,-1],lty=1, type="l",
         ylab="ridge estimates", xlab="log-lambda")
abline(h=0,lty=3)

lm(ln_death_risk~ln_events+ln_fert+ln_pop+hdi,data=vul)

# Call:
# lm(formula = ln_death_risk ~ ln_events + ln_fert + ln_pop + hdi,     data = vul)

# Coefficients:
# (Intercept)    ln_events      ln_fert       ln_pop          hdi
#     -5.3485       1.3708       2.1961      -0.5672       1.9922

cor(vul[,3:6])

#            ln_events     ln_fert         hdi      ln_pop
# ln_events  1.0000000 -0.15641675  0.14891515  0.74320022
# ln_fert   -0.1564168  1.00000000 -0.84119616 -0.09589724
# hdi        0.1489151 -0.84119616  1.00000000 -0.01559138
# ln_pop     0.7432002 -0.09589724 -0.01559138  1.00000000
```


ln_fert
hdi
ln_events
ln_pop
ridge estimates
2.0
1.5
1.0
0.5
0.0
−0.5
0
200
400
600
800
1000
lambda


ln_fert
hdi
ln_events
ln_pop
ridge estimates
2.0
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
6
7
log−lambda

Example

The ridge traces all start at the least squares values for the estimates ($\lambda = 0$) and then eventually work their way to zero as the constraint tightens

Notice that unlike the other three variables, HDI changes its sign as you increase the penalty -- This behavior is not uncommon when you have correlated predictors

In this case `ln_fert` and `HDI` are negatively correlated and so they can be thought of as representing the same factor, but with opposite signs -- As such, it doesn't seem reasonable (although we better think about the particulars of the variables) that their contributions should have the same sign

Ridge regression

We can link the "penalized regression" problem

$$\sum_{i=1}^{n}(\widetilde{y}_i - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

to Hoerl and Kennard's solution by taking the same approach we followed for the standard least squares problem -- Taking derivatives and forming an analog of the normal equations yields (ignoring the intercept)

$$\widehat{\beta}^* = (\widetilde{M}^t\widetilde{M} + \lambda I)^{-1}\widetilde{M}\widetilde{y} \quad \text{and} \quad \widehat{\mu}^* = \widetilde{M}(\widetilde{M}^t\widetilde{M} + \lambda I)^{-1}\widetilde{M}\widetilde{y}$$

Notice that the solution can be written as $S(\lambda)y$ for $S(\lambda) = \widetilde{M}(\widetilde{M}^t\widetilde{M} + \lambda I)^{-1}\widetilde{M}$

Example

The one difficulty with these plots is that they are not well calibrated in terms of choosing the penalty parameter -- We have selected a large range of values, knowing that for 0 we have OLS and for something big, we have essentially an intercept-only model

What scale might make it easier to interpret the action of our constraint?

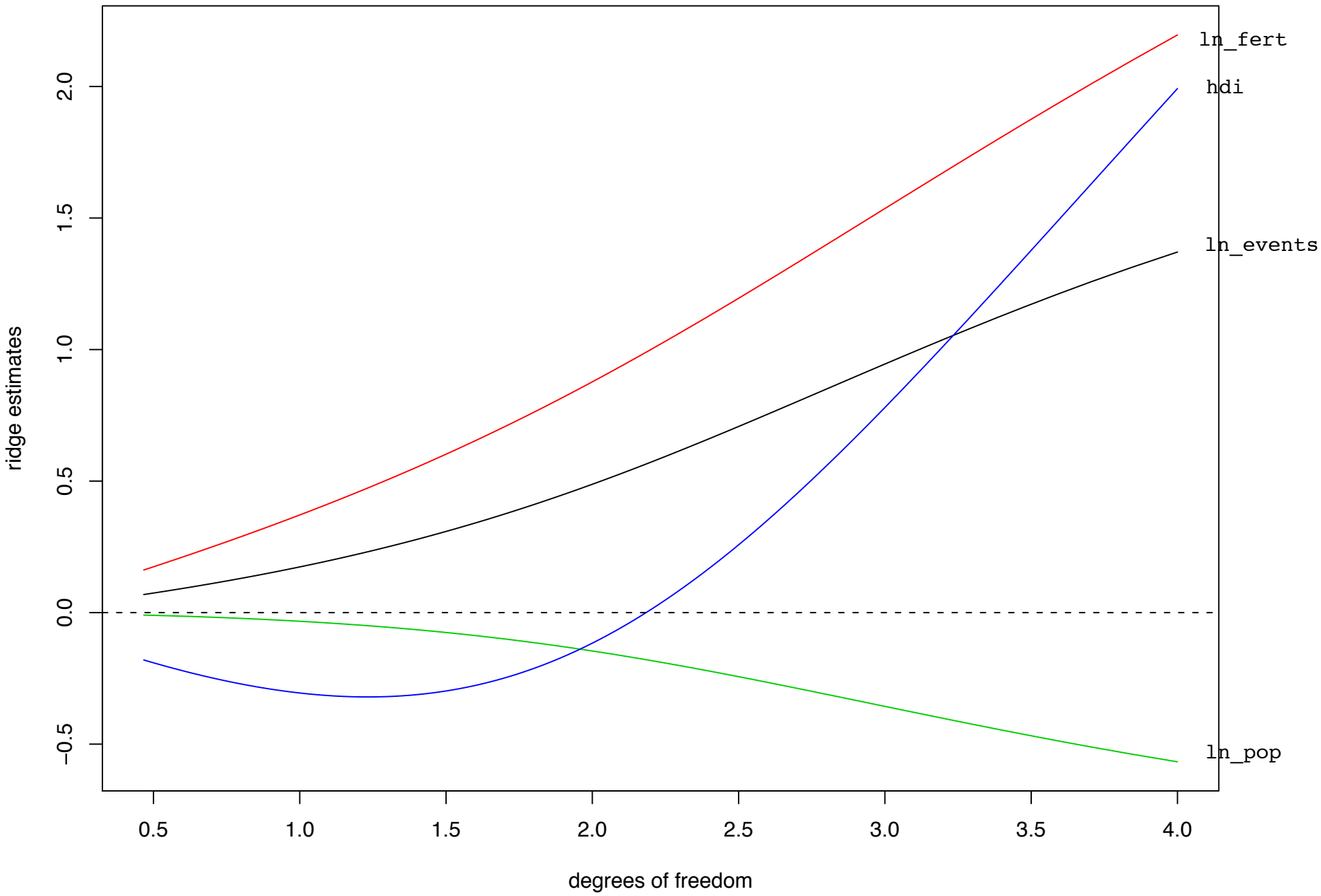
Degrees of freedom

The form of the predictor suggests a handful of “standard” techniques for expressing the degrees of freedom in the fit -- Here, our predictions are given by an “additive model” $\mathsf{S}(\lambda)\mathsf{y}$ (“additive” in that each conditional mean is a linear combination of the responses y) and

$$\mathsf{df}(\lambda) = \mathsf{trace}\,[\mathsf{S}(\lambda)] = \mathsf{trace}\,[\widetilde{\mathsf{M}}(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda\,\mathsf{I})^{-1}\widetilde{\mathsf{M}}]$$

is often used as a rough guide for the degrees of freedom in the fit (more on this in a moment)

The plot on the next page expresses the traces as a function of the degrees of freedom -- Again, we are ignoring the intercept and technically should add 1 to get the actual degrees of freedom in the complete fit


ln_fert
hdi
ln_events
ln_pop
ridge estimates
degrees of freedom
2.0
1.5
1.0
0.5
0.0
−0.5
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Ridge regression

What can we say about the difference between the solution to the OLS problem

$$\widehat{\beta} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y}$$

and our ridge regression estimate

$$\widehat{\beta}^{*} = (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda\mathsf{I})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y}$$

Character of the solution: Orthonormal predictors

Assume initially that $\widetilde{\mathsf{M}}$ has **orthonormal columns** so that $\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} = \mathsf{I}$, and recall that the OLS estimates are just $\widehat{\beta} = \widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{y}}$

Then, following our nose(s) with the expression for the ridge regression, we find that the ridge solutions are given by

$$\begin{aligned}
\widehat{\beta}^{*} &= (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda\mathsf{I})^{-1}\widetilde{\mathsf{M}}\widetilde{\mathsf{y}} \\
&= (\mathsf{I} + \lambda\mathsf{I})^{-1}\widehat{\beta} \\
