Penalized Regression

regularized regression

November 11, 2021

overview

1. James Stein estimator
2. Lasso and ridge regression
3. interpreting ridge (might jump)
4. Example in genetics
5. Bayesian connection
6. A word of my own research

Stein Paradox

- We will start with one of the arguably surprising result in statistics, namely the Stein Paradox.
- In 1956, it was shown that for a simple example the regular Maximal likelihood estimator is not optimal.
- We will look a strictly better shrinkage estimator from 1961.

References

- C. Stein (1956) *Inadmissibility of the usual estimator of the mean of a multivariate normal distribution*, Proc. Third Berkeley Symposium, 1, 197–206, Univ. California Press
- W. James and C. Stein (1961), *Estimation with quadratic loss*, Proc. Fourth Berkeley Symposium, 1, 361–380.

James Stein estimator

- Suppose that
$$Y \sim \mathcal{N}_p(\boldsymbol{\mu}, I)$$
- Goal find an estimator of $\boldsymbol{\mu}$ given we observed a single observation, $Y = y$.
- What is the best estimator in terms of squared error
$$L(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = ||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||^2 = \sum_{i=1}^{p} (\hat{\mu}_i - \mu_i)^2$$
- The Maximum likelihood is the sample mean $\hat{\boldsymbol{\mu}}^{mle} = y$ (recall $n = 1$).

James Stein estimator

- For $\hat{\boldsymbol{\mu}}^{\text{mle}}(\mathsf{Y}) = \mathsf{Y}$ we can analyse the expected loss

$$\mathbb{E}_{\mathsf{Y}}\left[||\hat{\boldsymbol{\mu}}^{\text{mle}}(\mathsf{Y}) - \boldsymbol{\mu}||^2\right] = \mathbb{E}_{\mathsf{Y}}\left[||\mathsf{Y} - \boldsymbol{\mu}||^2\right]$$

  Using $\mathsf{Y} = \boldsymbol{\mu} + \mathsf{Z}$ where $\mathsf{Z} \sim \mathcal{N}(0, \mathbf{I})$ we get

$$\mathbb{E}_{\mathsf{Y}}\left[||\mathsf{Y} - \boldsymbol{\mu}||^2\right] = \mathbb{E}_{\mathsf{Z}}\left[||\mathsf{Z}||^2\right] = \mathrm{p}.$$

- For $\mathrm{p} = 1, 2$ this is the optimal estimator, however for $\mathrm{p} \geq 3$ it is not the case!

James Stein estimator

Theorem (James and Stein (1961))

Let $Y \sim \mathcal{N}_p\left(\boldsymbol{\mu}, \sigma^2 I\right)$, *and* $L(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \mathbb{E}_Y\left[||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||^2\right]$ *then for* $p \geq 3$

$$L(\hat{\boldsymbol{\mu}}^{JS}, \boldsymbol{\mu}) \leq L(\hat{\boldsymbol{\mu}}^{MLE}, \boldsymbol{\mu}).$$

Here $\hat{\boldsymbol{\mu}}^{JS} = \left(1 - \sigma^2 \frac{p-2}{||Y||^2}\right) Y$.

[1]

- One can further prove that $\hat{\boldsymbol{\mu}}^{JS+} = \left(1 - \sigma^2 \frac{p-2}{||Y||^2}\right)_+ Y$ is even better.

---

[1] Samworth, Richard J., and Statslab Cambridge. "Stein's paradox." eureka 62 (2012): 38-41.

Baseball data

- We want to predict the batting average of eighteen baseball players the season 1970. We will use the betting average of the players for each players first 45 bats.
- Number of hits $H_i \sim \text{Bin}(n = 45, p_i)$.
- The MLE estimator is $\hat{p}_i = \frac{h_i}{n}$.

```
library(Rgbp)
data(baseball)
p <- baseball$Hits/baseball$At.Bats
p.true <- baseball$RemainingAverage
p.MLE <- p
```

simple James Stein estimator I

- To use the James Stein estimator we need to know the standard deviation which we estimate from the data.
- Using $\mathbb{V}\left[\frac{H_i}{n}\right] = \frac{1}{n}p_i(1-p_i)$ if $H_i \sim \text{Bin}(n, p_i)$
- Pool the estimate.

```
pbar  <- mean(p)
sigma2 <- pbar * (1-pbar)/baseball$At.Bats
p.JS <- (1 -sigma2/(length(p)-2) ) * p
```

simple James Stein estimator II

If we now compare square root of the mean square error:

```
Loss.MLE  = sqrt(mean((p.MLE - p.true)^2))
Loss.JS = sqrt(mean((p.JS - p.true)^2))
cat('Loss.MLE = ',  round(Loss.MLE,digits = 4),'\n'
```

```
## Loss.MLE =  0.069
```

