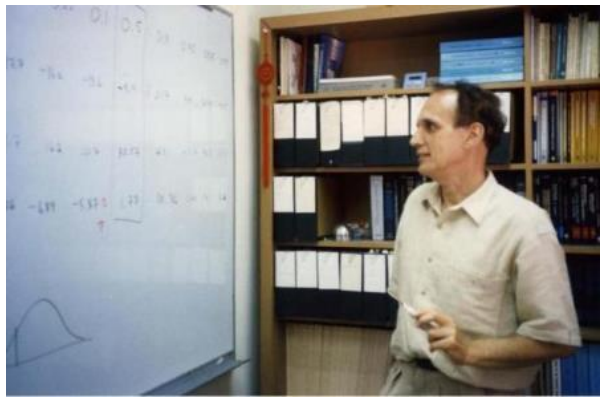
Prediction, Estimation, and Attribution

Statistical Learning

CLAMSES - University of Milano-Bicocca

Aldo Solari



Bradley Efron working in his classic office, circa 1996.

References

This material reproduces the following

- Efron, B. (2020). Prediction, Estimation, and Attribution. Journal of the American Statistical Association, 115(530), 636-655. With Discussion and Rejoinder.
- Efron's Slides
- Recorded presentation for the 62nd ISI World Statistics Congress in Kuala Lumpur [46 mins]

Table of Contents

Regression

Gauss (1809), Galton (1877)

- *Prediction: the prediction of new cases*
  e.g. random forests, boosting, support vector machines, neural nets, deep learning
- *Estimation: the estimation of regression surfaces*
  e.g. OLS, logistic regression, GLM (MLE)
- *Attribution: the assignment of significance to individual predictors*
  e.g. Fisher's ANOVA, Neyman-Pearson

How do the pure prediction algorithms relate to traditional regression methods?

That is the central question pursued in what follows.

Table of Contents

We will assume that the data $\mathcal{D}$ available to the statistician has this structure:

$$\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$$

- x_i is a p-dimensional vector of predictors taking its value in a known space $\mathcal{X}$ contained in $\mathbb{R}^p$;
- y_i is a real valued response;
- the n pairs are assumed to be independent of each other.

More concisely we can write

$$\mathcal{D} = \{X, y\}$$

where X is the $n \times p$ matrix having x_i^t as the ith row, and $y = (y_1, \ldots, y_n)^t$.

Regression surface

- The regression model is

$$y_i = s(x_i, \beta) + \epsilon_i \quad i = 1, \ldots, n \tag{1}$$

  $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$ where $s(x, \beta)$ is some functional form that, for any fixed value of the parameter vector β, gives expectation $\mu = s(x, \beta)$ as a function of $x \in \mathcal{X}$;
- The *regression surface* is

$$\mathcal{S} = \{s(x, \beta), x \in \mathcal{X}\}$$

  Most traditional regression methods depend on some sort of surface plus noise formulation;
- The surface describes the scientific truths we wish to learn, but we can only observe points on the surface obscured by noise;
- The statistician's traditional estimation task is to learn as much as possible about the surface from the data $\mathcal{D}$.

The left panel of the Figure shows the surface representation of Newton's second law of motion,

acceleration = force / mass

The right panel shows a picture of what experimental data might have looked like.

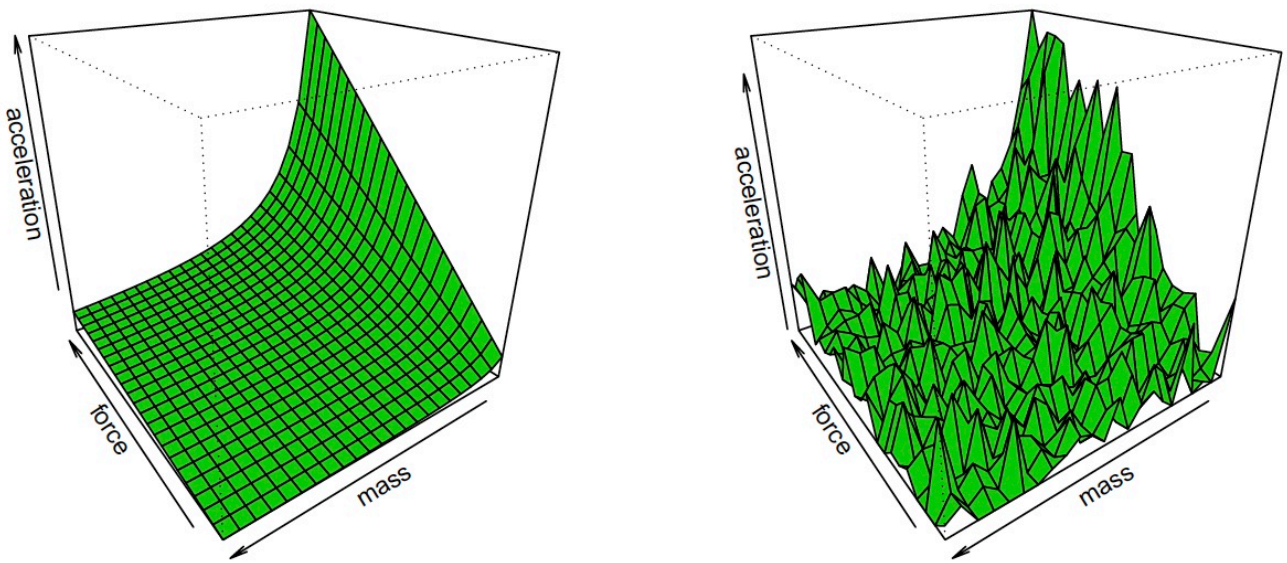


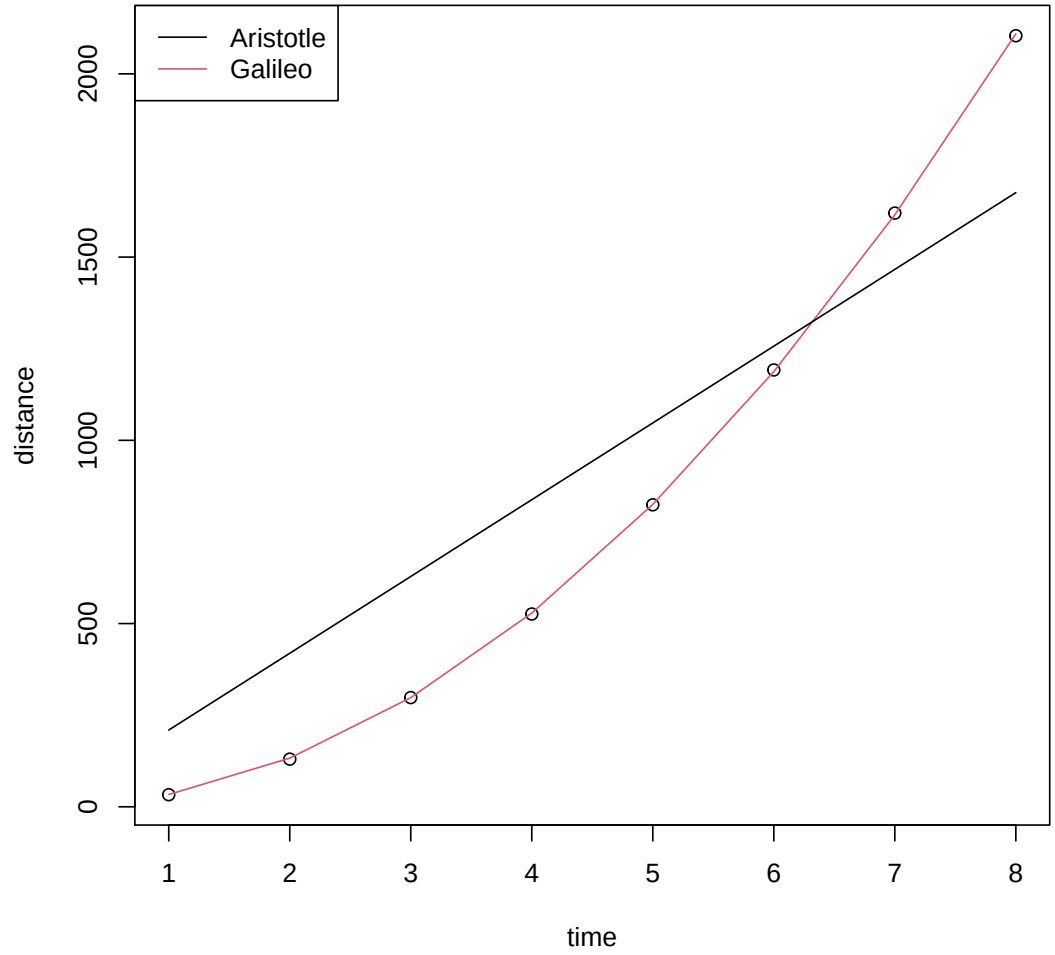
Figure 2. On left, a surface depicting Newton's second law of motion, acceleration = force/mass; on right, a noisy version.

Galileo's inclined plane experiment (1604)

- If a ball rolls down a ramp, what is the relationship between time (x) and distance (y)?
- Aristotle: Constant velocity (zero acceleration): distance $\propto$ time
- Galileo : Increasing velocity (constant acceleration): distance $\propto$ time^2
- Experimental data:

time	1	2	3	4	5	6	7	8
distance	33	130	298	526	824	1192	1620	2104

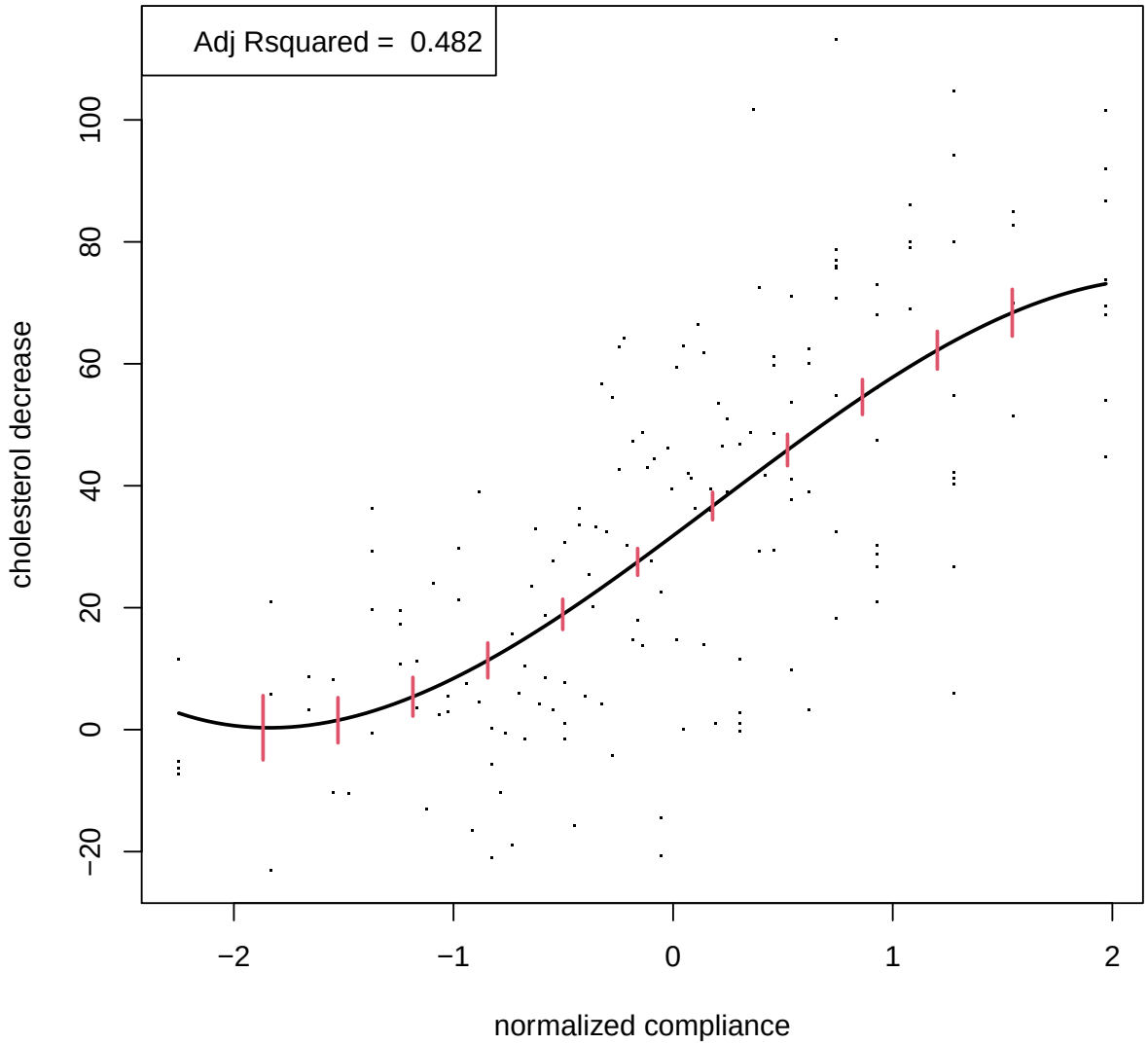
MacDougal, D. W. (2012). Galileo's Great Discovery: How Things Fall. In Newton's Gravity (pp. 17-36). Springer



https://github.com/aldosolari/SL/blob/master/docs/RCODE/EfronPEA.R

Cholesterol data

- Cholestyramine, a proposed cholesterol lowering drug, was administered to 164 male doctors for an average of seven years each (Efron and Feldman, 1991)
- The response variable (y_i) is a man's decrease in cholesterol level over the course of the experiment.
- The single predictor is compliance (x_i), the fraction of intended dose actually taken. Compliance, the proportion of the intended dose actually taken, ranged from 0% to 100%, −2.25 to 1.97 on the normalized scale. It was hoped to see larger cholesterol decreases for the better compliers.
- https://hastie.su.domains/CASI_files/DATA/cholesterol.html


Adj Rsquared = 0.482
cholesterol decrease
100
80
60
40
20
0
−20
−2
−1
0
1
2
normalized compliance

- A normal regression model was fit, with

$$s(x_i, \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3$$

  in other words, a cubic regression model.
- The black curve is the estimated surface

$$\hat{\mathcal{S}} = \{s(x, \hat{\beta}), x \in \mathcal{X}\}$$

  fit by maximum likelihood or, equivalently, by ordinary least squares (OLS).
- The vertical bars indicate one standard error for the estimated values $s(x, \hat{\beta})$, at 11 choices of x, showing how inaccurate $\hat{\mathcal{S}}$ might be as an estimate of the true $\mathcal{S}$
- Only $\hat{\beta}_0$ and $\hat{\beta}_1$ were significantly nonzero. The adjusted R^2 was 0.482, a traditional measure of the model's predictive power.