&= \frac{1}{1+\lambda}\widehat{\beta}
\end{aligned}$$

What do you notice?

Character of the solution: Orthonormal predictors

Since $\lambda \geq 0$, we obtain our ridge regression estimates by “shrinking” their OLS cousins -- Ridge regression can be thought of as a shrinkage procedure, a class of techniques that gained popularity in the 1960s with the discovery of James-Stein estimation

We will consider the estimation properties of ridge regression in a moment -- For now we want to continue to think about the solution, and, in particular, what it means

The original motivation

Recall that Hoerl and Kennard were concerned with the expected squared error in the least squares coefficients

$$\mathsf{E}\|\widehat{\beta} - \beta\|^2 = \sum_{j=1}^{p} (\widehat{\beta}_j - \beta_j)^2$$

We can rewrite this expression into a (one of several) bias-variance decompositions

$$\sum_{j=1}^{p} \mathsf{E}(\widehat{\beta}_j - \mathsf{E}\widehat{\beta}_j + \widehat{\beta}_j - \beta_j)^2 = \sum_{j=1}^{p} (\mathsf{E}\widehat{\beta}_j - \beta_j)^2 + \sum_{j=1}^{p} \mathsf{E}(\widehat{\beta}_j - \mathsf{E}\widehat{\beta}_j)^2$$

The original motivations

Assuming our linear model is correct and that our predictors are orthogonal, we recall that the least squares estimates $\widehat{\beta}$ have mean β and variance-covariance matrix $\sigma^2 \mathrm{I}_{\mathrm{p}\times\mathrm{p}}$

With this, the last expression above becomes simply $\mathrm{p}\sigma^2$

The original motivation: Orthogonal predictors

From a previous slide, $\widehat{\beta}^* = \widehat{\beta}/(1+\lambda)$ and so our ridge solutions are biased -- It's not hard to show that the bias-variance decomposition becomes

$$\left(\frac{\lambda}{1+\lambda}\right)^2 \sum_{j=1}^{p} \beta_j^2 + \frac{p\sigma^2}{(1+\lambda)^2}$$

The value of λ that minimizes this expression is $\lambda^* = p\sigma^2 / \sum_{j=1}^{p} \beta_j^2$ and the minimum is

$$\frac{p\sigma^2}{1 + p\sigma^2 / \sum_{j=1}^{p} \beta_j^2}$$

What do you think?

Character of the solution: General case

If we set $\mathsf{A} = \widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}}$, then we can write

$$\begin{aligned}
\widehat{\beta}^{*} &= (\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}} + \lambda \mathsf{I})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y} \\
&= (\mathsf{A} + \lambda \mathsf{I})^{-1}\mathsf{A}\mathsf{A}^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y} \\
&= [\mathsf{A}(\mathsf{I} + \lambda \mathsf{A}^{-1})]^{-1}\mathsf{A}[(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}\widetilde{\mathsf{M}}^{\mathsf{t}}\mathsf{y}] \\
&= (\mathsf{I} + \lambda \mathsf{A}^{-1})^{-1}\mathsf{A}^{-1}\mathsf{A}\widehat{\beta} \\
&= [\mathsf{I} + \lambda(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}]^{-1}\widehat{\beta}
\end{aligned}$$

and taking expectations we again have a biased estimate

$$\mathsf{E}\widehat{\beta}^{*} = [\mathsf{I} + \lambda(\widetilde{\mathsf{M}}^{\mathsf{t}}\widetilde{\mathsf{M}})^{-1}]^{-1}\mathsf{E}\widehat{\beta} \neq \beta$$

Character of the solution: General case

When we don't have a special orthonormal model matrix, we can appeal to the singular value decomposition, writing $\widetilde{M} = UDV^t$, to help us interpret the solution

$$\begin{aligned}
\widehat{\mu}^* &= \widetilde{M}\widehat{\beta}^* \\
&= \widetilde{M}(\widetilde{M}^t\widetilde{M} + \lambda I)^{-1}\widetilde{M}^t\widetilde{y} \\
&= UDV^t(VDU^t\,UDV^t + \lambda I)^{-1}VDU^t\widetilde{y} \\
&\quad \text{(more grinding noises)} \\
&= UD(D^2 + \lambda I)^{-1}DU^t\widetilde{y}
\end{aligned}$$

Expanding this last expression we find that

$$\widehat{\mu}^* = \sum_{j=1}^{p} u_j \frac{s_j^2}{s_j^2 + \lambda}(u_j^t\widetilde{y})$$

Character of the solution: General case

Recall that we could also write our least squares solutions in terms of the elements of the SVD

$$\widehat{\mu} = UU^t y = \sum_{j=1}^{p} u_j (u_j^t \widetilde{y})$$

which we can now directly compare to our ridge solutions

$$\widehat{\mu}^* = \sum_{j=1}^{p} u_j \frac{s_j^2}{s_j^2 + \lambda} (u_j^t \widetilde{y})$$

The difference is the shrinkage factors $0 < s_j^2/(s_j^2 + \lambda) < 1$

Character of the solution: The general case

Recall that the columns of U are normalized principal components -- Each column u_j can be thought of as $u_j = z_j/\|z_j\|$ where $z_1, \ldots, z_p$ are our p principal components

Therefore, our OLS fit can be thought of in terms of this set of orthonormal basis vectors $u_1, \ldots, u_p$ -- The ridge fit also uses these directions, but shrinks the OLS estimates according to factors that depend on the eigenvalues of $\widetilde{M}^t\widetilde{M}$

