Shrinkage methods

for regression

Motivation

more accurate predictions

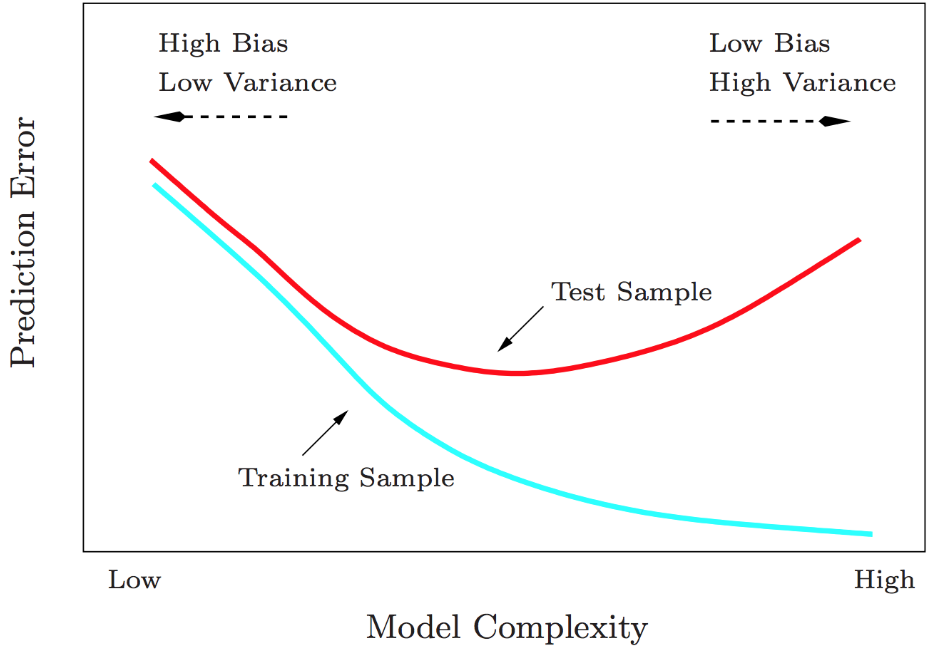


Image credit: Elements of Statistical Learning

Motivation

sift through many candidate predictors

enable inference when $p \gg n$, e.g:

- Feature engineering
- Genome Wide Association Studies (GWAS)

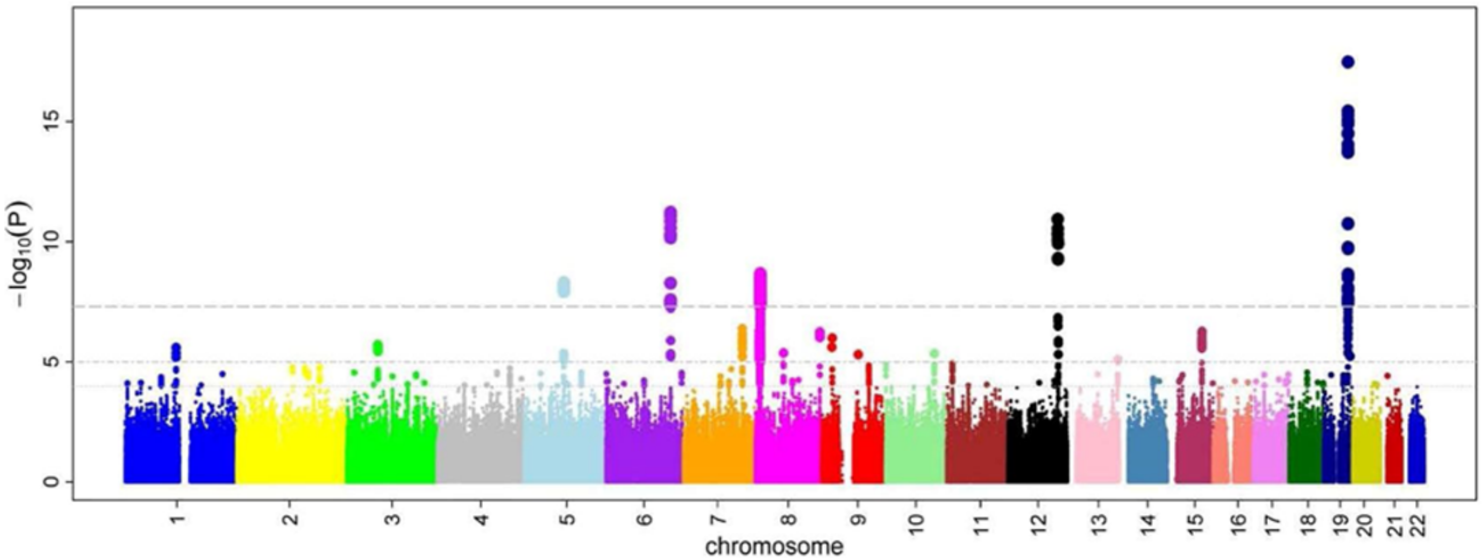


Image credit: Wikimedia Commons

The idea

we want a model that fits the data, but we also don't want coefficients to be too big