Table of Contents

- Random Forests, Boosting, Deep Learning, etc.
- Data
$$\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$$
- Prediction rule $f(x, \mathcal{D})$
- New $(x, ?)$ gives $\hat{y} = f(x, \mathcal{D})$
- Strategy: Go directly for high predictive accuracy; forget (mostly) about surface + noise

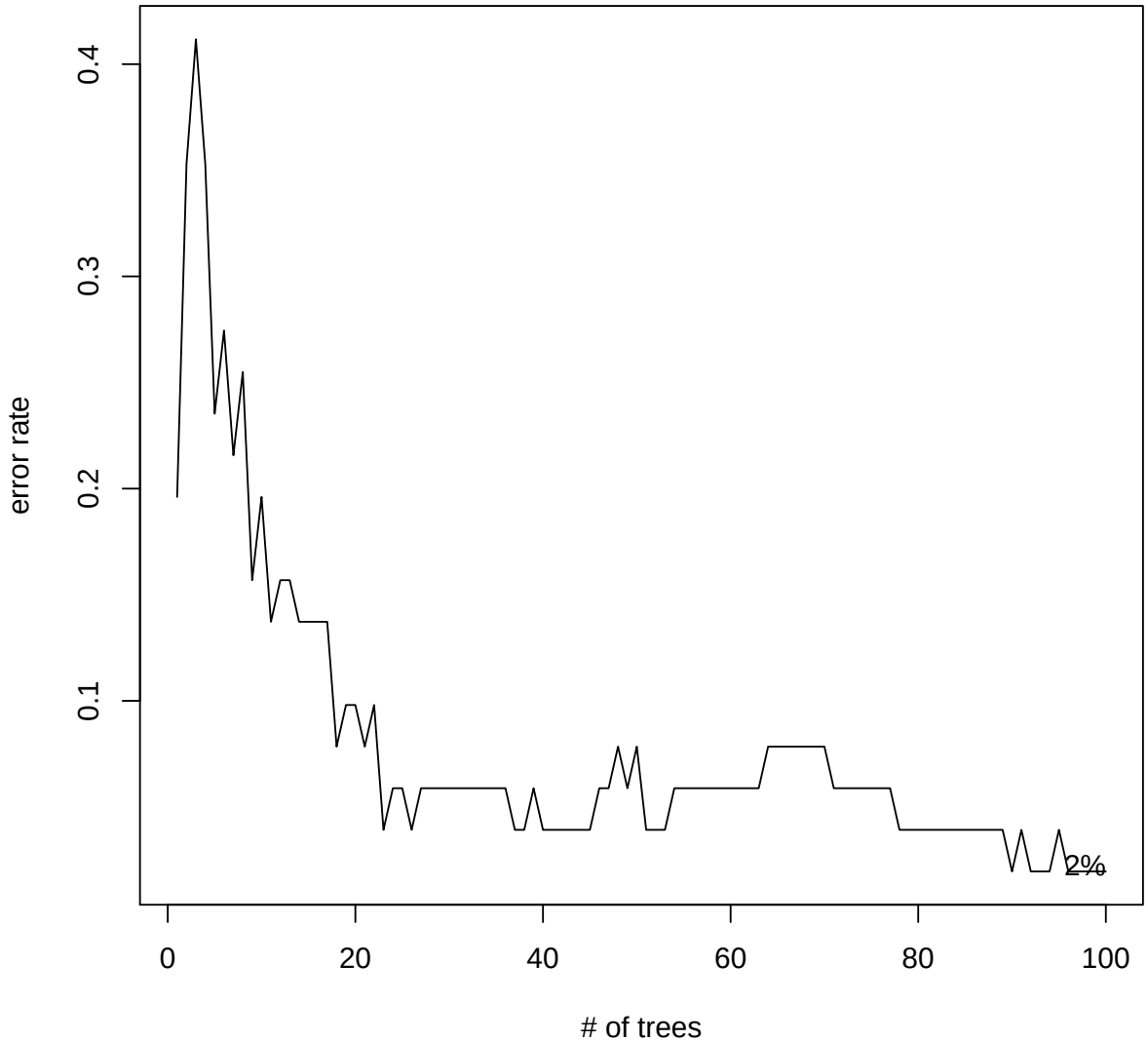
Table of Contents

The Prostate Cancer Microarray Study

- https://hastie.su.domains/CASI_files/DATA/prostate.html
- $n = 102$ men: 52 prostate cancer, 50 normal controls
- For each man measure activity of $p = 6033$ genes
- Data set D is 102×6033 matrix ("wide")
- Wanted: Prediction rule $f(x, \mathcal{D})$ that inputs new 6033-vector x and outputs $\hat{y}$ correctly predicting cancer/normal

Random forest

- Randomly divide the 102 subjects into:
  - training set of 51 subjects (26 + 25)
  - test set of 51 subjects (26 + 25)
- Run R program `randomForest` on the training set
- Use its rule $f(x_i, D)$ on the test set and see how many errors it makes


error rate
0.1
0.2
0.3
0.4
0
20
40
60
80
100
# of trees
2%

Boosting

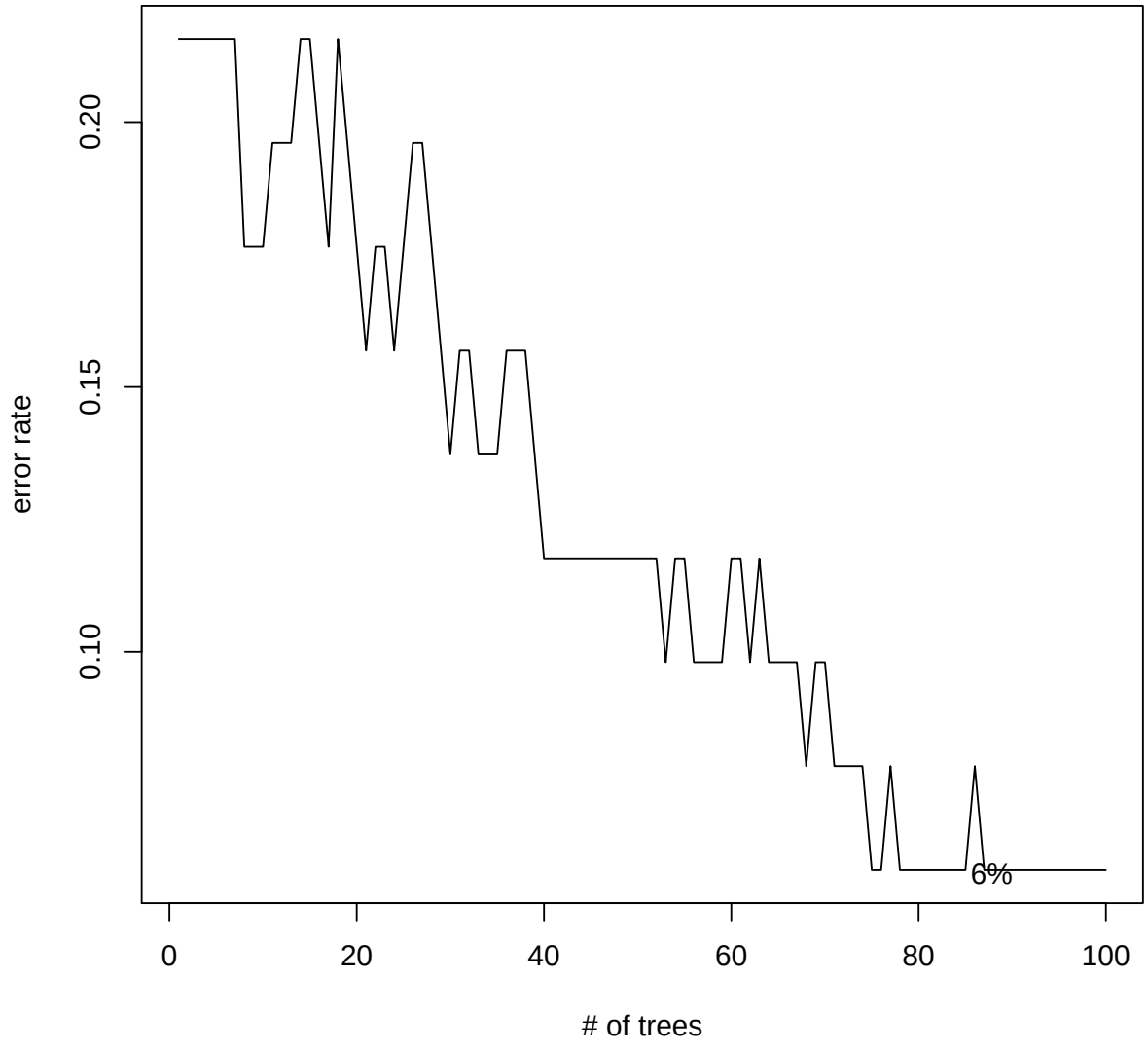
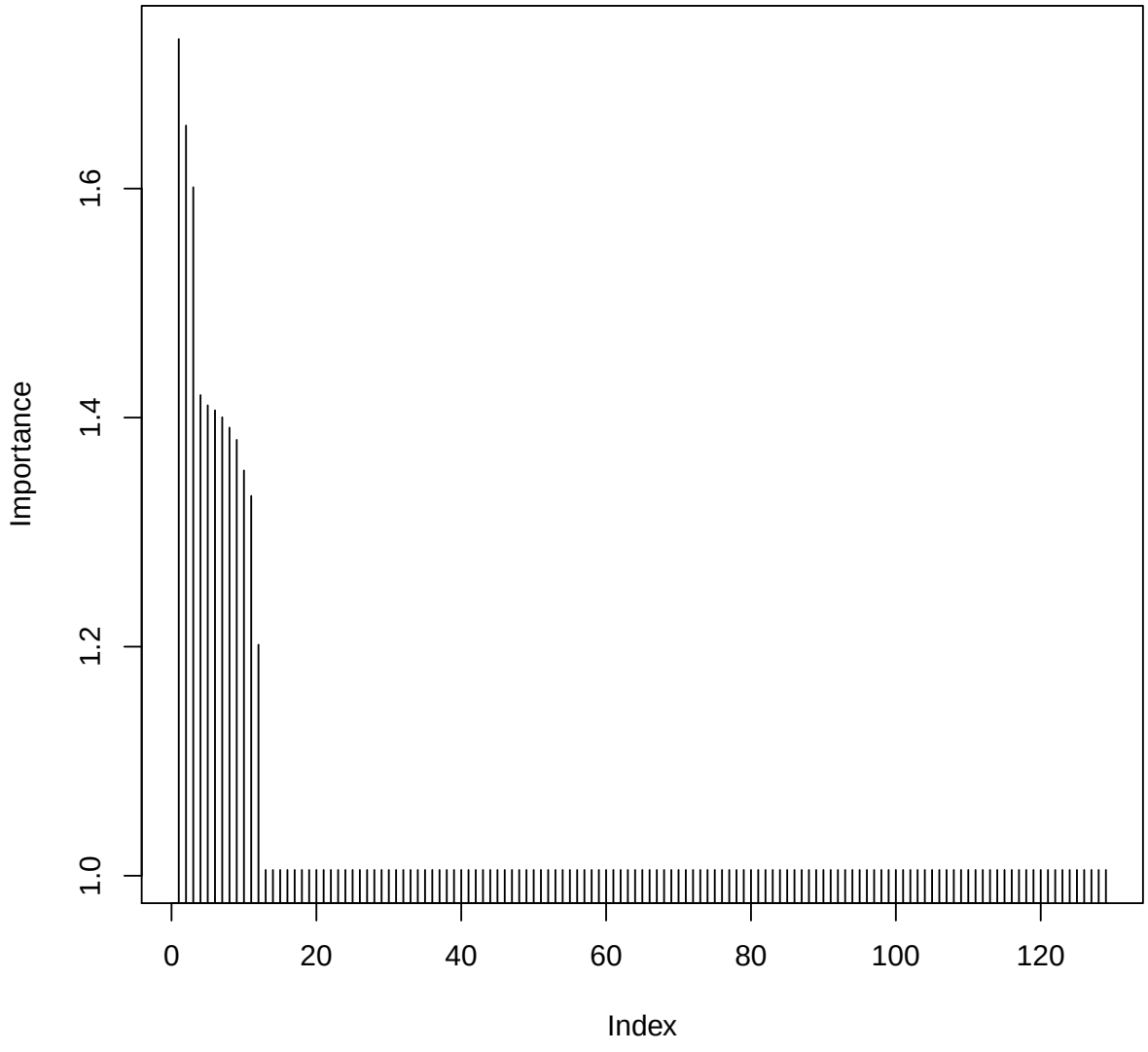