$$\frac{d_j}{d_j + \lambda}$$

Therefore, a greater amount of shrinkage is applied to directions associated with smaller values of d_j , the “thinner” direction of our ellipse in the previous data example

Aside: Degrees of freedom

One form of the degrees of freedom here is essentially a trace

$$\mathrm{df}(\lambda) = \mathrm{trace}\,[\widetilde{M}(\widetilde{M}^t\widetilde{M} + \lambda I)^{-1}\widetilde{M}^t] = \sum_{j=1}^{p} \frac{d_j}{d_j + \lambda}$$

Notice that if we are not doing any shrinkage ($\lambda = 0$), then the degrees of freedom is just p as you would hope -- With increasing λ the degrees of freedom drop to zero

Keep in mind that all of these calculations have been done omitting the intercept from the calculations -- This has been allowed to pass “as is” without any shrinkage

Character of the solution: The general case

In the end, we see that ridge regression protects against the instability of gradients estimated in these thin directions -- Implicitly, we're hoping that the response will tend to vary most in the directions of high variance for the inputs

This is often a reasonable assumption, but does not need to hold in general -- Recall that the principal components are entirely a function of the input space and does not involve the actual response at all

The original motivation: The general case

In general, Hoerl and Kennard show that in the general case, the squared bias is a continuous and monotonically increasing function of λ, while the variance is a continuous, monotonically decreasing function of λ

They then conclude that there is always a value of λ such that

$$E\|\widehat{\beta}^*_\lambda - \beta\|^2 < E\|\widehat{\beta} - \beta\|^2 = \sigma^2 \sum_{j=1}^{p} \frac{1}{d_j}$$

The trick, of course, is how you find it!

An alternative

As we have observed, ridge regression (the “shrinkage estimate”) could perform better in a squared loss sense than OLS -- We derived an explicit bias-variance tradeoff and saw that some degree of shrinkage could reduce variance at the expense of a small (hopefully) increase in bias

This discussion suggests that we are paying a price by trying to estimate “small” coefficients (principal component directions) -- Bringing them closer to zero improves the variance of our estimate

Subset selection often has a similar motivation -- Including unnecessary predictors in our model adds noise to the estimation of other quantities that interest us

From another direction, one easy way to deal with collinearity is to simply not include variables that contain essentially the same information -- From a practical perspective, this kind of redundancy means predictions from our model require unnecessary data collection

Variable selection

In general, we might want to perform variable selection to simplify the final model -- We are looking for a parsimonious model, a small one that seems to fit the data well

This strategy is behind a number of the hypothesis-based testing methods for model building -- That is, fit a model and drop (sequentially) terms that are not significant (backward elimination)

Ridge regression is often introduced as a competitor to variable selection because (in all but the extreme case of just a few strong predictors) it can reduce variance -- Unfortunately, it doesn't simplify the model and we have the same (potentially large) number of predictors in our model

Model selection

Let's assume our "true" data are generated by a linear model of the form

$$\begin{aligned} y_i &= f(x_i) + \epsilon_i \\ &= \beta_1^* x_{i1} + \cdots \beta_p^* x_{ip} + \epsilon_i \end{aligned}$$

We assume our predictors are orthonormal so that the design matrix M is given by $M = [x_1 | \cdots | x_p]$, where $x_j = (x_{1j}, \ldots, x_{nj})^t$, and satisfies $M^t M = I$ -- Then we find the OLS via

$$\widehat{\beta_j} = x_j^t y$$

We can show that these estimates have a normal distribution with mean β_j^* and variance σ^2

Model selection

Now, suppose that we form a model using only a subset of the variables -- For simplicity, we'll assume that there is some ordering to the variables (as in the case of principal components regression or orthogonal polynomials)

For a new data point $m_0 = (x_{01}, \ldots, x_{0p})^t$, we form an estimate of the conditional mean using just the first q variables

$$\widehat{f}(m_0) = \widehat{\beta}_1 x_{01} + \cdots + \widehat{\beta}_q x_{0q}$$

where again the "truth" is

$$f(m_0) = \beta_1^* x_{01} + \cdots + \beta_p^* x_{0p}$$

Variable selection

Finally, we'll consider a bias-variance decomposition

Variance: The variance of the estimate $\widehat{f}(m_0)$ is simply