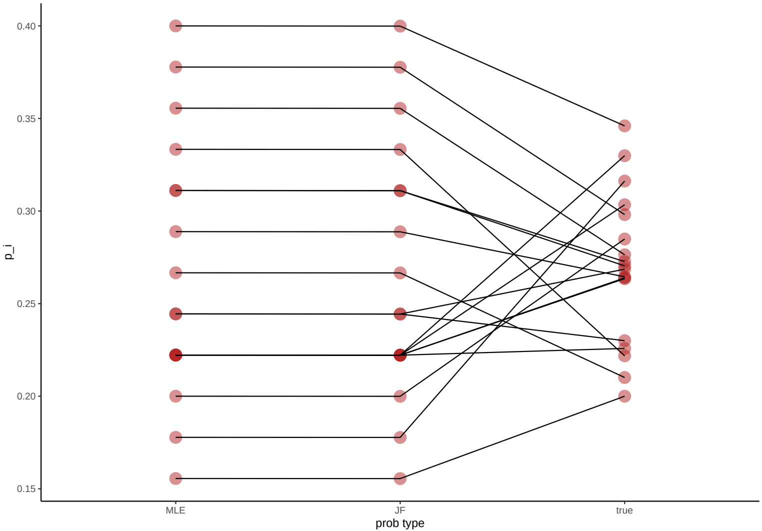
```
cat('Loss.JS = ', round(Loss.JS,digits=4),'\n')
```

```
## Loss.JS =  0.069
```

```
cat('RATIO = ',round(Loss.JS/Loss.MLE,6),'\n')
```

```
## RATIO =  0.999837
```

simple James Stein estimator III



James Stein estimator

- This is a shrinkage estimator, it pulls our estimator towards 0.
- But 0 is arbitrary and one can use any arbitrary point, $\boldsymbol{\mu}_0$ and get

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \boldsymbol{\mu}_0 + \left(1 - \sigma^2 \frac{p-2}{||Y - \boldsymbol{\mu}_0||^2}\right)(Y - \boldsymbol{\mu}_0).$$

- Can we use this in any practical way? Can you think about some better point to contract towards?

simple James Stein estimator IV

```
pbar <- mean(p)
p.JS <- pbar + (1 - sigma2*(length(p)-2)/sum((p-pbar)^2)
```

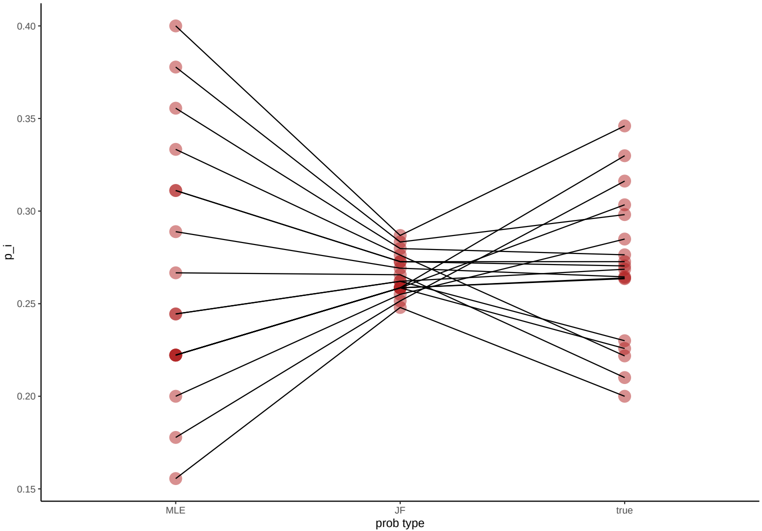
```
cat('Loss.JS = ', round(Loss.JS,digits=4),'\n')
```

```
## Loss.JS =  0.0384
```

```
cat('RATIO = ',round(Loss.JS/Loss.MLE,6),'\n')
```

```
## RATIO =  0.555981 (compared to  0.999837)
```

simple James Stein estimator VI



Regularization methods

- Now we will look at the two main regularizes in linear regression. So the base setting is