Table of Contents

Variable importance

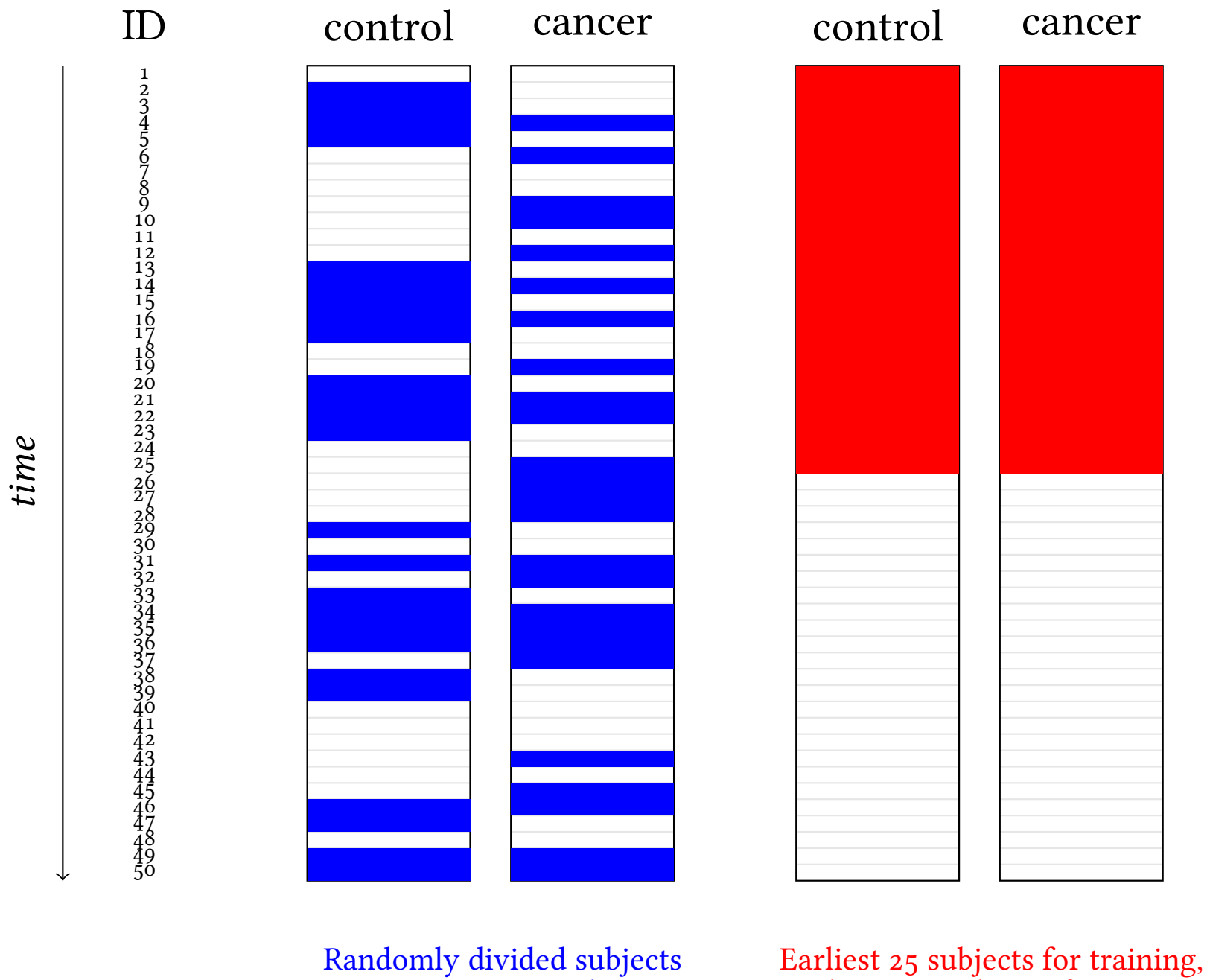


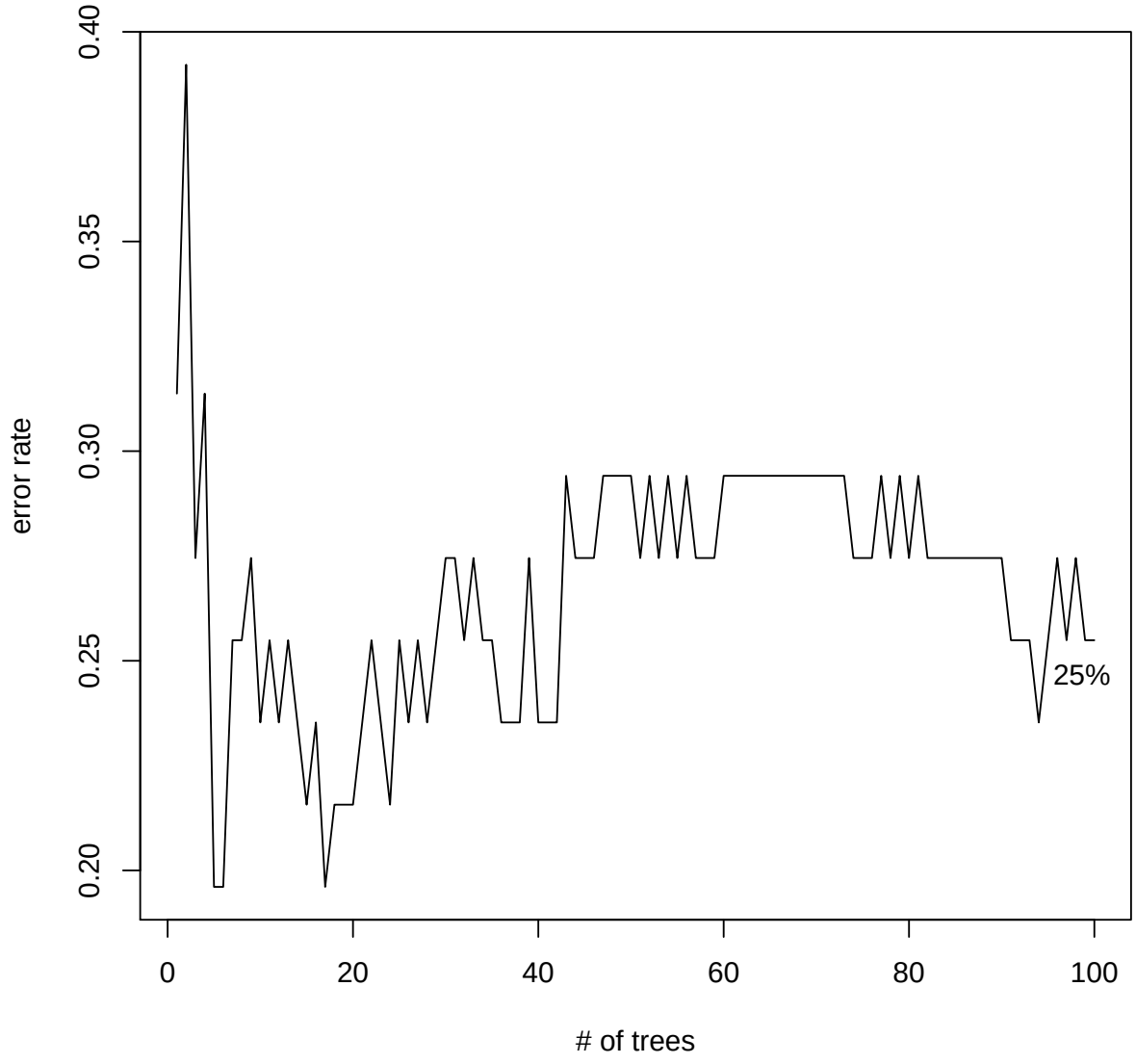
- Importance measure is computed for each of the p predictor variables.
- Of the $p = 6033$ genes, 129 had positive scores, these being the genes that ever were chosen as splitting variables.
- Can we use the importance scores for attribution?
- The answer seems to be no. Removing the most important 100 had similarly minor effects on the number of test set prediction errors
- Evidently there are a great many genes weakly correlated with prostate cancer, which can be combined in different combinations to give near-perfect predictions.

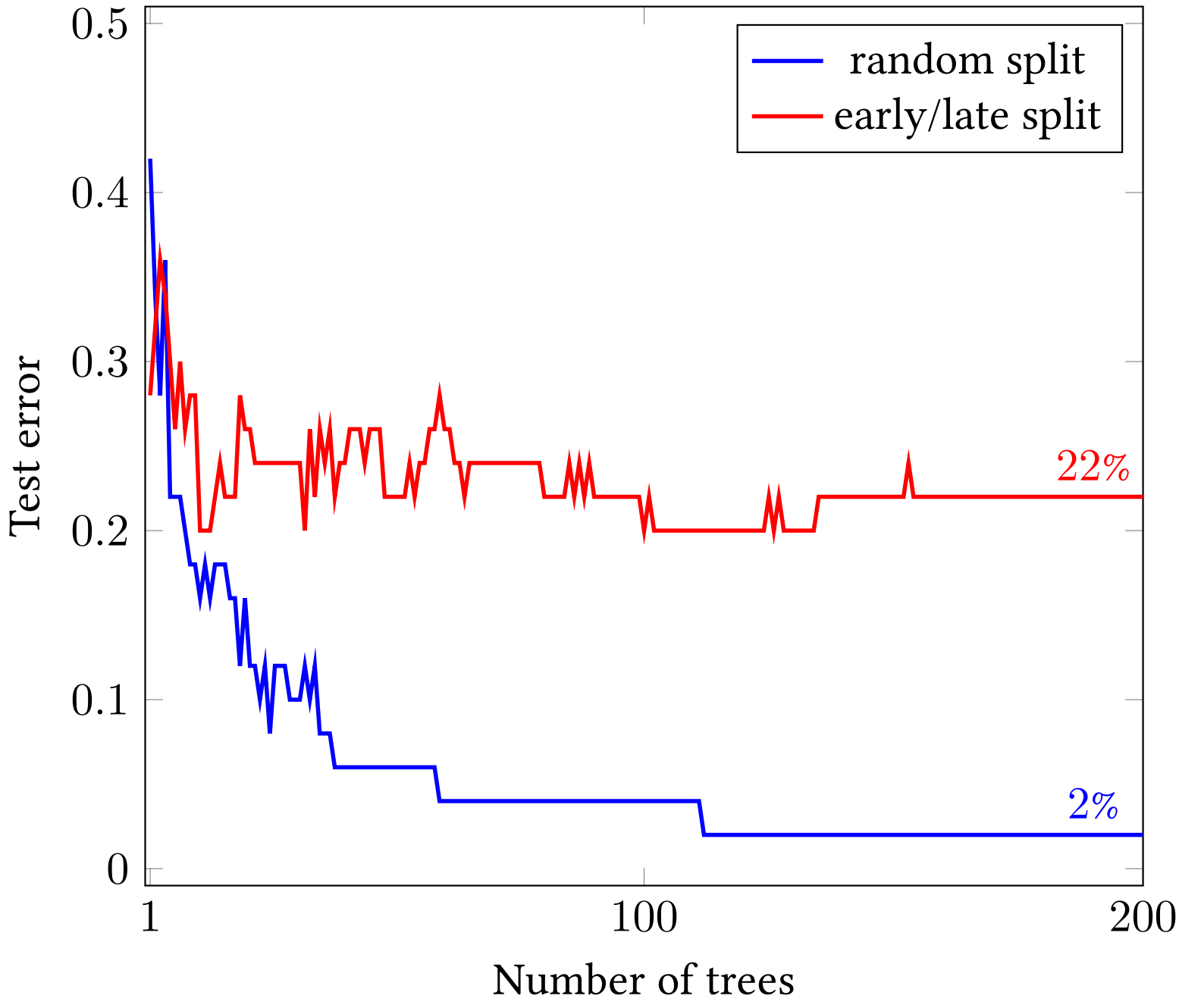
Table of Contents

Were the Test Sets Really a Good Test?

- Prediction can be highly context-dependent and fragile
- Before Randomly divided subjects into training and test
- Next:
  - 51 earliest subjects for training (25 control + 26 cancer with lowest ID numbers)
  - 51 latest subjects for test
- Study subjects might have been collected in the order listed, with some small methodological differences creeping in as time progressed (concept drift)


ID
control
cancer
control
cancer
time
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
Randomly divided subjects into training and test
Earliest 25 subjects for training, latest 25 subjects for test


error rate
0.20
0.25
0.30
0.35
0.40
0
20
40
60
80
100
# of trees
25%


0.5
0.4
0.3
0.2
0.1
0
Test error
1
100
200
Number of trees
random split
early/late split
22%
2%