we don't care about obtaining an unbiased estimator of the coefficients

shrink coefficients toward zero by adding a penalty on their size

The idea

linear regression:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\{\mathrm{RSS}(\beta)\}$$

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\hat{\mathbf{y}} = \alpha + \mathbf{X}\beta$$

penalized regression:

$$\hat{\beta}^{\text{penalized}} = \underset{\beta}{\operatorname{argmin}}\{\mathrm{RSS}(\beta) + f(\beta)\}$$

Two penalties:

Ridge regression

Hoerl & Kennard (1970) - link

The lasso

Tibshirani (1996) - link

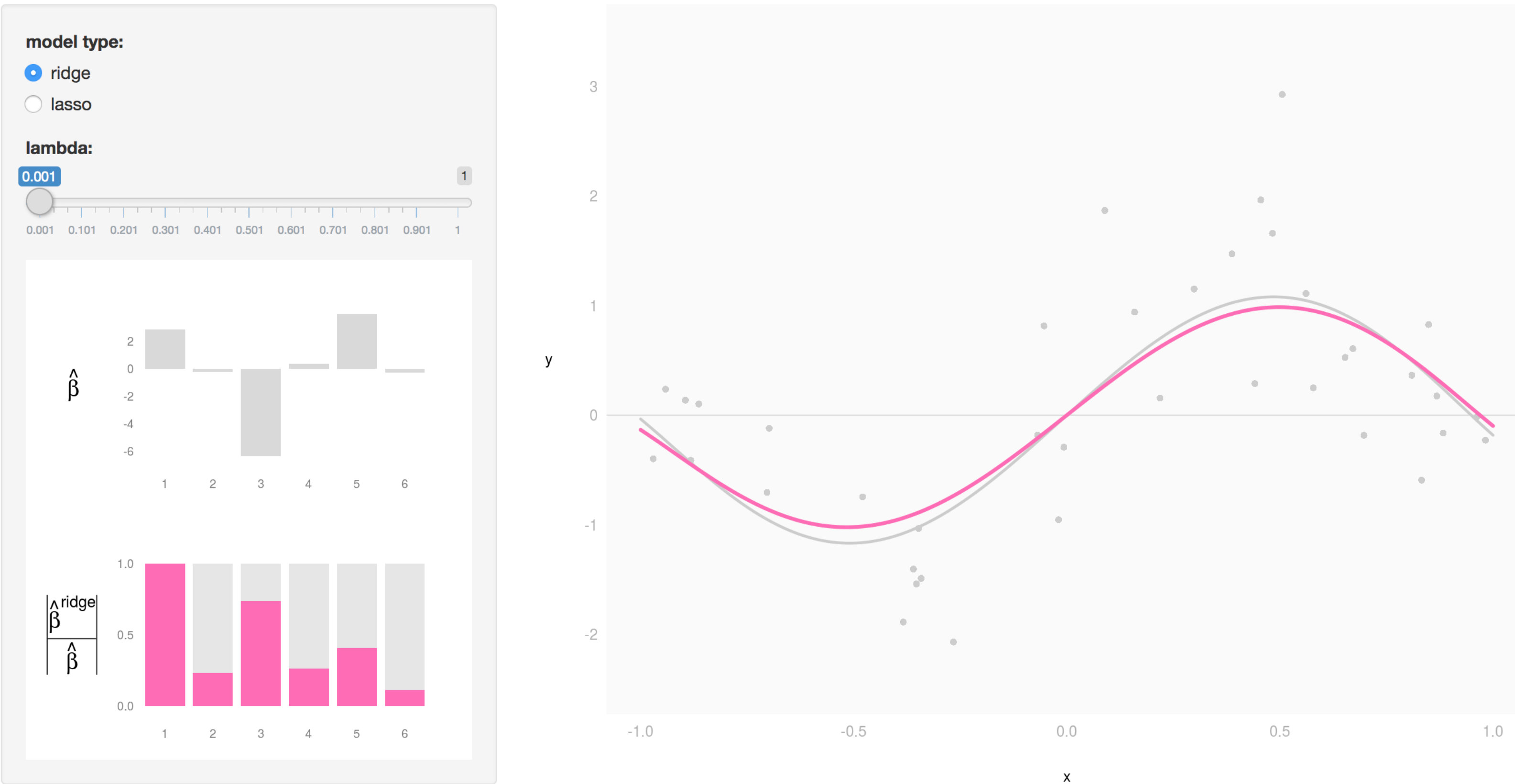
Ridge regression

penalise *sum of beta squared* (the L^2-norm)