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

- ridge regression one adds a quadratic penalty term to least square regression:

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.41}$$

Regularization methods 2

- Ridge regression solution:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.41}$$

- lasso

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.52}$$

Simple model

- To understand the the two methods one can examine the one dimensional problem $X = 1$ then we are solving

$$f(\beta) = (y - \beta)^2 + \lambda\beta^2$$

  for ridge regression. And

$$f(\beta) = (y - \beta)^2 + \lambda|\beta|$$

  for lasso.

- Thus

$$\hat{\boldsymbol{\beta}}^{\text{bridge}} = \frac{y}{1 + \lambda},$$

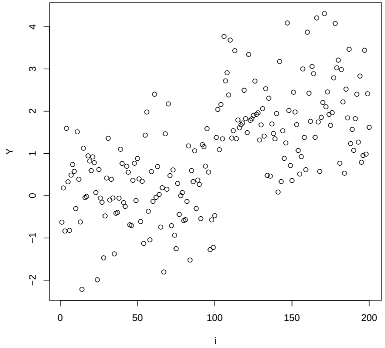
  and

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \text{sign}(y) \left(|y| - \lambda\right)_+ .$$

Simulation

Let $Y_i \sim \mathcal{N}(\beta_i, 1)$ where

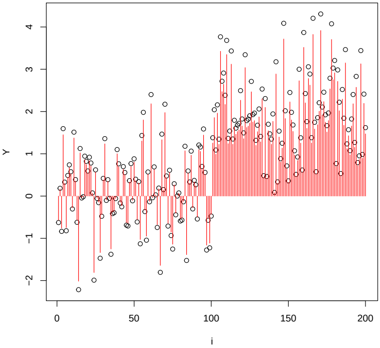
$$\beta_i = \begin{cases} 0 & \text{if } i \leq 100 \\ 2 & \text{if } 100 < i \leq 200 \end{cases}$$



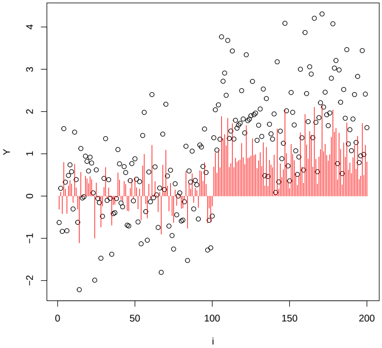
Ridge on Y

$$\hat{\beta}_i^{\text{bridge}} = \frac{y_i}{1+\lambda},$$

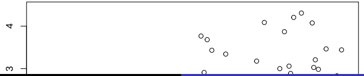
Ridge regression $\lambda = 0.1$



Ridge regression $\lambda = 1$

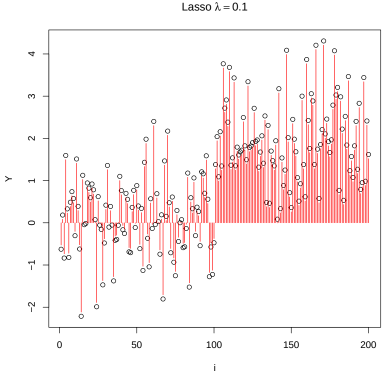


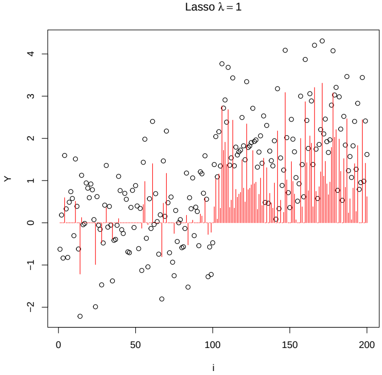
Ridge regression $\lambda = 3$



Lasso on Y

- $$\hat{\beta}_i^{\text{lasso}} = \text{sign}(y_i)\,(|y_i| - \lambda)_+ \,.$$
- Note that the sparsity is a variable selection tool: $S = \{j : \beta_j \neq 0\}$