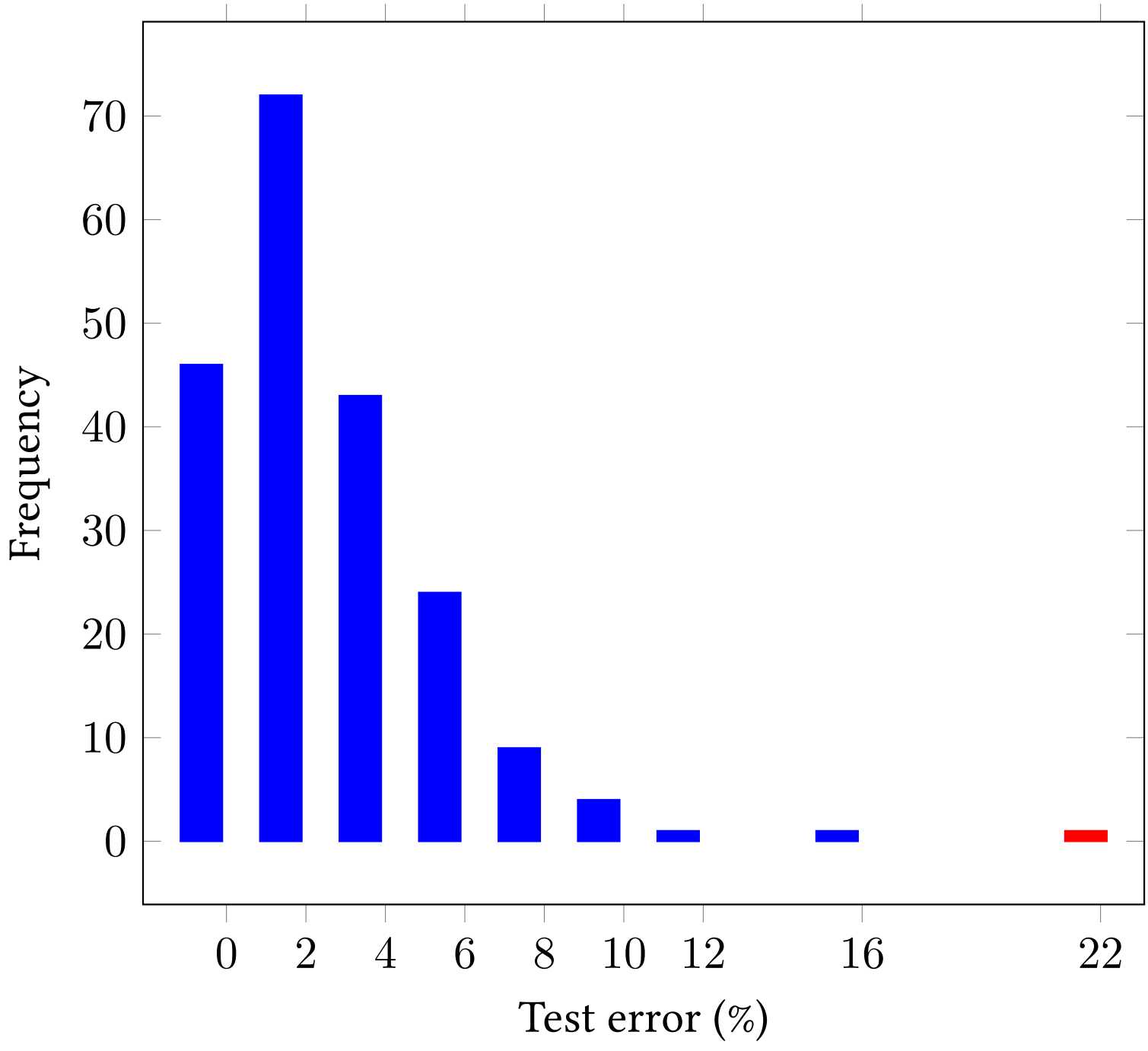
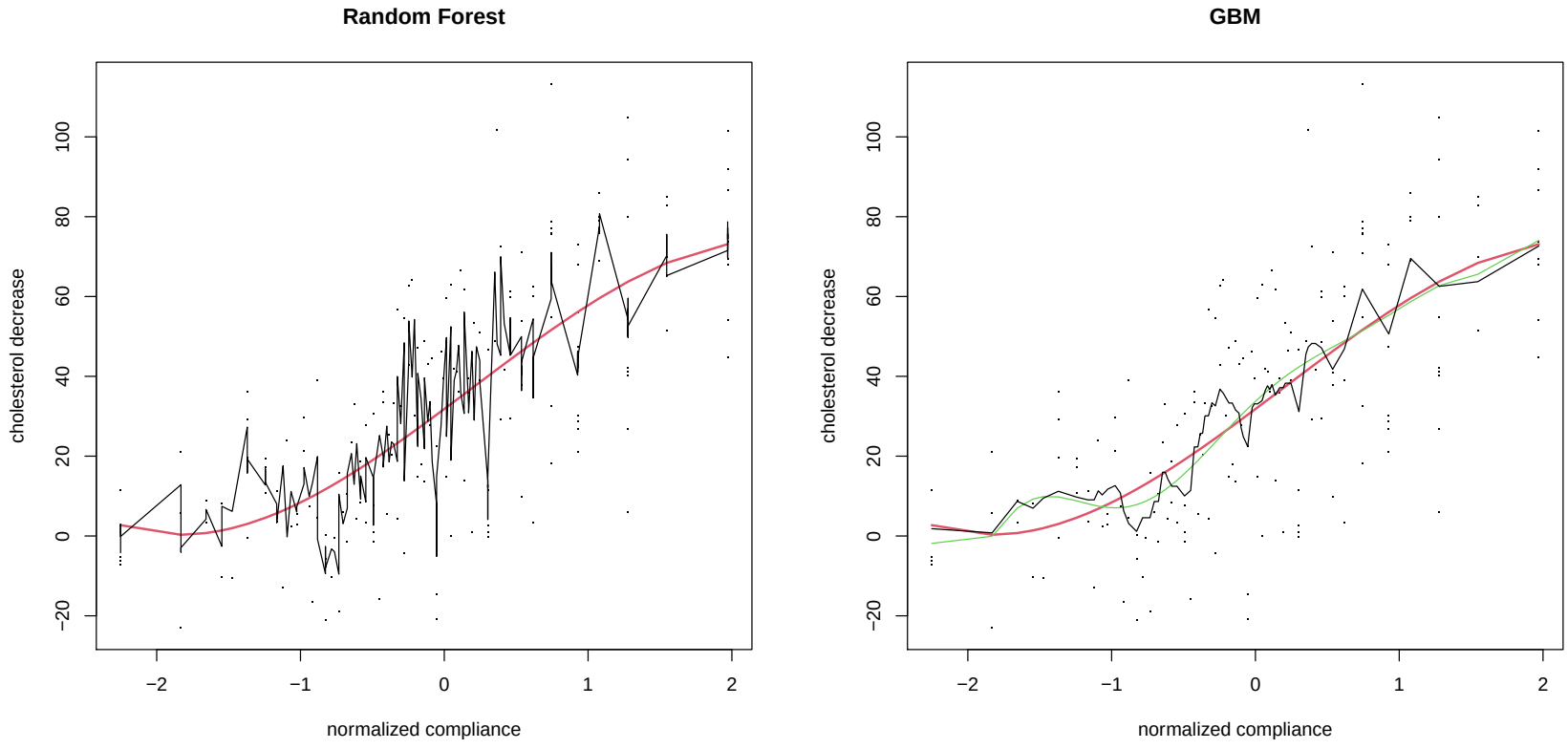

Frequency
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
16
22
Test error (%)

Table of Contents

- The parametric models of traditional statistical methodology enforce the smooth-world paradigm
- Looking back at the Cholesterol data, we might not agree with the exact shape of the cholostyramine cubic regression curve but the smoothness of the response seems unarguable
- The choice of cubic was made on the basis of a Cp comparison of polynomial regressions degrees 1 through 8, with cubic best.
- Smoothness of response is not built into the pure prediction algorithms.
- Random forest and algorithm `gbm` take X to be the 164×8 matrix `poly(c,8)` - an 8th degree polynomial basis



`randomForest` and `gbm` fits to the Cholesterol data. Heavy red curve is cubic OLS; dashed green curve in right panel is 8th degree OLS fit.

Table of Contents

	Traditional regressions methods	Pure prediction algorithms
1.	Surface plus noise models (continuous, smooth)	Direct prediction (possibly discrete, jagged)
2.	Scientific truth (long-term)	Empirical prediction accuracy (possibly short-term)
3.	Parametric modeling (causality)	Nonparametric (black box)
4.	Parsimonious modeling (researchers choose covariates)	Anti-parsimony (algorithm chooses predictors)
5.	X $n \times p$ with $p \ll n$ (homogeneous data)	$p \gg n$, both possibly enormous (mixed data)
6.	Theory of optimal inference (mle, Neyman–Pearson)	Training/test paradigm (Common Task Framework)

Table of Contents

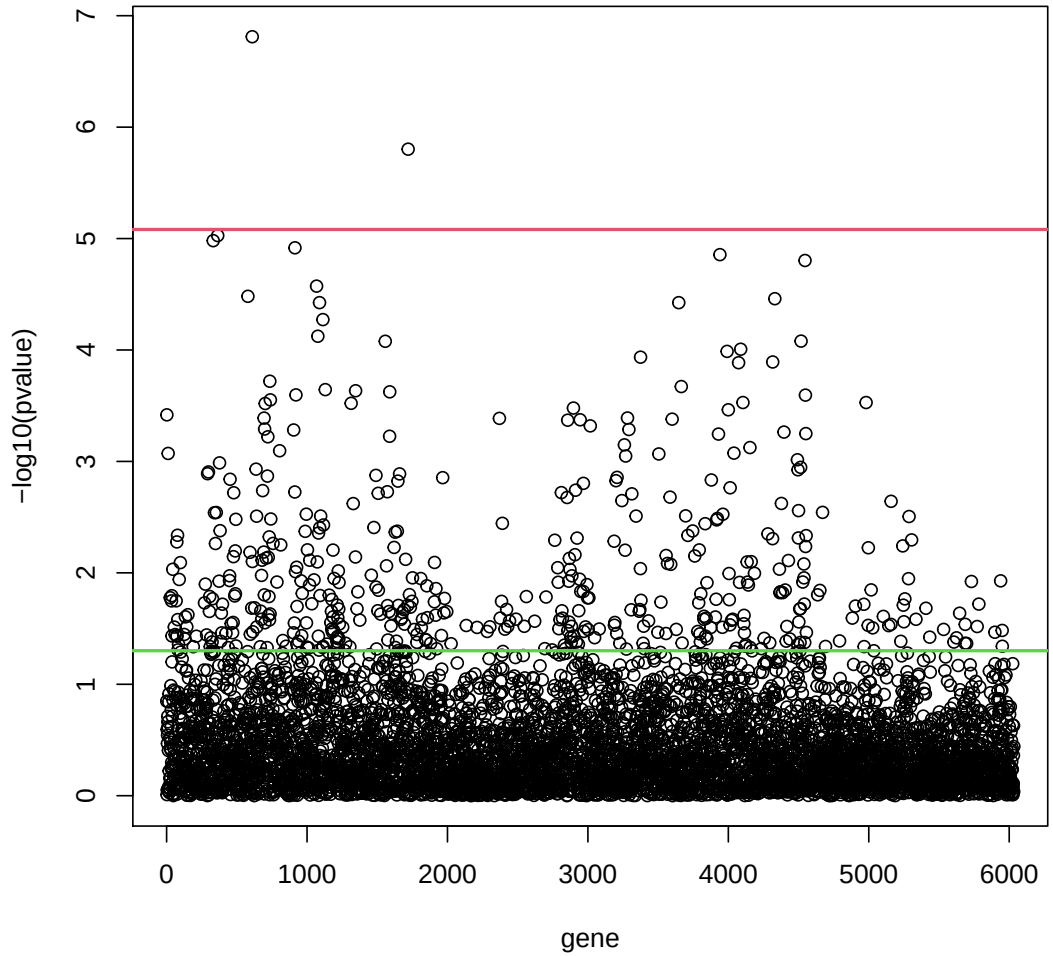
Estimation and Attribution in the Wide-Data Era

- Large p (the number of features) affects Estimation
  - MLE can be badly biased for individual parameters
  - "surface" if, say, $p = 6033$?
- Attribution still of interest. Compute p-value p_i for the null hypothesis H_i: no difference in gene expression between cancer and control at the ith gene
- The Bonferroni threshold for 0.05 significance is

$$p_i \leq 0.05/6033$$

$$\begin{aligned} \Pr(\text{at least one Type I error}) &= \Pr\left(\bigcup_{i \in I_0} \{p_i \leq \alpha/p\}\right) \\ &\leq \sum_{i \in I_0} \mathrm{P}(p_i \leq \alpha/p) \leq |I_0| \frac{\alpha}{p} \leq \alpha \end{aligned}$$

- Instead of performing a traditional attribution analysis with $p = 6033$ predictors, a microarray analysis performs 6033 analyses with $p = 1$


−log10(pvalue)
0
1
2
3
4
5
6
7
0
1000
2000
3000
4000
5000
6000
gene

- Sparsity offers another approach to wide-data estimation and attribution: we assume that most of the p predictor variables have no effect and concentrate effort on finding the few important ones.
- The lasso provides a key methodology. Estimate β, the p-vector of regression coefficients, by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^t\beta) + \lambda\|\beta\|_1$$

  where $\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$
- Here λ is a fixed tuning parameter: $\lambda = 0$ corresponds to the OLS solution for β (if $p \leq n$) while $\lambda = \infty$ makes $\hat{\beta} = 0$. For large values of λ only a few of the coordinates $\hat{\beta}_j$ will be nonzero.
- The lasso produced biased estimates of β, with the coordinate values $\hat{\beta}_j$ shrunk toward zero.

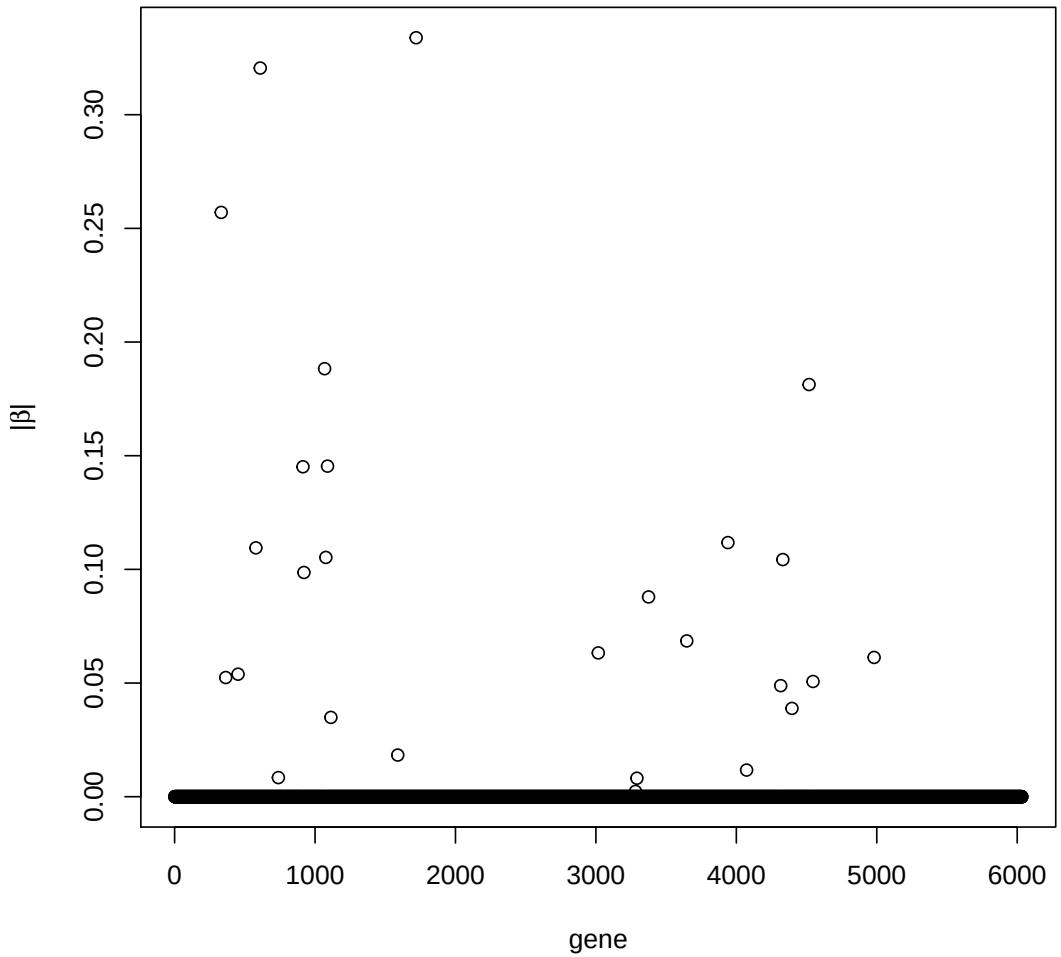

|β|
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1000
2000
3000
4000
5000
6000
gene

Table of Contents

- Making prediction algorithms better for scientific use
  - smoother
  - more interpretable
- Making traditional estimation/attribution methods better for large-scale (n, p) problems
  - more flexible
  - better scaled
- We do have optimality theory for estimation (MLE) and attribution (Neyman-Pearson), but we do not have an optimality theory for prediction.

