High-dimensional data analysis using penalized regression methods

Sahir Rai Bhatnagar

Department of Epidemiology, Biostatistics, and Occupational Health
Department of Diagnostic Radiology

https://sahirbhatnagar.com/

McGill Summer School in Health Data Analytics
May 8, 2019

Outline

- Classical statistical models
- Bet on sparsity
- A motivating example
- Background on the lasso and group lasso
- Generalizations

Classical Methods

Setting

- This lecture concerns the analysis of data in which we are attempting to predict an outcome Y using a number of explanatory factors $X_1, X_2, X_3, \ldots$, some of which may not be particularly useful

Setting

- This lecture concerns the analysis of data in which we are attempting to predict an outcome Y using a number of explanatory factors $X_1, X_2, X_3, \ldots$, some of which may not be particularly useful
- Although the methods we will discuss can be used solely for prediction (i.e., as a "black box"), I will adopt the perspective that we would like the statistical methods to be interpretable and to explain something about the relationship between the X and Y

Setting

- This lecture concerns the analysis of data in which we are attempting to predict an outcome Y using a number of explanatory factors $X_1, X_2, X_3, \ldots$, some of which may not be particularly useful
- Although the methods we will discuss can be used solely for prediction (i.e., as a "black box"), I will adopt the perspective that we would like the statistical methods to be interpretable and to explain something about the relationship between the X and Y
- Regression models are an attractive framework for approaching problems of this type, and the focus today will be on extending classical regression modeling to deal with high-dimensional data

Classical Methods

- A nice and powerful toolbox for analyzing the more traditional datasets where the sample size (N) is far **greater than** the number of covariates (p):
  - linear regression, logistic regression, LDA, QDA, glm,
  - regression spline, smoothing spline, kernel smoothing, local smoothing, GAM,
  - Neural Network, SVM, Boosting, Random Forest, ...

Classical Methods

- A nice and powerful toolbox for analyzing the more traditional datasets where the sample size (N) is far **greater than** the number of covariates (p):
  - linear regression, logistic regression, LDA, QDA, glm,
  - regression spline, smoothing spline, kernel smoothing, local smoothing, GAM,
  - Neural Network, SVM, Boosting, Random Forest, ...

$$\mathbf{X}_{n\times p} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{12} & \cdots & X_{1p} \\ X_{31} & X_{12} & \cdots & X_{1p} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{12} & \cdots & X_{np} \end{bmatrix}$$

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa

Classical Linear Regression

Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ iid from

$$y = \mathbf{x}^T \boldsymbol{\beta} + \epsilon$$

where $E(\epsilon|\mathbf{x}) = 0$, and $\dim(x) = p$. To include an intercept, we can set $\mathbf{x}_1 \equiv 1$. Using Matrix notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

The least squares estimator

$$\widehat{\boldsymbol{\beta}}_{LS} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

$$\widehat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Classical Linear Regression

Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ iid from

$$y = \mathbf{x}^T \boldsymbol{\beta} + \epsilon$$

where $E(\epsilon|\mathbf{x}) = 0$, and $\dim(x) = p$. To include an intercept, we can set $\mathbf{x}_1 \equiv 1$. Using Matrix notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

The least squares estimator

$$\widehat{\boldsymbol{\beta}}_{LS} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

$$\widehat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- **Question:** How to find the important variables $\mathbf{x}_j$?

Best-subset Selection (Beal et al. 1967, Biometrika)

Predictor set	model
None of $x_1\ x_2\ x_3\ x_4$	$E(Y) = \beta_0$
x_1	$E(Y) = \beta_0 + \beta_1 x_1$
x_2	$E(Y) = \beta_0 + \beta_2 x_2$
x_3	$E(Y) = \beta_0 + \beta_3 x_3$
x_4	$E(Y) = \beta_0 + \beta_4 x_4$
$x_1\ x_2$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
$x_1\ x_3$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_3 x_3$
$x_1\ x_4$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_4 x_4$
$x_2\ x_3$	$E(Y) = \beta_0 + \beta_2 x_2 + \beta_3 x_3$
$x_2\ x_4$	$E(Y) = \beta_0 + \beta_2 x_2 + \beta_4 x_4$
$x_3\ x_4$	$E(Y) = \beta_0 + \beta_3 x_3 + \beta_4 x_4$
$x_1\ x_2\ x_3$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
$x_1\ x_2\ x_4$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_4$
$x_1\ x_3\ x_4$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \beta_4 x_4$
$x_2\ x_3\ x_4$	$E(Y) = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$
$x_1\ x_2\ x_3\ x_4$	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$

Which variables are important?

- Scientists know only a small subset of variables (such as genes) are important for the response variable.
- An old Idea: try all possible subset models and pick the best one.
- Fit a subset of predictors to the linear regression model. Let S be the subset predictors, e.g., S = {1, 3, 7}.

$$C_p = \frac{\mathrm{RSS}_S}{\sigma^2} - (n - 2|S|) = \frac{\mathrm{RSS}_S}{\sigma^2} + 2|S| - n$$

- We pick the model with the smallest C_p value.

Model selection criteria

Minimizing C_p is equivalent to minimizing

$$\|\mathbf{y} - \mathbf{X}_S\widehat{\boldsymbol{\beta}}_S\|^2 + 2|S|\sigma^2.$$

which is AIC score.
Many popular model selection criteria can be written as

$$\|\mathbf{y} - \mathbf{X}_S\widehat{\boldsymbol{\beta}}_S\|^2 + \lambda|S|\sigma^2.$$

- BIC uses $\lambda = \sigma\sqrt{\log(n)/n}$.

Remarks

Best subset selection plus model selection criteria (AIC, BIC, etc.)

- Computing all possible subset models is a combinatorial optimization problem (NP hard)
- Instability in the selection process (Breiman, 1996)

Ridge Regression (Hoerl & Kennard 1970, Technometrics)

- $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_2^2$
- $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$

Ridge Regression
(Hoerl & Kennard 1970, Technometrics)

- $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_2^2$
- $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$
- $\widehat{\boldsymbol{\beta}}_{Ridge} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \rightarrow$ exact solution
- $\widehat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$

Ridge Regression (Hoerl & Kennard 1970, Technometrics)

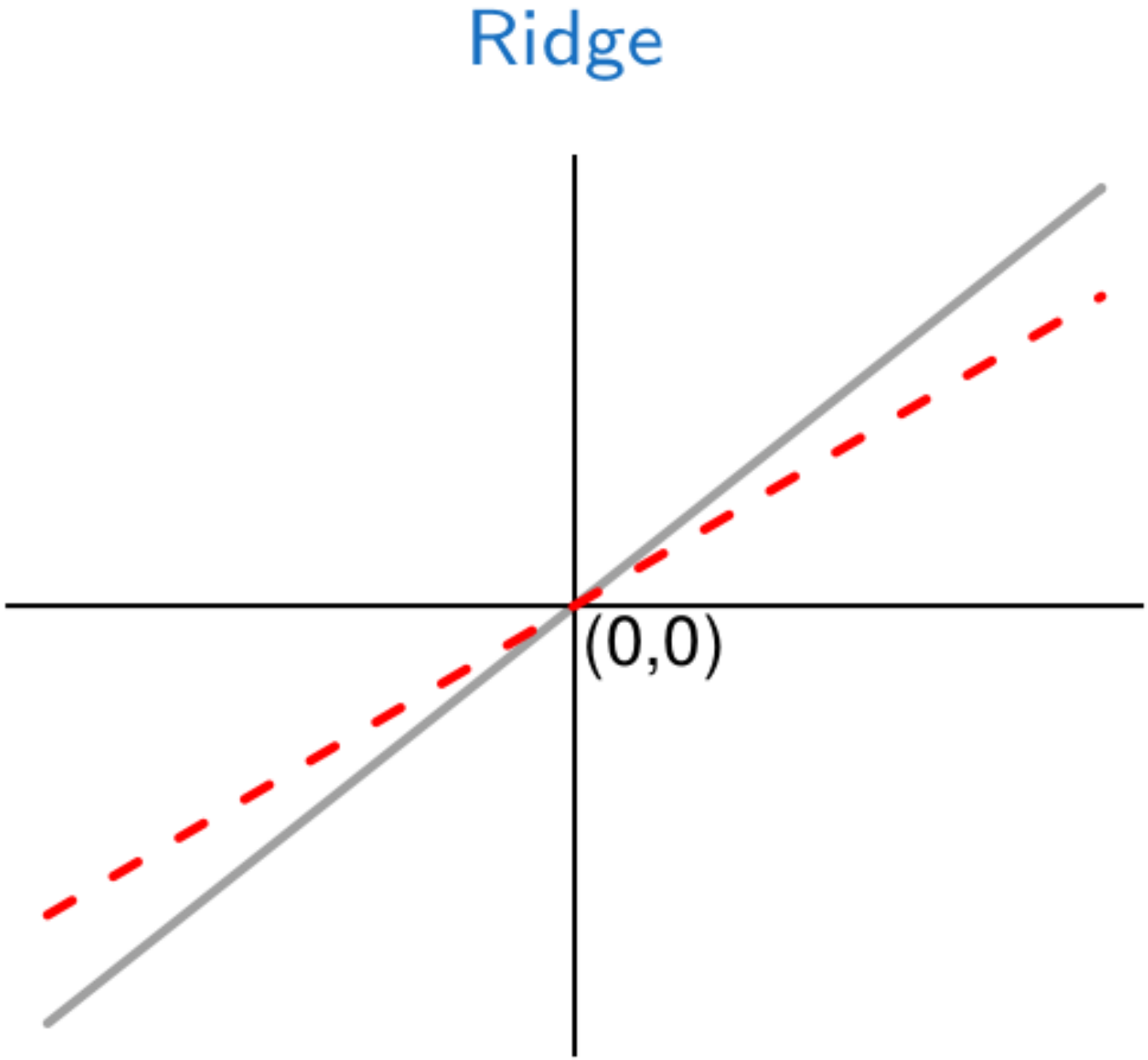
- $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_2^2$
- $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$
- $\widehat{\boldsymbol{\beta}}_{Ridge} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \rightarrow$ exact solution
- $\widehat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$
- Let $\mathbf{X}^\top\mathbf{X} = \mathbf{I}_{p\times p}$

Ridge Regression (Hoerl & Kennard 1970, Technometrics)

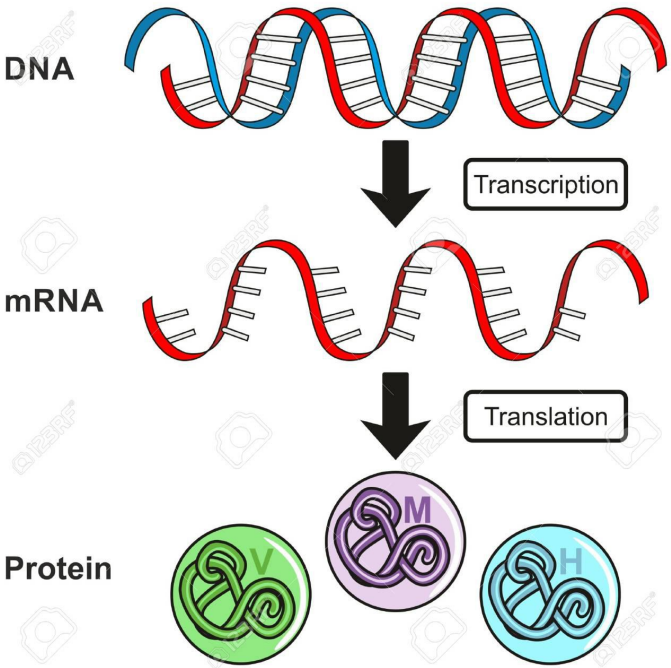
- $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_2^2$
- $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$
- $\widehat{\boldsymbol{\beta}}_{Ridge} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \rightarrow$ exact solution
- $\widehat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$
- Let $\mathbf{X}^\top\mathbf{X} = \mathbf{I}_{p\times p}$

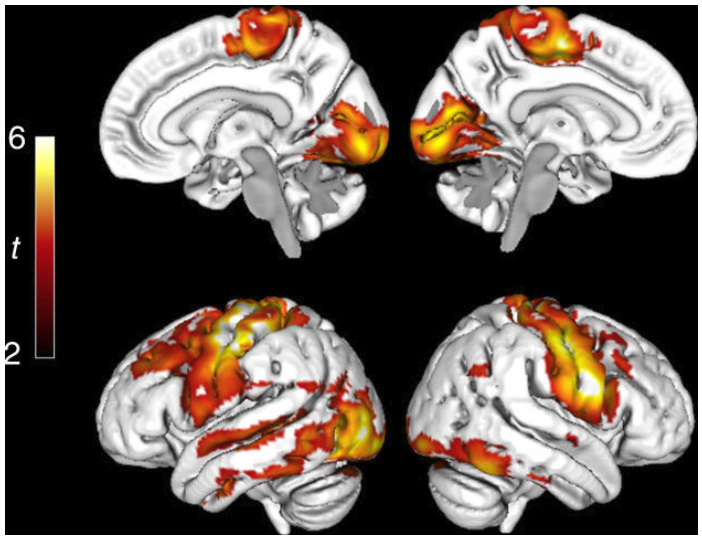
$$\hat{\beta}_{j\,(Ridge)} = \frac{\hat{\beta}_{j\,(MCO)}}{1+\lambda}$$

Least squares vs. Ridge



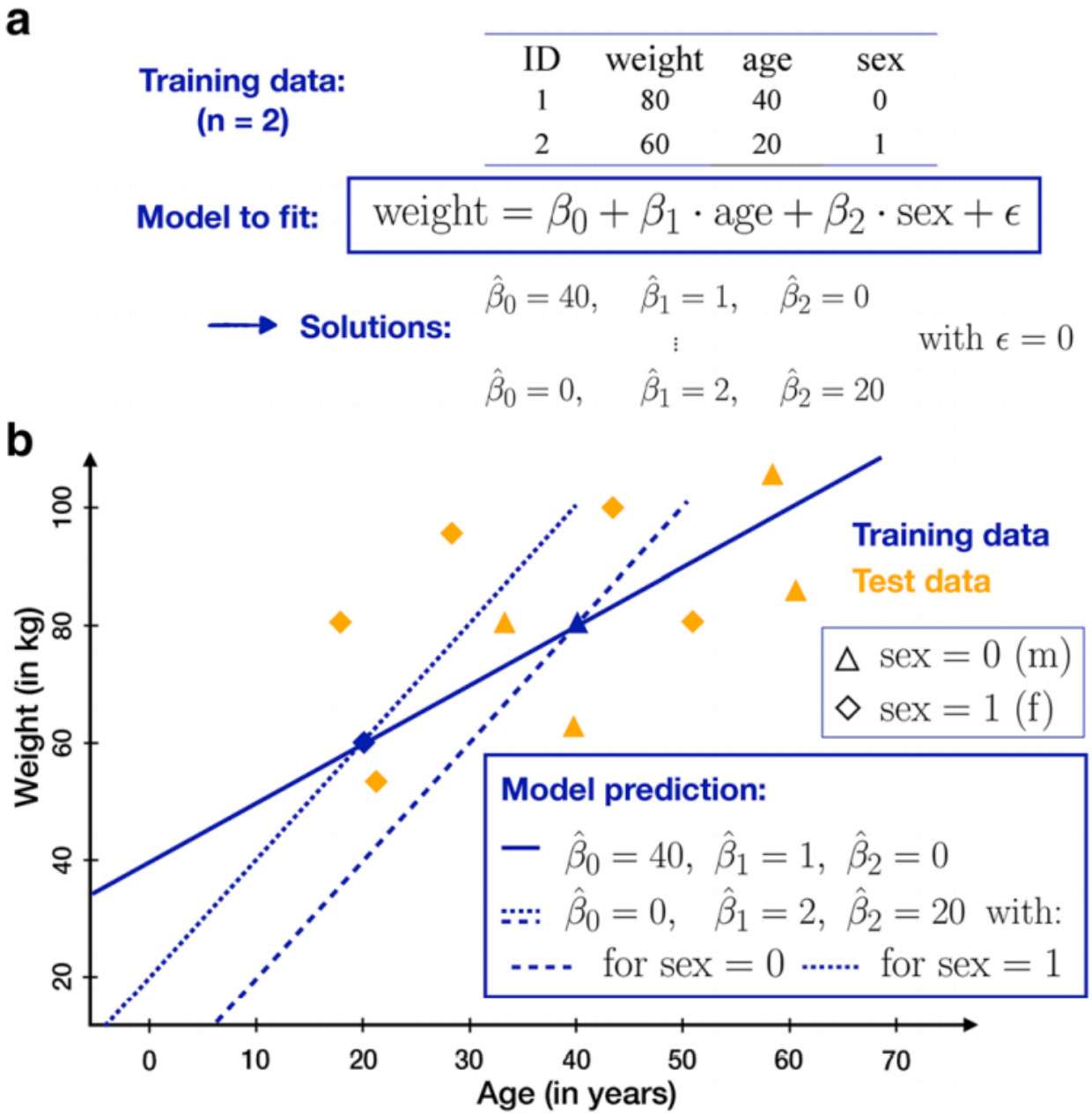
High-dimensional data ($n << p$)





$$\mathbf{X}_{n\times p} = \begin{bmatrix} X_{11} & X_{12} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & X_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{12} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & X_{np} \end{bmatrix}$$

Why can't we fit OLS to High-dimensional data?



Boulesteix et al., Human Genetics, 2019

High-dimensional data ($n << p$)

- Throughout the course, we will let
  - n denote the number of independent sampling units (e.g., number of patients)
  - p denote the number of features recorded for each unit

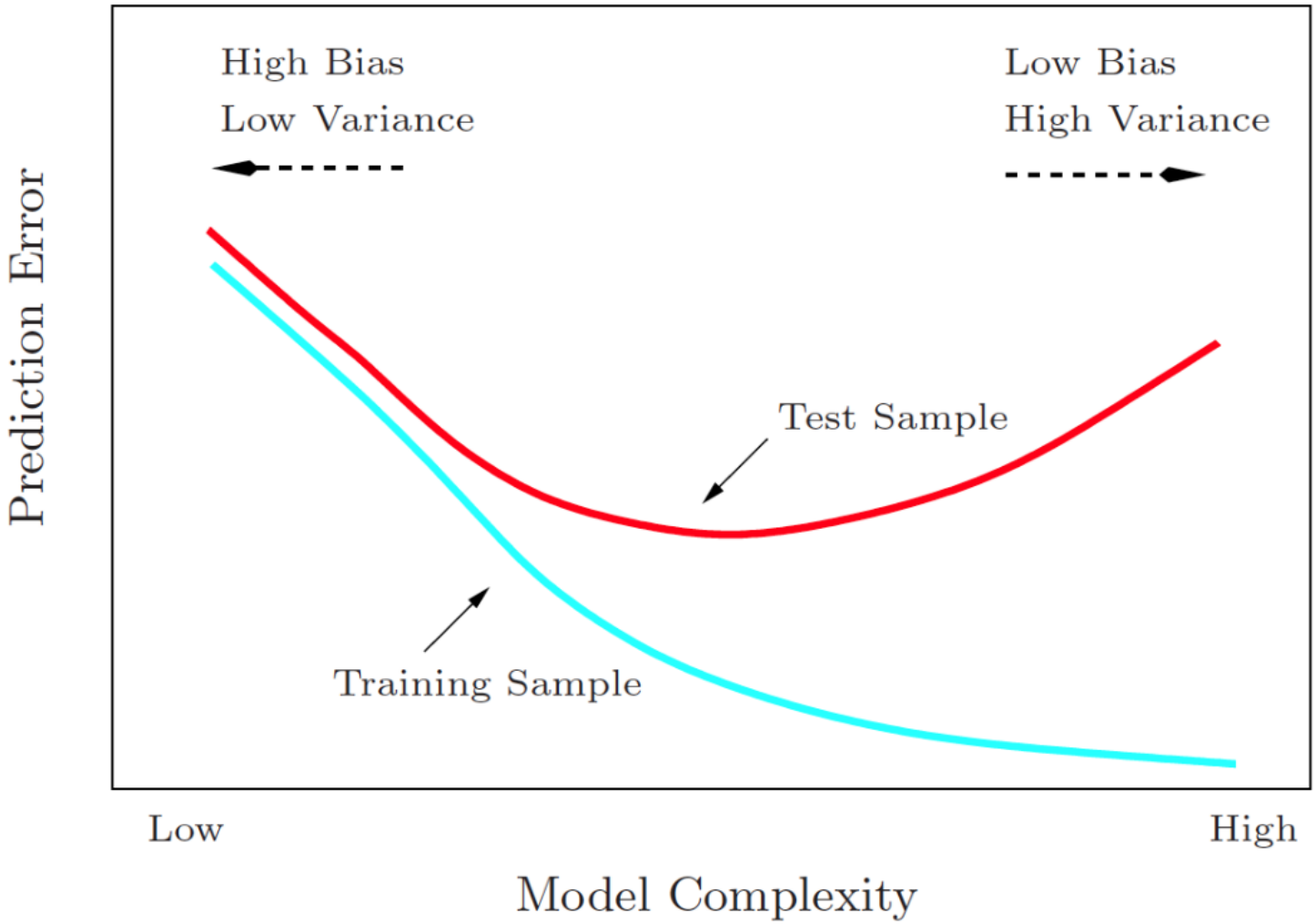
High-dimensional data ($n << p$)

- Throughout the course, we will let
  - n denote the number of independent sampling units (e.g., number of patients)
  - p denote the number of features recorded for each unit
- In high-dimensional data, p is large with respect to n
  - This certainly includes the case where $p > n$

High-dimensional data ($n << p$)

- Throughout the course, we will let
  - n denote the number of independent sampling units (e.g., number of patients)
  - p denote the number of features recorded for each unit
- In high-dimensional data, p is large with respect to n
  - This certainly includes the case where $p > n$
  - However, the ideas we discuss in this course are also relevant to many situations in which $p < n$; for example, if $n = 100$ and $p = 80$, we probably don't want to use ordinary least squares