Multivariate

- We know will look adding l2 (ridge) and l1 (lasso) penalty to the general OLS/Normal:

$$L(\beta) = \frac{1}{2}(y - X\boldsymbol{\beta})^T(y - X\boldsymbol{\beta})$$

centering and scaling

- We center the data (remove the mean) and scale the data, i.e.

$$X_{i.} = \frac{X_{i.} - \bar{X}_{i.}}{\widehat{sd}(X_{i.})} \;\; \forall\, i,$$

and $y = y - \bar{y}$

Ridge regression

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\arg\min} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (3.41)$$

- One can rewrite equation (3.41) to

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\arg\min} \, (y - X\boldsymbol{\beta})^T (y - X\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta} \qquad (3.43)$$

- Some calculus then gives

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \left(X^T X + \lambda I \right)^{-1} X^T y \qquad (3.44)$$

Thus $\hat{\boldsymbol{\beta}}^{\text{ridge}}$ has an explicit solution as function of λ.

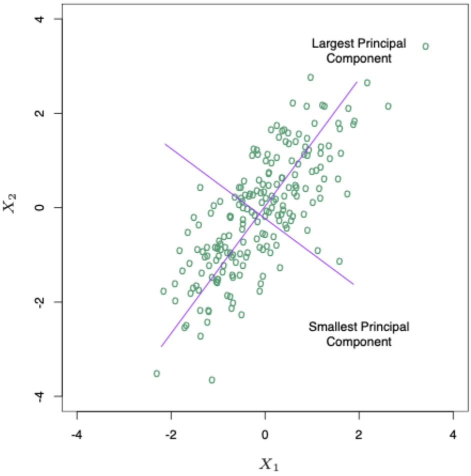
Ridge regression interpretation

- From our previous one dimensional example one could see that λ will pull the coefficients towards zero.
- The larger λ the less the data affects $\hat{\beta}^{\text{ridge}}$.
- It is easier to interpret how the shrinkage affects $\hat{\mathrm{y}} = \mathrm{X}\hat{\beta}^{\text{ridge}}$ than the coefficients.
- The interpretation builds one the singular value decomposition of X:

$$\mathrm{X} = \mathrm{UDV}^{\mathrm{T}}$$

Here U a $\mathrm{n} \times \mathrm{p}$ matrix and D is a $\mathrm{p} \times \mathrm{p}$ matrix which is different from the full SVD.

SVD



- The matrix DU has columns $d_i U_i$ which are known as the principal components of X.

SVD regression

- For OLS:

$$\hat{y} = X\hat{\beta}^{OLS} = UU^T y$$

- For Ridge regression:

$$\hat{y} = X\hat{\beta}^{OLS} = U\mathrm{Diag}\left(\frac{d_i^2}{d_i^2 + \lambda}\right)U^T y$$

Ridge regression interpretation

- For linear regression we have the classical result:

$$\mathrm{df}(\hat{y}) = \mathrm{tr}(H) = \mathrm{tr}\left(X\left(X^T X\right)^{-1} X^T\right) = \mathrm{tr}\left(UU^T\right) = p.$$

- For ridge

$$\begin{aligned}\mathrm{df}(\hat{y}) &= \mathrm{tr}(H_\lambda) = \mathrm{tr}\left(X\left(X^T X + \lambda I\right)^{-1} X^T\right) \\ &= \ldots = \sum_{i=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}.\end{aligned}$$

Application

- One of the main advantage with Lasso and ridge is they can be used for problems when $p > n$.
- A typical application is genetics. Here
  - X_i– samples were scanned with a microarray, that measures the expression of 10000s of genes simultaneously.
  - y_i– time for severe breast cancer to metastasize.
  - The goal is to identify patients with poor prognosis in order to administer more aggressive follow-up treatment for them.
- Two typical genetic ”models”
  - quantitative trait loci (QTL) a single or few important gene.
  - Polygene: many genes with small individual effect.