$$f(\beta) = \lambda \sum_{i=1}^{p} \beta^2$$

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}}\{\text{RSS}(\beta) + \lambda \sum_{i=1}^{p} \beta^2\}$$

Ridge regression demo

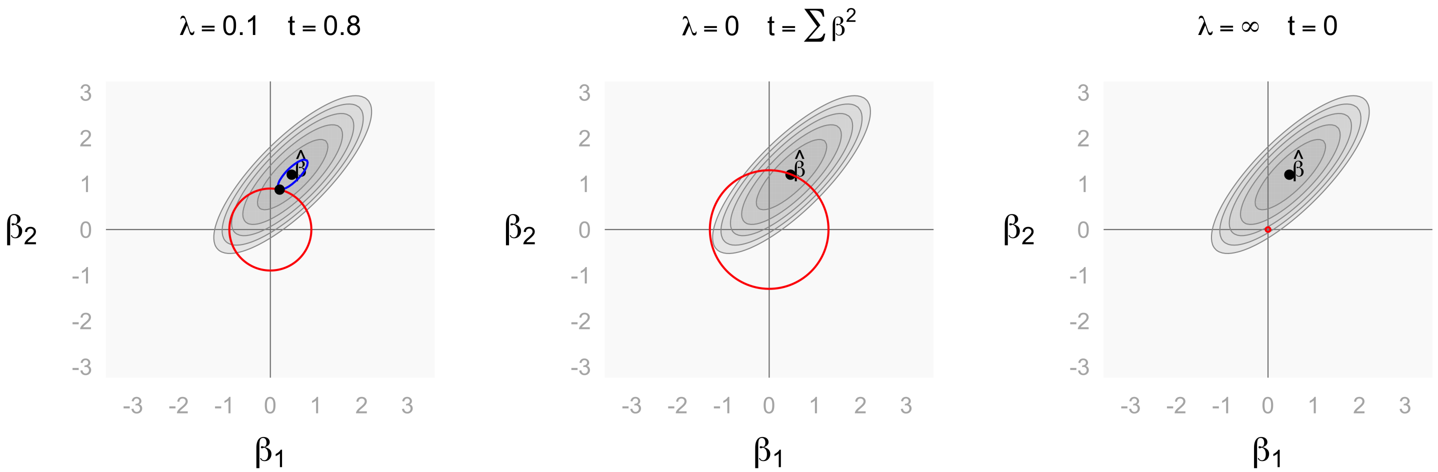


goldingn.shinyapps.io/shrinkage_demo

Ridge regression

as constrained optimisation

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}}\{\text{RSS}(\beta)\} \quad \text{s. t.} \sum_{i=1}^{p} \beta^2 < t$$



The lasso

penalise *sum of modulus of beta* (the L^1-norm)