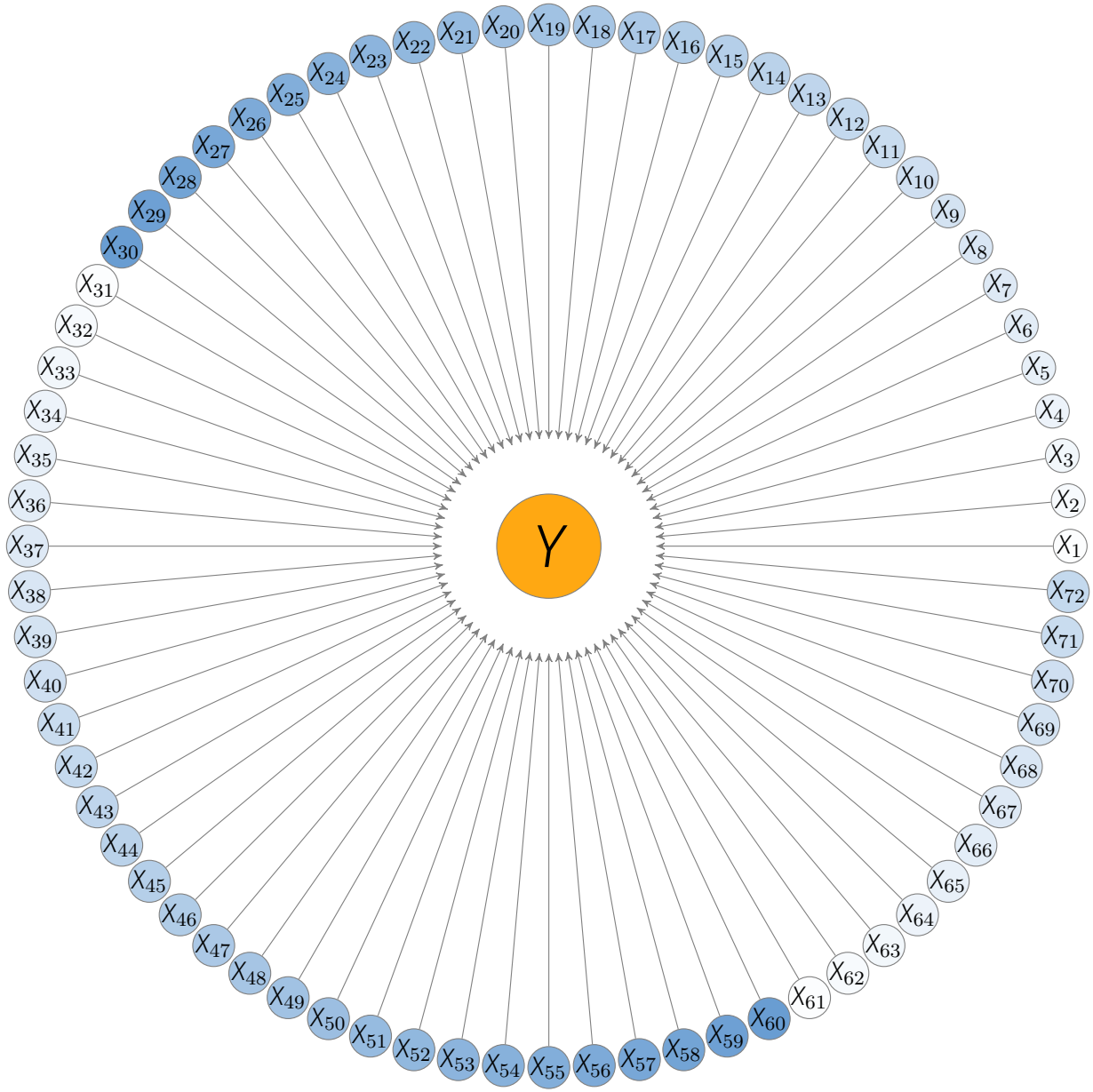
A fundamental picture for data science



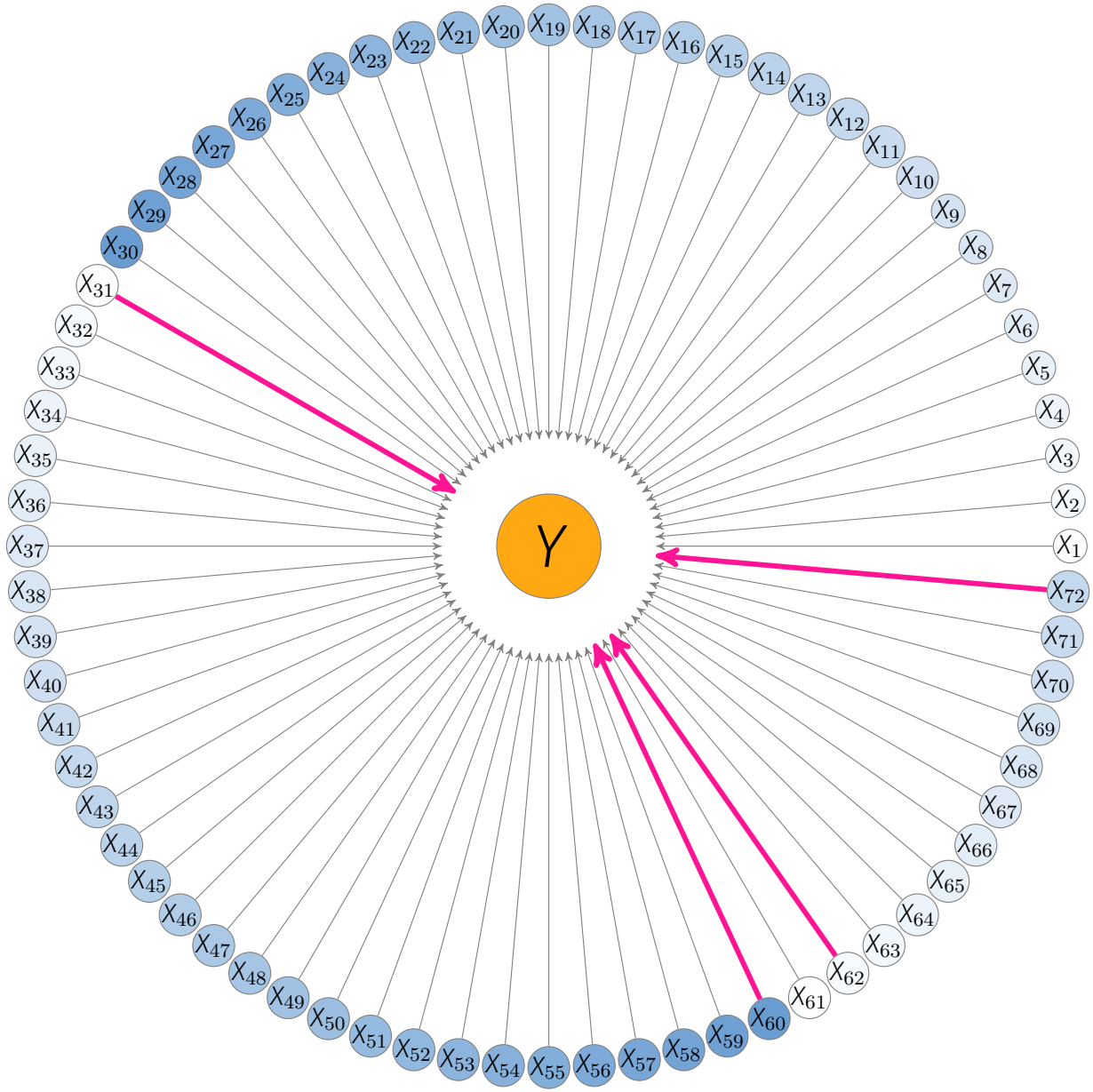
ESL, Hastie et al. 2009

Betting on Sparsity

Bet on Sparsity Principle



Bet on Sparsity Principle



Bet on Sparsity Principle

Use a procedure that does well in sparse problems, since no procedure does well in dense problems.[1]

[1] The elements of statistical learning. Springer series in statistics, 2001.

Bet on Sparsity Principle

Use a procedure that does well in sparse problems, since no procedure does well in dense problems.[1]

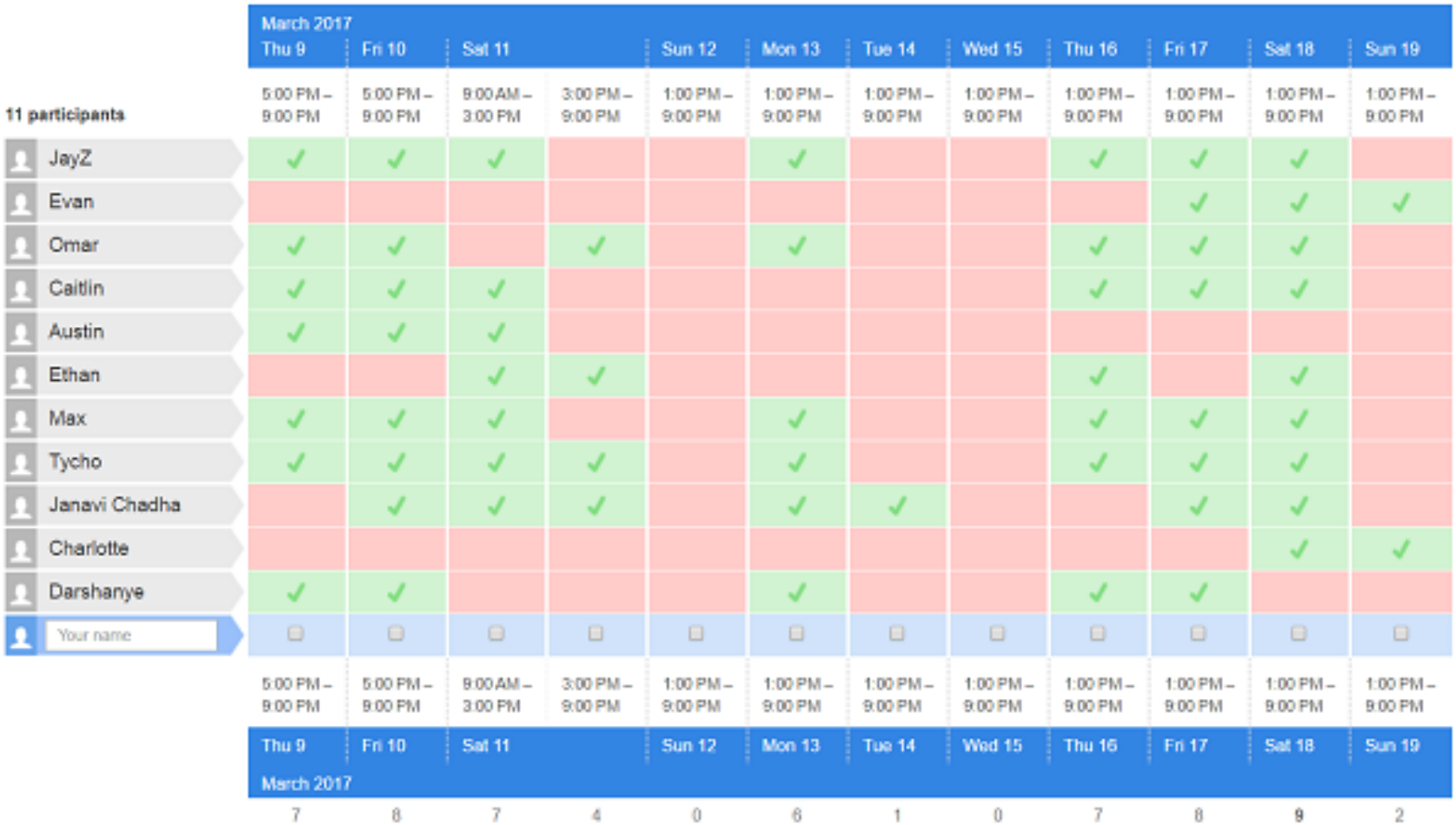
- We often don't have enough data to estimate so many parameters
- Even when we do, we might want to identify a relatively small number of predictors ($k < N$) that play an important role
- Faster computation, easier to understand, and stable predictions on new datasets.

[1] The elements of statistical learning. Springer series in statistics, 2001.

A Thought Experiment

How would you schedule a meeting of 20 people?

How would you schedule a meeting of 20 people?



Doctors Bet on Sparsity Also

Doctors Bet on Sparsity Also

Motivating Example

Predictors of NHL Salary[2]

[2] https://www.kaggle.com/camnugent/nhl-salary-data-prediction-cleaning-and-modelling