Conformal prediction

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Lei, G'Sell, Rinaldo, Tibshirani, Wasserman (2018) Distribution-free predictive inference for regression. JASA,113:1094–1111
- Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511.
- A Tutorial on Conformal Prediction `https://www.youtube.com/watch?v=nql000Lu_iE` (Part 1); `https://www.youtube.com/watch?v=TRx4a2u-j7M` (Part 2); `https://www.youtube.com/watch?v=37HKrmA5gJE` (Part 3)

Table of Contents

Suppose we have fitted a Gaussian linear model based on the training data $(\mathbf{y}, \mathbf{X})$, obtaining the estimates

$$\hat{\beta} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y}, \quad \hat{\sigma}^2 = \|\mathbf{y} - \mathbf{X}\hat{\beta}\|^2/(n-p)$$

There are (at least) two levels at which we can make predictions

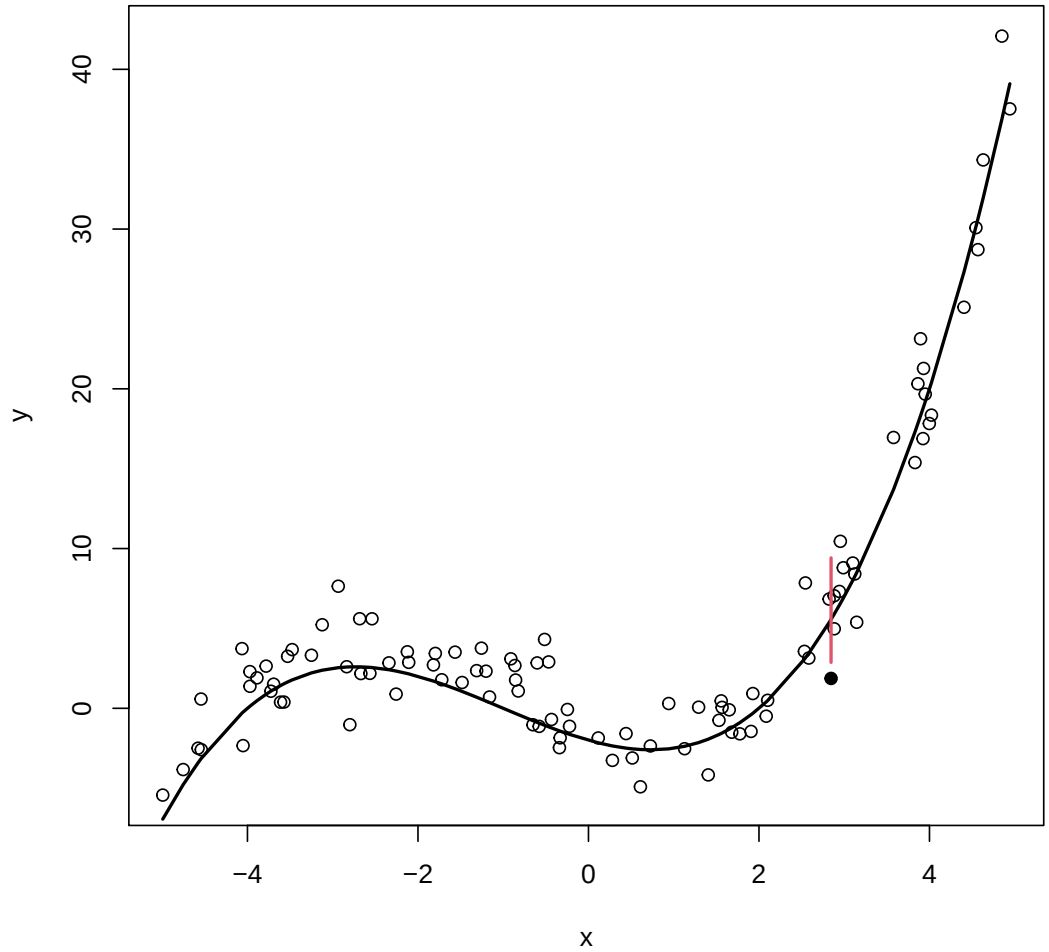
1. A *point prediction* is a single best guess about what a new Y will be when $X = x$
2. A *prediction interval*

   $$C_\alpha(x) = x^t\hat{\beta} \pm t_{n-p}^{1-\alpha/2}\hat{\sigma}\sqrt{x^t(\mathbf{X}^t\mathbf{X})^{-1}x + 1}$$

   for $Y|X = x$ with $(1-\alpha)$ *conditional coverage* guarantee, i.e.

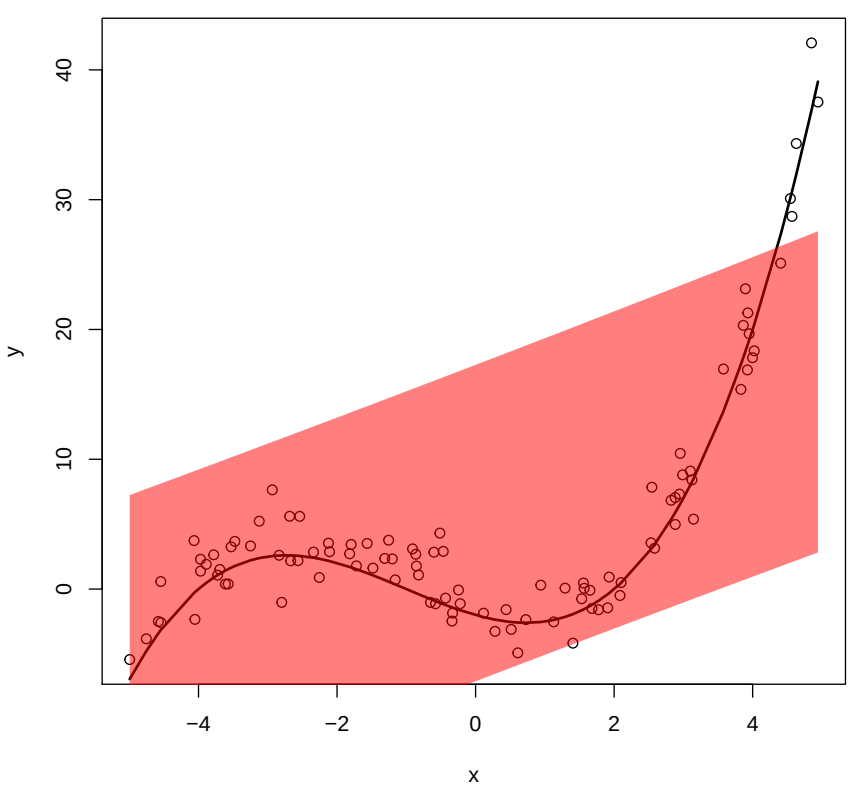
   $$\mathrm{P}(Y \in C_\alpha(x)|X = x) = 1 - \alpha$$

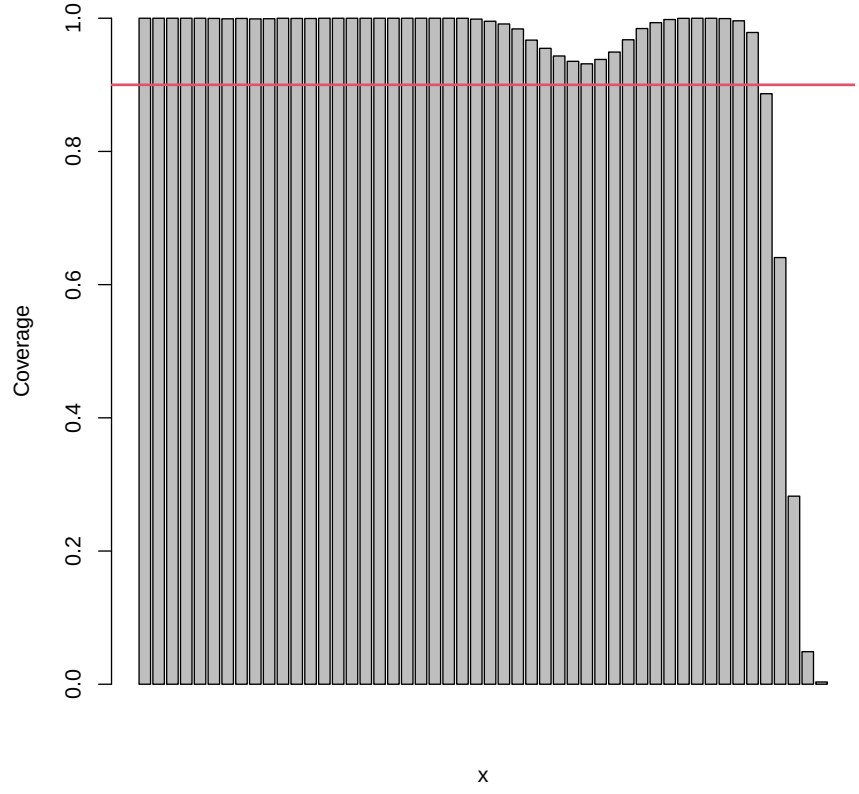
   where the probability is with respect to the training data $(X_1, Y_1), \ldots, (X_n, Y_n)$, and the new response Y at a fixed test point $X = x$



$$f(x) = \frac{1}{4}(x+4)(x+1)(x-2)$$

Model miss-specification





$1 - \alpha = 90\%$, marginal coverage $\approx 93\%$

Table of Contents

Marginal and conditional coverage

- $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$ follows some *unknown* joint distribution P_{XY}
- Training $(X_1, Y_1), \ldots, (X_n, Y_n)$ and test (X_{n+1}, Y_{n+1}) i.i.d. (X, Y)
- C_α satisfies *distribution-free marginal coverage* at level $1 - \alpha$ if

$$\mathrm{P}(Y_{n+1} \in C_\alpha(X_{n+1})) \geq 1 - \alpha \qquad \forall\ P_{XY}$$

  where the probability is w.r.t. $(X_1, Y_1), \ldots, (X_n, Y_n)$ and (X_{n+1}, Y_{n+1})
- C_α satisfies *distribution-free conditional coverage* at level $1 - \alpha$ if

$$\mathrm{P}(Y_{n+1} \in C_\alpha(X_{n+1}) | X_{n+1} = x) \geq 1 - \alpha \qquad \forall\ P_{XY},\ \ \forall\ x$$

  where the probability is w.r.t. $(X_1, Y_1), \ldots, (X_n, Y_n)$, and Y_{n+1} at a fixed test point $X_{n+1} = x$

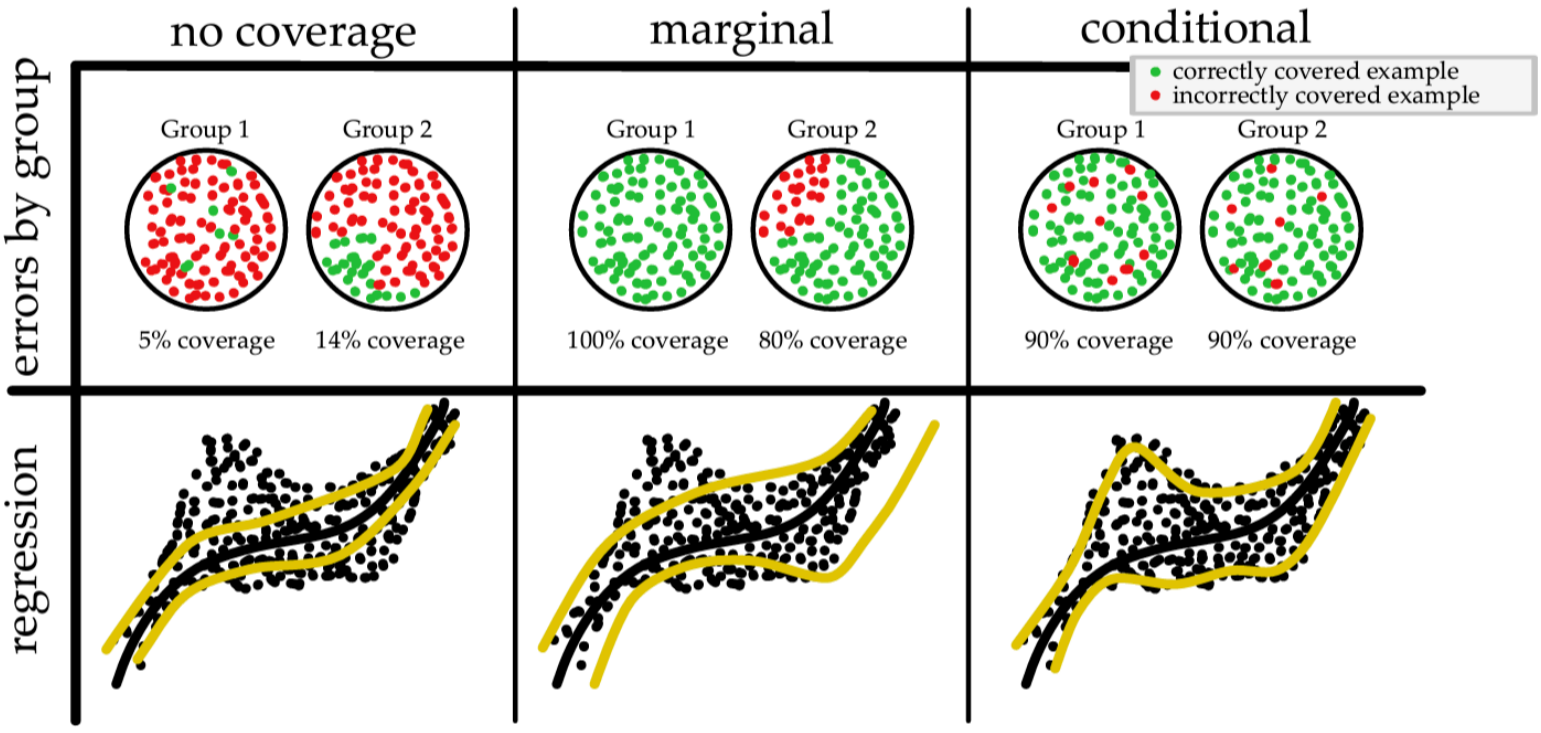


Figure 10: Prediction sets with various notions of coverage: *no coverage, marginal coverage, or conditional coverage (at a level of 90%). In the marginal case, all the errors happen in the same groups and regions in X-space. Conditional coverage disallows this behavior, and errors are evenly distributed.*

From: Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511.