$$f(\beta) = \lambda \sum_{i=1}^{p} |\beta|$$

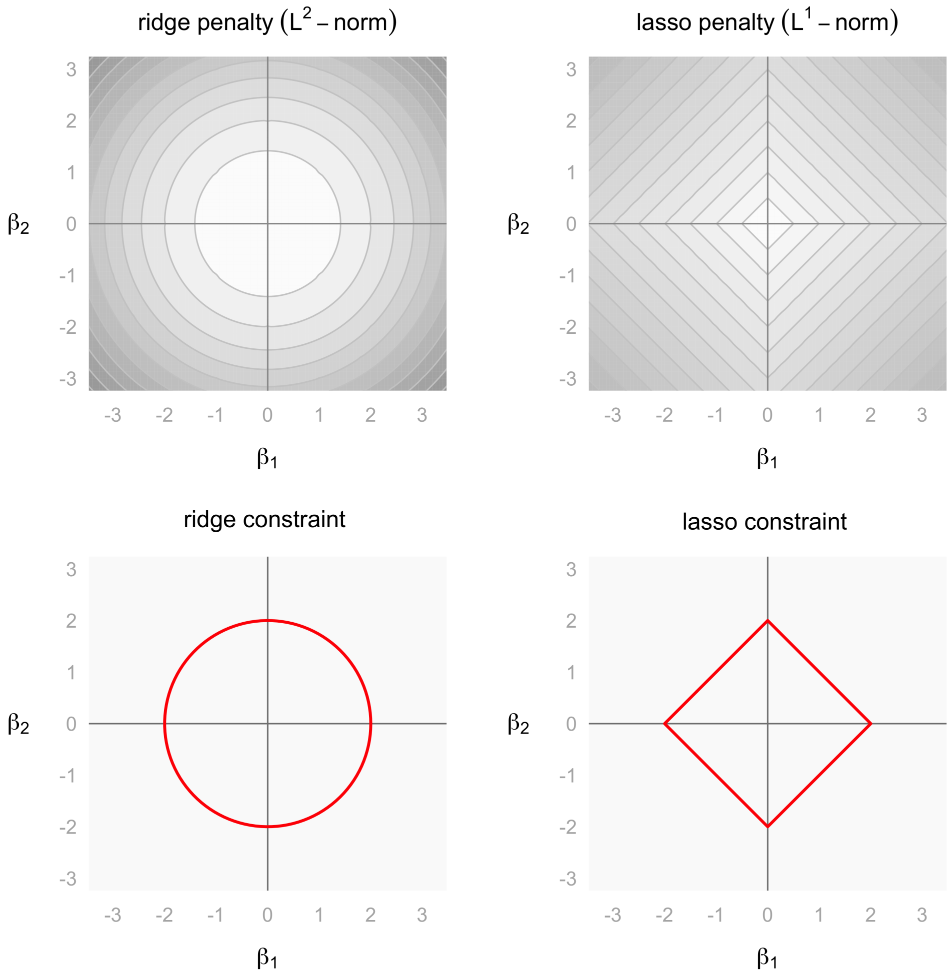
so

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \{\mathrm{RSS}(\beta) + \lambda \sum_{i=1}^{p} |\beta|\}$$