$$\begin{aligned} \text{var}\, \widehat{f}(m_0) &= E\left[(\widehat{\beta}_1 - \beta_1^*)\, x_{01} + \cdots + (\widehat{\beta}_q - \beta_q^*)\, x_{0q}\right]^2 \\ &= \sigma^2(x_{01}^2 + \cdots + x_{0q}^2) \end{aligned}$$

Bias: Because the OLS estimates for this orthogonal case are unbiased no matter what terms we include in the model, the "bias" of our estimate $\widehat{f}(m_0)$ comes from having left out the terms with indices larger than q

$$\left[\beta_{q+1}^* x_{0,q+1} + \cdots + \beta_p^* x_{0p}\right]^2$$

Variable selection

To be clear, what we've done here is use facts about the sampling distribution of our OLS estimates -- All of these results were derived **conditional on the input variables**

Therefore, our bias-variance decomposition is

$$E\left[\widehat{f}(m_0) - f(m_0)\right]^2 = E\left[\widehat{f}(m_0) - E\widehat{f}(m_0)\right]^2 + E\left[f(m_0) - E\widehat{f}(m_0)\right]^2$$

where the expectation is with respect to the response vector y (consider repeating your experiment, say)

Finally, then, we have

$$E\left[\widehat{f}(m_0) - f(m_0)\right]^2 = \sigma^2(x_{01}^2 + \cdots + x_{0q}^2) + (\beta_{q+1}^* x_{0,q+1} + \cdots + \beta_p^* x_{0p})^2$$

Variable selection

Given that our results are conditional on the input variables, it is common to consider **the errors across our inputs** -- We refer to the resulting quantity as an “in-sample” measure of error

Substituting $m_i = (x_{i1}, \ldots, x_{ip})^t$ for m_0 and summing we find

$$\sum_{i=1}^{n} E\left[\widehat{f}(m_i) - f(m_i)\right]^2$$

Variable selection

Consider the bias and variance terms after summing...

Variance: Because our predictors are orthonormal, each has a length of 1 (sum of squares) so that

$$\sum_{i=1}^{n} \sigma^2(x_{i1}^2 + \cdots + x_{iq}^2) = \sigma^2(\sum_{i} x_{i1}^2 + \cdots + \sum_{i} x_{iq}^2) = q\sigma^2$$

Bias: Orthonormal predictors again intervene to simplify the squared bias term -- The squared terms are all 1 and the cross terms are zero

$$\sum_{i=1}^{n} (\beta_{q+1}^* x_{i,q+1} + \cdots + \beta_p^* x_{ip})^2 = \beta_{q+1}^{*2} + \cdots + \beta_p^{*2}$$

Variable selection

Pulling it together, we can derive **the bias-variance decomposition for any orthonormal regression** (dropping the need for pre-ordered predictors) that estimates q predictors and sets the remaining coefficients to zero can be written as

$$\sigma^2 q + (\text{squared coefficients of omitted terms})$$

The intuition here is that each term $\widehat{\beta}_j$ we include in our model charges us σ^2 in terms of estimation accuracy (the price we pay for choosing to estimate β_j^*) -- Leaving it out will cost us β_j^{*2} in bias

In general, the greater the complexity of our model, the more we are charged in terms of variance -- But if our model is not complex enough and we are missing big effects, we lose out in terms of bias

Variable selection

This simple estimation setting highlights the basic reasons we might not want to include all the variables in our model

Prediction accuracy. As we have seen, the more variables we include, the greater the error we incur

Interpretability. A smaller number of predictors will help us “see” what’s is going on in the data more easily -- Which effects are strongest in some sense

Subset selection, then, can help us on both fronts -- But is the expression on the previous slide useful from a practical perspective?

Variable selection

If we had an “oracle” that would tell us both the regression coefficients as well as the error variance, could we pick an “optimal” model?

What would it be?

Variable selection

To compare this to ridge regression, we could derive the in-sample error for ridge a ridge estimate $\widetilde{f}$ (using mostly the same expressions from the previous slide) to be

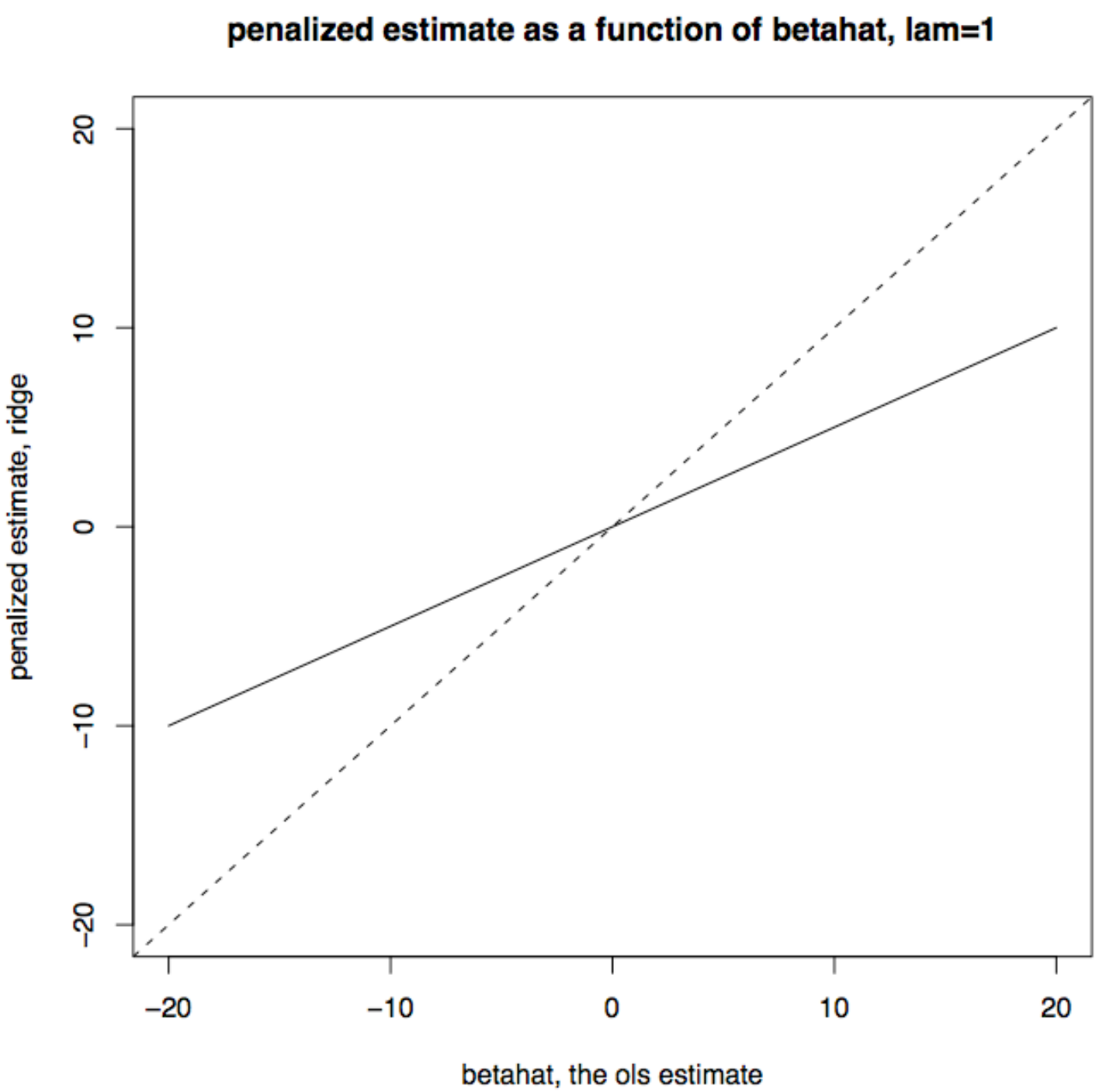
$$\sum_{i=1}^{n} E\left[\widetilde{f}(m_i) - f(m_i)\right]^2 = p\sigma^2\left(\frac{1}{1+\lambda}\right)^2 + \left(\frac{\lambda}{1+\lambda}\right)^2 \sum_{j=1}^{p} \beta_j^{*2}$$

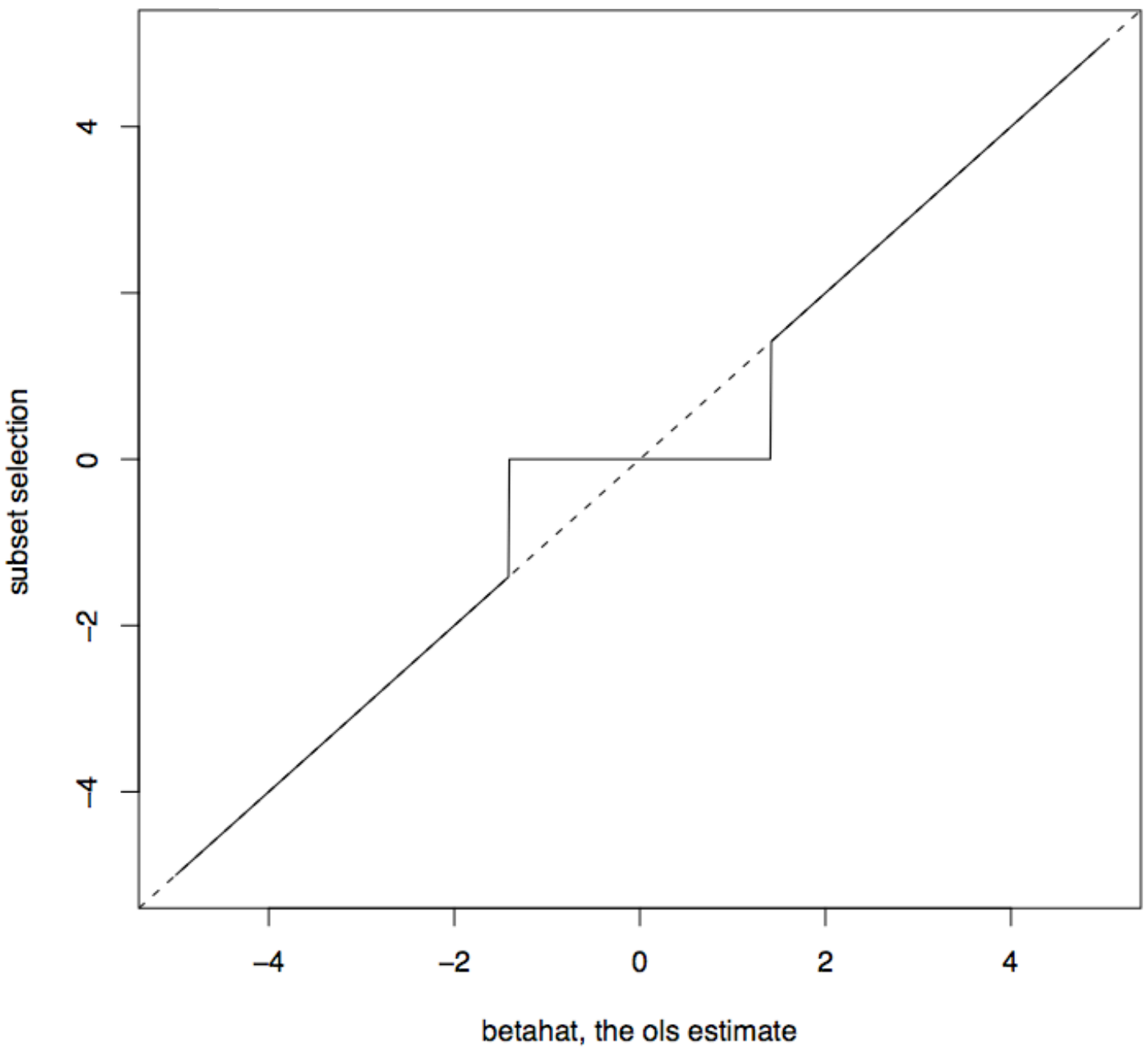
Does it “match” the least squares expression?

Interpretation

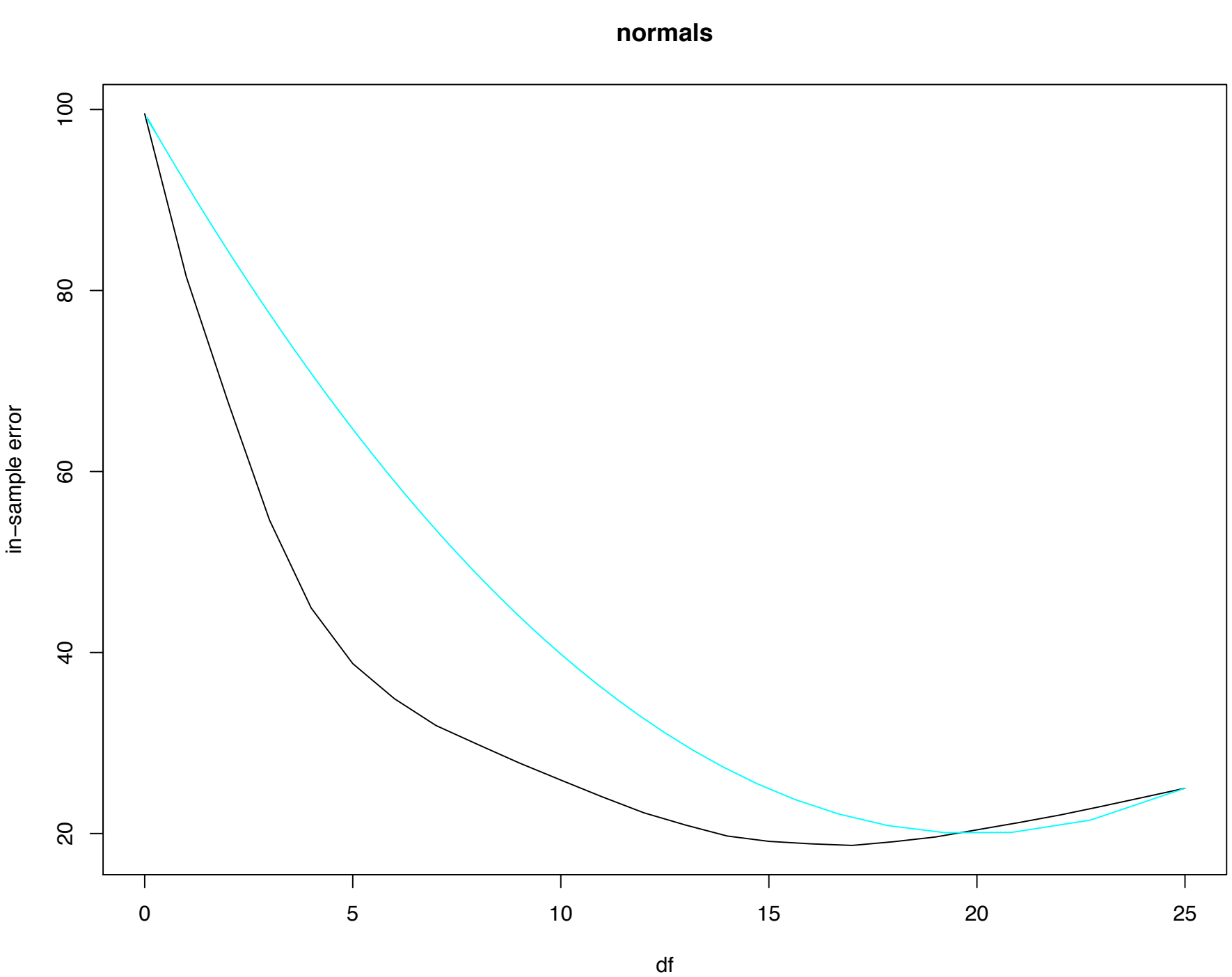
On the following two slides, we attempt an interpretation of what the two approaches (shrinkage and selection) do to the least squares coefficients in the orthogonal case

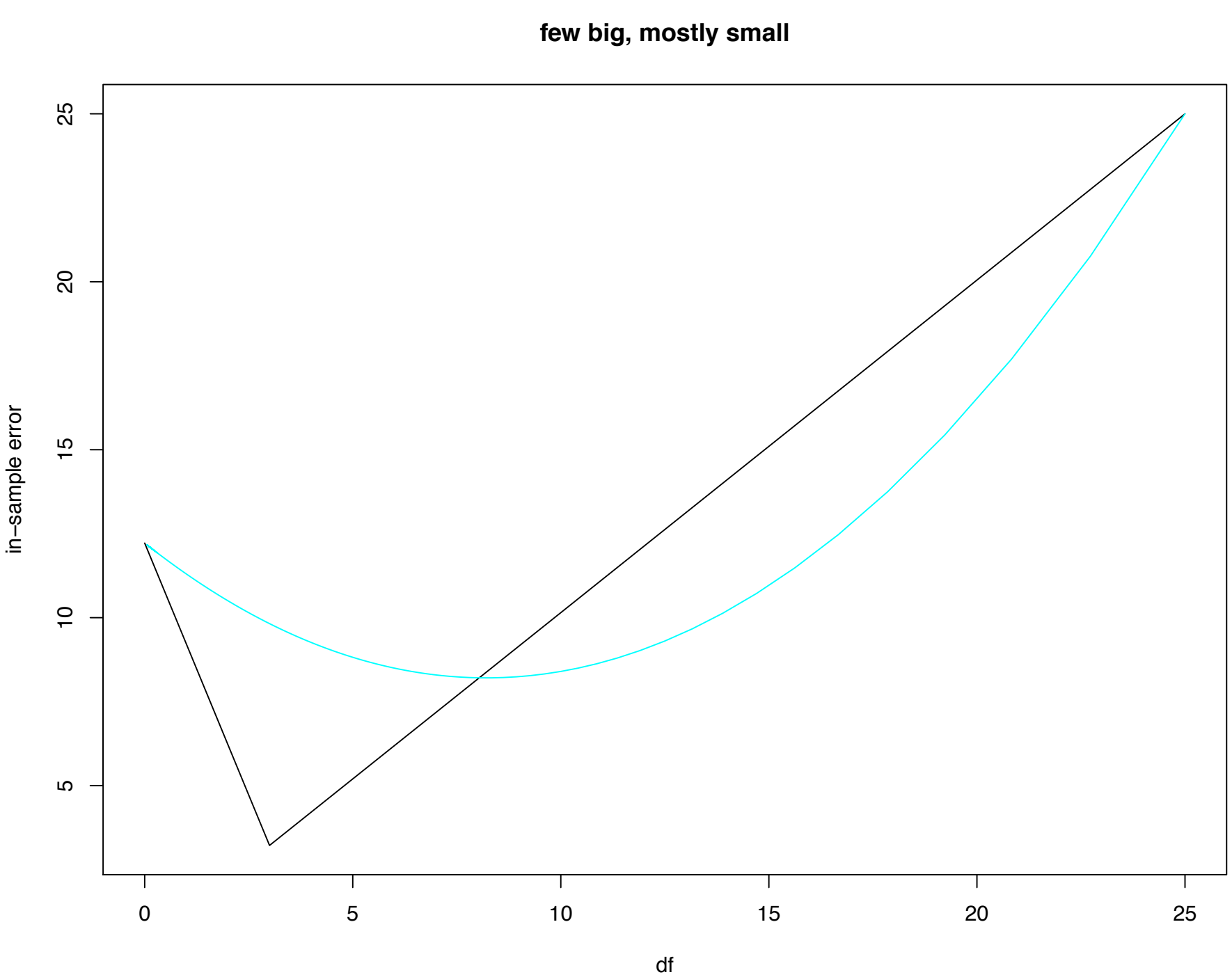
What do you think?

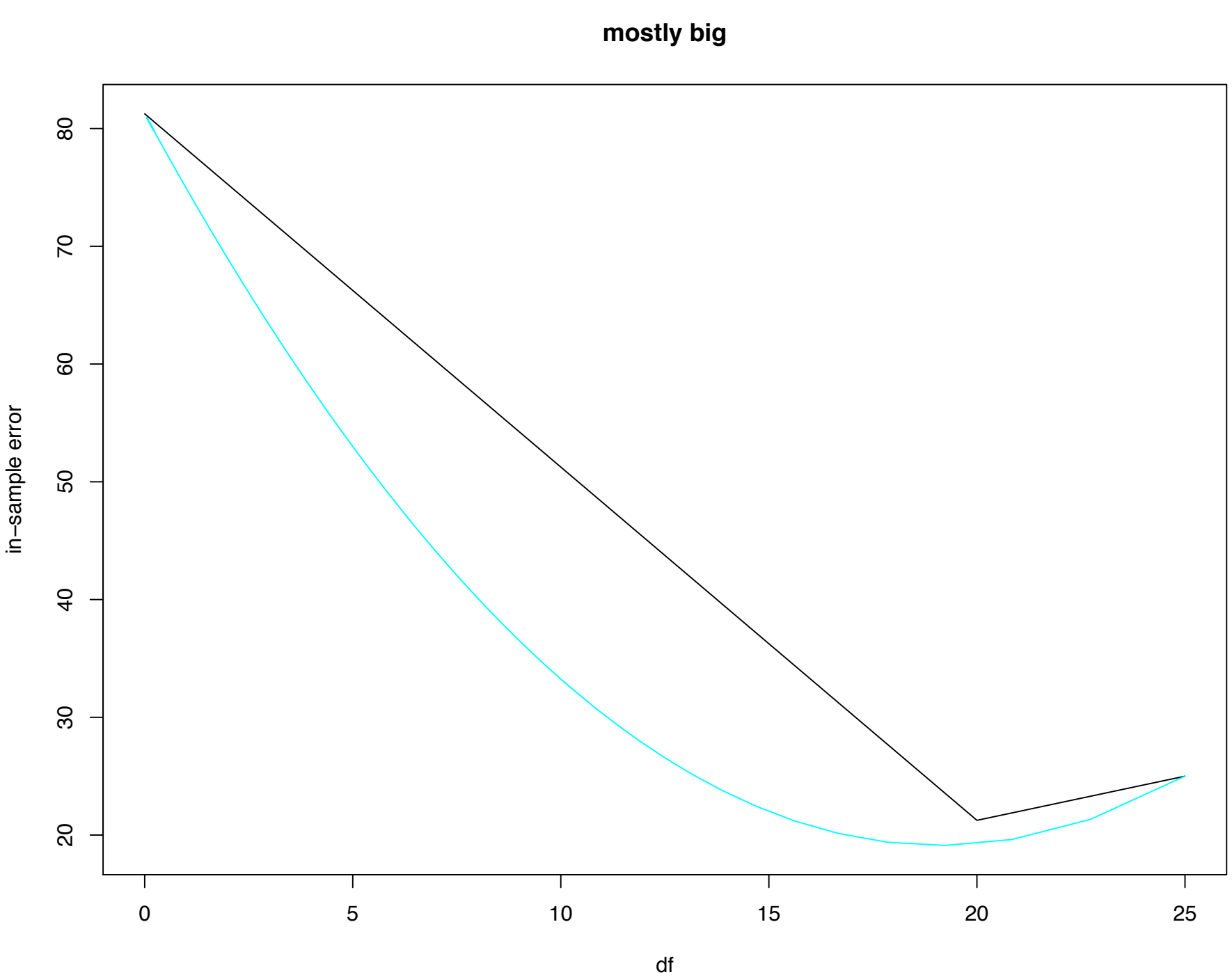

penalized estimate as a function of betahat, lam=1
penalized estimate, ridge
20
10
0
−10
−20
−20
−10
0
10
20
betahat, the ols estimate

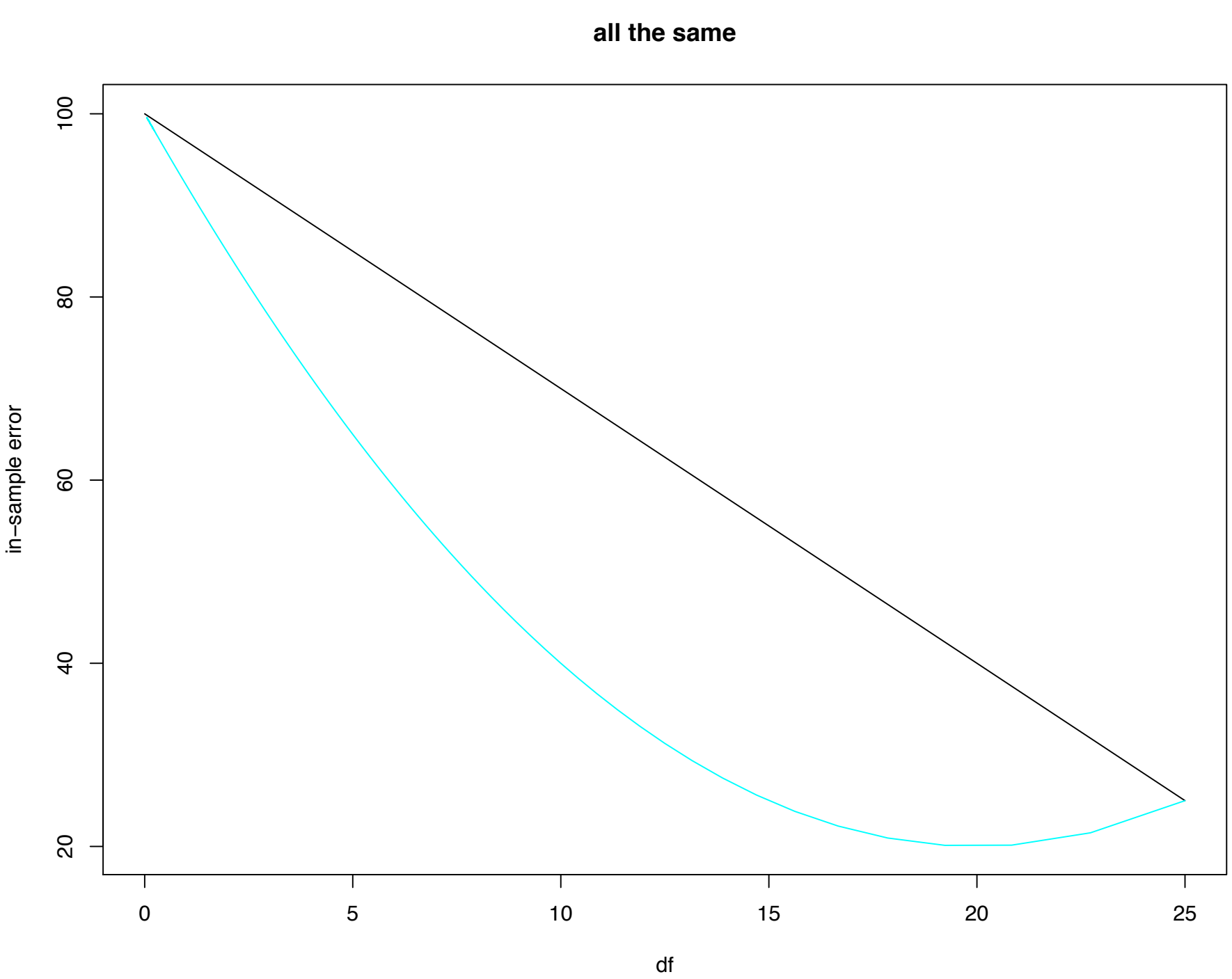

The effect of subset selection
subset selection
4
0
−2
−4
−4
−2
0
2
4
betahat, the ols estimate