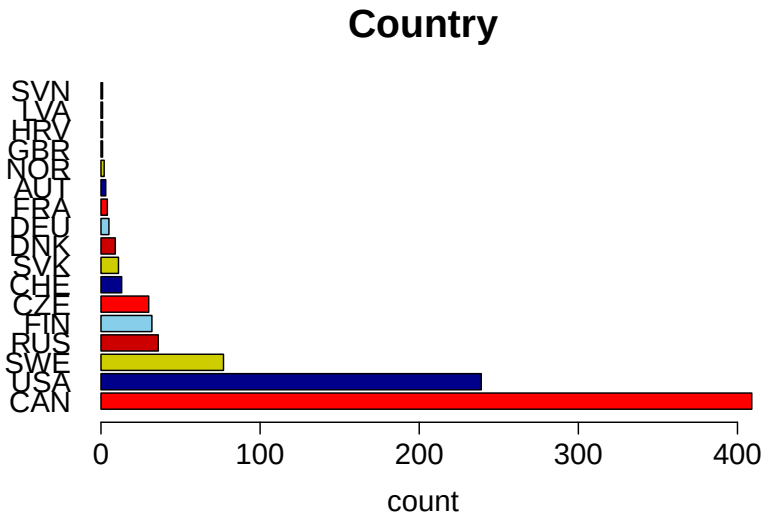
Supervised Learning

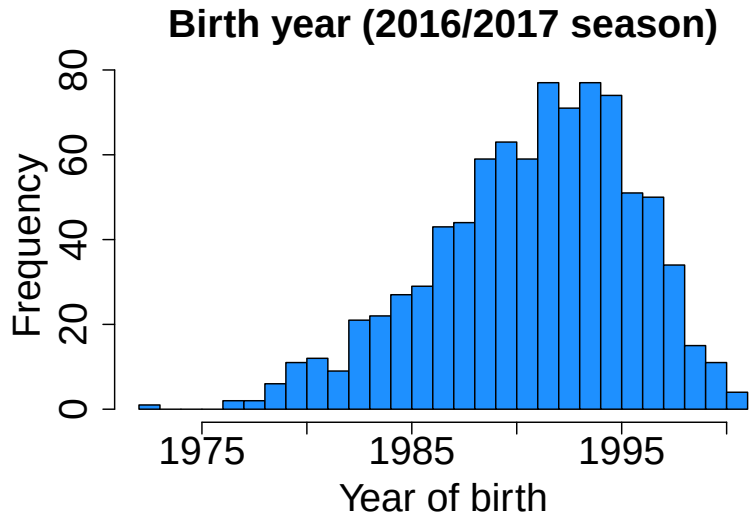
- Learn the function *f*

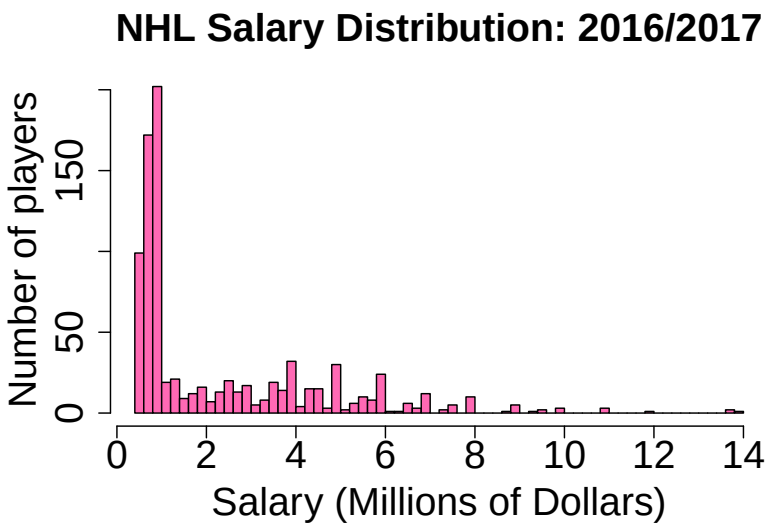
f

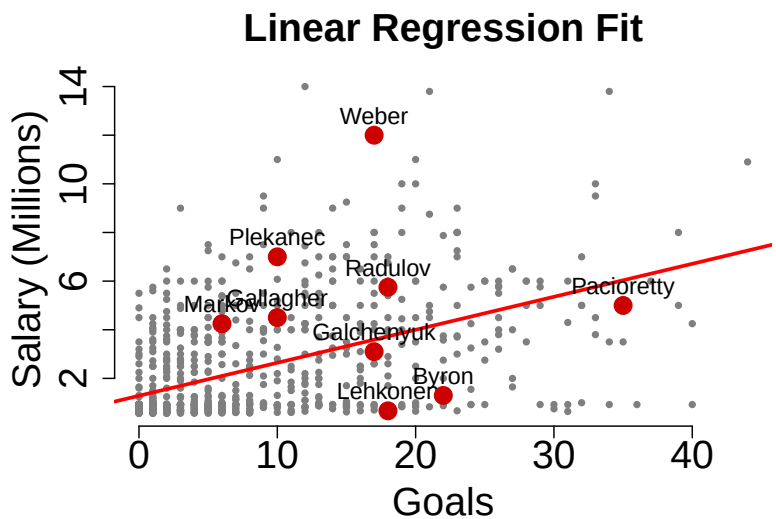
=

Predictors of NHL Salary

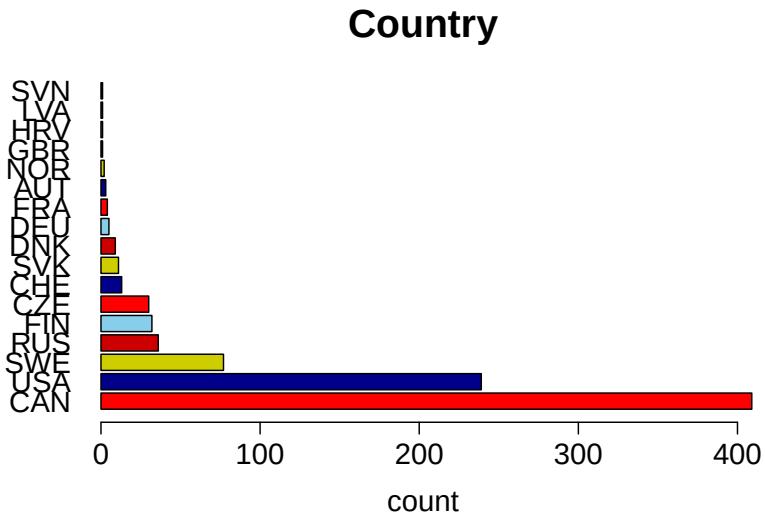


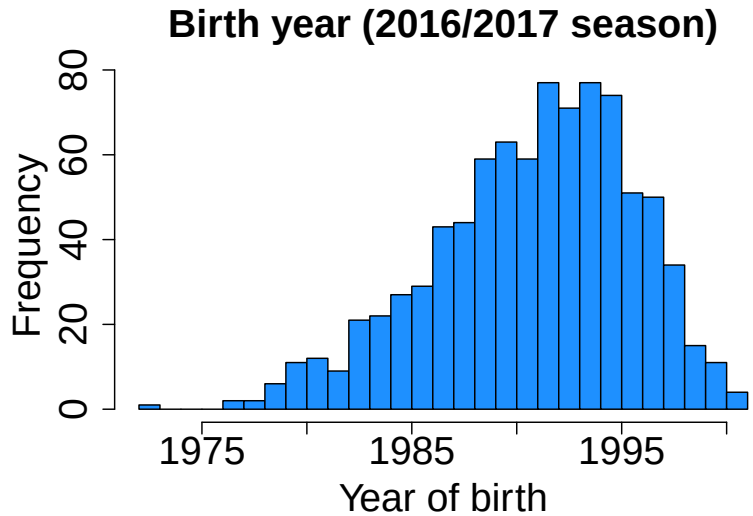


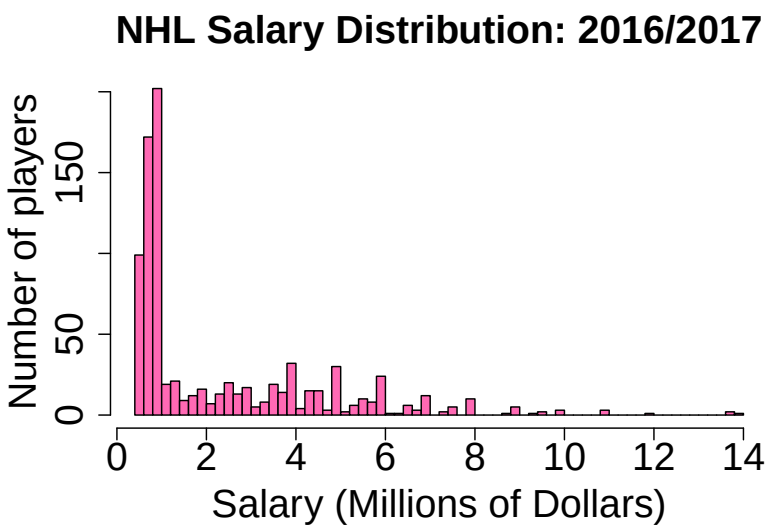


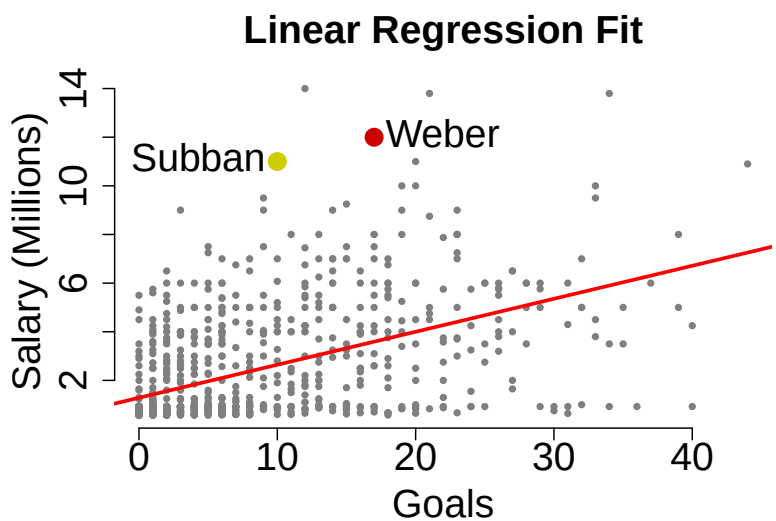


Predictors of NHL Salary

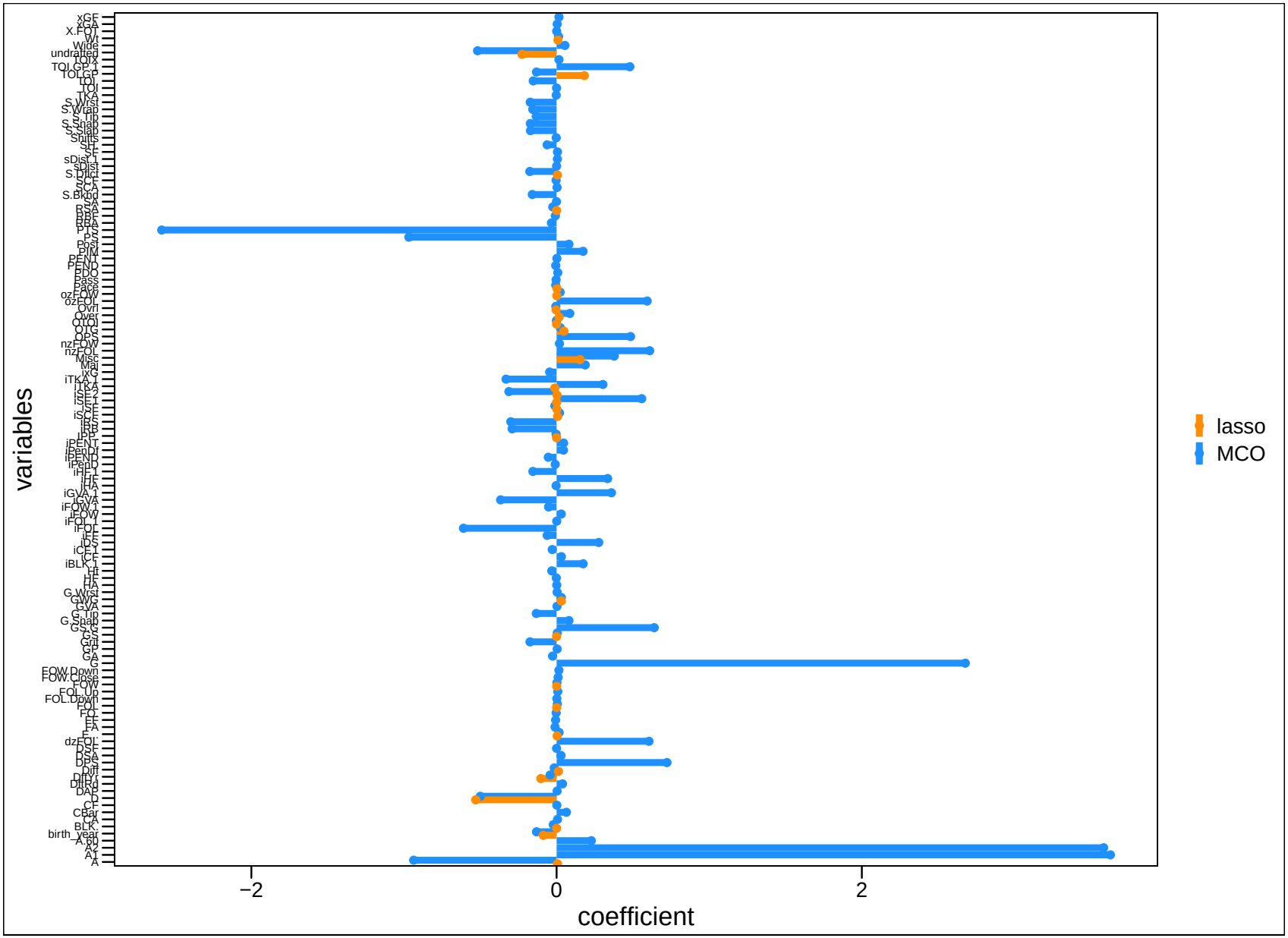




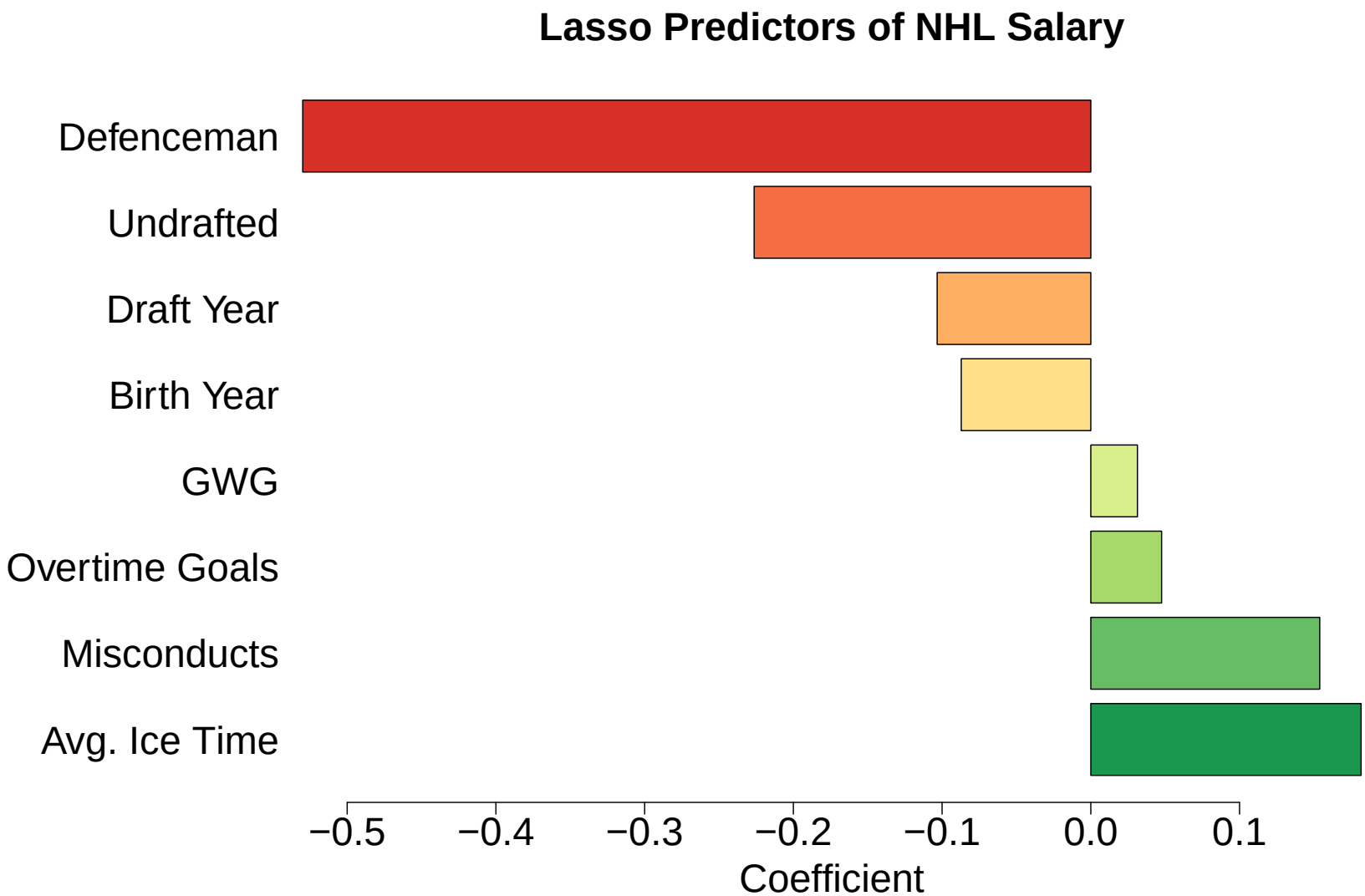




OLS vs. Lasso Coefficients



Lasso Selected Predictors



Background on the lasso (Tibshirani. *JRSSB*, 1996)

Bridge regression (Frank and Friedman, 1993)

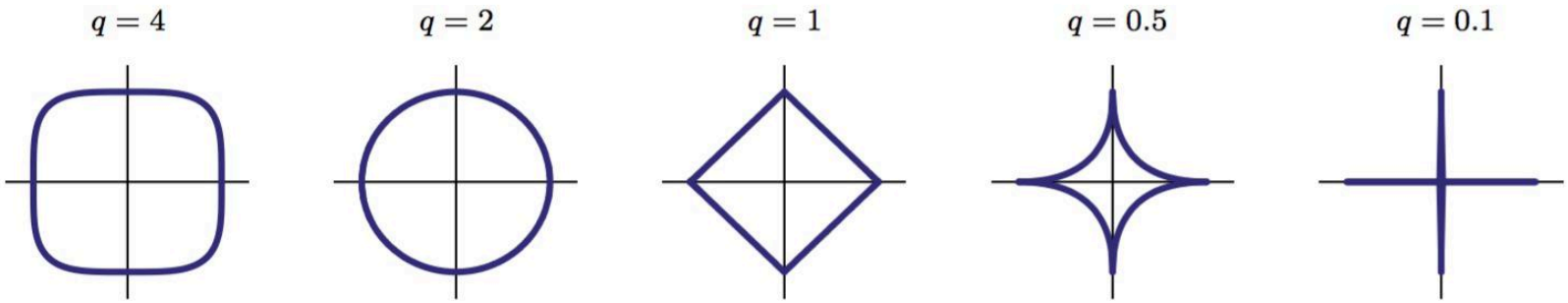
$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_q \qquad 0 \leq q \leq 2.$$

Its constrained formulation

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

$$\text{subject to } \|\boldsymbol{\beta}\|_q = \sum_{j=1}^{p} |\beta_j|^q \leq s$$

Bridge regression (Frank and Friedman, 1993)



Contours of equal value for the L_q penalty for difference values of q. For $q < 1$, the constraint region is **nonconvex**.

- $q = 0$, $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^{p} |\beta_j|^0 = \sum_{j=1}^{p} I(\beta_j \neq 0)$
- $q = 1$, $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ convex

Background on the Lasso

- Predictors x_{ij}, $j = 1, \ldots, p$ and outcome values y_i for the ith observation, $i = 1, \ldots, n$
- Assume x_{ij} are standardized so that $\sum_i x_{ij}/n = 0$ and $\sum_i x_{ij}^2 = 1$.

[1] Tibshirani. JRSSB (1996)

Background on the Lasso

- Predictors x_{ij}, $j = 1, \ldots, p$ and outcome values y_i for the ith observation, $i = 1, \ldots, n$
- Assume x_{ij} are standardized so that $\sum_i x_{ij}/n = 0$ and $\sum_i x_{ij}^2 = 1$. The lasso[1] solves

$$\widehat{\beta}^{lasso} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq s, \qquad s > 0$$

[1] Tibshirani. JRSSB (1996)

Background on the Lasso

- Predictors x_{ij}, $j = 1, \ldots, p$ and outcome values y_i for the ith observation, $i = 1, \ldots, n$
- Assume x_{ij} are standardized so that $\sum_i x_{ij}/n = 0$ and $\sum_i x_{ij}^2 = 1$. The lasso[1] solves

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq s, \qquad s > 0$$

- Equivalently, the Lagrange version of the problem, for $\lambda > 0$

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

[1] Tibshirani. JRSSB (1996)

Inspection of the Lasso Solution

- Consider a single predictor setting based on the observed data $\{(x_i, y_i)\}_{i=1}^n$. The problem then is to solve

$$\widehat{\beta}^{lasso} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda|\beta| \tag{1}$$

Inspection of the Lasso Solution

- Consider a single predictor setting based on the observed data $\{(x_i, y_i)\}_{i=1}^n$. The problem then is to solve

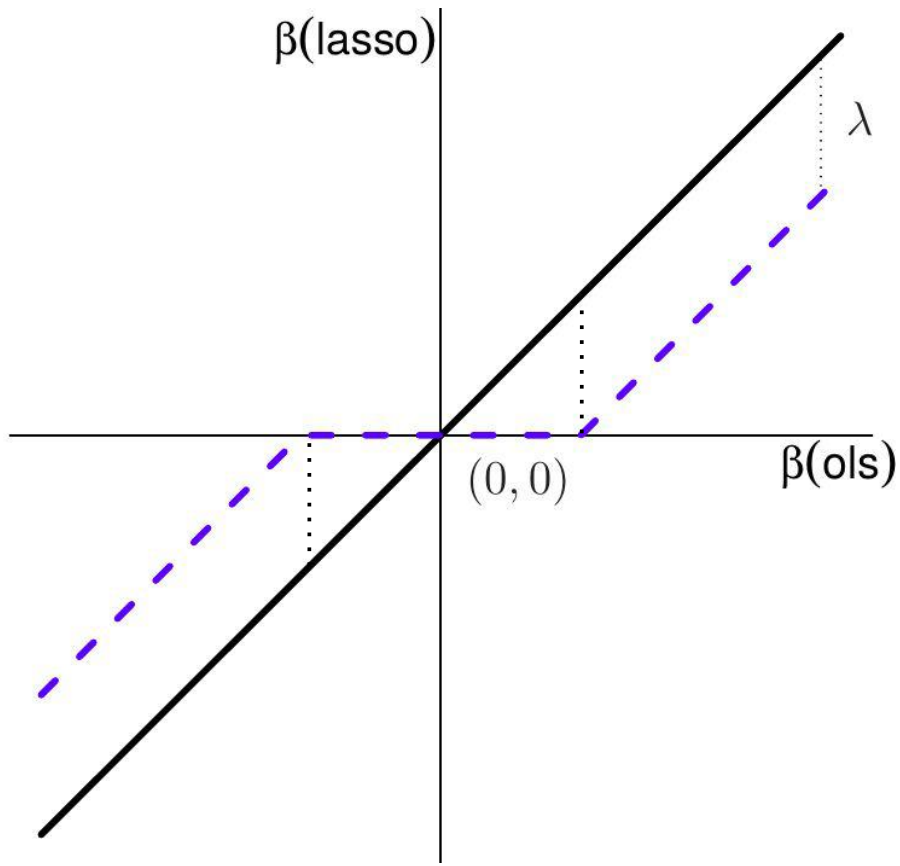
$$\widehat{\beta}^{lasso} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda|\beta| \tag{1}$$

- With a standardized predictor, the lasso solution (1) is a **soft-thresholded** version of the least-squares (LS) estimate $\widehat{\beta}^{LS}$

$$\begin{aligned}
\widehat{\beta}^{lasso} &= S_\lambda\left(\widehat{\beta}^{LS}\right) = \text{sign}\left(\widehat{\beta}^{LS}\right)\left(|\widehat{\beta}^{LS}| - \lambda\right)_+ \\
&= \begin{cases} \widehat{\beta}^{LS} - \lambda, & \widehat{\beta}^{LS} > \lambda \\ 0 & |\widehat{\beta}^{LS}| \leq \lambda \\ \widehat{\beta}^{LS} + \lambda & \widehat{\beta}^{LS} \leq -\lambda \end{cases}
\end{aligned}$$

Inspection of the Lasso Solution

- When the data are standardized, the lasso solution **shrinks the LS estimate toward zero** by the amount λ



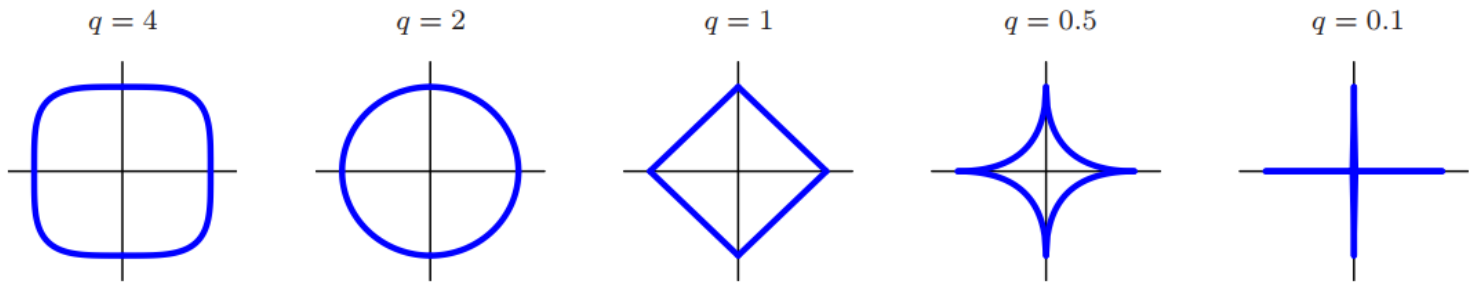
[1] Hastie et al. Statistical learning with sparsity: the lasso and generalizations

Why the ℓ_1 norm?

- For $q \geq 0$, evaluate the criteria

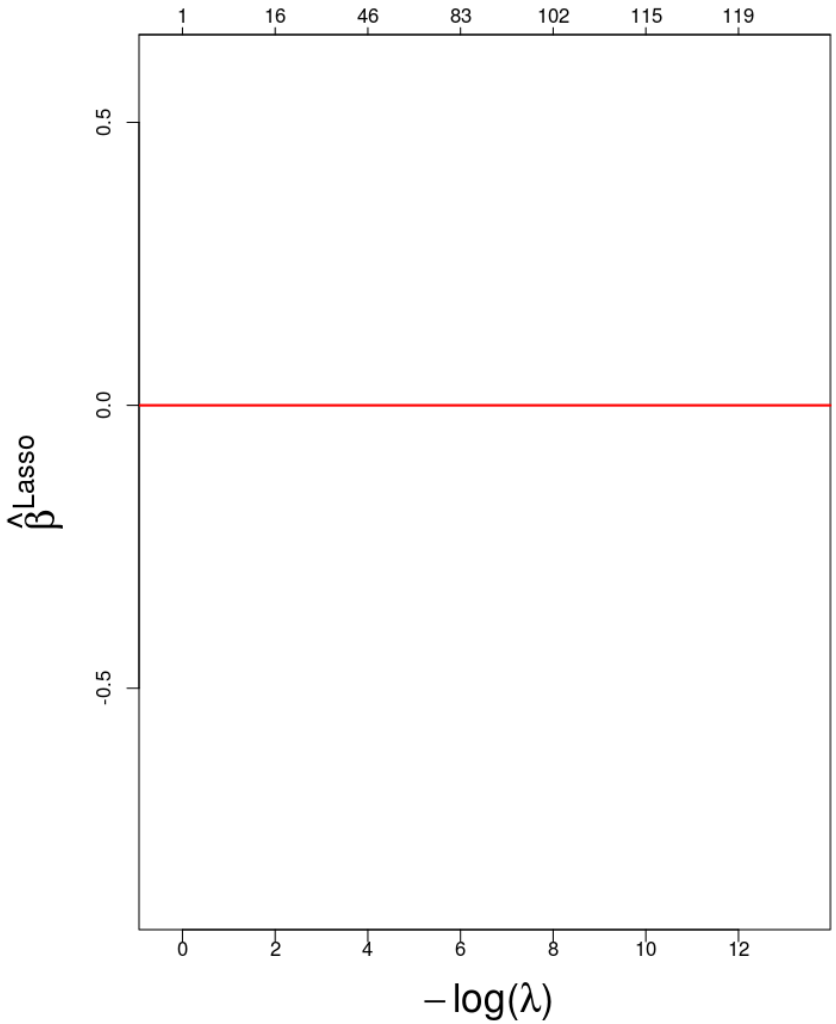
$$\widetilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}$$

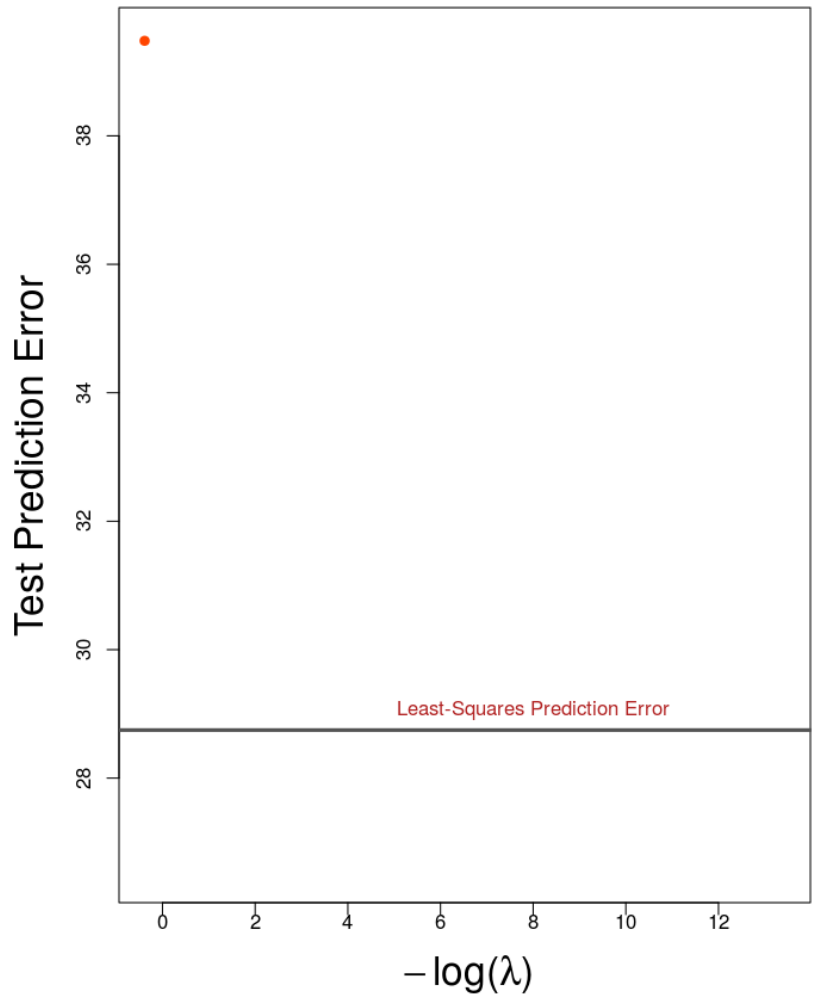
- Why do we use the ℓ_1 and not $q = 2$ (Ridge) or any other norm ℓ_q?