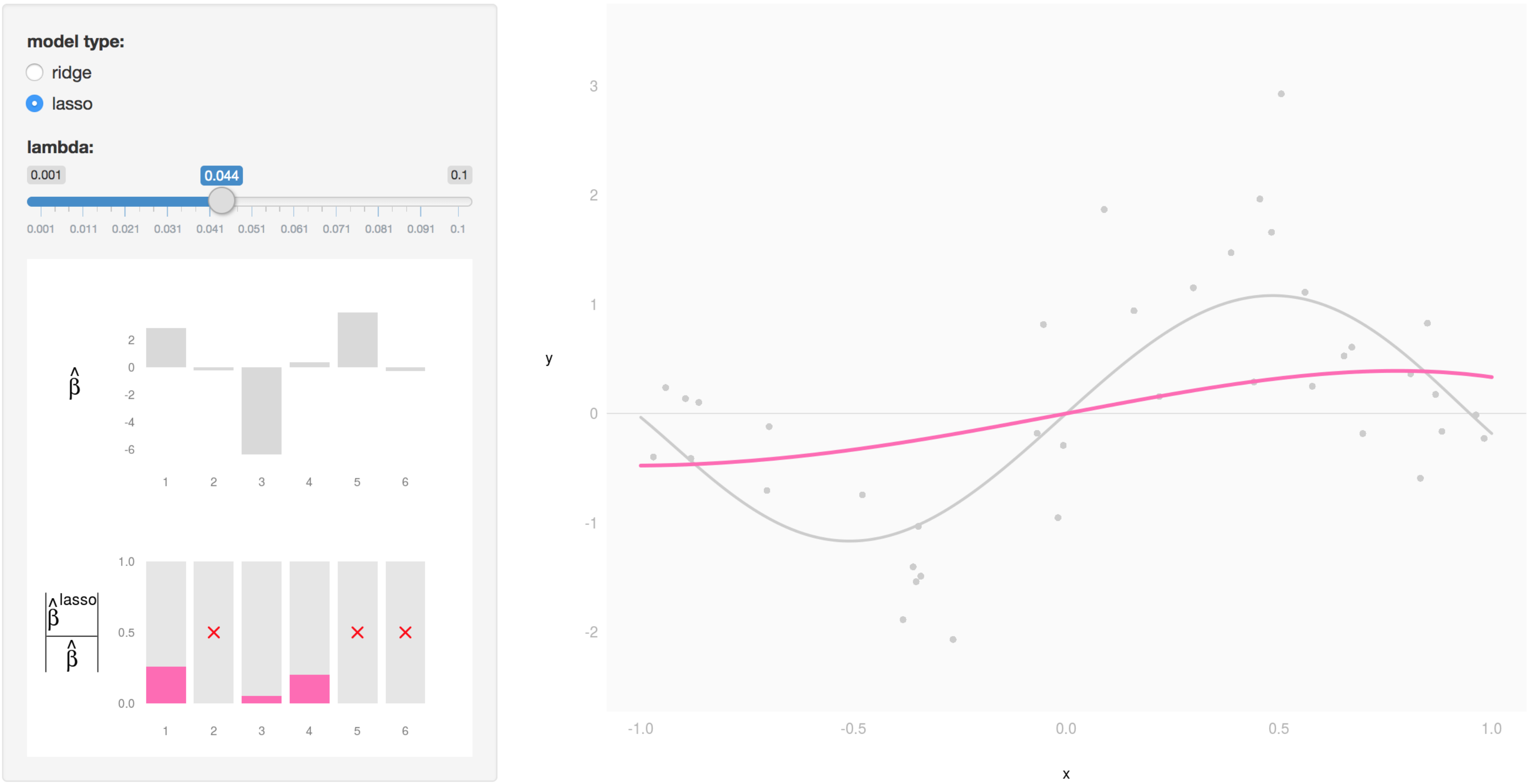
or equivalently

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \{\mathrm{RSS}(\beta)\} \quad \text{s. t.} \ \sum_{i=1}^{p} |\beta| < t$$

ridge vs. lasso

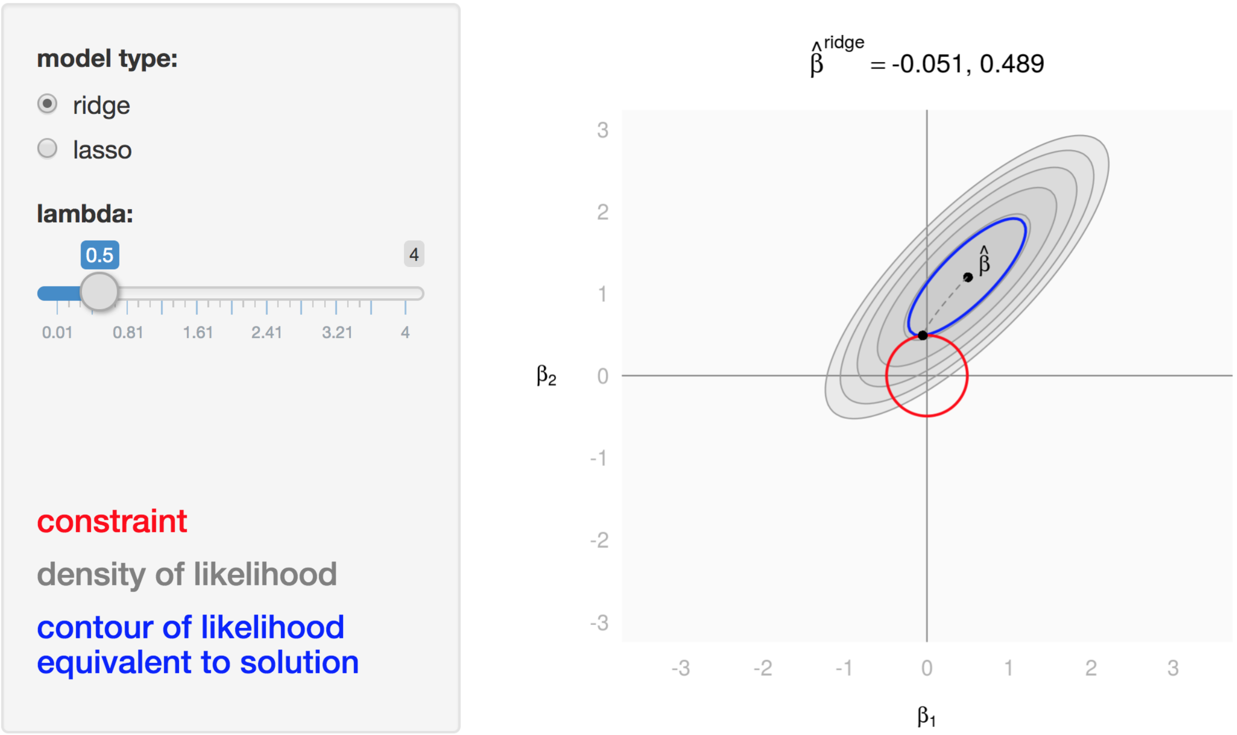


Lasso shrinks coefficients to zero!