Which model is lasso which model is ridge?

```r
D = read.table("vantveer.txt", header = T)
print(dim(D))
X = as.matrix(D[,2:ncol(D)])
y = D$Months
```

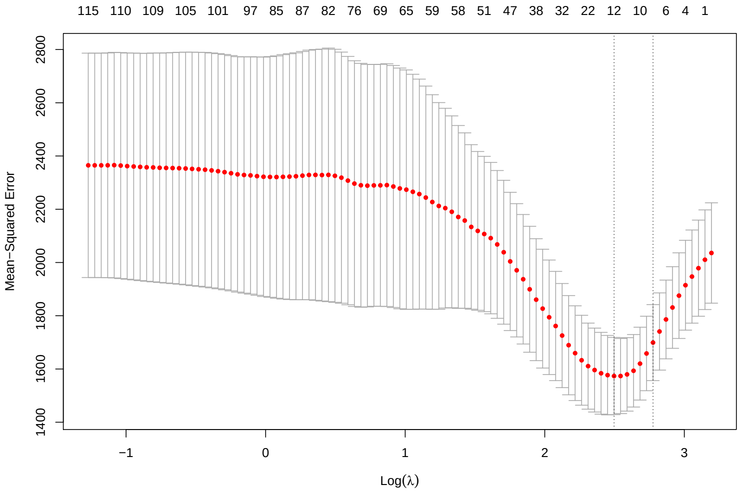
```
## [1]    98 24189
```

fit λ (lasso)

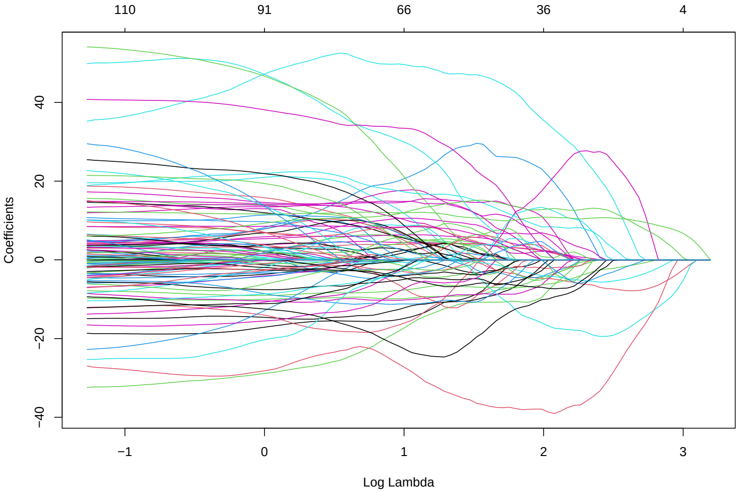
- Go to package in R **glmnet** for both lasso and ridge.
- Need to find λ, use k-fold cross-validation.
- We start with lasso ($\alpha = 1$)

```
library(glmnet)
cvfit <- cv.glmnet(X, y, alpha=1, nfolds = 10, intercept = T, standardize = T)
plot(cvfit)
```

fit λ II (lasso)



fit λ III (lasso)