- $q = 1$ is the smallest value which gives sparse solutions AND is **convex** $\rightarrow$ scales well to high-dimensions
- For $q < 1$ the constrained region is **not-convex**

Choosing Model Complexity

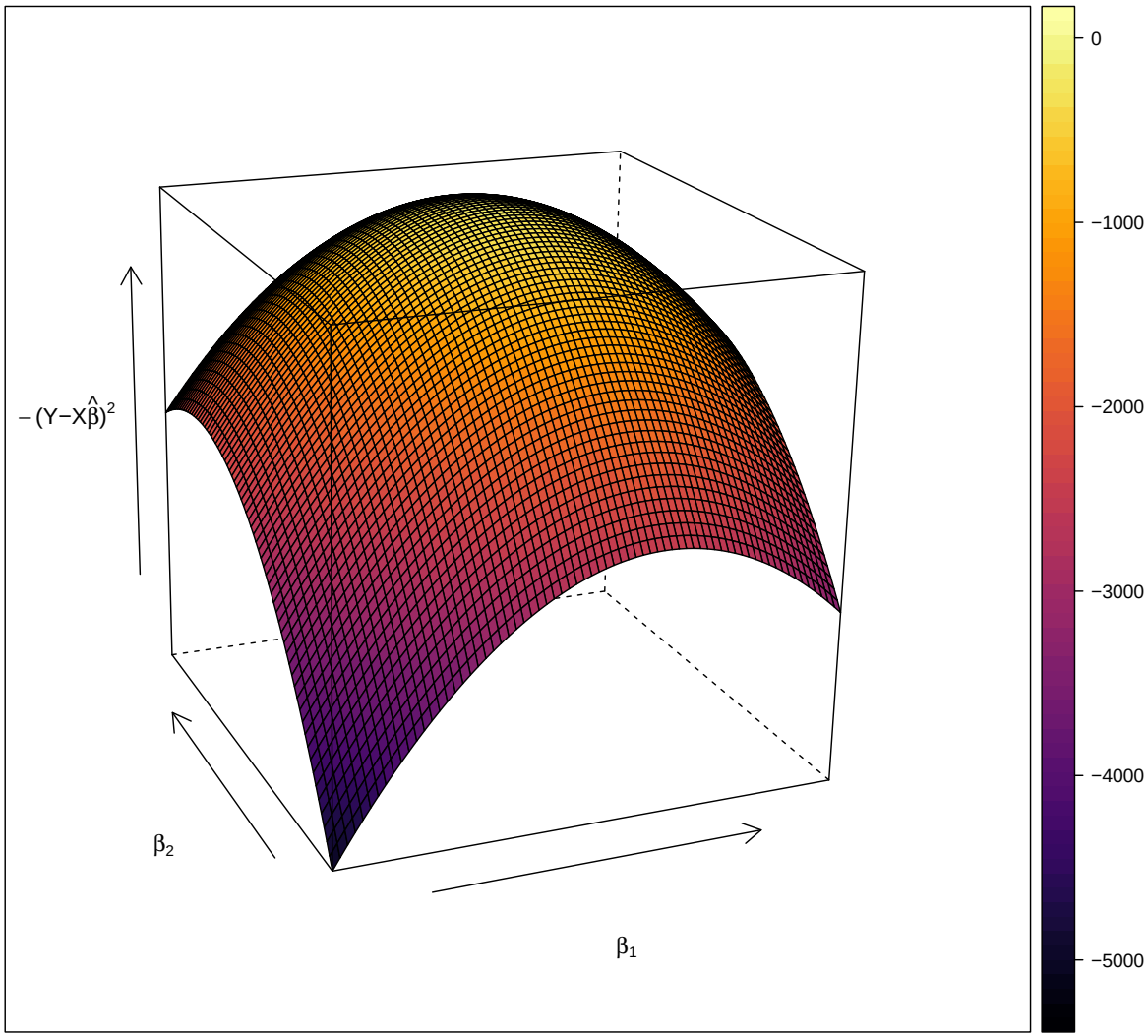




Least-squares regression surface

- Consider the following model with two predictors ($\mathbf{y}$ is centered)

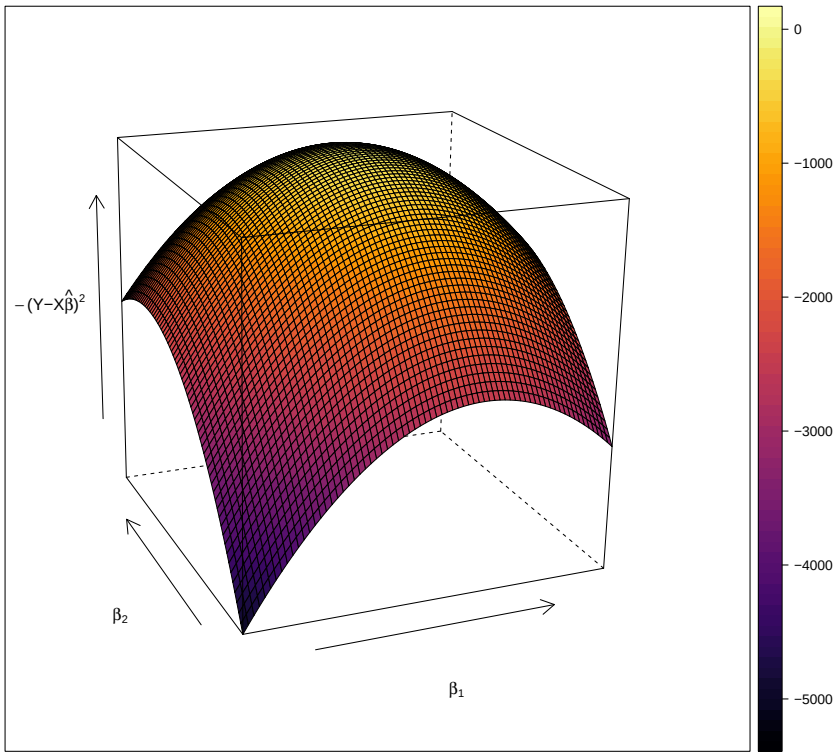
$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \boldsymbol{\varepsilon}$$

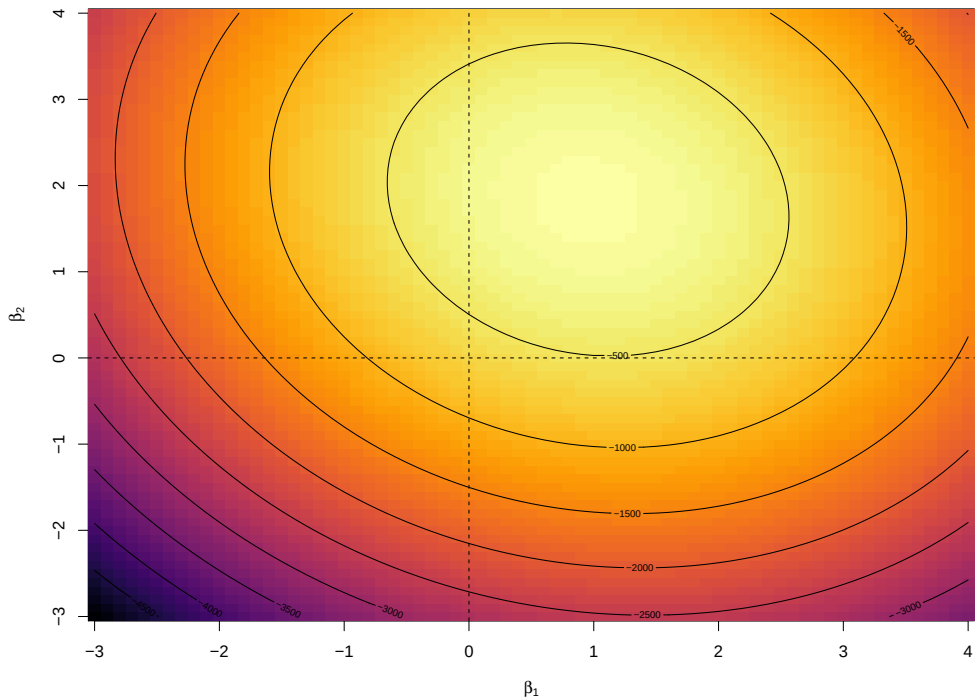


code to generate previous plot

```
pacman::p_load(viridis,fields,lattice,latex2exp,plotrix)

set.seed(12345)
b0 <- 0
b1 <- 1
b2 <- 2
X <- cbind(1,replicate(2, rnorm(100)))
y <- X %*% matrix(c(b0,b1,b2)) + sqrt(2)*rnorm(100)

# Define function for RSS
MyRss <- function(beta0, beta1) {
        b <- c(0, beta0, beta1)
        rss <- crossprod(y - X %*% b)
        return(rss)
}

b0 <- seq(-3, 4, by=0.1)
b1 <- seq(-3, 4, by = 0.1)
z <- outer(b0, b1, function(x,y) mapply(MyRss, x, y))

wireframe(-z,drape = TRUE, colorkey = TRUE, screen = list(z = 20, x = -70, y = 3),
        xlab = TeX("$\\beta_1$"), ylab = TeX("$\\beta_2$"),
        zlab = TeX("$-(Y-X\\hat{\\beta})^2$"), col.regions = viridis::inferno(100))
```

Contours of the least-squares regression surface

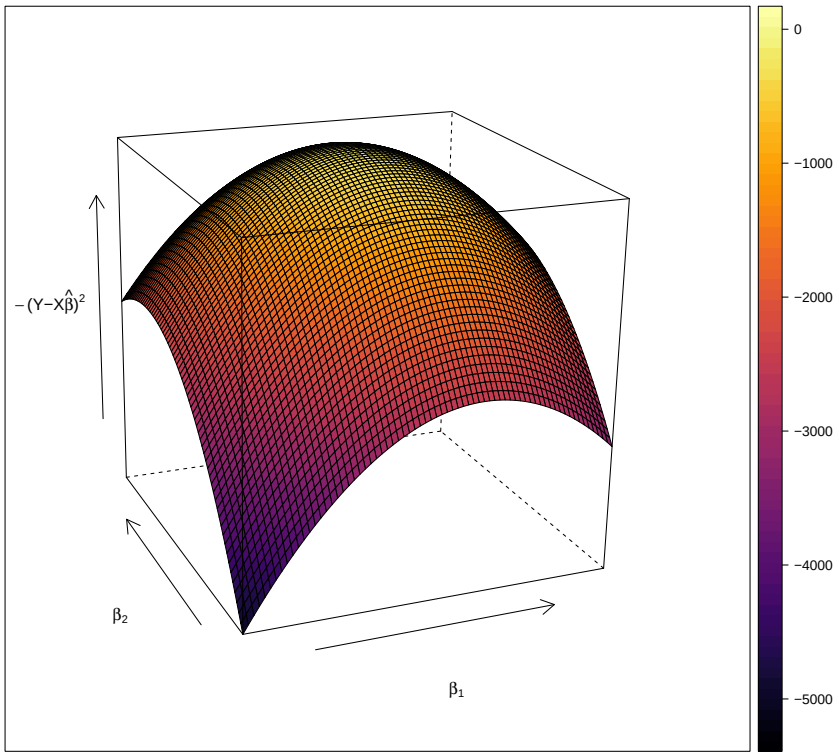


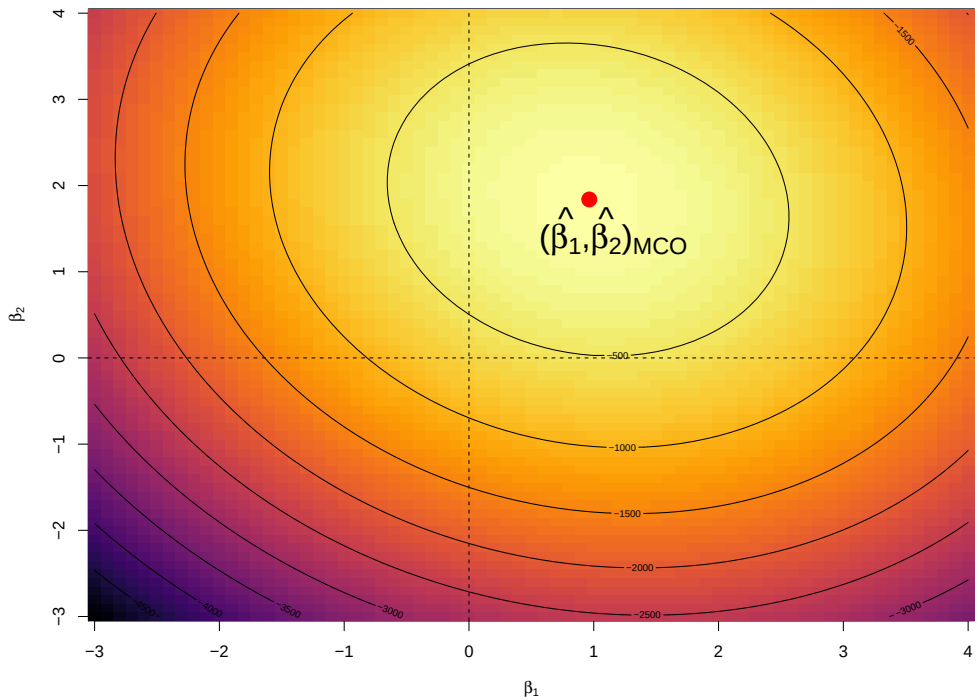


code to generate previous plot

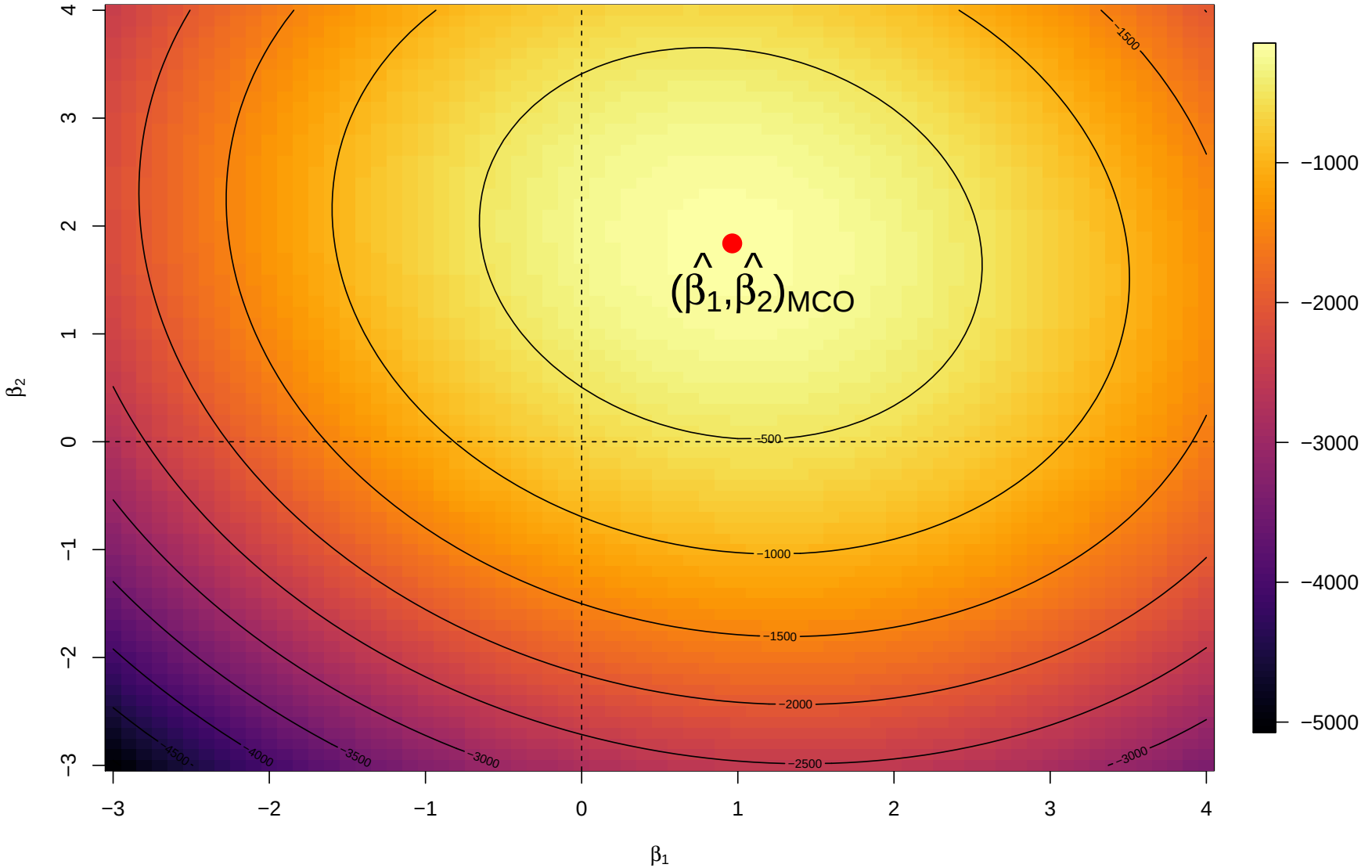
```
pacman::p_load(viridis,fields,lattice,latex2exp,plotrix)

set.seed(12345)
b0 <- 0
b1 <- 1
b2 <- 2
X <- cbind(1,replicate(2, rnorm(100)))
y <- X %*% matrix(c(b0,b1,b2)) + sqrt(2)*rnorm(100)

# Define function for RSS
MyRss <- function(beta0, beta1) {
b <- c(0, beta0, beta1)
rss <- crossprod(y - X %*% b)
return(rss)
}

b0 <- seq(-3, 4, by=0.1)
b1 <- seq(-3, 4, by = 0.1)
z <- outer(b0, b1, function(x,y) mapply(MyRss, x, y))

fields::image.plot(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"), ylab = TeX("$\\beta_2$"),
        col = viridis::inferno(100))
contour(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"), ylab = TeX("$\\beta_2$"),
        nlevels = 10, add=TRUE)
abline(v = 0, lty=2)
abline(h = 0, lty=2)
```