```
lambda <- seq(0,500,len=5000)

# set up the problem

p <- 25        # number of predictors
sigma <- 1     # error variance

#beta <- rnorm(p,sd=1.5)               # normals
#beta <- c(rep(2,3),rep(0.1,p-3))      # a few big, mostly small
#beta <- c(rep(2,p-5),rep(0.5,5))      # mostly big
beta <- rep(2,p)                      # all the same

# compute squared bias and variance

b <- (lambda/(1+lambda))^2*sum(beta^2)
v <- p*sigma^2*(1/(1+lambda))^2
df <- p/(1+lambda)

subs <- (p:0)*sigma^2 + cumsum(c(0,sort(beta^2)))

plot(c(0,p),range(c(subs,b+v)),type="n",main="all the same",xlab="df",ylab="in-sample error"
lines(p:0,subs)
lines(df,b+v,col=5)
```


normals
in−sample error
100
80
60
40
20
0
5
10
15
20
25
df


few big, mostly small
in−sample error
5
10
15
20
25
0
5
10
15
20
25
df


mostly big
in−sample error
20
30
40
50
60
70
80
0
5
10
15
20
25
df


all the same
in−sample error
100
80
60
40
20
0
5
10
15
20
25
df

A detour (slight)

The work we've done here is largely theoretical, idealized and essentially motivation -- In the next lecture we'll see how we can make these ideas operational, translating our understanding to general regression problems (ones for which we don't know the coefficients and the error variance before we start fitting!)

For the moment, however, we are going to consider a second kind of penalized approach to regression, one that appeared about 25 years after Hoerl and Kennard's original paper...

Penalties

Suppose that instead of penalizing the sum of squared coefficients, we apply another loss function -- There are various norms we might consider