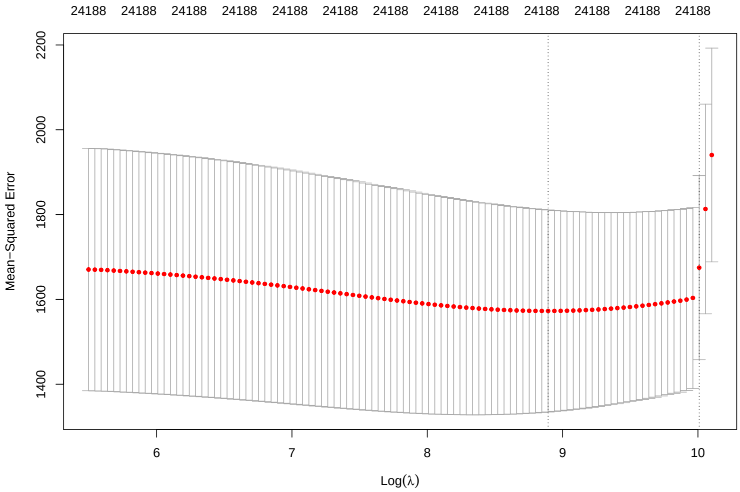


ridge

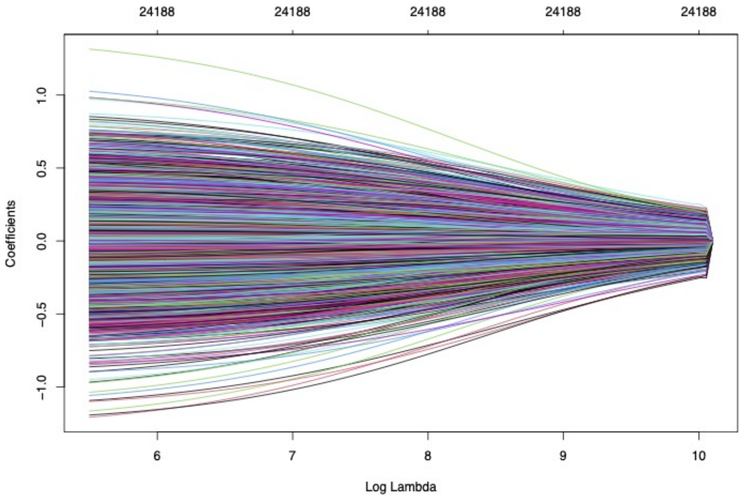
- Then ridge ($\alpha = 0$), fitting λ the same way

```r
cvfit <- cv.glmnet(X, y, alpha=0, nfolds = 10, intercept = T, standardize = T)
plot(cvfit)
```

ridge II


24188 24188 24188 24188 24188 24188 24188 24188 24188 24188 24188 24188 24188
Mean−Squared Error
1400
1600
1800
2000
2200
6
7
8
9
10
Log(λ)

ridge III


24188
24188
24188
24188
24188
1.0
0.5
0.0
−0.5
−1.0
Coefficients
6
7
8
9
10
Log Lambda

The Bayesian connection I:ridge

- Ridge regression solution:

$$\hat{\boldsymbol{\beta}}^{\text{bridge}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (3.41)$$

- A Probabilistic interpretation of the regularization is

$$\frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2 = \frac{\lambda}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} = \frac{\lambda}{2} (\boldsymbol{\beta} - 0)^T (\boldsymbol{\beta} - 0) .$$

Thus the ridge penalty can be considered a prior

$$\boldsymbol{\beta} \sim \mathcal{N}\left(\boldsymbol{\beta}; 0, \frac{1}{\lambda} I\right)$$

- And the ridge solution is thus the MAP (maximum a posteriori) estimate of:

$$\pi(\boldsymbol{\beta}|y, \lambda) \propto \mathcal{N}(y; X\boldsymbol{\beta}, I)\, \mathcal{N}\left(\boldsymbol{\beta}; 0, \frac{1}{\lambda} I\right)$$

The Bayesian connection II:lasso

- Lasso:

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad (3.52)$$

- A Probabilistic interpretation of the regularization

$$\frac{\lambda}{2} \sum_{j=1}^{p} |\beta_j|$$

is that is log density of p independent variables with Laplace distributions

$$\boldsymbol{\beta} \sim \prod_{i=1}^{p} \text{Laplace}\left(0, \frac{1}{\lambda}\right)$$

- And the lasso is thus the MAP estimate of:

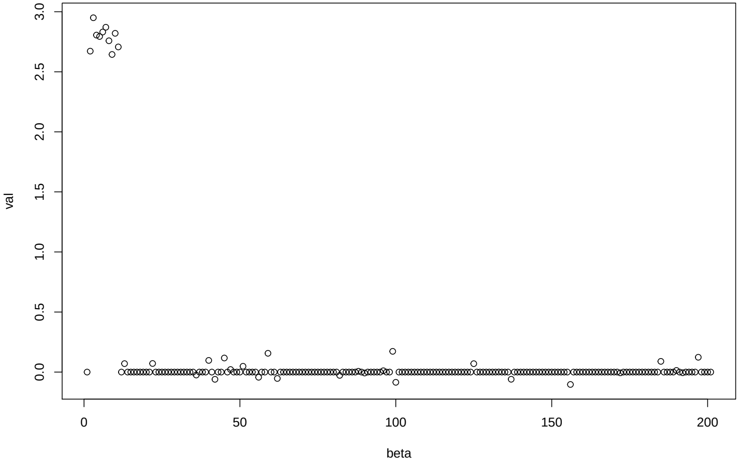
$$\pi(\boldsymbol{\beta}|y, \lambda) \propto \mathcal{N}(y; X\boldsymbol{\beta}, I) \prod_{i=1}^{p} \text{Laplace}\left(0, \frac{1}{\lambda}\right)$$

The Bayesian connection III:lasso

- Be careful with thinking of lasso as Laplace prior.
- In the Bayesian paradigm we are interested in the posterior distribution, and we make inference by generating draws from the posterior distribution.