Contours of the least-squares regression surface

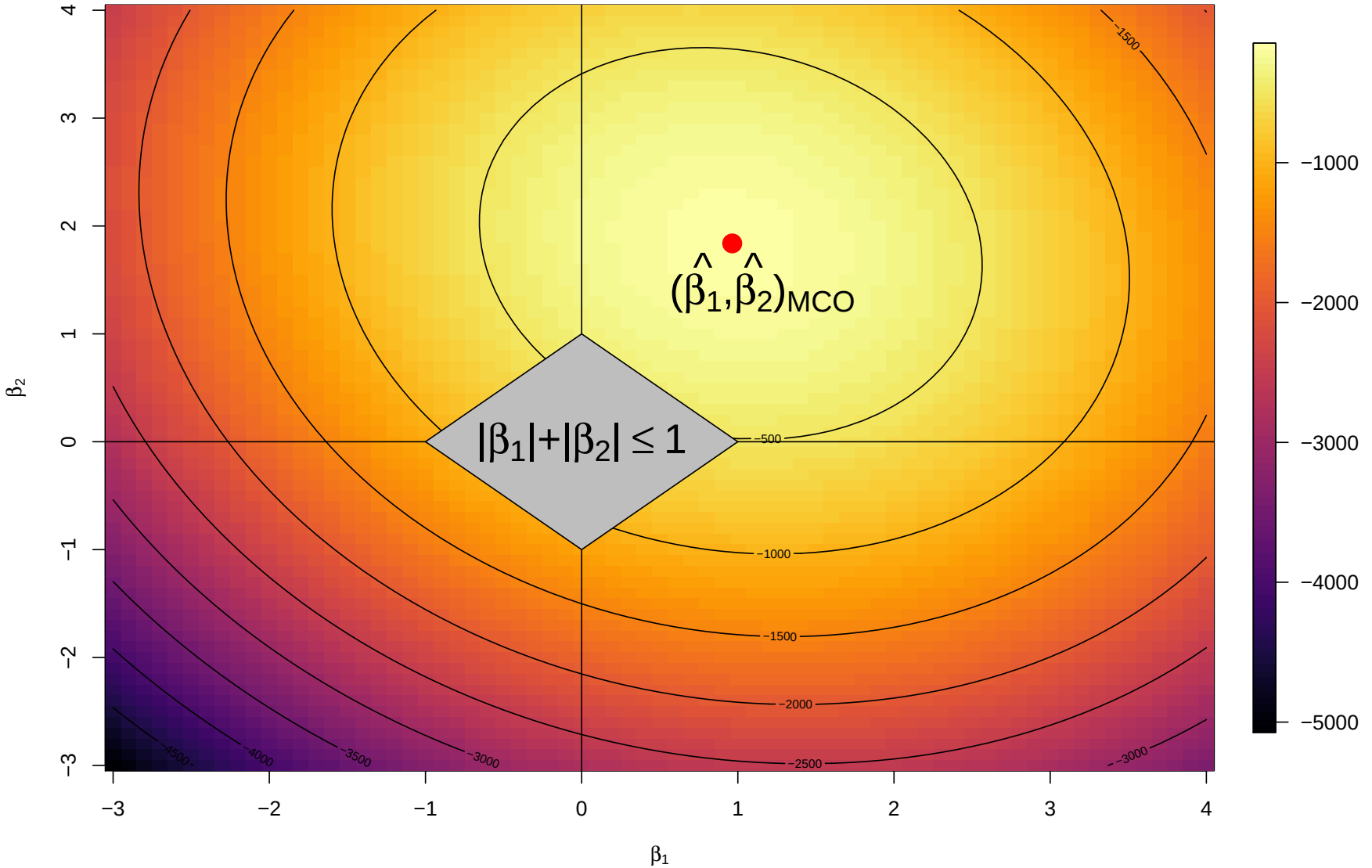




Contours of the least-squares regression surface



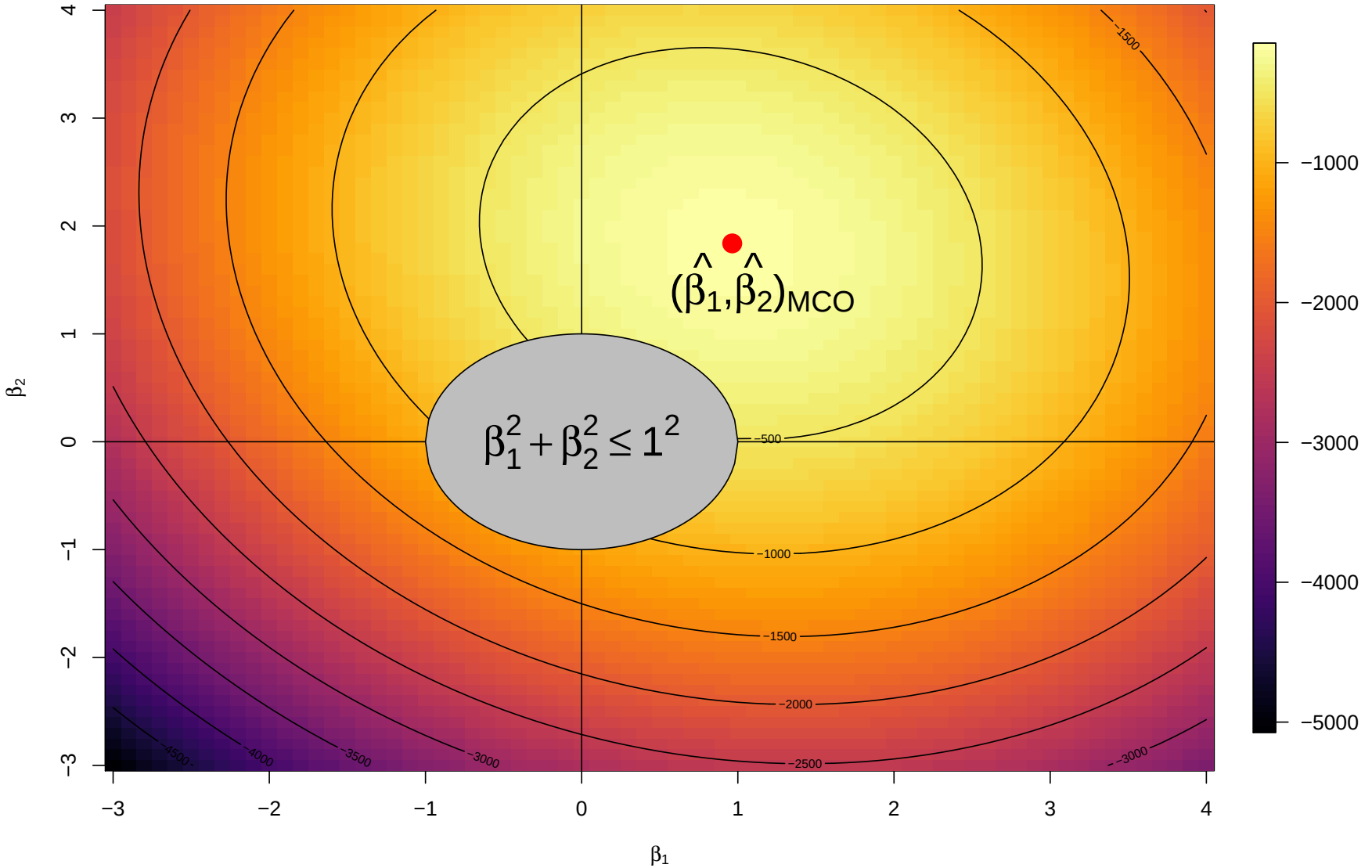
Constraint region of the lasso



code to generate previous plot

```
fields::image.plot(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"),
        ylab = TeX("$\\beta_2$"),
        col = viridis::inferno(100))
contour(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"), ylab = TeX("$\\beta_2$"),
        nlevels = 10, add=TRUE)
points(x = lm.fit(x = X, y = y)$coef[2], y = lm.fit(x = X, y = y)$coef[3],
        pch = 19, cex=2, col = "red")
text(x = lm.fit(x = X, y = y)$coef[2]*1.2,
        y = lm.fit(x = X, y = y)$coef[3]*0.80,
        labels = TeX("$(\\hat{\\beta_1},\\hat{\\beta_2})_{MCO}$"),
        cex = 2)
abline(v = 0)
abline(h = 0)

conditions <- function(x,y) {
        c1 <- (abs(x) + abs(y)) <= 1
        return(c1)}

zz <- expand.grid(x=b0,y=b1)
zz <- zz[conditions(zz$x,zz$y),]

polygon(c(zz$x[which.min(zz$x)],zz$x[which.max(zz$y)],
        zz$x[which.max(zz$x)], zz$x[which.min(zz$y)]),
        c(zz$y[which.min(zz$x)],zz$y[which.max(zz$y)],
        zz$y[which.max(zz$x)], zz$y[which.min(zz$y)]),
        col = "grey")
text(x = 0, y= 0,
        labels = TeX("$|\\beta_1|+|\\beta_2| \\leq 1$"), cex = 2)
```

Constraint region of the ridge



code to generate previous plot

```
fields::image.plot(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"),
        ylab = TeX("$\\beta_2$"),
        col = viridis::inferno(100))
contour(x = b0, y = b1, z = -z,xlab = TeX("$\\beta_1$"), ylab = TeX("$\\beta_2$"),
        nlevels = 10, add=TRUE)
points(x = lm.fit(x = X, y = y)$coef[2], y = lm.fit(x = X, y = y)$coef[3],
        pch = 19, cex=2, col = "red")
text(x = lm.fit(x = X, y = y)$coef[2]*1.2,
        y = lm.fit(x = X, y = y)$coef[3]*0.80,
        labels = TeX("$(\\hat{\\beta_1},\\hat{\\beta_2})_{MCO}$"), cex = 2)
abline(v = 0)
abline(h = 0)

beta2 <- function(x,r=1) {
        y <- sqrt(r^2 - x^2)
        return(y)}

xseq <- seq(-1,1, length.out = 100)
polygon(cbind(c(xseq, rev(xseq)),c(beta2(x=xseq), -beta2(x=xseq))), col = "grey")
text(x = 0, y= 0,
        labels = TeX("$\\beta_1^2+\\beta_2^2 \\leq 1^2$"), cex = 2)
```

Lasso vs. ridge

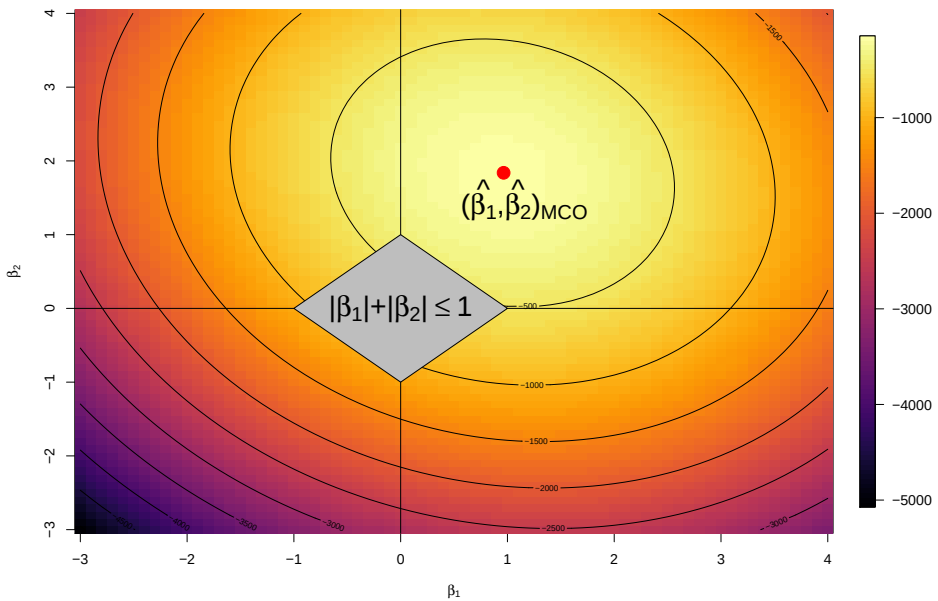


Fig. 1: `lasso`

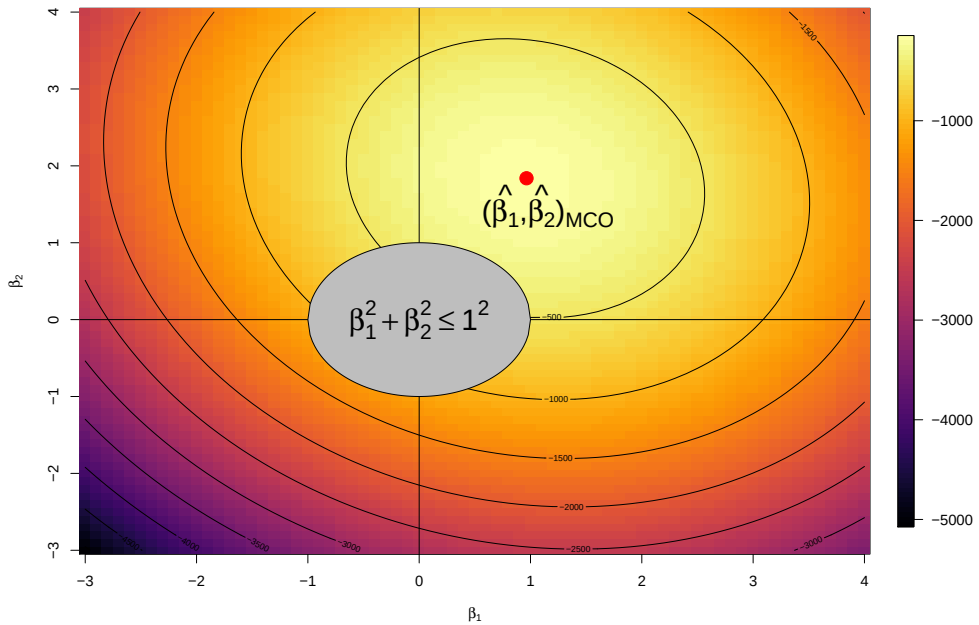
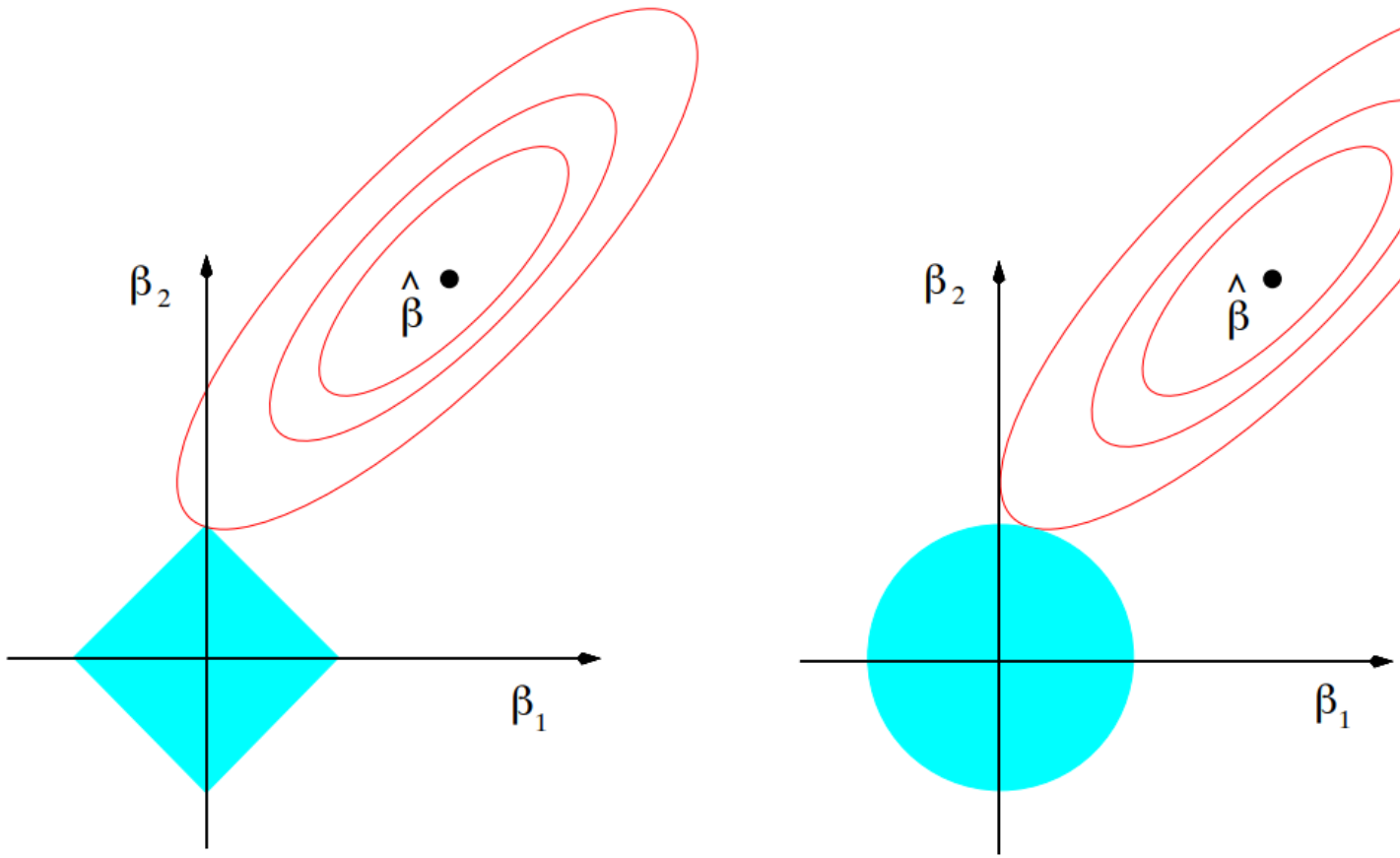


Fig. 2: `ridge`

Classic version of the previous figure



Optimality Conditions

Score functions and penalized score functions

- In classical statistical theory, the derivative of the log-likelihood function $\mathcal{L}(\theta)$ is called the score function, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the likelihood equations (or score equations):

$$0 = \frac{\partial}{\partial \theta} \mathcal{L}(\theta)$$

Score functions and penalized score functions

- In classical statistical theory, the derivative of the log-likelihood function $\mathcal{L}(\theta)$ is called the score function, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the likelihood equations (or score equations):

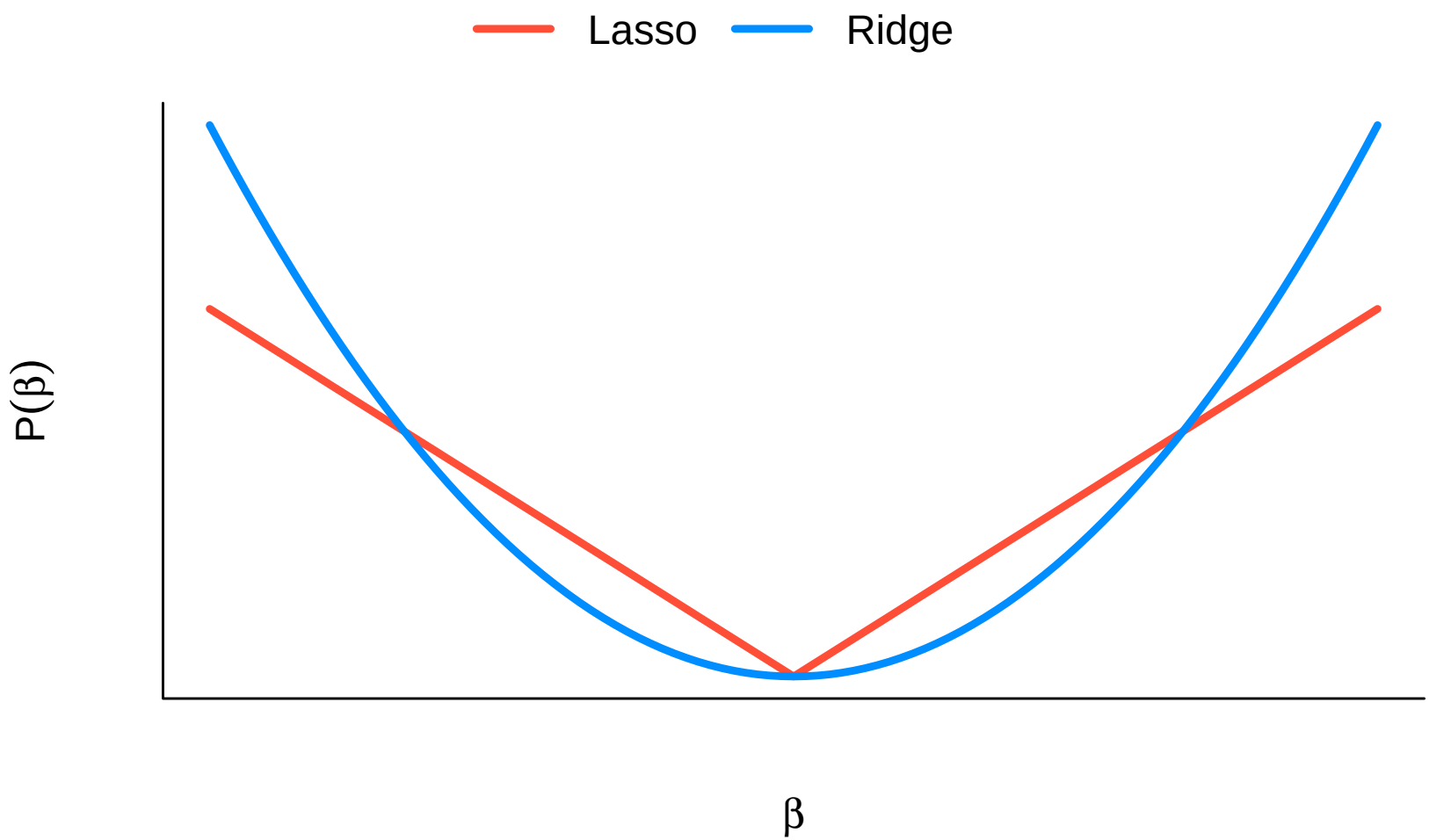
$$0 = \frac{\partial}{\partial \theta} \mathcal{L}(\theta)$$

- Extending this idea to penalized likelihoods involves taking the derivatives of objective functions of the form:

$$\mathbf{Q}(\theta) = \underbrace{\mathcal{L}(\theta)}_{\text{likelihood}} + \underbrace{P(\theta)}_{\text{penalty}}$$

yielding the penalized score function

Ridge vs. Lasso penalty



Penalized likelihood equations

- For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward

$$\widehat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_2^2$$

- For the lasso, the penalized likelihood is not differentiable - specifically, not differentiable at zero - and *subdifferentials* are needed to characterize them

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

---

http://myweb.uiowa.edu/pbreheny/7240/s19/notes/2-13.pdf

Penalized likelihood equations

- For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward

$$\widehat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_2^2$$

- For the lasso, the penalized likelihood is not differentiable - specifically, not differentiable at zero - and *subdifferentials* are needed to characterize them

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Letting $\partial\mathbf{Q}(\theta)$ denote the subdifferential of $\mathbf{Q}$, penalized likelihood equations are:

$$0 \in \partial\mathbf{Q}(\theta)$$

---

http://myweb.uiowa.edu/pbreheny/7240/s19/notes/2-13.pdf

Karush-Kuhn-Tucker (KKT) Conditions

- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions

Karush-Kuhn-Tucker (KKT) Conditions

- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions
- For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution

Karush-Kuhn-Tucker (KKT) Conditions

- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions
- For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution
- The idea is simple: to solve for $\widehat{\boldsymbol{\beta}}^{lasso}$, we simply replace the derivative with the subderivative and the likelihood with the penalized likelihood

Subdifferential for $|x|$

The subdifferential for $f(x) = |x|$ is:

$$\partial|x| = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

$$\frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| \leq \lambda \qquad \widehat{\beta}_j = 0$$

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

$$\frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| \leq \lambda \qquad \widehat{\beta}_j = 0$$

- In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

$$\frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| \leq \lambda \qquad \widehat{\beta}_j = 0$$

- In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model

- When this correlation is below λ, $\widehat{\beta}_j = 0$

Some remarks

- If we set

$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$$

  then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions

- That is, for any $\lambda \geq \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

Some remarks

- If we set

$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$$

  then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions

- That is, for any $\lambda \ge \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

- On the other hand, if we set $\lambda = 0$, the KKT conditions are simple the normal equations for OLS

$$\frac{1}{n} \mathbf{x}_j^\top (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0 \cdot \operatorname{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

Some remarks

- If we set

$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$$

  then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions

- That is, for any $\lambda \ge \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

- On the other hand, if we set $\lambda = 0$, the KKT conditions are simple the normal equations for OLS

$$\frac{1}{n} \mathbf{x}_j^\top (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0 \cdot \operatorname{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

- Thus, the coefficient path for the lasso starts at $\lambda_{\max}$ and continues until $\lambda = 0$ if $\mathbf{X}$ is full rank; otherwise the solution will fail to be unique for λ values below some point $\lambda_{\min}$