$$L_2: \ \|\beta\|^2 = \sum_{j=1}^{p} \beta_j^2$$

$$L_1: \ \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$$

$$L_0: \ \|\beta\|_0 = \sum_{j=1}^{p} I_{\beta_j \neq 0} = \text{count of } \beta_j \text{ that are nonzero}$$

The lasso

Like ridge regression, this procedure introduces a penalty on the size of the coefficients

$$\sum_{i=1}^{n}(\widetilde{y}_i - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

where we could also view this problem in terms of minimizing

$$\sum_{i=1}^{n}(\widetilde{y}_i - \beta_1\widetilde{x}_{i1} - \cdots - \beta_p\widetilde{x}_{ip})^2$$

subject to the constraint that $\sum_{j=1}^{p}|\beta_j| < s$

The lasso

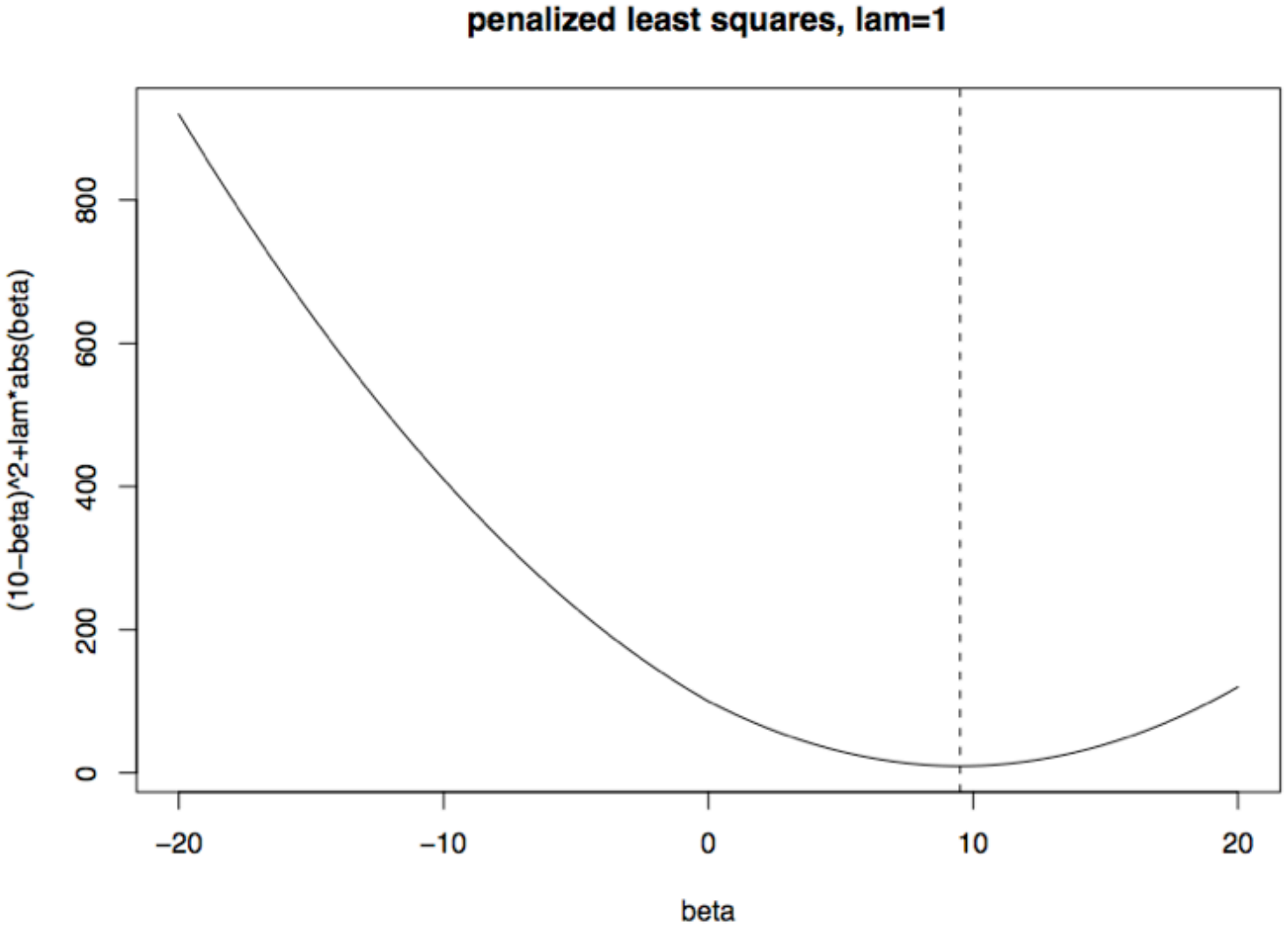
Unlike ridge regression, however, there is not a closed form solution to this problem in general -- We can, however, trot out our good old orthogonal predictors for some insight into the character of the solution

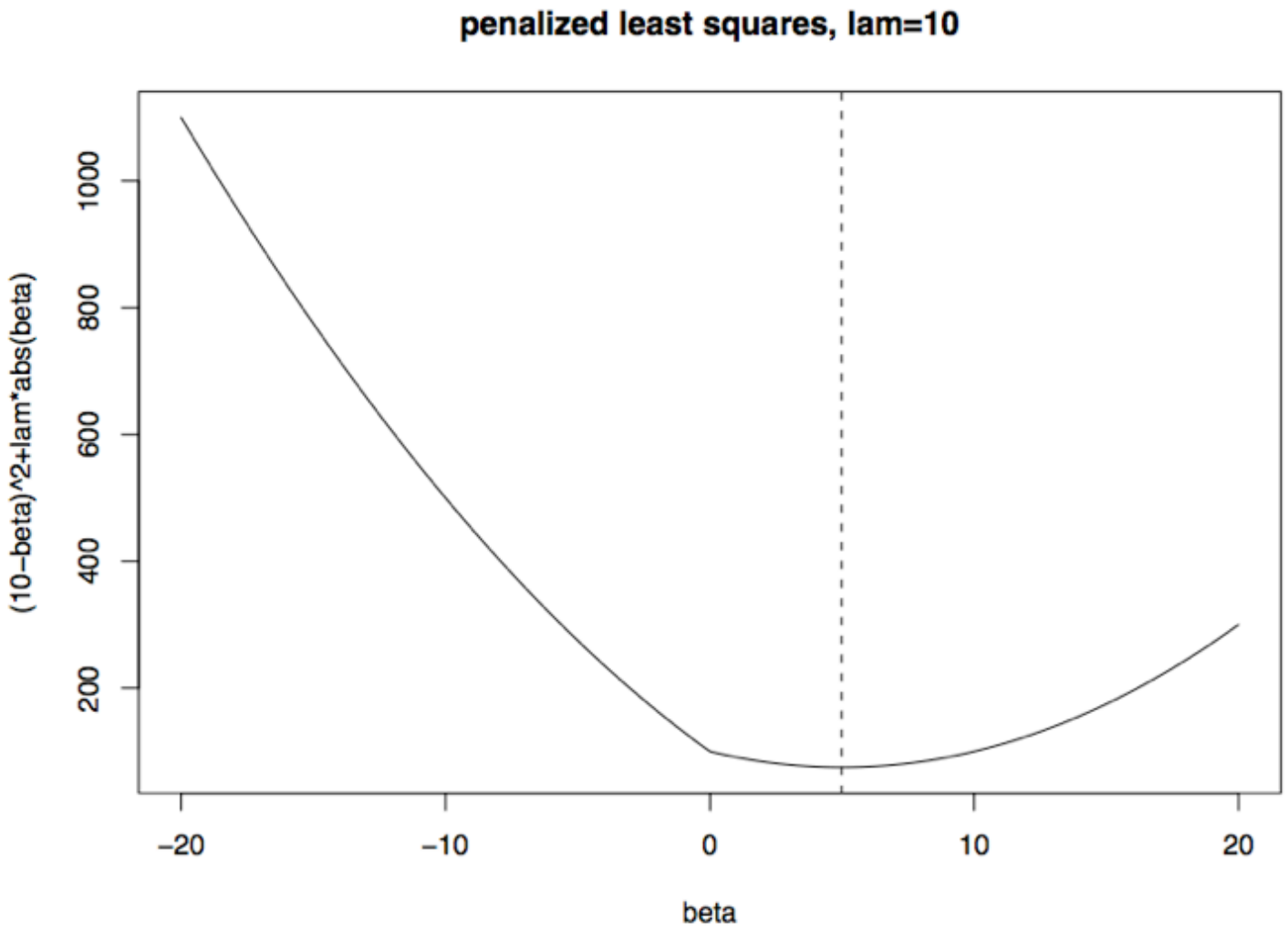
For orthonormal predictors, this quantity simplifies and we can write

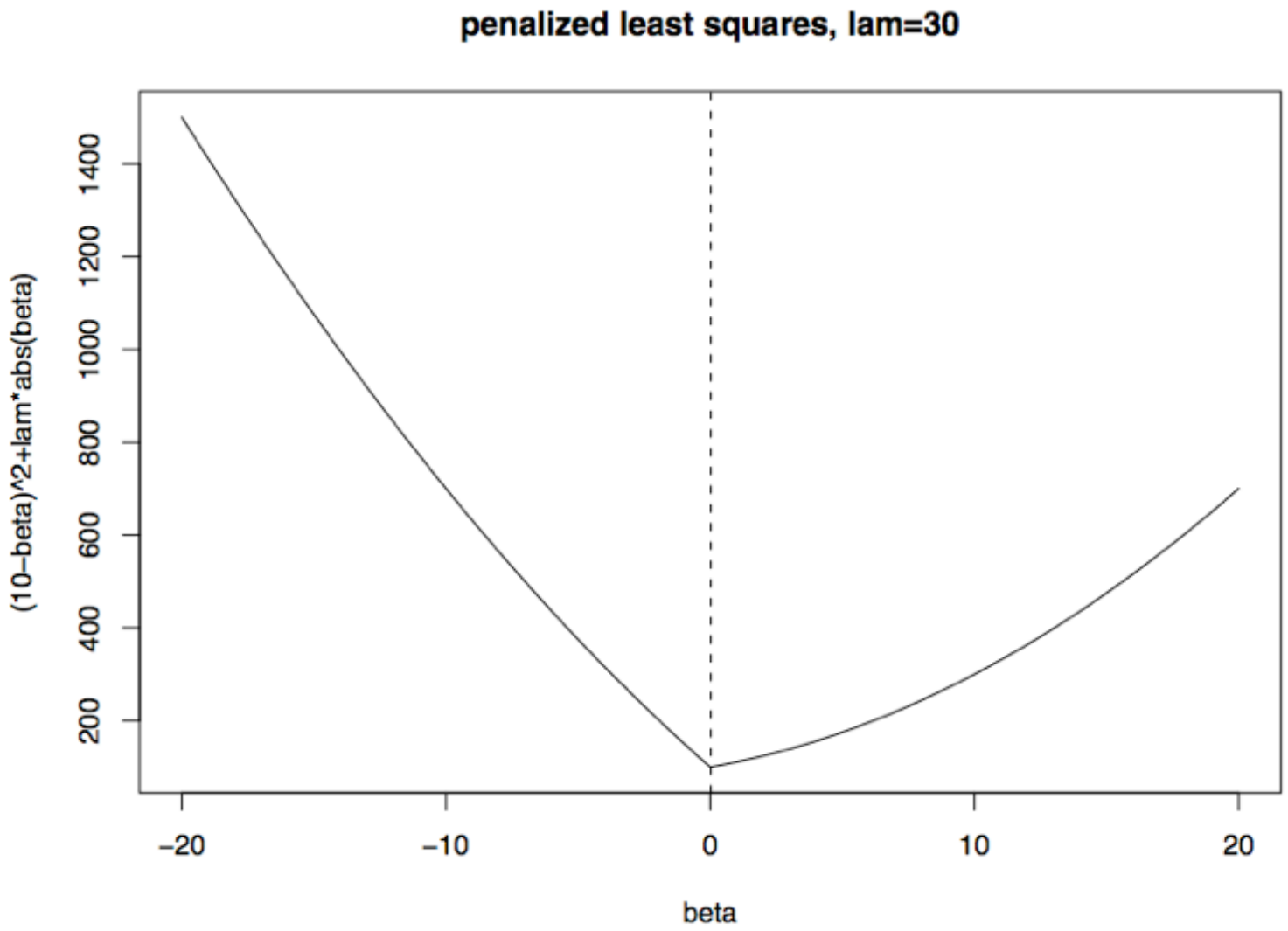
$$\sum_{i=1}^{n} y_i^2 - \sum_{j=1}^{p} \widehat{\beta}_j^2 + \sum_{j=1}^{p} (\beta_j - \widehat{\beta}_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

We can then minimize the expression one coefficient at a time, in each case considering the expression

$$(\beta_j - \widehat{\beta}_j)^2 + \lambda |\beta_j|$$


penalized least squares, lam=1
(10−beta)^2+lam*abs(beta)
0
200
400
600
800
−20
−10
0
10
20
beta


penalized least squares, lam=10