Table of Contents

Conformal Prediction

Conformal prediction (Vovk, Gammerman, Saunders, Vapnik, 1996-1999) is a general framework for constructing prediction sets $\hat{C}_n$ with

1. Finite-sample coverage guarantee (exact)
2. For any data distribution (distribution-free)
3. For any predictive model (model-free)

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_n(X_{n+1})\} = 1 - \alpha$$

Two main limitations:

1. Marginal coverage
2. Exchangeability assumption

Full conformal and split conformal

Two main algorithms:

- *Full* conformal prediction
- *Split* conformal prediction

Inductive or split conformal prediction addresses the very high computational cost of (full) conformal prediction, but at the cost of introducing extra randomness due to a one-time random split of the data.

Algorithm 1 Full conformal prediction

Require: Training $(x_1, y_1), \ldots, (x_n, y_n)$, test x_{n+1}, algorithm $\hat{\mu}$, level α, grid of values $\mathcal{Y} = \{y, y', y'', \ldots\}$

for $y \in \mathcal{Y}$ **do**
    Train $\hat{\mu}^y(x) = \hat{\mu}(x; (x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, y))$
    Compute $R_i^y = |y_i - \hat{\mu}^y(x_i)|$ for $i = 1, \ldots, n$
    Sort $R_1^y, \ldots, R_n^y$ in increasing order: $R_{(1)}^y \leq \ldots \leq R_{(n)}^y$
    Compute $R_\alpha^y = R_{(k)}^y$ with $k = \lceil (1-\alpha)(n+1) \rceil$
    Compute $R^y = |y - \hat{\mu}^y(x_{n+1})|$
end for
$C_\alpha(x_{n+1}) = \{y \in \mathcal{Y} : R^y \leq R_\alpha^y\}$

- Assume that (X_i, Y_i), $i = 1, \ldots, n+1$ are i.i.d. from a probability distribution P_{XY} on the sample space $\mathbb{R}^p \times \mathbb{R}$. This is the only assumption of the method
- The prediction interval

$$C_\alpha(x_{n+1}) = \{y \in \mathbb{R} : R^y \leq R^y_\alpha\},$$

  satisfies

$$\mathrm{P}(Y_{n+1} \in C_\alpha(X_{n+1})) = 1 - \alpha$$

  if and only if $\alpha \in \{1/(n+1), 2/(n+1), \ldots, n/(n+1)\}$
- Informally, the null hypothesis that the random variable Y_{n+1} will have the outcome y, i.e.

$$H_y : Y_{n+1} = y$$

  is rejected when $R^y > R^y_\alpha$

Nonparametric Statistics

- Machine Learning has strong historical roots in Nonparametric Statistics
- K-Nearest Neighbors was introduced by two statisticians (students of Jerzy Neyman), Evelyn Fix and Joseph Hodges (Fix and Hodges, 1951)
- Conformal Prediction turns out to have roots in Permutation Testing (Fisher, 1925; Efron, 2021)

Prediction interval for Y_{n+1} (Vovk et al., 2005)	Confidence interval for Δ (Lehmann, 1963)
Supervised learning Training set $(X_1, Y_1), \ldots, (X_n, Y_n)$ Test point (X_{n+1}, Y_{n+1})	Two-sample location shift model $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} F(x)$ $Y_1, \ldots, Y_m \overset{\text{i.i.d.}}{\sim} F(y - \Delta)$
$H_y : Y_{n+1} = y$	$H_d : \Delta = d$
$(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, y)$	$x_1, \ldots, x_n, y_1 - d, \ldots, y_m - d$
$\hat{C} = \{y : p_y^* > \alpha\}$	$\hat{C} = \{d : p_d^* > \alpha\}$

Table of Contents

Algorithm 2 Split conformal prediction

Require: Training $(x_1, y_1), \ldots, (x_n, y_n)$, x_{n+1}, algorithm $\hat{\mu}$, validation sample size m, level α

1: Split $\{1, \ldots, n\}$ into L of size w and I of size $m = n - w$
2: Train $\hat{\mu}_L(x) = \hat{\mu}(x; (x_l, y_l), l \in L)$
3: Compute $R_i = |y_i - \hat{\mu}_L(x_i)|$ for $i \in I$
4: Sort $\{R_i, i \in I\}$ in increasing order: $R_{(1)} \leq \ldots \leq R_{(m)}$
5: Compute $R_\alpha = R_{(k)}$ with $k = \lceil (1-\alpha)(m+1) \rceil$

$$\begin{aligned} C_\alpha(x_{n+1}) &= \{y \in \mathbb{R} : |y - \hat{\mu}_L(x_{n+1})| \leq R_\alpha\} \\ &= [\hat{\mu}_L(x_{n+1}) - R_\alpha, \hat{\mu}_L(x_{n+1}) + R_\alpha] \end{aligned}$$

- Assume that (X_i, Y_i), $i = 1, \ldots, n+1$ are i.i.d. from a probability distribution P_{XY} on the sample space $\mathbb{R}^p \times \mathbb{R}$
- The prediction interval

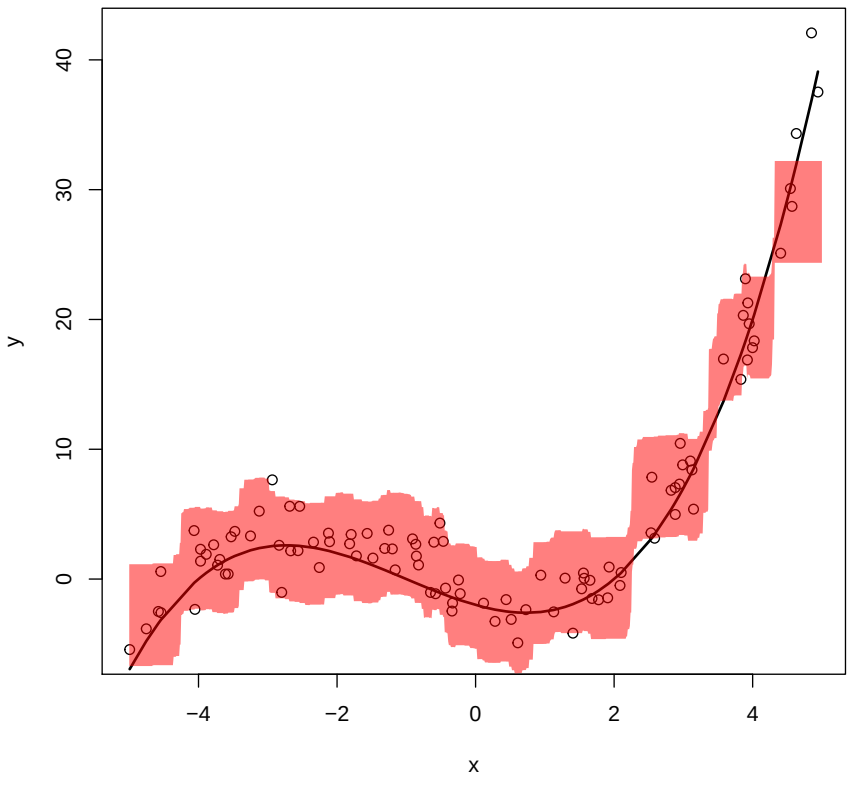
$$C_\alpha(x_{n+1}) = [\hat{\mu}_L(x_{n+1}) - R_\alpha, \hat{\mu}_L(x_{n+1}) + R_\alpha]$$

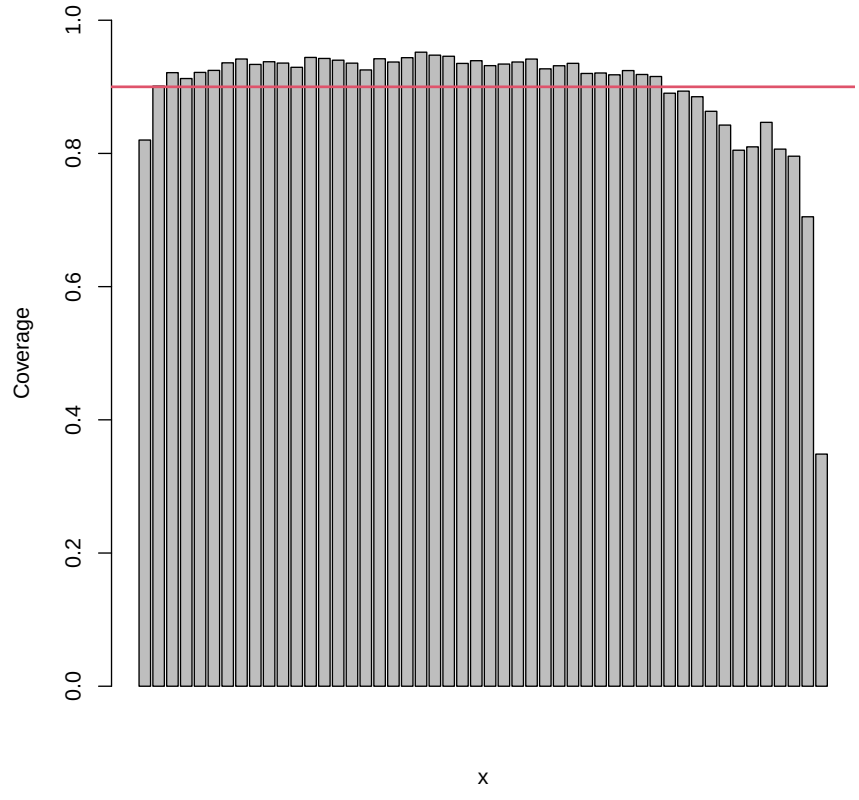
  satisfies

$$\mathrm{P}(Y_{n+1} \in C_\alpha(X_{n+1})) = 1 - \alpha$$

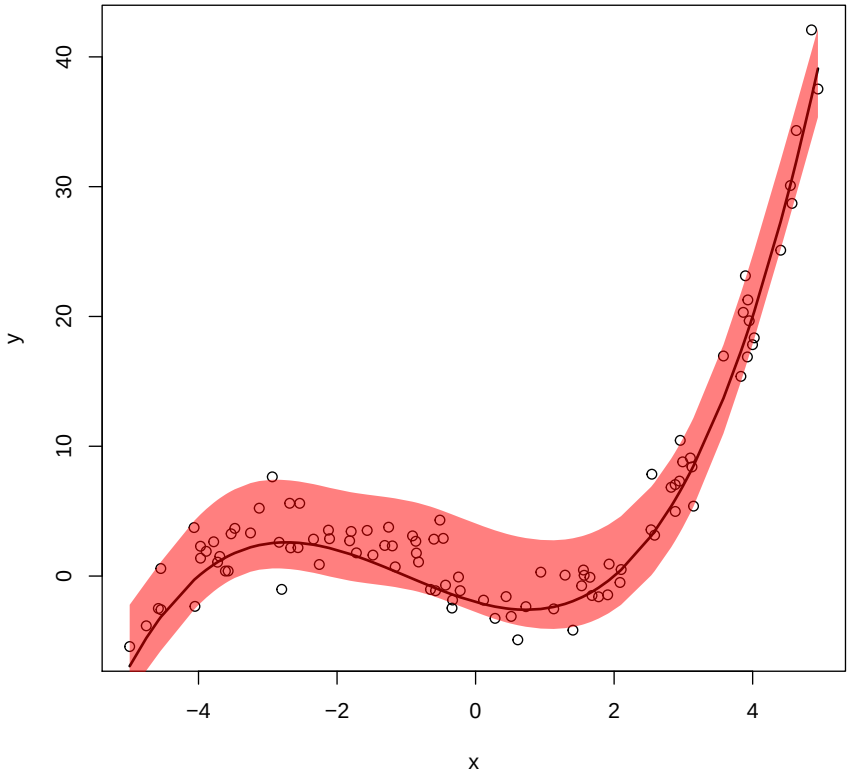
  if and only if $\alpha \in \{1/(m+1), 2/(m+1), \ldots, m/(m+1)\}$
- Note that in computing the critical value $R_\alpha = R_{(k)}$ with $k = \lceil (1-\alpha)(m+1) \rceil$, we need to have $k \leq m$, which happens if $\alpha \geq 1/(m+1)$ (otherwise if $k > m$ we need to set $R_\alpha = +\infty$)

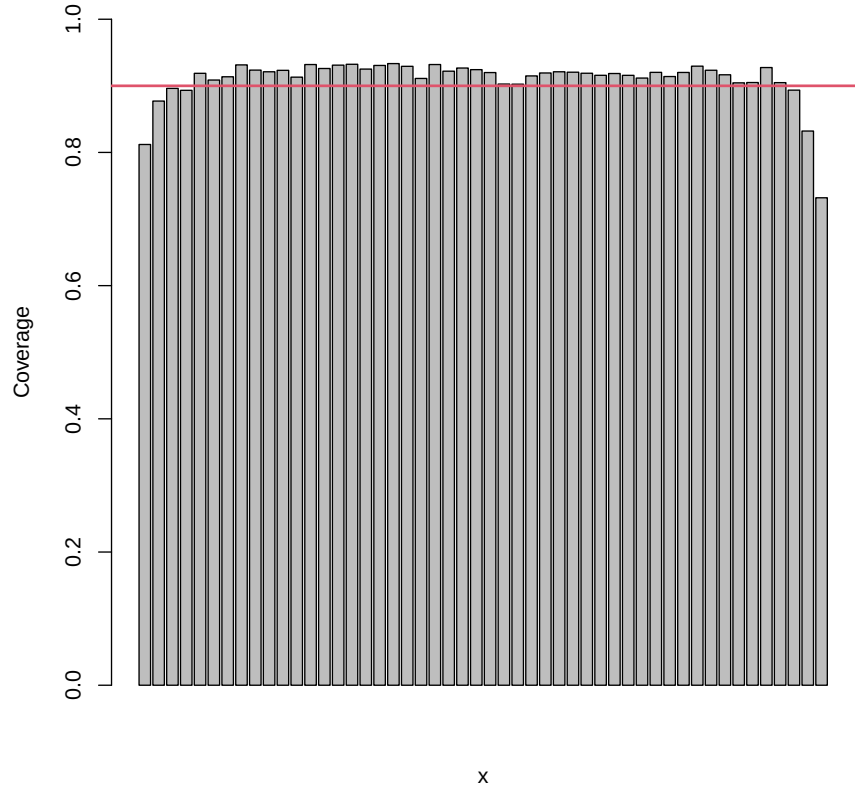
Random Forest





Smoothing splines





Conformity scores

- In the previous algorithm we used a statistic, called *conformity score*, which is the absolute value of the residual