Recall the Lasso Solution in the Orthonormal Design

- When the design matrix $\mathbf{X}$ is orthonormal, i.e., $n^{-1}\mathbf{X}^\top\mathbf{X} = \mathbf{I}$, the lasso estimate is a **soft-thresholded** version of the least-squares (LS) estimate $\widehat{\beta}^{LS}$

$$\widehat{\beta}^{lasso} = S_\lambda\left(\widehat{\beta}^{LS}\right) = \text{sign}\left(\widehat{\beta}^{LS}\right)\left(|\widehat{\beta}^{LS}| - \lambda\right)_+$$

$$= \begin{cases} \widehat{\beta}^{LS} - \lambda, & \widehat{\beta}^{LS} > \lambda \\ 0 & |\widehat{\beta}^{LS}| \leq \lambda \\ \widehat{\beta}^{LS} + \lambda & \widehat{\beta}^{LS} \leq -\lambda \end{cases}$$

- where $\widehat{\beta}^{LS} = \mathbf{x}_j^\top\mathbf{y}/n$

Probability that $\hat{\beta}_j = 0$

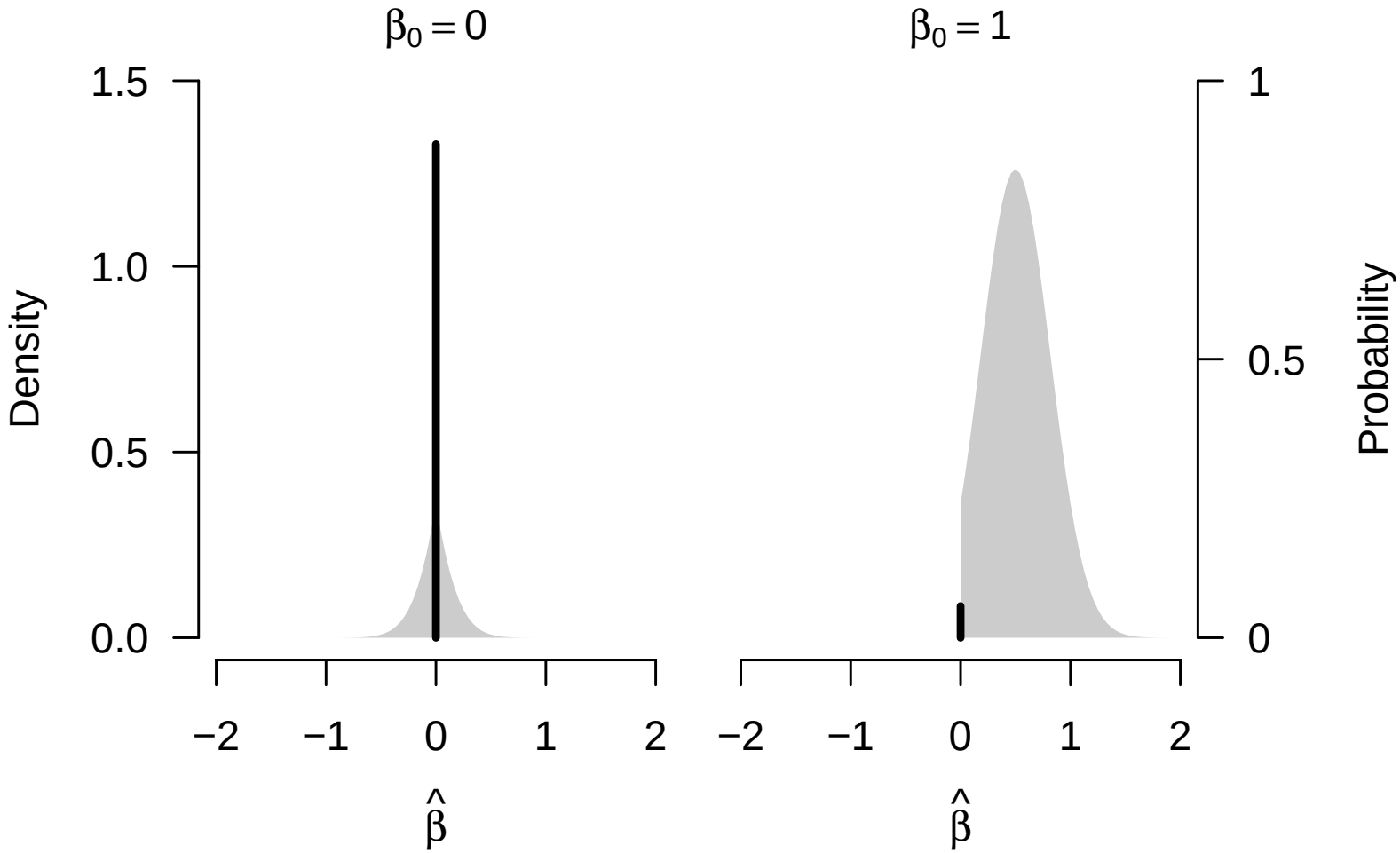
- With soft thresholding, it is clear that the lasso has a positive probability of yielding an estimate of exactly 0 - in other words, of producing a sparse solution
- Specifically, the probability of dropping $\mathbf{x}_j$ from the model is $\mathbb{P}\left(\left|\beta_j^{LS}\right| \leq \lambda\right)$
- Under the assumption that $\epsilon_i \overset{\perp\!\!\!\perp}{\sim} \mathrm{N}\left(0, \sigma^2\right)$, we have $\beta_j^{LS} \sim \mathcal{N}(\beta, \sigma^2/n)$ and

$$\mathbb{P}\left(\widehat{\beta}_j(\lambda) = 0\right) = \Phi\left(\frac{\lambda - \beta}{\sigma/\sqrt{n}}\right) - \Phi\left(\frac{-\lambda - \beta}{\sigma/\sqrt{n}}\right)$$

where Φ is the Gaussian CDF

Sampling Distribution

For $\sigma = 1$, $n = 10$, and $\lambda = 1/2$:



Why standard inference is invalid?

- This sampling distribution is very different from that of a classical MLE:
  - The distribution is mixed: a portion is continuously distributed, but there is also a point mass at zero
  - The continuous portion is not normally distributed
  - The distribution is asymmetric (unless $\beta = 0$)
  - The distribution is not centered at the true value of β

Algorithms

Algorithms for the lasso

- The KKT conditions only allow us to check a solution

Algorithms for the lasso

- The KKT conditions only allow us to check a solution
- They do not necessarily help us to find the solution in the first place

Coordinate descent[1]

- The idea behind coordinate descent is, simply, to optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached

[1] Fu (1998), Friedman et al. (2007), Wu and Lange (2008)

Coordinate descent[1]

- The idea behind coordinate descent is, simply, to optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached
- Coordinate descent is particularly suitable for problems, like the lasso, that have a simple closed form solution in a single dimension but lack one in higher dimensions

[1] Fu (1998), Friedman et al. (2007), Wu and Lange (2008)

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

$$\widetilde{\beta}_j = \arg\min_{\beta_j} \mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = S_\lambda(\tilde{z}_j) = \begin{cases} \tilde{z}_j - \lambda, & \tilde{z}_j > \lambda \\ 0 & |\tilde{z}_j| \leq \lambda \\ \tilde{z}_j + \lambda & \tilde{z}_j < -\lambda \end{cases}$$

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

$$\widetilde{\beta}_j = \arg\min_{\beta_j} \mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = S_\lambda(\tilde{z}_j) = \begin{cases} \tilde{z}_j - \lambda, & \tilde{z}_j > \lambda \\ 0 & |\tilde{z}_j| \leq \lambda \\ \tilde{z}_j + \lambda & \tilde{z}_j < -\lambda \end{cases}$$

- $\tilde{r}_{ij} = y_i - \sum_{k\neq j} x_{ik}\widetilde{\beta}_k$ $\qquad\qquad$ $\tilde{z}_j = n^{-1}\sum_{i=1}^{n} x_{ij}\tilde{r}_{ij}$

- $\{\tilde{r}_{ij}\}_{i=1}^{n}$ are the partial residuals with respect to the j^{th} predictor, and $\tilde{z}_j$ OLS estimator based on $\{\tilde{r}_{ij}, x_{ij}\}_{i=1}^{n}$

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

  has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:
  - the function $\mathcal{L}(\boldsymbol{\beta})$ is differentiable and
  - the penalty function $P_\lambda(\boldsymbol{\beta})$ is separable $\rightarrow P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

  has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:
  - the function $\mathcal{L}(\boldsymbol{\beta})$ is differentiable and
  - the penalty function $P_\lambda(\boldsymbol{\beta})$ is separable $\rightarrow P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$

- Lasso-penalized linear regression satisfies both of these criteria

Convergence

- Furthermore, because the lasso objective is a convex function, the sequence of the objective functions $\left\{Q\left(\widetilde{\boldsymbol{\beta}}^{(s)}\right)\right\}$ converges to the global minimum
- However, because the lasso objective is not strictly convex, there may be multiple solutions
- In such situations, coordinate descent will converge to one of those solutions, but which solution it converges to is essentially arbitrary, as it depends on the order of the features

Coordinate descent, pathwise optimization, warm starts

- We are typically interested in determining $\widehat{\boldsymbol{\beta}}^{Lasso}$ for a range of values of λ, thereby obtaining the coefficient path
- In applying the coordinate descent algorithm to determine the lasso path, an efficient strategy is to compute solutions for decreasing values of λ, starting at $\lambda_{\max} = \max_{1 \leq j \leq p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$, the point at which all coefficients are 0
- Warm starts $\rightarrow$ By continuing along a decreasing grid of λ values, we can use the solutions $\widehat{\boldsymbol{\beta}}(\lambda_k)$ as initial values when solving for $\widehat{\boldsymbol{\beta}}(\lambda_{k+1})$

Group Lasso

Motivating Dataset

	ID	Response	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6
1	2610781	−1.255	1	2	0	0	0	1
2	4114347	−0.339	1	2	0	2	0	1
3	4399930	−0.6	1	2	1	1	0	1
4	2081319	0.809	1	2	0	1	0	2
5	1347380	0.279	2	2	0	0	0	0
6	3262449	−0.421	2	2	0	1	0	1
7	4870063	−0.454	2	2	0	0	0	2
8	1141212	1.383	2	2	1	1	1	0
9	2997954	−2.29	1	2	0	0	0	1
10	5805218	2.289	1	2	0	1	1	1

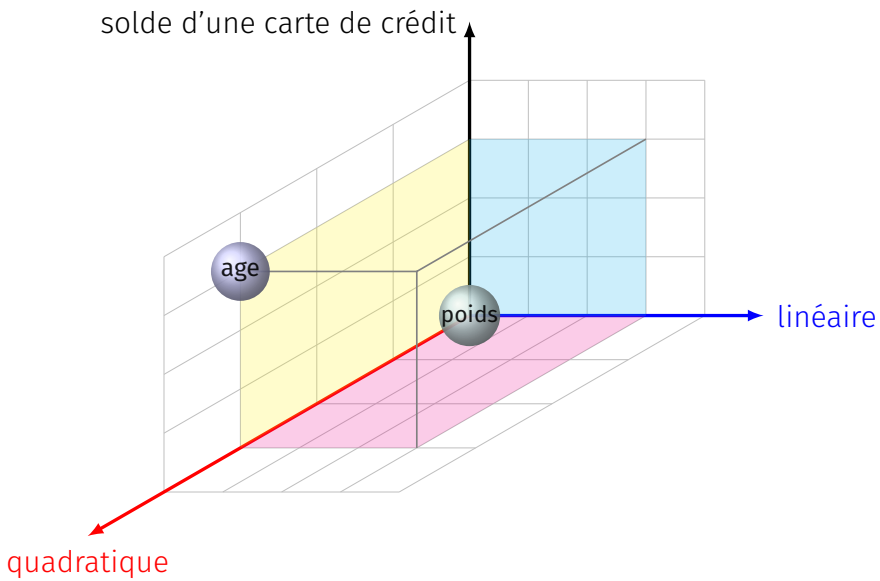
Groups of Predictors Affect the Response

	ID	Response	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6
1	2610781	−1.255	1	2	0	0	0	1
2	4114347	−0.339	1	2	0	2	0	1
3	4399930	−0.6	1	2	1	1	0	1
4	2081319	0.809	1	2	0	1	0	2
5	1347380	0.279	2	2	0	0	0	0
6	3262449	−0.421	2	2	0	1	0	1
7	4870063	−0.454	2	2	0	0	0	2
8	1141212	1.383	2	2	1	1	1	0
9	2997954	−2.29	1	2	0	0	0	1
10	5805218	2.289	1	2	0	1	1	1

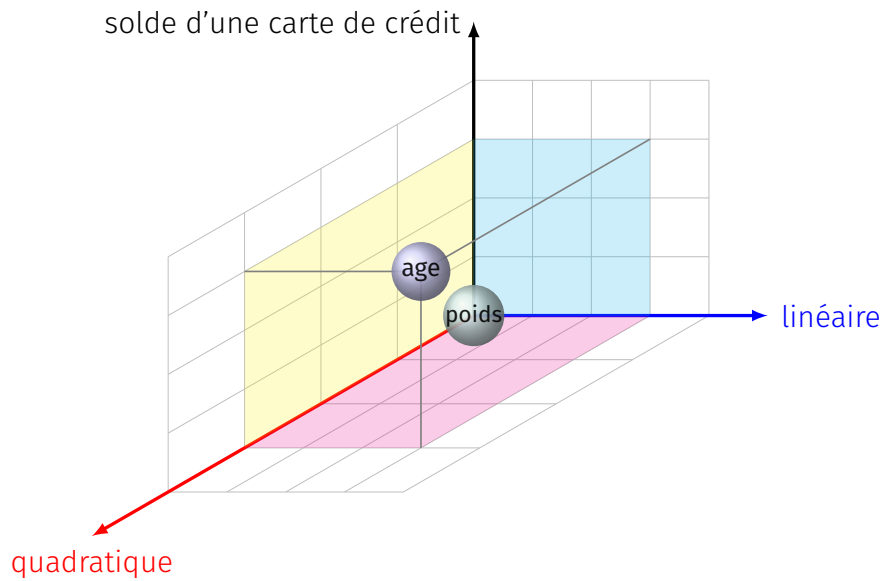
Group lasso for Categorical variables and Basis expansions

Useful for groups of variables (factor with > 2 categories, *Age*, Age^2). **Group lasso** estimator is:

$$\min_{(\beta_0,\boldsymbol{\beta})} \frac{1}{2} \|\mathbf{y} - \beta_0 - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k} \|\boldsymbol{\beta}^{(k)}\|_2 \qquad p_k - \text{taille de groupe}$$



(a) Lasso



(b) Groupe lasso

Group Lasso Model

- Assume the predictors in $\mathbf{X} \in \mathbb{R}^{n \times p}$ belong to K **non-overlapping groups** with **pre-defined** group membership and cardinality p_k
- Let $\boldsymbol{\beta}_{(k)}$ to denote the segment of $\boldsymbol{\beta}$ corresponding to group k

Group Lasso Model

- Assume the predictors in $\mathbf{X} \in \mathbb{R}^{n \times p}$ belong to K **non-overlapping groups** with **pre-defined** group membership and cardinality p_k
- Let $\boldsymbol{\beta}_{(k)}$ to denote the segment of $\boldsymbol{\beta}$ corresponding to group k
- We consider the group lasso penalized estimator

$$\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}|\mathbf{D}) + \lambda \sum_{k=1}^{K} w_k \|\boldsymbol{\beta}_{(k)}\|_2, \tag{2}$$

- where

$$L(\boldsymbol{\beta} \mid \mathbf{D}) = \frac{1}{2} \left[\mathsf{Y} - \widehat{\mathsf{Y}}\right]^\top \mathbf{W} \left[\mathsf{Y} - \widehat{\mathsf{Y}}\right] \tag{3}$$

$\widehat{\mathsf{Y}} = \sum_{j=1}^{p} \beta_j X_j$, $\mathbf{D}$ is the working data $\{\mathsf{Y}, \mathbf{X}\}$, and $\mathsf{W}_{n \times n}$ is an observation weight matrix

Groupwise Descent: Exploiting Sparsity Structure

Minimize the objective function

$$\frac{1}{2}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]^{\top}\mathbf{W}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]+\lambda\sum_{k=1}^{K}w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

Groupwise Descent: Exploiting Sparsity Structure

Minimize the objective function

$$\frac{1}{2}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]^{\top}\mathbf{W}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]+\lambda\sum_{k=1}^{K}w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

During each sub-iteration only optimize $\boldsymbol{\beta}^{(k)}$. Set $\boldsymbol{\beta}^{(k')}=\widetilde{\boldsymbol{\beta}}^{(k')}$ for $k'\neq k$ at their current value.

1. Initialization: $\widetilde{\boldsymbol{\beta}}$

Groupwise Descent: Exploiting Sparsity Structure

Minimize the objective function

$$\frac{1}{2}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]^{\top}\mathbf{W}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]+\lambda\sum_{k=1}^{K}w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

During each sub-iteration only optimize $\boldsymbol{\beta}^{(k)}$. Set $\boldsymbol{\beta}^{(k')}=\widetilde{\boldsymbol{\beta}}^{(k')}$ for $k'\neq k$ at their current value.

1. Initialization: $\widetilde{\boldsymbol{\beta}}$
2. Cyclic groupwise descent: for $k=1,2,\ldots,K$, update $\boldsymbol{\beta}^{(k)}$ by minimizing the objective function

$$\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})\leftarrow\arg\min_{\boldsymbol{\beta}^{(k)}}L(\boldsymbol{\beta}\mid\mathbf{D})+\lambda w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

Groupwise Descent: Exploiting Sparsity Structure

Minimize the objective function

$$\frac{1}{2}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]^{\top}\mathbf{W}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]+\lambda\sum_{k=1}^{K}w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

During each sub-iteration only optimize $\boldsymbol{\beta}^{(k)}$. Set $\boldsymbol{\beta}^{(k')}=\widetilde{\boldsymbol{\beta}}^{(k')}$ for $k'\neq k$ at their current value.

1. Initialization: $\widetilde{\boldsymbol{\beta}}$
2. Cyclic groupwise descent: for $k=1,2,\ldots,K$, update $\boldsymbol{\beta}^{(k)}$ by minimizing the objective function

$$\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})\leftarrow\arg\min_{\boldsymbol{\beta}^{(k)}}L(\boldsymbol{\beta}\mid\mathbf{D})+\lambda w_k\|\boldsymbol{\beta}^{(k)}\|_2$$

3. Repeat (2) till convergence.

Quadratic Majorization Condition

$$\underset{\boldsymbol{\beta}^{(k)}}{\arg\min} \frac{1}{2} \left[\mathsf{Y} - \widehat{\mathsf{Y}} \right]^{\top} \mathbf{W} \left[\mathsf{Y} - \widehat{\mathsf{Y}} \right] + \lambda \sum_{k=1}^{K} w_k \| \boldsymbol{\beta}^{(k)} \|_2 \tag{4}$$

- Unfortunately, there is no closed form solution to (4)

[1] Yang and Zou. Statistical Computing (2014)

Quadratic Majorization Condition

$$\underset{\boldsymbol{\beta}^{(k)}}{\arg\min}\ \frac{1}{2}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]^{\top}\mathbf{W}\left[\mathsf{Y}-\widehat{\mathsf{Y}}\right]+\lambda\sum_{k=1}^{K}w_k\|\boldsymbol{\beta}^{(k)}\|_2 \tag{4}$$

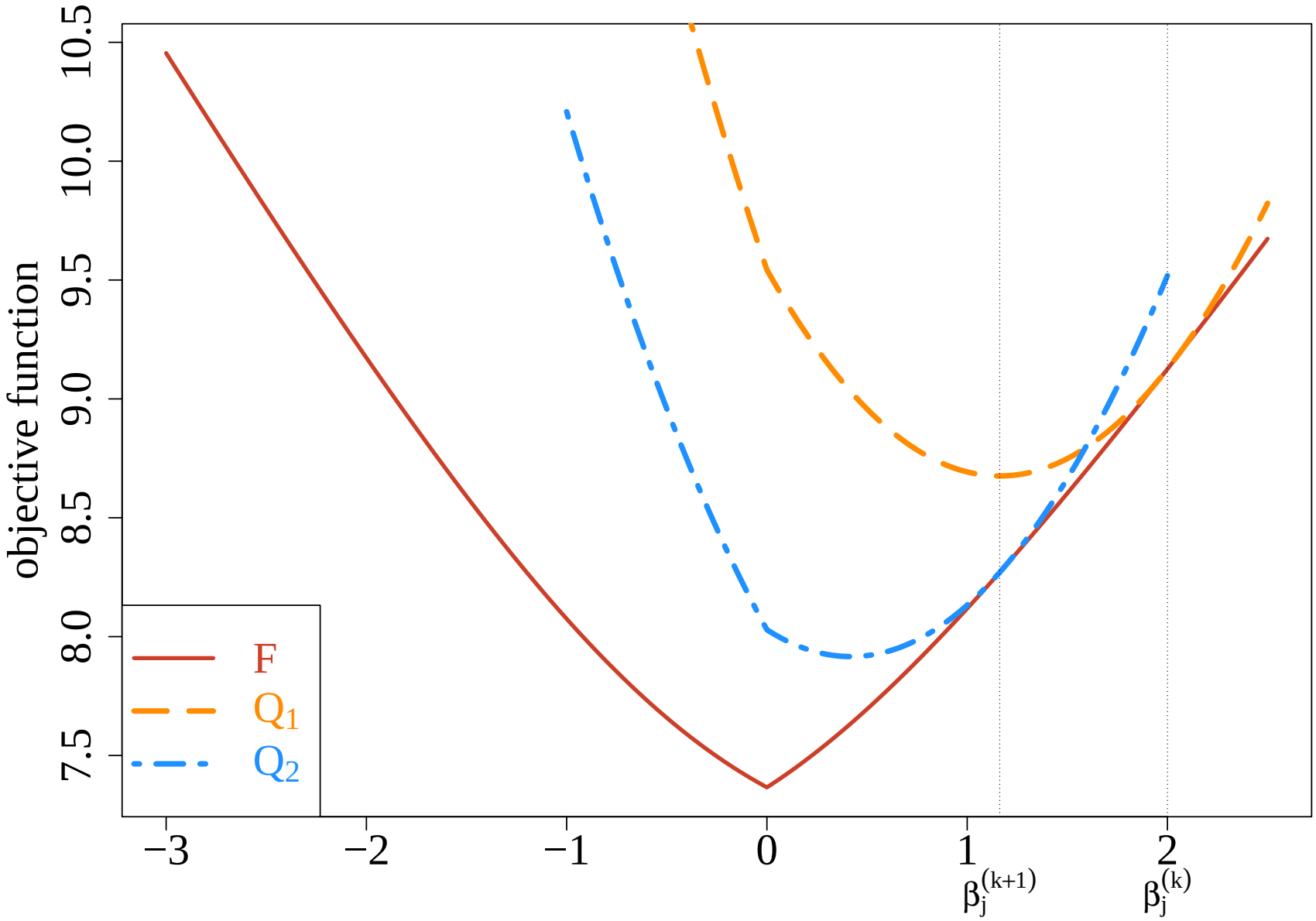
- Unfortunately, there is no closed form solution to (4)
- However, the loss function $L(\boldsymbol{\beta}|\mathbf{D})$ satisfies the quadratic majorization (QM) condition[1], since there exists
  - a $p \times p$ matrix $\mathbf{H} = \mathbf{X}^{\top}\mathbf{W}\mathbf{X}$, and
  - $\nabla L(\boldsymbol{\beta}|\mathbf{D}) = -\left(Y-\hat{Y}\right)^{\top}\mathbf{W}\mathbf{X}$

  which may only depend on the data $\mathbf{D}$, such that for all $\boldsymbol{\beta}, \boldsymbol{\beta}^*$,

$$L(\boldsymbol{\beta}\mid\mathbf{D}) \leq L(\boldsymbol{\beta}^*\mid\mathbf{D})+(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^{\intercal}\nabla L(\boldsymbol{\beta}^*|\mathbf{D})+\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^{\intercal}\mathbf{H}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)$$