(10−beta)^2+lam*abs(beta)
200
400
600
800
1000
−20
−10
0
10
20
beta

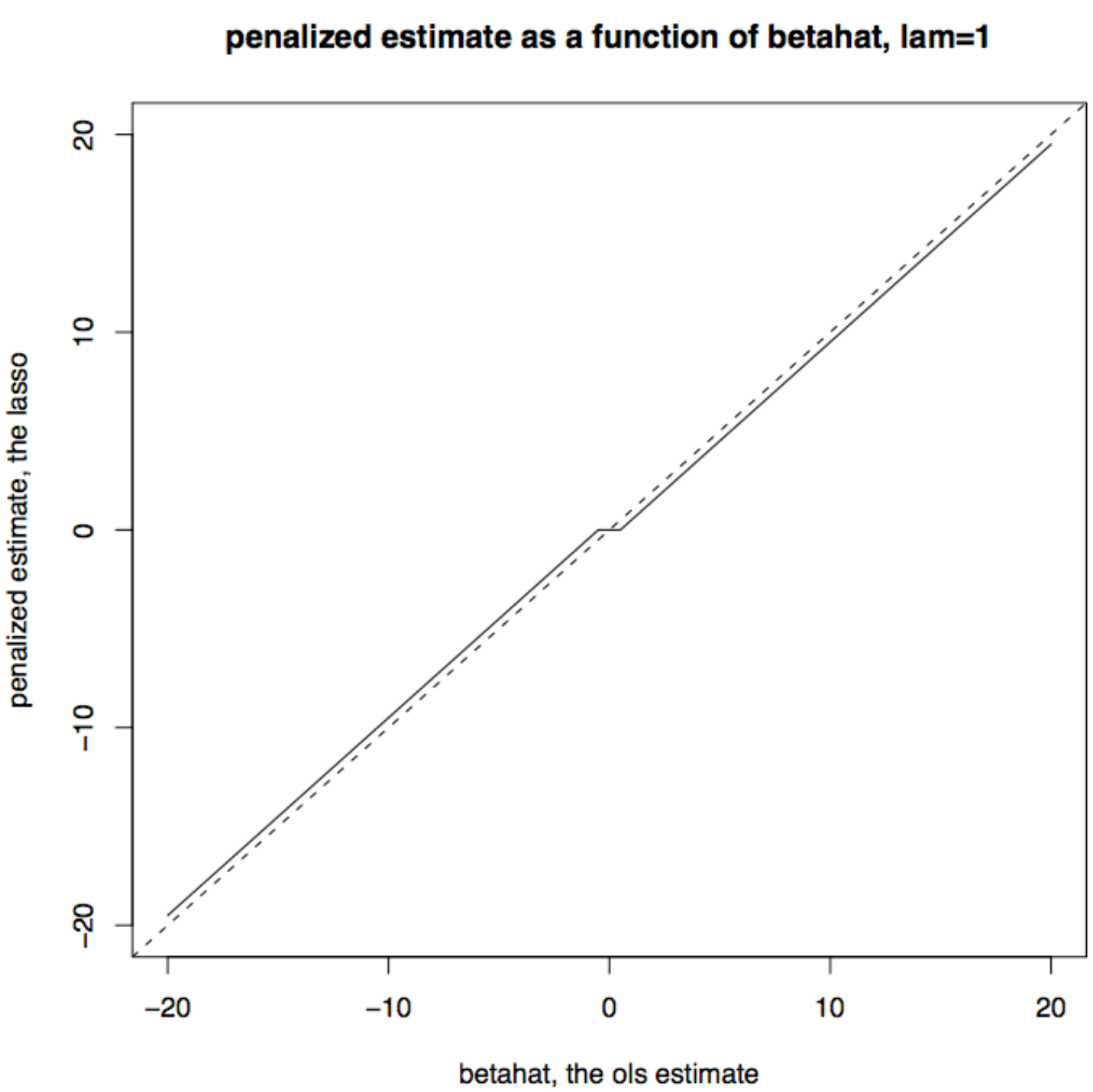

penalized least squares, lam=30
(10−beta)^2+lam*abs(beta)
200
400
600
800
1000
1200
1400
−20
−10
0
10
20
beta

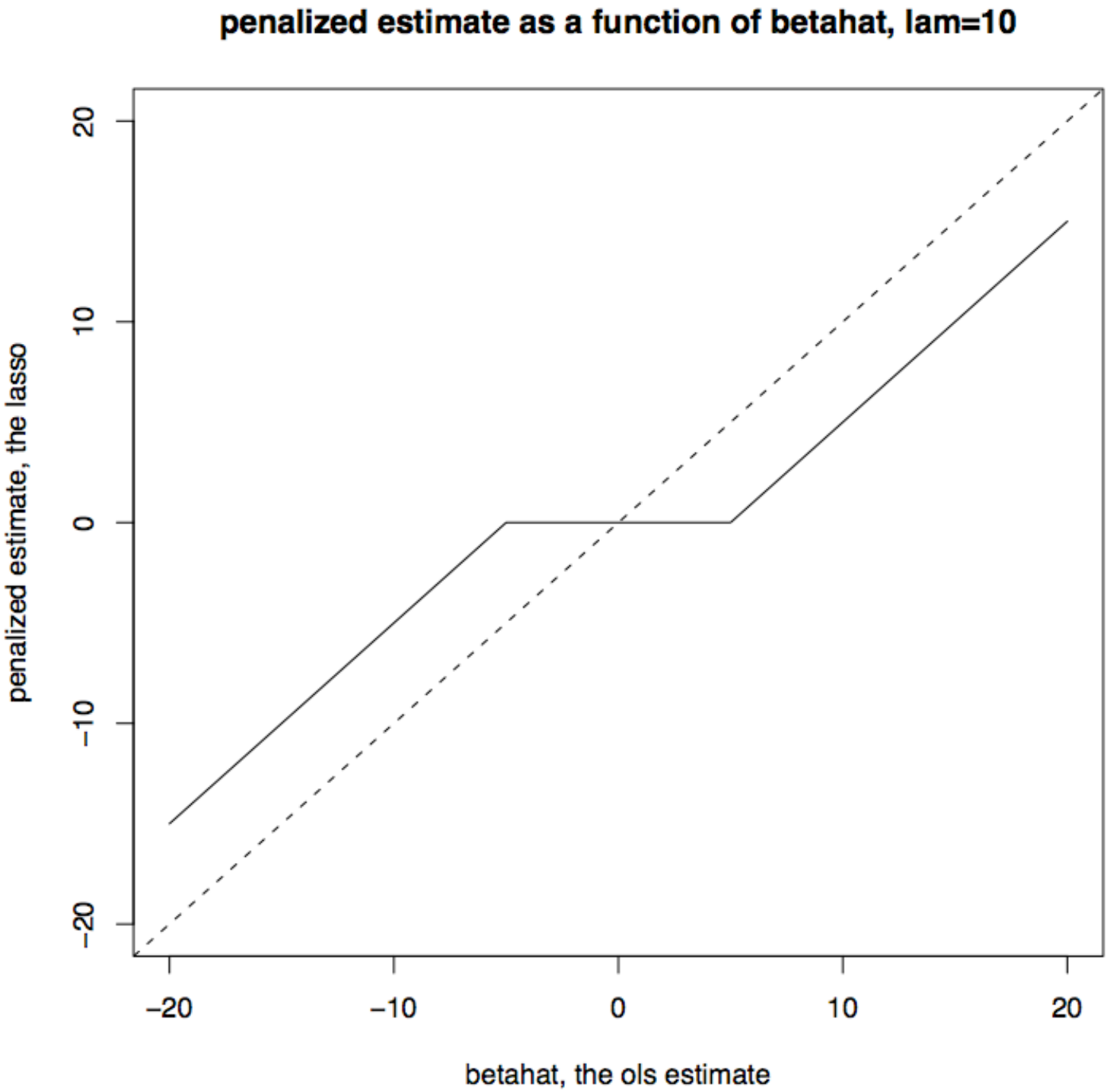
The lasso

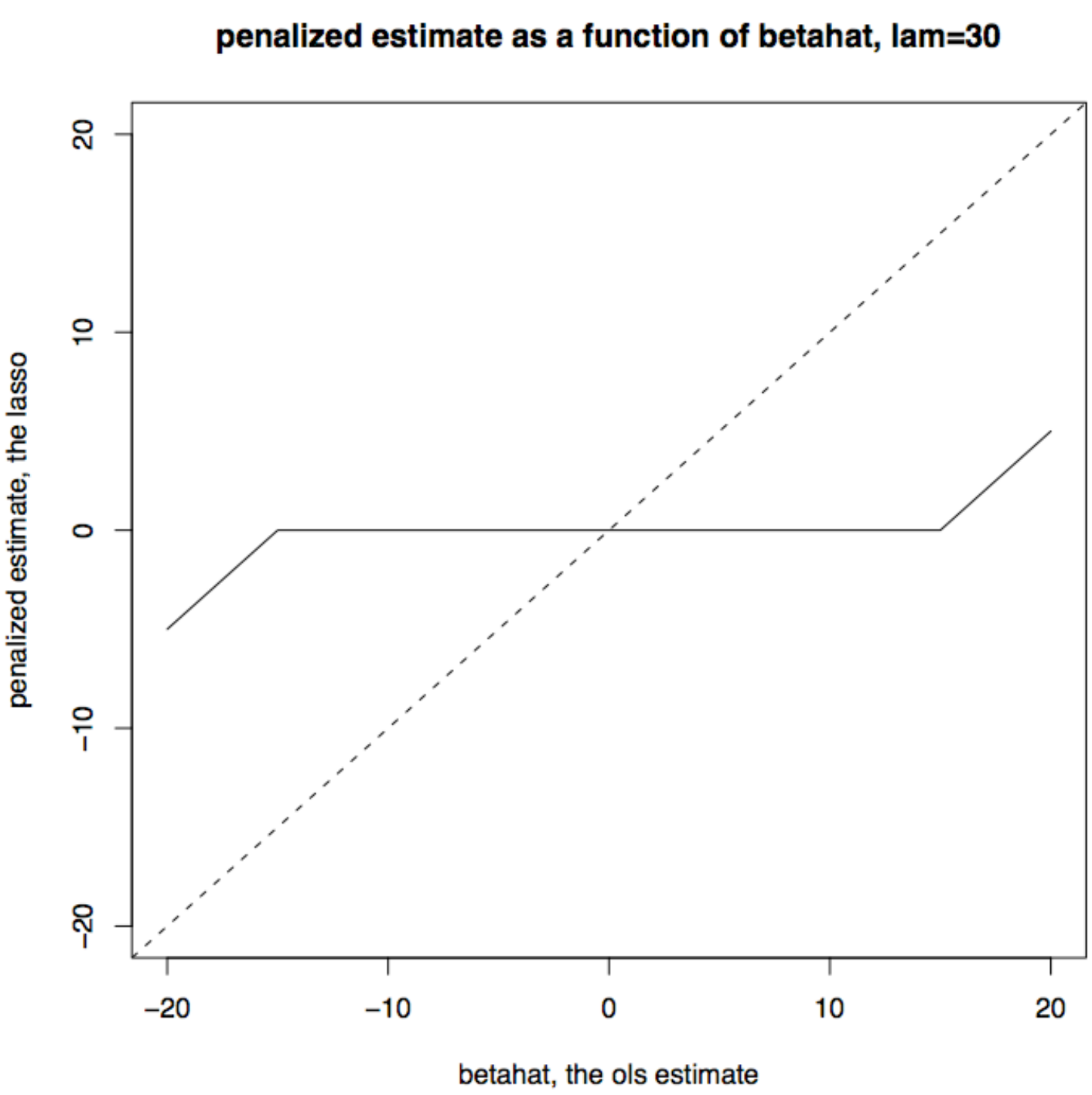
While this criterion is discontinuous at 0, we can look for the minimum on either size of 0 and compare it to the value at 0 -- After the dust settles we end up with the expression

$$\widetilde{\beta}_j = \text{sign}(\widehat{\beta}_j)(\,|\widehat{\beta}_j| - \lambda/2)_+$$

where $(z)_+ = \max(z, 0)$ and sign(z) is +1 or -1 depending on whether z is larger or smaller than zero


penalized estimate as a function of betahat, lam=1
penalized estimate, the lasso
betahat, the ols estimate
20
10
0
−10
−20
−20
−10
0
10
20


penalized estimate as a function of betahat, lam=10
penalized estimate, the lasso
betahat, the ols estimate
20
10
0
−10
−20
−20
−10
0
10
20

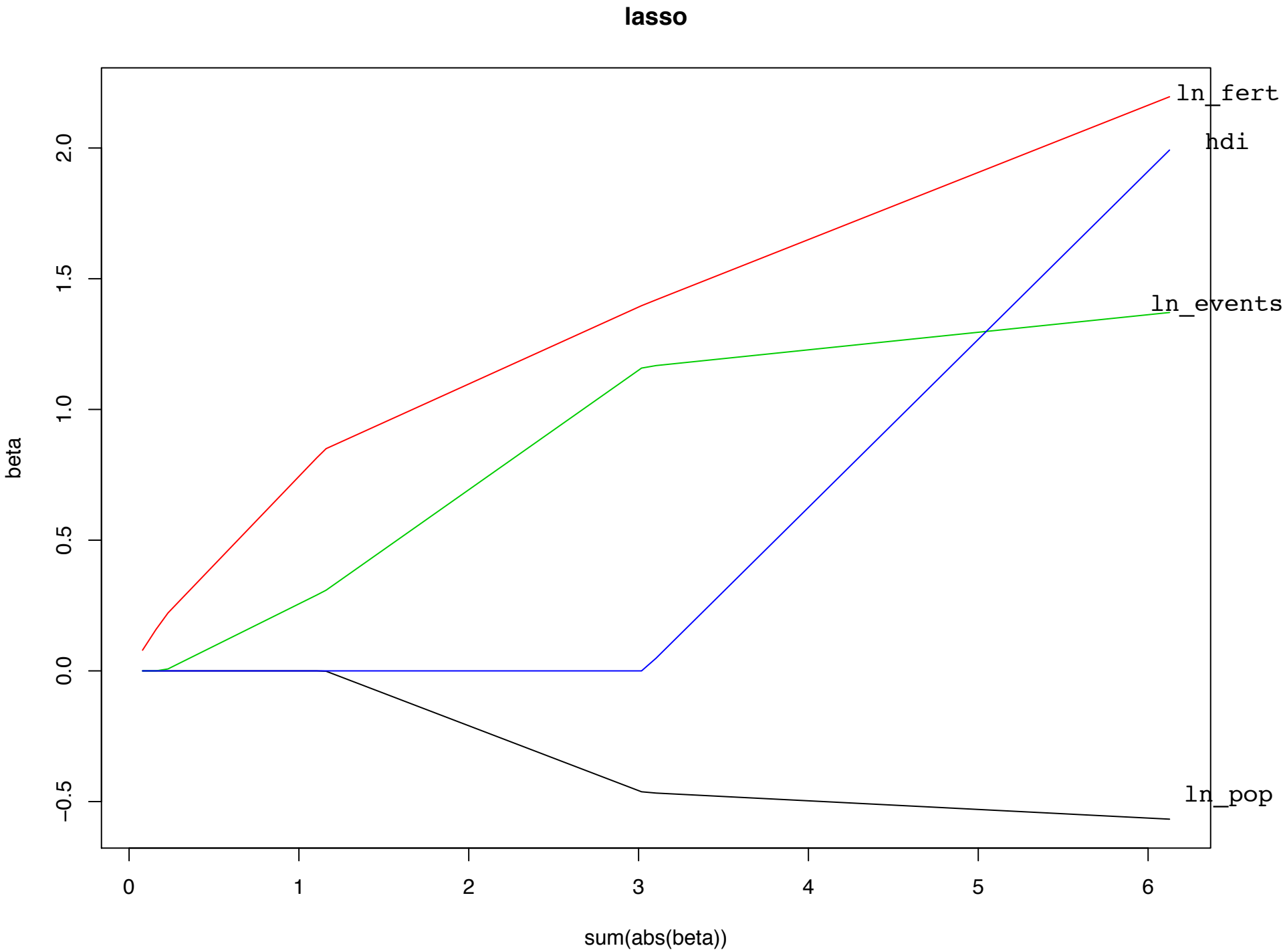

penalized estimate as a function of betahat, lam=30
penalized estimate, the lasso
betahat, the ols estimate
20
10
0
−10
−20
−20
−10
0
10
20

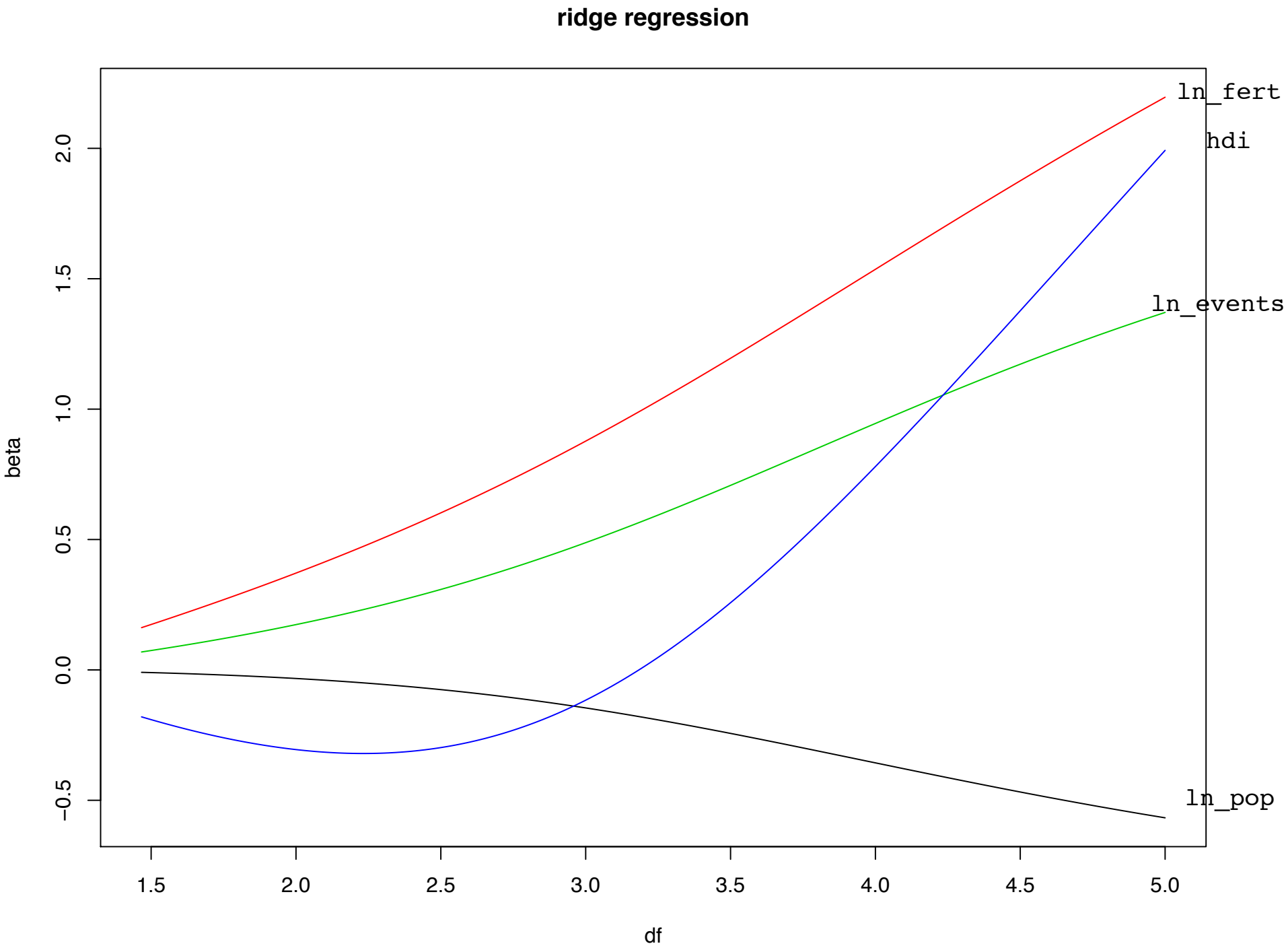
The lasso

For orthogonal problems we see that **some of the coefficients are explicitly set to zero** (mimicking subset selection) and **the others are pulled closer to zero** (like ridge)

The same is true for non-orthogonal problems -- Let's consider a lasso fit to the vulnerability data and in the process look at two different R packages for working with these models