$$R_i = |y_i - \hat{\mu}_L(x_i)|, \quad i \in I$$

  where $\hat{\mu}_L$ is an estimator of $\mathbb{E}(Y \mid X)$ based on $\{(X_i, Y_i), i \in L\}$
- The oracle knows the conditional distribution of $Y \mid X$. The oracle prediction interval

$$C^*_\alpha(x) = [q^{\alpha/2}(x), q^{1-\alpha/2}(x)]$$

  where $q^\gamma(x)$ is the γ-quantile of $Y \mid X = x$, guarantees exact conditional coverage

$$\mathrm{P}(Y \in C^*_\alpha(X) | X = x) = 1 - \alpha \quad \forall\ x$$

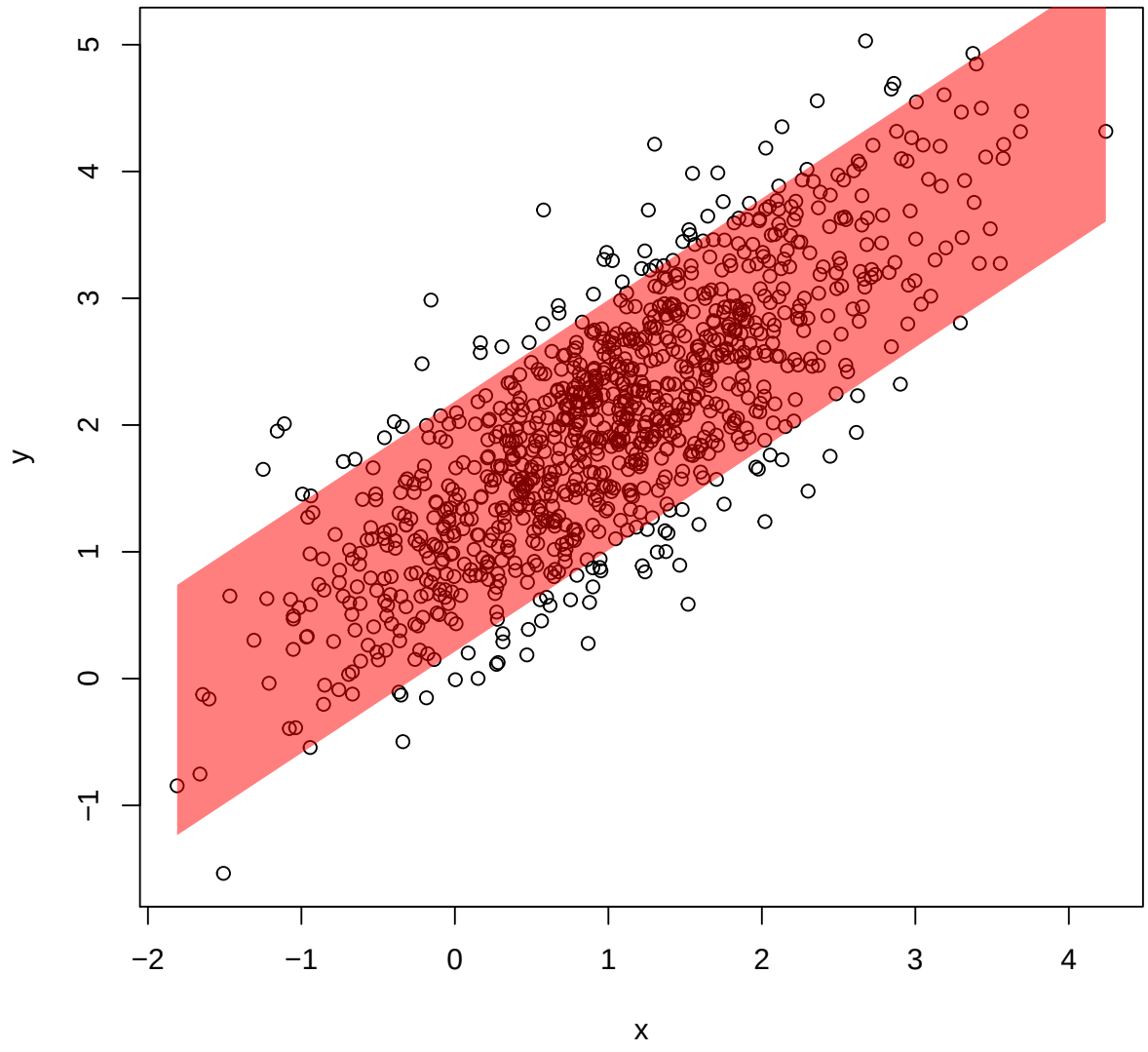
Suppose that

$$\begin{pmatrix} Y \\ X \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_y \\ \mu_x \end{pmatrix}, \begin{pmatrix} \sigma_y^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_x^2 \end{pmatrix}\right)$$

then the conditional distribution of $Y \mid X = x$ is

$$(Y|X = x) \sim N\Big(\mu_y + \rho\frac{\sigma_y}{\sigma_x}(x - \mu_x), \sigma_y^2(1 - \rho^2)\Big)$$

from which we can compute the quantile $q^\gamma(x)$



$C^*_\alpha(x) = [q^{\alpha/2}(x), q^{1-\alpha/2}(x)]$ as a function of x

Table of Contents

Conformal quantile regression

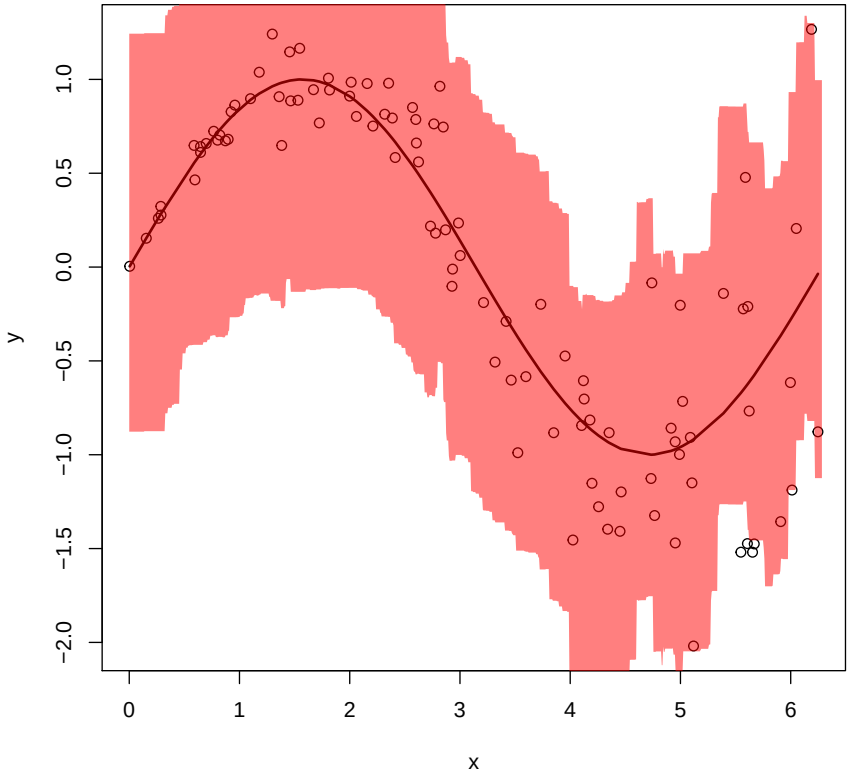
- Compute conformity scores

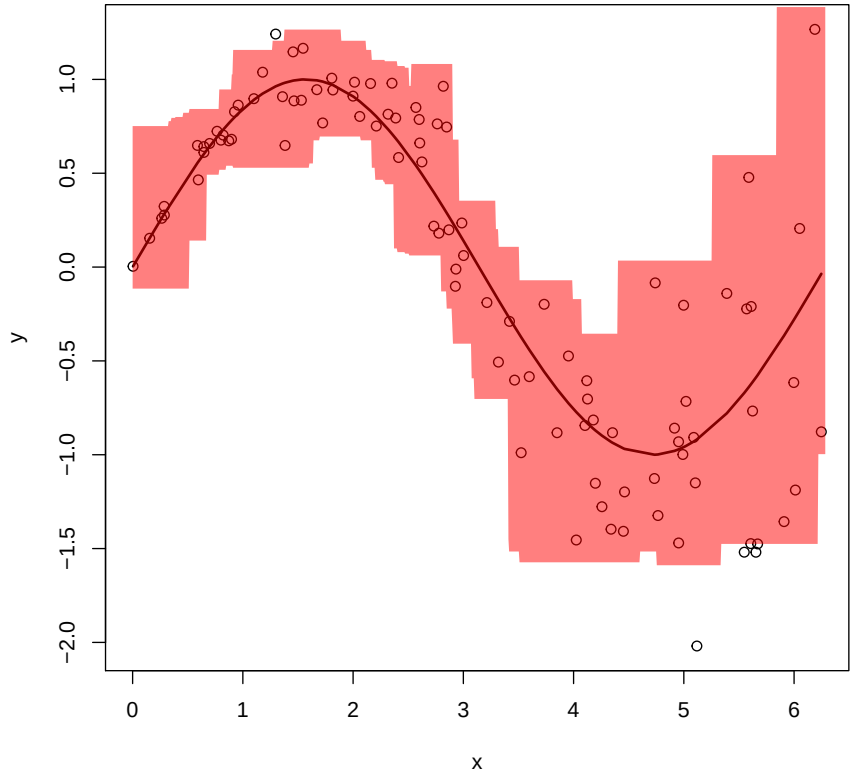
$$R_i = \max\left\{\hat{q}_L^{\gamma}(X_i) - Y_i, Y_i - \hat{q}_L^{1-\gamma}(X_i)\right\}, \quad i \in I$$

  where $\hat{q}_L^{\gamma}$ is an estimator of the γ-quantile of $Y \mid X$ based on $\{(X_i, Y_i), i \in L\}$
- Sort $\{R_i, i \in I\}$ in increasing order, obtaining $R_{(1)} \leq \ldots \leq R_{(m)}$, and compute $R_\alpha = R_{(k)}$ with $k = \lceil (1-\alpha)(m+1) \rceil$
- Compute the prediction interval

$$\begin{aligned} C_\alpha(x_{n+1}) &= \{y \in \mathbb{R} : \max\left\{\hat{q}_L^{\gamma}(x_{n+1}) - y, y - \hat{q}_L^{1-\gamma}(x_{n+i})\right\} \leq R_\alpha\} \\ &= [\hat{q}_L^{\gamma}(x_{n+1}) - R_\alpha, \hat{q}_L^{1-\gamma}(x_{n+1}) + R_\alpha] \end{aligned}$$

  or $C_\alpha(x_{n+1}) = \emptyset$ if $R_\alpha < (1/2)(\hat{q}_L^{\gamma}(x_{n+1}) - \hat{q}_L^{1-\gamma}(x_{n+1}))$





$$X_i \sim U(0, 2\pi), \epsilon_i \sim N(0, 1), Y_i = \sin(X_i) + \frac{\pi |X_i|}{20} \epsilon_i$$

James-Stein estimation

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Samworth (2012). Stein's paradox. eureka, 62:38–41
- Candés (2022) Lecture notes (Stats 300C - Theory of Statistics)

- A very surprising result arises in a remarkably simple estimation problem.
- Let $X_1, \ldots, X_p$ be independent random variables, with $X_i \sim N(\mu_i, 1)$ for $i = 1, \ldots, p$. Writing $X = (X_1, \ldots, X_p)^t$, suppose we want to find a good estimator $\hat{\mu} = \hat{\mu}(X)$ of $\mu = (\mu_1, \ldots, \mu_p)^t$
- Squared error loss function:

$$L(\hat{\mu}, \mu) = \|\hat{\mu} - \mu\|^2 = \sum_{i=1}^{p} (\hat{\mu}_i - \mu_i)^2$$

  where $\|\cdot\|$ denotes the Euclidean norm
- Risk function: $R(\hat{\mu}, \mu) = \mathbb{E}[L(\hat{\mu}, \mu)]$

Inadmissible estimators

- If $\hat{\mu}$ and $\tilde{\mu}$ are both estimators of μ, we say that $\hat{\mu}$ strictly dominates $\tilde{\mu}$ if $R(\hat{\mu}, \mu) \leq R(\tilde{\mu}, \mu)$ for all μ, with strict inequality for some value of μ. In this case we say that $\tilde{\mu}$ is *inadmissible*.
- If $\hat{\mu}$ is not strictly dominated by any estimator of μ, it is said to be admissible. Note that admissible estimators are not necessarily sensible: for $p = 1$, the estimator $\hat{\mu} = 37$ (which ignores the data!) is *admissible*.
- On the other hand decision theory dictates that inadmissible estimators can be discarded
- $\hat{\mu} = X$ is a very obvious estimator of μ: it is the maximum likelihood estimator and the uniform minimum variance unbiased estimator with

$$R(\hat{\mu}, \mu) = p \quad \forall\ \mu \in \mathbb{R}^p$$

since $\|X - \mu\|^2 \sim \chi^2_p$

James-Stein estimator

- It has been proved that $\hat{\mu} = X$ is admissible for $p = 1, 2$
- James and Stein (1961) showed that the estimator

$$\hat{\mu}_{JS} = \left(1 - \frac{p-2}{\|X\|^2}\right) X$$

strictly dominates $\hat{\mu} = X$ for $p \geq 3$:

$$R(\hat{\mu}_{JS}, \mu) = p - (p-2)^2 \mathbb{E}\left(\frac{1}{\|X\|^2}\right) < p \quad \forall\ \mu \in \mathbb{R}^p$$

$\|X\|^2 = \sum_{i=1}^p X_i^2$ follows a noncentral χ^2 distribution with p degrees of freedom and noncentrality parameter $\|\mu\|^2$. Using a result about noncentral χ^2 variables, we can write

$$\|X\|^2 \sim \chi^2_{p+2K}$$

where $K \sim \text{Poisson}(\|\mu\|^2/2)$.