[1] Yang and Zou. Statistical Computing (2014)

Generalized Coordinate Descent (GCD)



Groupwise Majorization Descent

- Update $\boldsymbol{\beta}$ in a groupwise fashion

$$\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}} = (\underbrace{0, \ldots, 0}_{k-1}, \boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}, \underbrace{0, \ldots, 0}_{K-k})$$

Groupwise Majorization Descent

- Update $\boldsymbol{\beta}$ in a groupwise fashion

$$\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}} = (\underbrace{0, \ldots, 0}_{k-1}, \boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}, \underbrace{0, \ldots, 0}_{K-k})$$

- Only need to compute the majorization function on group level

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^{\intercal} U^{(k)} + \frac{1}{2}\gamma_k (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^{\intercal} (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})$$

$$U^{(k)} = \frac{\partial}{\partial \boldsymbol{\beta}_{(k)}} L(\boldsymbol{\beta} \mid \mathbf{D}) = -\left(Y - \hat{Y}\right)^{\top} \mathbf{W}\mathbf{X}_{(k)}$$

$$\mathbf{H}^{(k)} = \frac{\partial^2}{\partial \boldsymbol{\beta}_{(k)} \partial \boldsymbol{\beta}_{(k)}^{\top}} L(\boldsymbol{\beta} \mid \mathbf{D}) = \mathbf{X}_{(k)}^{\top} \mathbf{W}\mathbf{X}_{(k)}$$

- $\gamma_k = \mathrm{eigen}_{\max}(\mathbf{H}^{(k)})$

Groupwise Majorization Descent

- Update $\boldsymbol{\beta}$ in a groupwise fashion

$$\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}} = (\underbrace{0, \ldots, 0}_{k-1}, \boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}, \underbrace{0, \ldots, 0}_{K-k})$$

- Only need to compute the majorization function on group level

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^{\intercal} U^{(k)} + \frac{1}{2}\gamma_k (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^{\intercal} (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})$$

$$U^{(k)} = \frac{\partial}{\partial \boldsymbol{\beta}_{(k)}} L(\boldsymbol{\beta} \mid \mathbf{D}) = -\left(Y - \hat{Y}\right)^{\top} \mathbf{W}\mathbf{X}_{(k)}$$

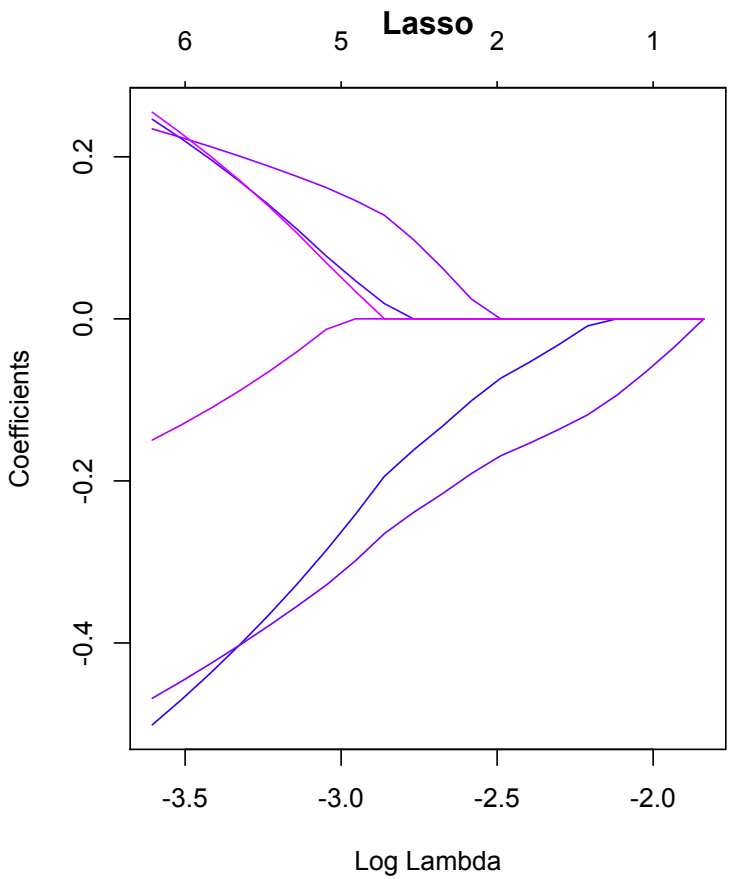
$$\mathbf{H}^{(k)} = \frac{\partial^2}{\partial \boldsymbol{\beta}_{(k)} \partial \boldsymbol{\beta}_{(k)}^{\top}} L(\boldsymbol{\beta} \mid \mathbf{D}) = \mathbf{X}_{(k)}^{\top} \mathbf{W}\mathbf{X}_{(k)}$$

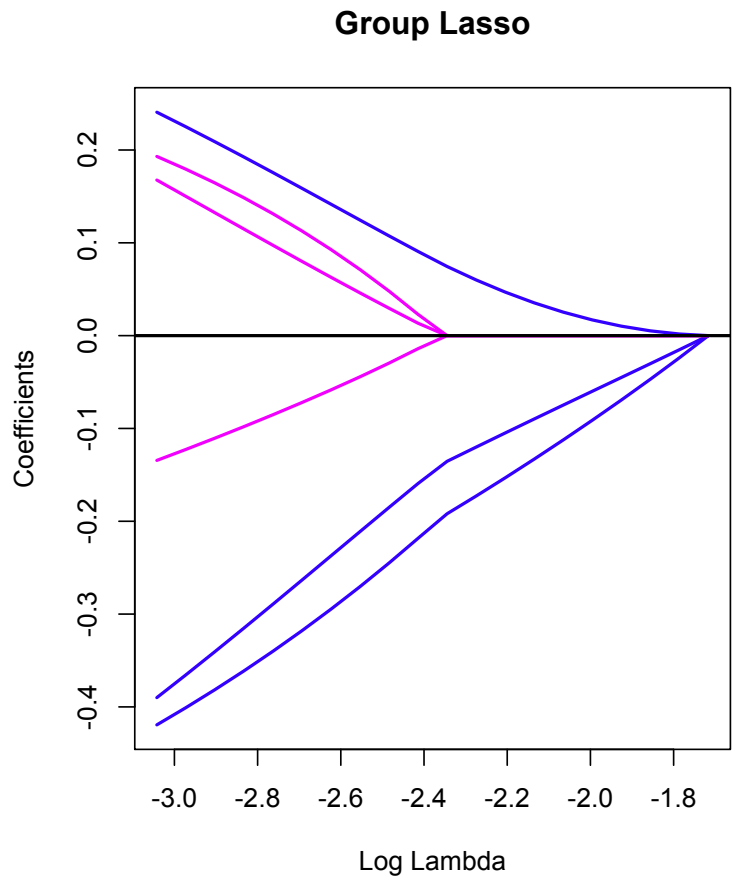
- $\gamma_k = \mathrm{eigen}_{\max}(\mathbf{H}^{(k)})$
- Update $\widetilde{\boldsymbol{\beta}}^{(k)}$ with a fast operation:

$$\widetilde{\boldsymbol{\beta}}^{(k)}(\mathrm{new}) = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda w_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right)_+$$

Lasso vs. Group Lasso

- Logistic regression with group lasso: $n = 50$, $p = 6$.
- Group lasso: specify $(\beta_1, \beta_2, \beta_3)$, $(\beta_4, \beta_5, \beta_6)$. Variable selection at the group level.
- Solution path: view $\boldsymbol{\beta}$ as function of λ.





Generalizations of the Lasso

Generalizations of the Lasso Penalty

Generalized penalties arise in a wide variety of settings:

- **Group lasso, Hierarchical group lasso:** handle structurally grouped features. e.g. dummy variables.
- **Adaptive lasso:** a lasso with the Oracle property.
- **Elastic net:** handle highly correlated features. e.g. genes.
- **SCAD and MCP:** non-convex penalties with the Oracle property.
- **Multitask lasso:** handle between-tasks sparsity while allowing within-task sparsity.

Asymptotic Properties

- Consider $y_i = \mathbf{x}_i^\top \boldsymbol{\beta}^* + \epsilon_i$, $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)$, $\epsilon_i \sim D(0, \sigma^2)$.
- $\mathcal{A}^* = \{j : \beta_j^* \neq 0\}$ – the support of $\boldsymbol{\beta}^*$
- $\mathcal{A}_n = \{j : \hat{\beta}_j \neq 0\}$ – the support of the penalized estimator $\hat{\boldsymbol{\beta}}_n = (\hat{\beta}_{n,1}, \ldots, \hat{\beta}_{n,p})$.
- **Oracle Property**: an important property that any penalized estimator $\hat{\boldsymbol{\beta}}_n$ should possess
  - *Variable selection consistency:*

$$\lim_{n \to \infty} P(\mathcal{A}_n \to \mathcal{A}^*) = 1$$

  - *$\sqrt{n}$-estimation consistency:*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{n,\mathcal{A}^*} - \boldsymbol{\beta}_{A^*}^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_0)$$

    where $\boldsymbol{\Sigma}_0$ is the covariance matrix knowing the true subset model.

Adaptive Lasso

The adaptive lasso estimator

$$\widehat{\boldsymbol{\beta}}^{\text{alasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j |\beta_j|, \tag{5}$$

where $\hat{w}_j = \frac{1}{|\hat{\beta}_j|^\gamma}$ for some $\gamma > 0$ and a $\sqrt{n}$-consistent estimator $\hat{\beta}_j$ of β_j.

- For Lasso, if an irrelevant variable is highly correlated with variables in the true model, the lasso may fail to distinguish it from the true variables even with large n.
- As $n \to \infty$, the weights corresponding to insignificant variables tend to infinity, while the weights corresponding to significant variables converge to a finite constant.
- Zou (2006) showed that, under certain regularity conditions, the adaptive lasso has the **oracle property**.

Elastic Net

The **elastic net** (Zou and Hastie, 2005) solves the convex program

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \left[\frac{1}{2}(1-\alpha)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1\right]$$

where $\alpha \in [0, 1]$ is a parameter. The penalty applied to an individual coefficient (disregarding the regularization weight $\lambda > 0$) is given by

$$\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|.$$

- The coefficients are selected approximately together in their groups.
- The coefficients approximately share their values equally.

An illustration example

- Two independent "hidden" factors $\mathbf{z}_1$ and $\mathbf{z}_2$

$$\mathbf{z}_1 \sim U(0, 20), \qquad \mathbf{z}_2 \sim U(0, 20)$$

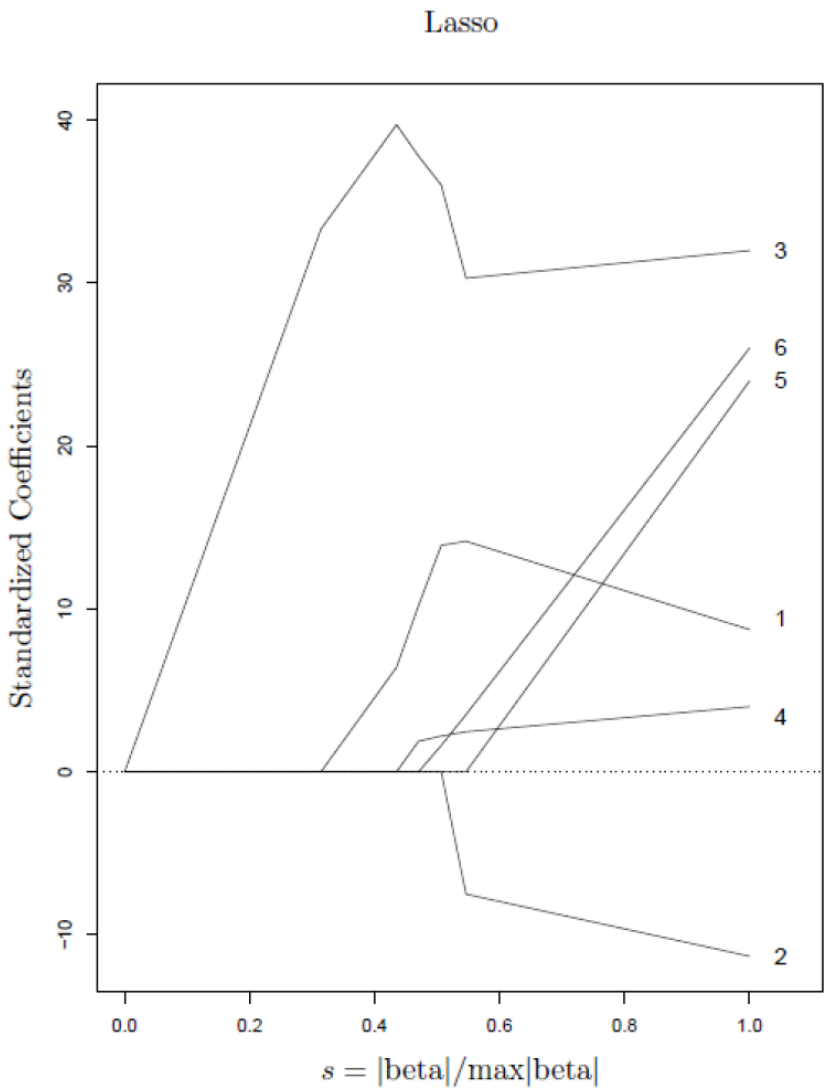
- Generate the response vector $\mathbf{y} = \mathbf{z}_1 + 0.1 \cdot \mathbf{z}_2 + N(0, 1)$
- Suppose only observe predictors

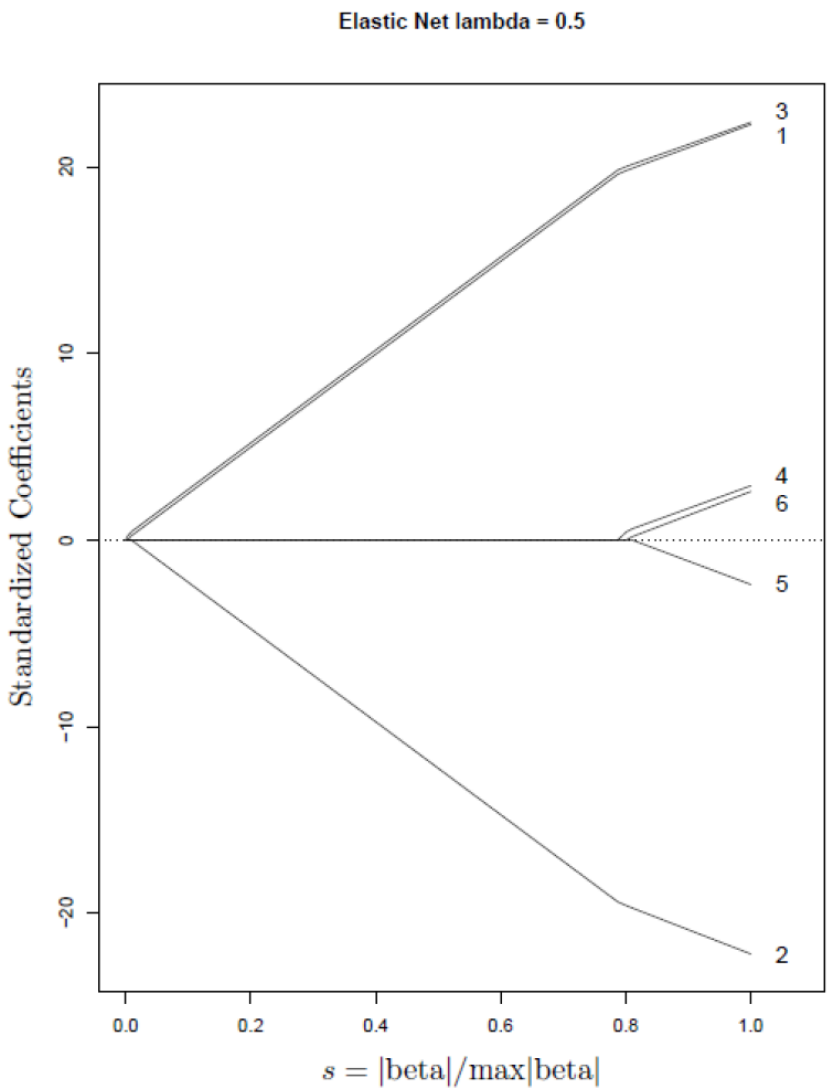
$$\mathbf{x}_1 = \mathbf{z}_1 + \epsilon_1, \quad \mathbf{x}_2 = \mathbf{z}_1 + \epsilon_2, \quad \mathbf{x}_3 = \mathbf{z}_1 + \epsilon_3$$

$$\mathbf{x}_4 = \mathbf{z}_2 + \epsilon_4, \quad \mathbf{x}_5 = \mathbf{z}_2 + \epsilon_5, \quad \mathbf{x}_6 = \mathbf{z}_2 + \epsilon_6$$

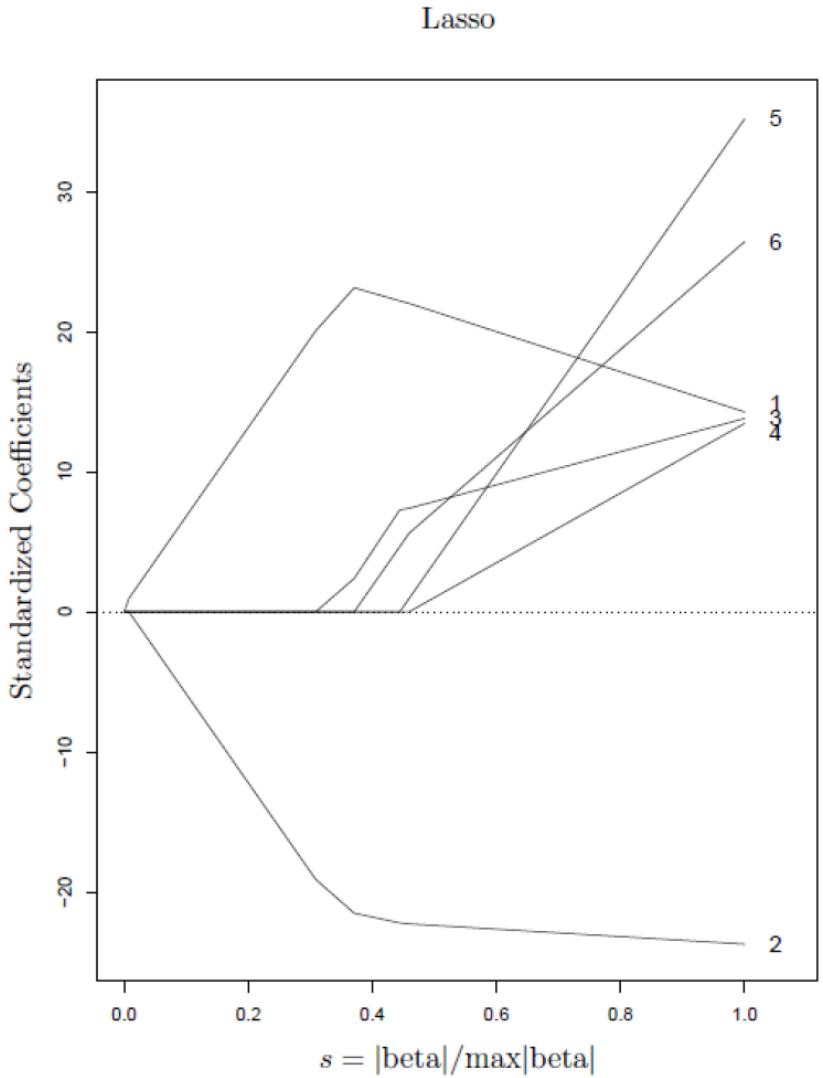
- Fit the model on $(\mathbf{X}, \mathbf{y})$
- An "oracle" would identify $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ (the $\mathbf{z}_1$ group) as the most important variables.