```
library(lasso2)

# this library works directly with the constraints of the form
# sum(abs(beta)) <= bound -- they also, by default, use the fact
# that the OLS fits are the largest (in 1-norm) you can have so
# that the "bound" is relative to this largest value and takes on
# any number between 0 and 1 (0 meaning all beta's are forced to
# 0 and 1 meaning the beta's are free to take on their OLS values)

fit <- l1ce(ln_death_risk~ln_pop+ln_fert+ln_events+hdi,
            data=vul,bound=(1:100)/100)

cc <- coefficients(fit)[,-1]

matplot(apply(cc,1,function(x) sum(abs(x))),coefficients(fit)[,-1],
         type="l",lty=1,xlab="sum(abs(beta))",ylab="beta",main="lasso")


# now let's recall ridge regression on the same set of variables

library(MASS)

fit <- lm.ridge(ln_death_risk~ln_pop+ln_fert+ln_events+hdi,
                data=vul,lambda=0:1000)

matplot(fit$df,coefficients(fit)[,-1],
         type="l",lty=1,xlab="df",ylab="beta",main="ridge regression")
```


lasso
ln_fert
hdi
ln_events
ln_pop
beta
sum(abs(beta))
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6

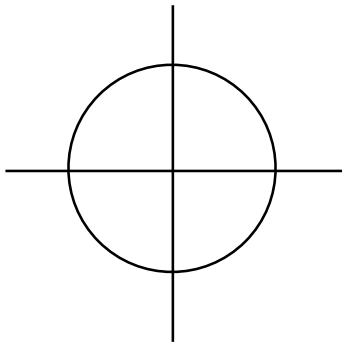

ridge regression
ln_fert
hdi
ln_events
ln_pop
beta
df
2.0
1.5
1.0
0.5
0.0
−0.5
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

The lasso

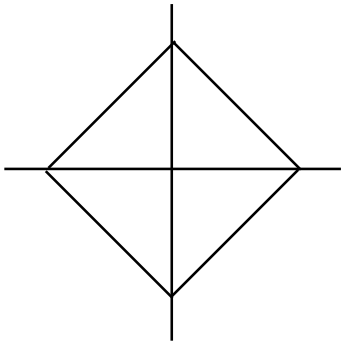
As expected, the penalty has the effect of explicitly setting some coefficients to zero while making the others smaller in absolute value than their least squares counterparts

The intuition for this behavior comes from looking at the “unit balls” for each of the two penalties -- In two dimensions (two predictors), for example, the standard Euclidean norm is a circle

$$\|\beta\| = \sqrt{\beta_1^2 + \beta_2^2} = 1 \quad \text{or equivalently} \quad \beta_1^2 + \beta_2^2 = 1$$



while the 1-norm is a diamond

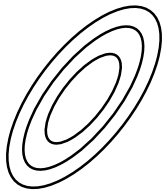


$$\|\beta\|_1 = |\beta_1| + |\beta_2| = 1$$

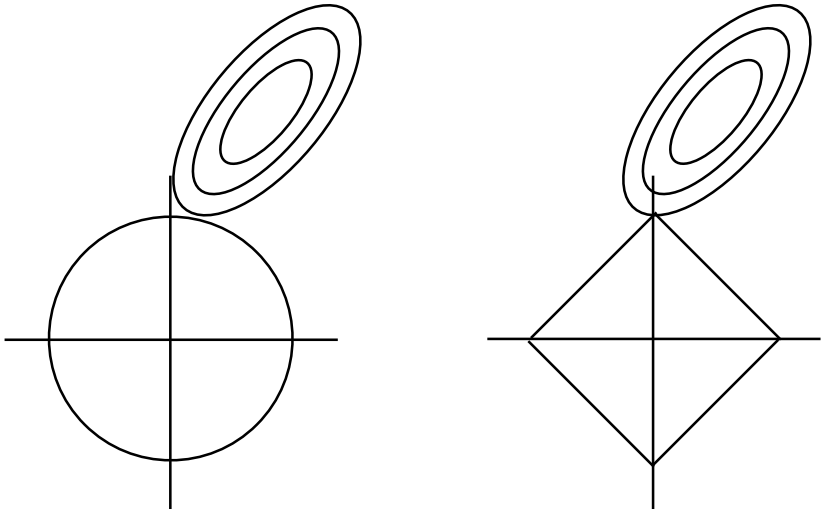
The lasso

These two penalties are then put into competition with the elliptical contours of the loss function

$$\|\mathsf{y} - \mathsf{M}\beta\|^2 = (\mathsf{y} - \mathsf{M}\beta)^{\mathsf{t}}(\mathsf{y} - \mathsf{M}\beta)$$



The solution to the penalized problem, then, occurs where the constraint region touches the elliptical contours of the OLS criterion -- For the lasso, there's a chance this will be at one of the corners, yielding a zero



Interpretations

We can relate the penalized approaches to the Bayesian linear model and interpret he penalties function as priors (Gaussian, double exponential) -- We will see how these ideas can be used to motivate so-called model averaging (rather than hard selection)

We will also see how the lasso relates to stagewise regression (a variant on stepwise) through a relatively recent technique known as least angle regression or LARS