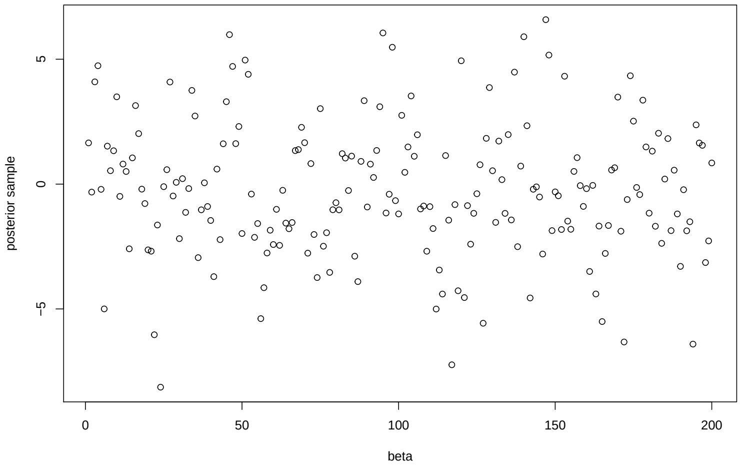
```
set.seed(2)
p <- 200
n <- 100
sigma <- 1
X <- matrix(rnorm(n * p), nrow = n, ncol = p )
beta <- rep(0,p)
beta[1:10] <- 3
y <- X%*%beta + sigma * rnorm(n)
```

lasso cv

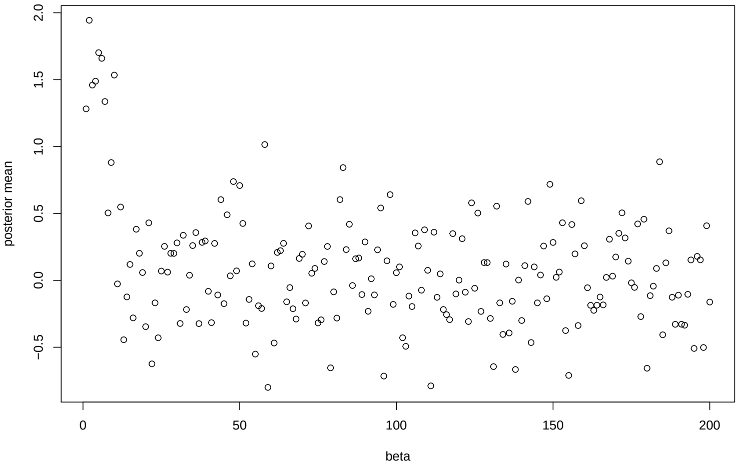


- For modern variable selection methods see knockoffs.

posterior sample



posterior mean



Bayesian alternative

- Heavy tails stronger and shrinkage towards zero[2]
- Mixture distribution [3]
- Non parametric, try to learn the distribution of β with a Dirichlet process[4]

[2] C. Carvalho, Nicholas. Polson, and J. Scott. The horseshoe estimator for sparse signals. Biometrika (2010)

[3] V. Rockova, and G. Edward. The spike-and-slab lasso. JASA 2018

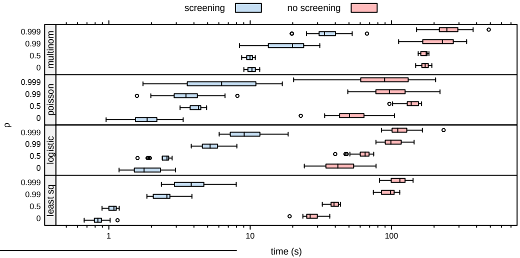
[4] D. Yekutieli, and A. Weinstein. Hierarchical Bayes Modeling for Large-Scale Inference (2021)

Own research

- SLOPE (OWL) Replace the l1-norm with the sorted l1-norm

$$\hat{\boldsymbol{\beta}}^{\text{SLOPE}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \sum_{j=1}^{p} \lambda_j |\beta|_{(j)}$$

- Many advantages, however much much slower to fit to data. When fitting an entire path one can use previous solution when solving the path[5]



[5] J. Larsson, M. Bogdan, and J. Wallin. "The Strong Screening Rule for SLOPE." NeurIPS (2020)