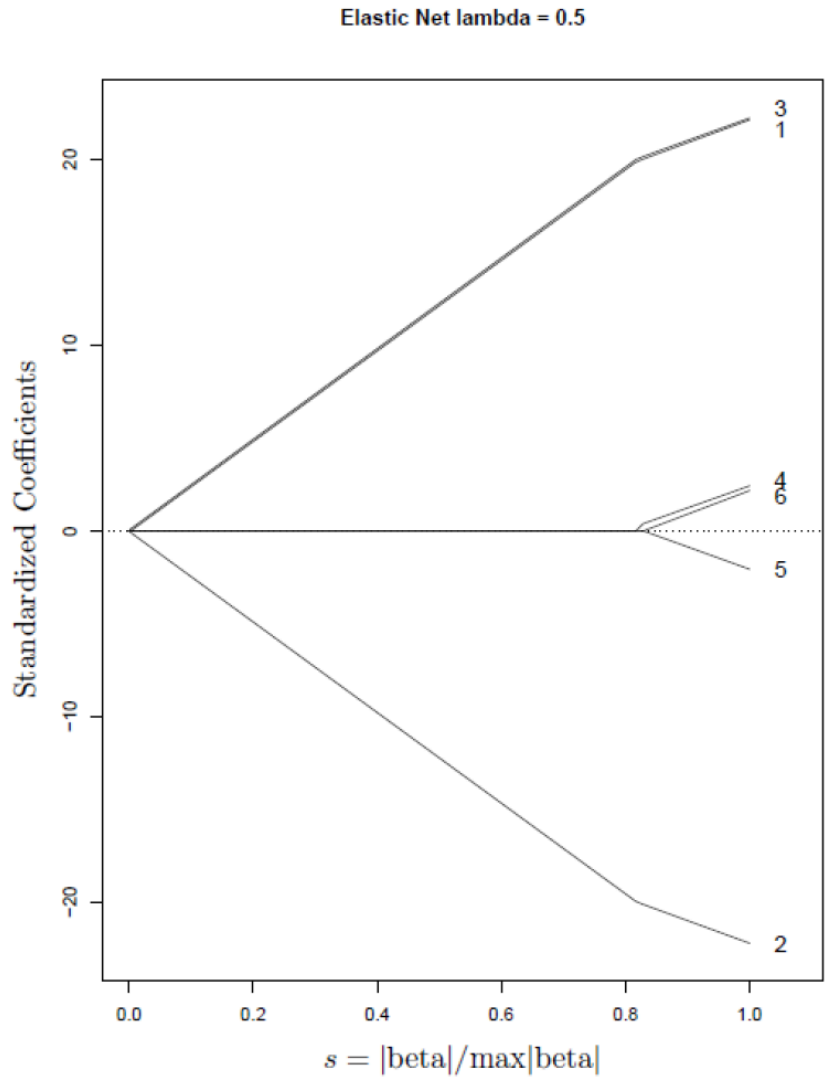
Simulation 1



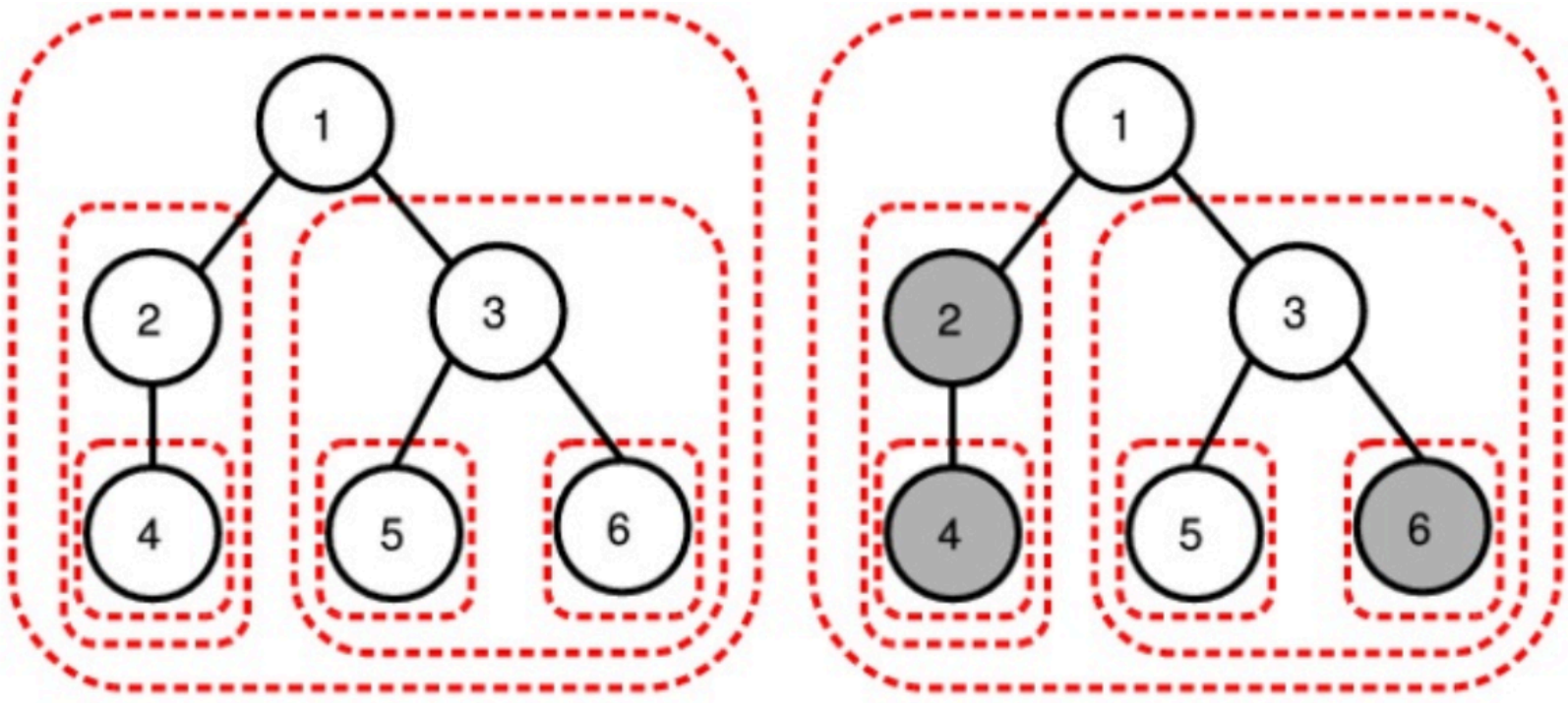


Simulation 2





Hierarchical Group Lasso



A node can be active only if its **ancestors are active**.
The selected patterns are **rooted subtrees.**

Optimization via efficient proximal methods (same cost as ℓ_1)

(Jenatton, Mairal, Obozinski, and Bach 2010)

Multitask Lasso

Suppose that we have K regression tasks

$$Y^{(k)} = \mathbf{X}^{(k)}\boldsymbol{\beta}^{(k)} + \epsilon^{(k)}, \qquad k = 1, \ldots, K.$$

- The k-th task has n_k observations for $k = 1, \ldots, K$
- $Y^{(k)} = (y_1^{(k)}, \ldots, y_{n_k}^{(k)})^\intercal$, $X_j^{(k)} = (x_{1j}^{(k)}, \ldots, x_{n_k j}^{(k)})^\top$
- $\mathbf{X}^{(k)} = (X_1^{(k)}, \ldots, X_p^{(k)})$ be the $n_k \times p$ design matrix for task k
- $\boldsymbol{\beta}^{(k)} = (\beta_1^{(k)}, \cdots, \beta_p^{(k)})^\top$ and $\boldsymbol{\beta}_j = (\beta_j^{(1)}, \cdots, \beta_j^{(K)})^\top$
- find commonly shared relevant covariates and retains the ability to recover covariates unique to individual data sources.

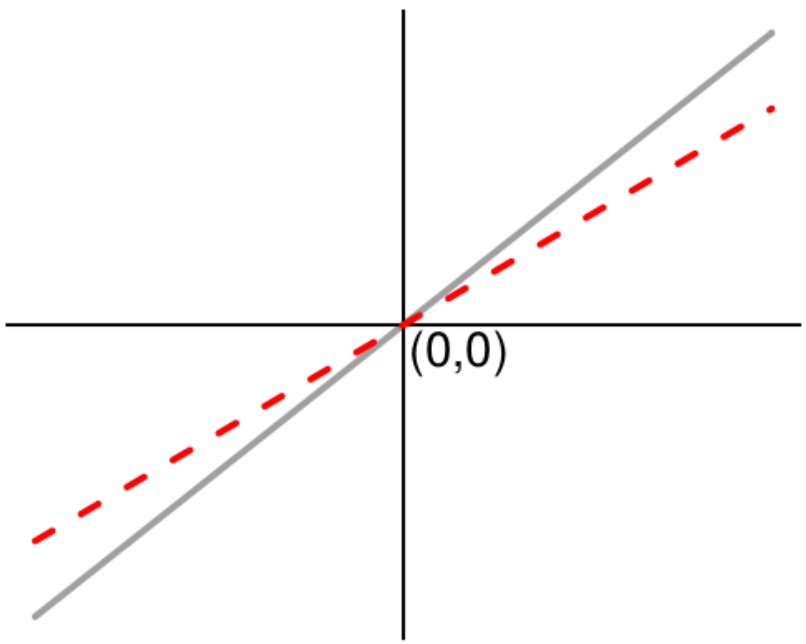
$$\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{k=1}^{K} \left\| Y^{(k)} - \mathbf{X}^{(k)}\boldsymbol{\beta}^{(k)} \right\|^2 + \lambda P_\alpha(\boldsymbol{\beta}),$$

$$P_\alpha(\boldsymbol{\beta}) = \sum_{j=1}^{p} w_j \left[(1-\alpha)||\boldsymbol{\beta}_j||_q + \alpha||\boldsymbol{\beta}_j||_1 \right]$$

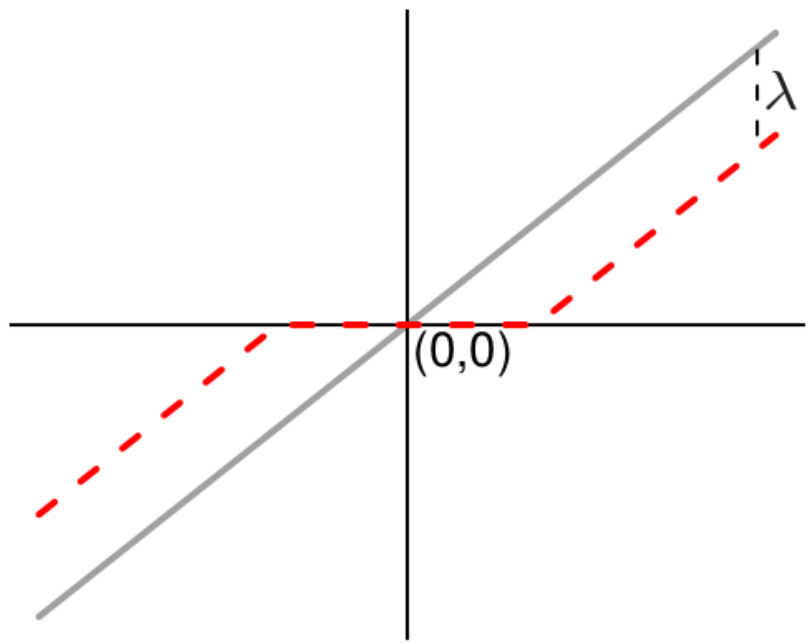
Recall the bias of the lasso

q	Estimator	Formula
1	Lasso	$\mathbf{sign}(\widehat{\beta}_j^{LS})(\lvert\widehat{\beta}_j^{LS}\rvert - \lambda)_+$
2	Ridge	$\widehat{\beta}_j^{LS}/(1+\lambda)$

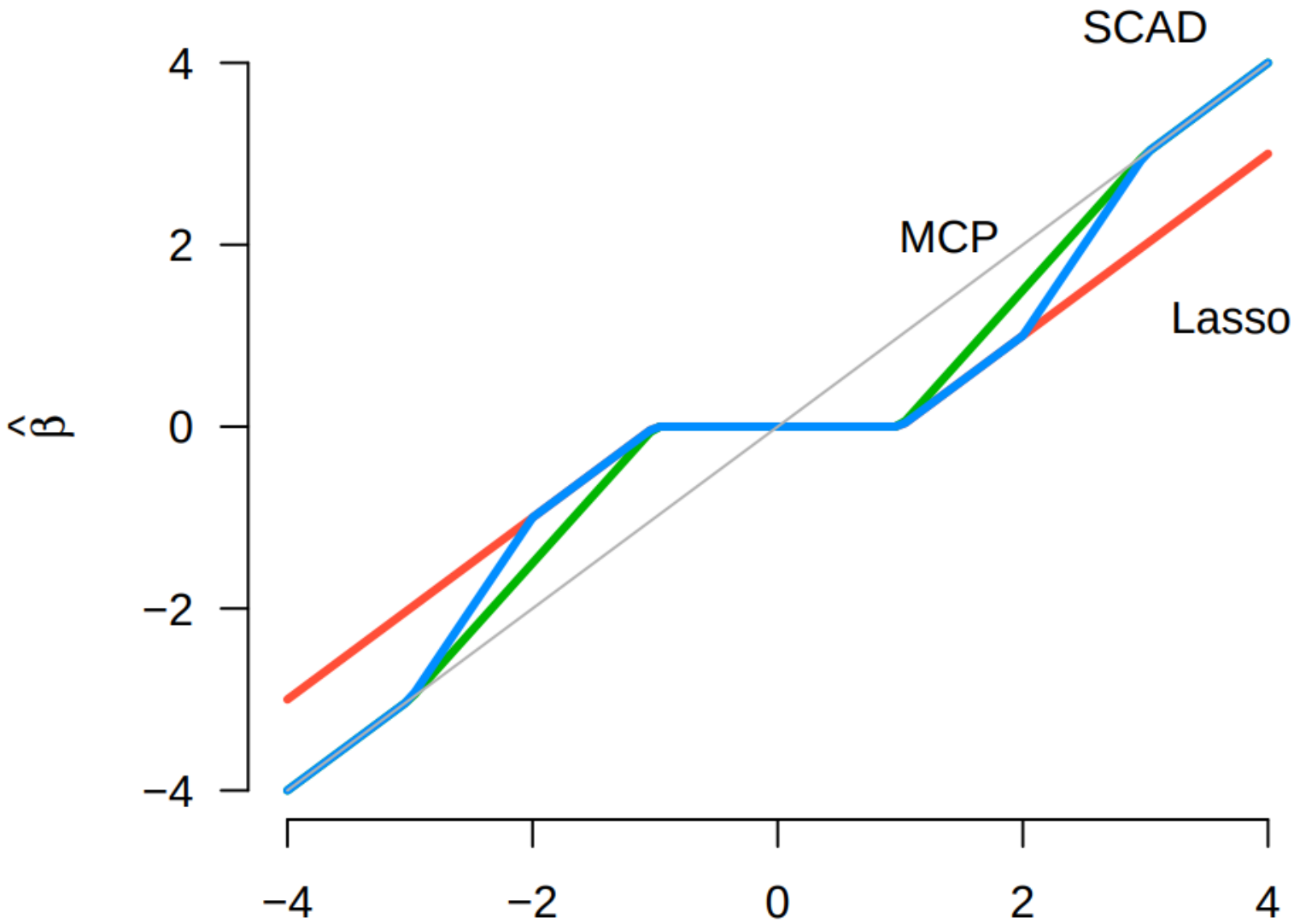
Ridge



Lasso



SCAD (Fan et Li, JASA, 2001), MCP (Zhang, Ann. Stat., 2010)



Discussion

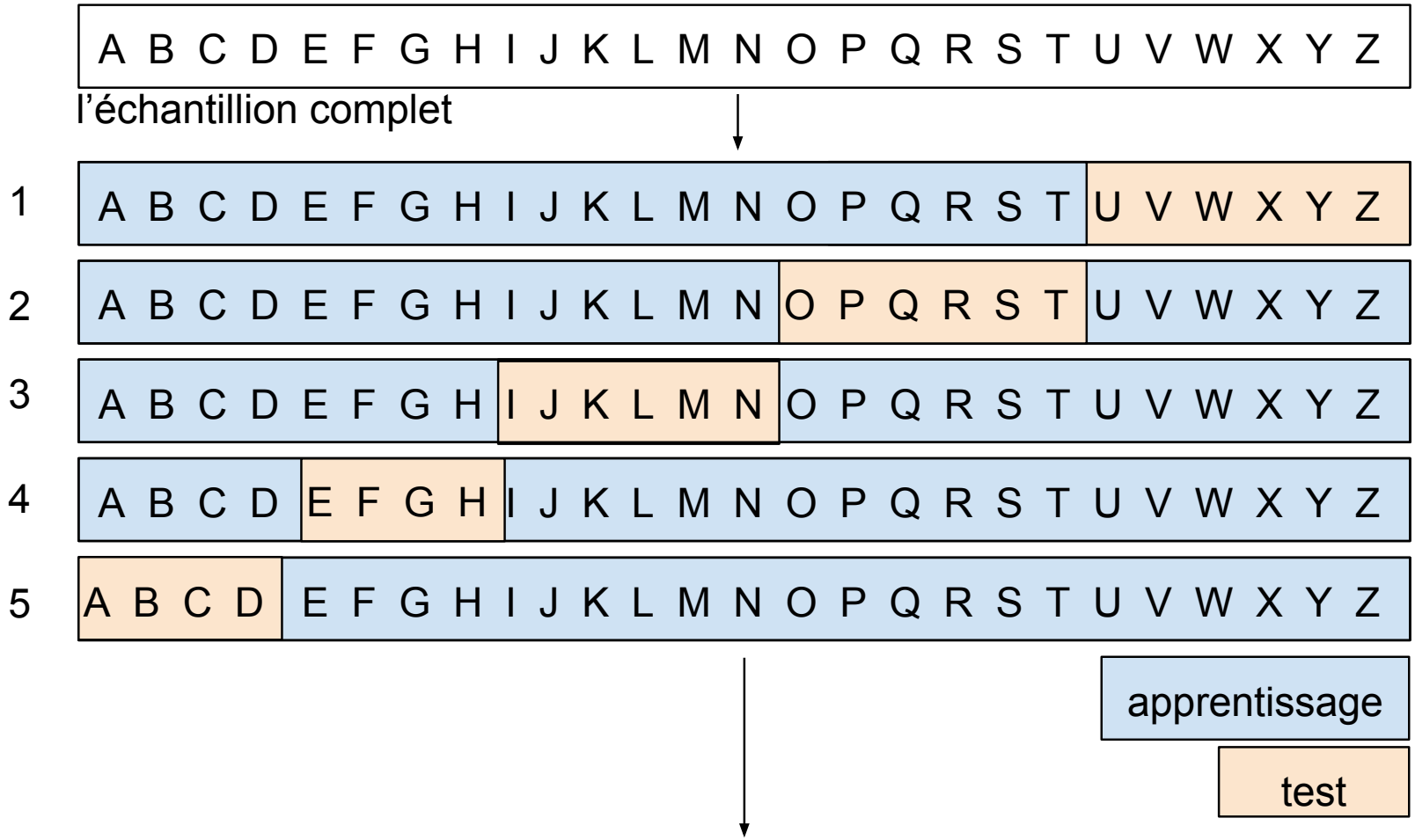
- Variable selection is an active area of research
- Few inference tools exist
- Robust software has been developed, but more scalable algorithms and implementations are needed

References

- Fan, J. and Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, 96(456), pp.1348-1360.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), pp.267-288.
- Friedman, J., Hastie, T., Höfling, H. and Tibshirani, R., 2007. Pathwise coordinate optimization. The annals of applied statistics, 1(2), pp.302-332.
- Buhlmann, P. & van de Geer, S. (2011), Statistics for High-Dimensional Data, Springer.
- Breheny, P. BIOS 7240 class notes (accessed March 15, 2019).
- Tibshirani, R. A Closer Look at Sparse Regression (accessed March 15, 2019).
- Gaillard, P. and Rudi, A. Introduction to Machine Learning (accessed March 15, 2019).
- Hastie, T., Tibshirani, R. & Friedman, J. (2009), The Elements of Statistical Learning; Data Mining, Inference and Prediction, Springer. Second edition.
- Hastie, T., Tibshirani, R. & Wainwright, M. (2015), Statistical Learning with Sparsity: the Lasso and Generalizations, Chapman & Hall.

slides available at
https://sahirbhatnagar.com/talks/

Contexte sur la validation croisée



$$CV(\alpha) = \frac{1}{5}\sum_{v=1}^{5} MSE_v^{(test)}$$

SCAD

$$p'(|\beta|;\lambda) = \lambda sign(\beta_j)\left\{I_{(|\beta_j|\leq\lambda)} + \frac{(a\lambda - |\beta_j|)_+}{(a-1)\lambda} I_{(|\beta_j|>\lambda)}\right\}, \quad a > 2$$

The penalty is expressed in terms of its derivative. The SCAD is a combination of the HARD, LASSO, and Clipped penalties. This leads to the solution

$$\widehat{\boldsymbol{\beta}}_{j,SCAD} = \begin{cases} sign(\widehat{\boldsymbol{\beta}}_{j,OLS})(|\widehat{\boldsymbol{\beta}}_{j,OLS}| - \lambda)_+ & |\widehat{\boldsymbol{\beta}}_{j,OLS}| \leq 2\lambda \\ \frac{(a-1)\widehat{\boldsymbol{\beta}}_{j,OLS} - sign(\widehat{\boldsymbol{\beta}}_{j,OLS})a\lambda}{a-2} & 2\lambda < |\widehat{\boldsymbol{\beta}}_{j,OLS}| \leq a\lambda \\ \widehat{\boldsymbol{\beta}}_{j,OLS} & |\widehat{\boldsymbol{\beta}}_{j,OLS}| > a\lambda \end{cases}$$

MCP, Zhang (2010)

$$p(|\beta|_j; \lambda, \gamma) = \begin{cases} \lambda|\beta_j| - \frac{|\beta_j|^2}{2\gamma} & |\beta_j| \leq \gamma\lambda \\ \frac{\gamma\lambda^2}{2} & |\beta_j| > \gamma\lambda \end{cases}$$