goldingn.shinyapps.io/shrinkage_demo

Why does lasso shrink to zero?



goldingn.shinyapps.io/constraint_app

Estimation

linear regression:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

ridge regression:

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

lasso has no closed-form solution, so we optimize numerically

practical issues

ridge and lasso estimates are influenced by scale of covariates, so we usually standardize covariates first

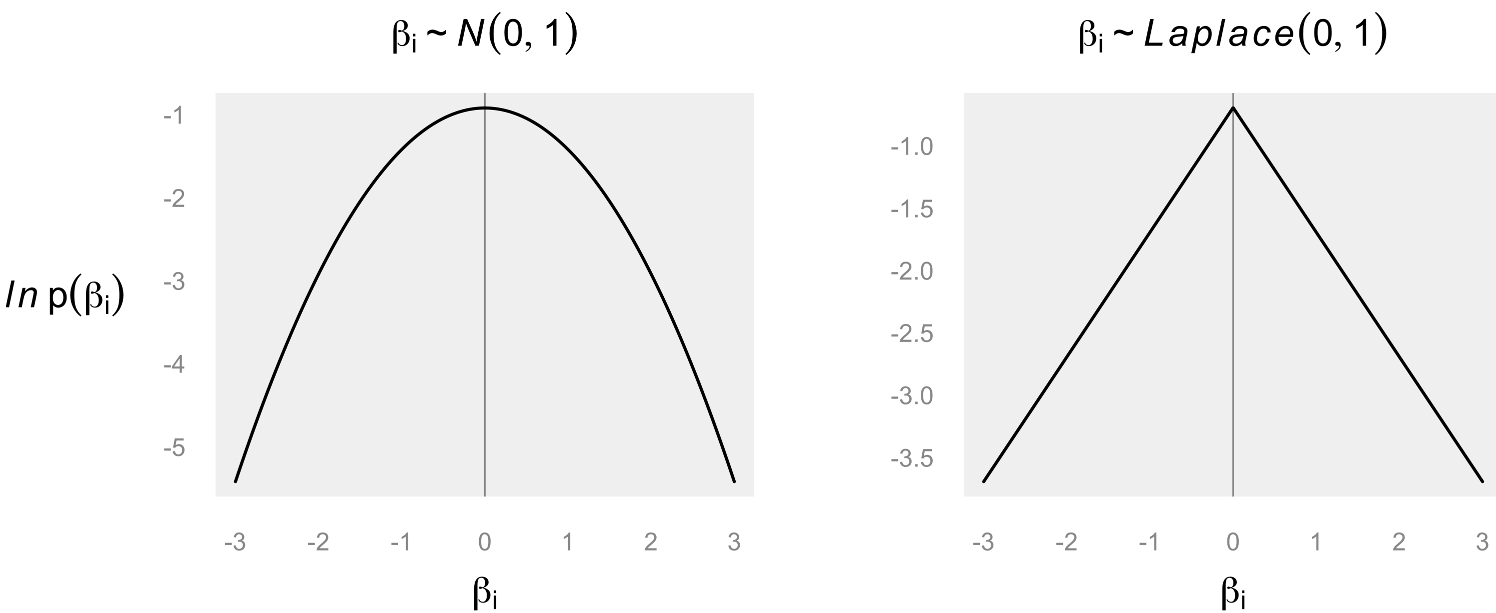
select lambda by cross-validation

```r
library(glmnet)

# lasso
cv.glmnet(x, y, alpha = 0, nfolds = 5)

# ridge
cv.glmnet(x, y, alpha = 1, nfolds = 5)
```

Bayesian equivalence

$$p(\beta|\mathbf{X}, y) \propto p(y|\mathbf{X}\beta)p(\beta)$$

$$\hat{\beta}_{MAP}(\mathbf{X}, y) = \underset{\beta}{argmin}\{RSS(\beta) + -ln\, p(\beta)\}$$



Other shrinkage methods

Least Angle Regression

closely related to lasso

Elastic net

a mixture of ridge and lasso penalties

$f(\beta) = \lambda \sum_{i=1}^{p} a\beta_i^2 + (1-a)|\beta_i|$

materials

slides, code, interactives

`github.com/goldingn/shrinkage_lecture`

glmnet R package

including introductory vignette