$$\begin{aligned}
\mathbb{E}\left(\frac{1}{\|X\|^2}\right) &= \mathbb{E}\left(\frac{1}{\chi^2_{p+2K}}\right) = \mathbb{E}\left\{\mathbb{E}\left(\frac{1}{\chi^2_{p+2K}}\right)|K\right\} \\
&= \mathbb{E}\left\{\frac{1}{(p-2)+2K}\right\} \geq \frac{1}{(p-2)+\|\mu\|^2}
\end{aligned}$$

with equality if $\mu = 0$, where we used $\mathbb{E}(1/\chi^2_p) = 1/(p-2)$ for $p > 2$ and Jensen's inequality. Then

$$R(\hat{\mu}_{\mathcal{JS}}, \mu) \leq p - \frac{p-2}{1+\|\mu\|^2/(p-2)}$$

Oracle linear estimator

- A linear estimator of the form

$$\tilde{\mu} = bX = (bX_1, \ldots, bX_p)^t$$

  with $0 \leq b \leq 1$ shrinks X towards the origin
- The risk of a linear estimator is

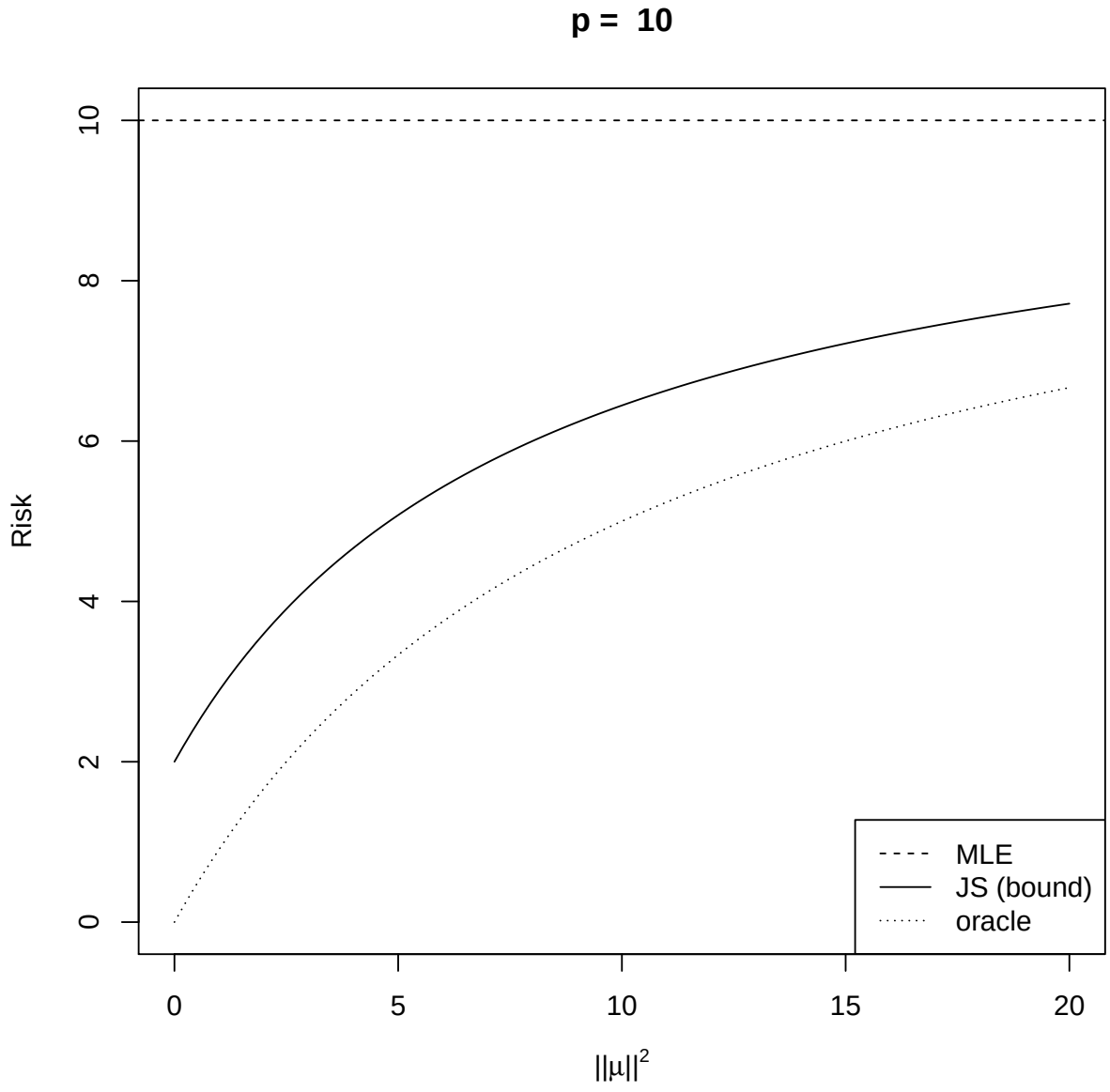
$$R(\tilde{\mu}, \mu) = (1-b)^2 \|\mu\|^2 + b^2 p$$

  minimized by

$$b^* = \frac{\|\mu\|^2}{p + \|\mu\|^2}$$

- The risk of the oracle linear estimator $\tilde{\mu}^* = b^* X$ is

$$R(\tilde{\mu}^*, \mu) = p\|\mu\|^2 / (p + \|\mu\|^2)$$


p = 10
Risk
0
2
4
6
8
10
0
5
10
15
20
||μ||2
MLE
JS (bound)
oracle

- Geometrically, the James-Stein estimator shrinks each component of X towards the origin, and the biggest improvement comes when μ is close to zero
- For $\mu = 0$ we have $R(\hat{\mu}_{JS}, 0) = 2$ for all $p \geq 2$
- As $\|\mu\|^2 \to \infty$, $R(\hat{\mu}_{JS}, \mu) \to p$

Stein's heuristic argument (1956)

- Stein argued that a good estimate should obey $\hat{\mu}_i \approx \mu_i$ for every i. Thus we should also have $\hat{\mu}_i^2 \approx \mu_i^2$, which further implies $\sum_i \hat{\mu}_i^2 \approx \sum_i \mu_i^2$
- Consider the estimator $\hat{\mu} = X$. For this estimator we have

$$\mathbb{E}\|X\|^2 = \mathbb{E}\sum_i X_i^2 = \mathbb{E}\sum_i (\mu_i + Z_i)^2 = \|\mu\|^2 + p$$

  where $Z_i \sim N(0, 1)$
- This suggests that for large p, $\|X\|^2$ is likely to be considerably larger than $\|\mu\|^2$, and hence we may be able to obtain a better estimator by shrinking the estimator $\hat{\mu} = X$ towards o.

Positive James-Stein estimator

- If the shrinkage in $\hat{\mu}_{JS}$ is too large, it is possible that the estimator switches to the other sign when $\|X\|^2 < p - 2$
- By precluding the possibility of a sign reversal, the positive JS estimator

$$\hat{\mu}_{JS}^{+} = \left(1 - \frac{p-2}{\|X\|^2}\right)_{+} X$$

  where $(a)_+ = \max(a, 0)$ denotes the positive part
- $\hat{\mu}_{JS}^{+}$ further improves upon the $\hat{\mu}_{JS}$ estimate, i.e., $R(\hat{\mu}_{JS}^{+}, \mu) < R(\hat{\mu}_{JS}, \mu)$ for all μ
- However, this estimator is not admissible either.

Shrinking toward an arbitrary point

- In terms of choosing a point to shrink towards, though, there is nothing special about the origin, and we could equally well shrink towards any pre-chosen $m \in \mathbb{R}^p$ using the estimator

$$\hat{\mu}^m_{JS} = m + \left(1 - \frac{p-2}{\|X-m\|^2}\right)(X-m)$$

- In this case, we have $R(\hat{\mu}^m_{JS}, \mu - m) = R(\hat{\mu}_{JS}, \mu)$, so $\hat{\mu}^m_{JS}$ still strictly dominates $\hat{\mu} = X$

Correlated data

- Assume that $X \sim N_p(\mu, \Sigma)$ where Σ is a known covariance matrix
- A a generalization of James-Stein estimator

$$\hat{\mu}_{JS}^{\Sigma} = \left(1 - \frac{c(\tilde{p} - 2)}{X^t \Sigma^{-1} X}\right) X$$

  with $0 < c < 2$ and $\tilde{p} = \text{tr}(\Sigma)/\lambda_{\text{max}}(\Sigma)$ is the effective dimension of the problem, where $\lambda_{\text{max}}(\Sigma)$ is the maximum eigenvalue of Σ
- If $\tilde{p} > 2$, then the generalization of the JS estimator $\hat{\mu}_{JS}^{\Sigma}$ dominates the MLE $\hat{\mu} = X$

Linear model

- We can apply the previous result to the case of linear regression $y \sim N_n(X\beta, \sigma^2 I_n)$, where the MLE is the OLS estimator $\hat{\beta} = (X^t X)^{-1} X^t y \sim N_p(\beta, \sigma^2 (X^t X)^{-1})$, so with $\mu = X\beta$ and $\hat{\mu} = X\hat{\beta}$ we have $R(\hat{\mu}, \mu) = \sigma^2 p$
- James-Stein estimator becomes

$$\hat{\beta}_{JS} = \left(1 - \frac{(p-2)\sigma^2}{\hat{\beta}^t X^t X \hat{\beta}}\right) \hat{\beta}$$

- Letting $\hat{\mu}_{JS} = X\hat{\beta}_{JS}$ and $\mu = X\beta$, the James–Stein Theorem guarantees that

$$R(\hat{\mu}_{JS}, \mu) \leq \sigma^2 p$$

no matter what β is, as long as $p \geq 3$

- It is natural to ask how crucial the normality and squared error loss assumptions are to the Stein phenomenon
- The normality assumption is not critical at all;
- The original result can also be generalised to different loss functions, but there is an important caveat here: the Stein phenomenon only holds when we are interested in simultaneous estimation of all components of μ. If our loss function were $L(\hat{\mu}, \mu) = (\hat{\mu}_1 - \mu_1)^2$ then we could not improve on $\hat{\mu} = X$

$p = 5$, $\mu = (\sqrt{p/2}, \sqrt{p/2}, 0, 0, 0)^t$, $\|\mu\|^2 = p$

10^4 repetitions

	Risk	Risk1	Risk2	Risk3	Risk4	Risk5
MLE	5.00	1.01	1.01	1.00	0.98	0.99
JS	3.65	1.08	1.07	0.50	0.49	0.50

$R(\hat{\mu}_{JS}, \mu) \leq p - (p-2)/(1 + p/(p-2)) = 3.875$

An Empirical Bayes interpretation

Bayesian setup

- Consider the Bayesian setup

$$\mu_i \sim N(0, \tau^2) \quad X|\mu \sim N(\mu, I_p) \tag{1}$$

- Given the data X, the posterior of μ is

$$\mu|X \sim N(\lambda X, \lambda I_p)$$

  where $\lambda = \tau^2/(1+\tau^2)$

- The Bayes estimator is simply the mean of the posterior

$$\hat{\mu}_B = \lambda X = \left(1 - \frac{1}{1+\tau^2}\right) X$$

- Assuming (1), the Bayes risk is $R(\hat{\mu}_B, \mu) = \lambda p$

Connection to James-Stein

- We cannot directly compute the shrinkage factor $\lambda = \tau^2/(1+\tau^2)$, but perhaps we can estimate it using the data
- Since $X_i = \mu_i + Z_i \sim N(0, 1+\tau^2)$, where $Z_i \sim N(0,1)$. This implies $\|X\|^2 \sim (1+\tau^2)\chi^2_p$
- Combining this result with $\mathbb{E}[(p-2)/\chi^2_p] = 1$, we arrive at an unbiased estimate for λ

$$\hat{\lambda} = \left(1 - \frac{(p-2)}{\|X\|^2}\right)$$

- Assuming (1), the Bayes risk is $R(\hat{\mu}_{JS}, \mu) = \left(1 + \frac{2}{p\tau^2}\right) R(\hat{\mu}_B, \mu)$

$p = 5$, $\tau^2 = 2$, $\mu_i \sim N(0, \tau^2)$

10^4 repetitions

	Bayes Risk	B.Risk1	B.Risk2	B.Risk3	B.Risk4	B.Risk5
MLE	5.01	1.01	1.01	1.00	0.99	1.00
BAYES	3.34	0.67	0.68	0.67	0.67	0.66
JS	4.02	0.81	0.82	0.80	0.80	0.79

$R(\hat{\mu}, \mu) = 5$, $R(\hat{\mu}_B, \mu) = 3.33$, $R(\hat{\mu}_{JS}, \mu) = 4$,

Shrinking Toward the Group Mean

- In practice, instead of arbitrarily picking some point, it might instead make sense to chose $m = \bar{X}$ as so as to adapt to the true center of μ_i
- Consider the Bayesian setup

$$\mu_i \sim N(m, \tau^2) \quad X|\mu \sim N(\mu, I_p) \tag{2}$$

  with m and τ^2 unknown
- The marginal distribution of our data is

$$X_i \overset{i.i.d.}{\sim} N(m, 1 + \tau^2)$$

  and the posterior mean is

$$\mu|X \sim N(m + \lambda(X - m), \lambda I_p)$$

- $\hat{\mu}_B = m + \lambda(X - m)$ but m is unknown. Taking the empirical Bayes approach, we can use the unbiased estimator $\bar{X}$ in its place
- Similarly, we can use the sample variance $S = \sum_i (X_i - \bar{X})^2 \sim (1 + \tau^2)\chi^2_{p-1}$ to estimate λ. Now we have $\mathbb{E}[(p-3)/\chi^2_{p-1}] = 1$
- This gives us the estimator

$$\hat{\mu}^{\bar{X}}_{JS} = \bar{X} + \left(1 - \frac{p-3}{S}\right)(X - \bar{X})$$

If $p > 3$, this estimator dominates the MLE everywhere

A baseball data example

Player	MLE	TRUTH
1	0.34	0.30
2	0.33	0.35
3	0.32	0.22
4	0.31	0.28
5	0.29	0.26
6	0.29	0.27
7	0.28	0.30
8	0.26	0.27
9	0.24	0.23
10	0.23	0.26
11	0.23	0.26
12	0.22	0.21
13	0.22	0.26
14	0.22	0.27
15	0.21	0.32
16	0.21	0.23
17	0.20	0.28
18	0.14	0.20

The column labelled MLE is the batting average for 18 players in the 1970 season, using the first 90 at bats.

The column labelled TRUTH is the batting average for the remainder of the 1970 season.

- Each player Batting average =(# hits / # at bats) value is a binomial proportion

$$Y_i \sim \text{Binomial}(n, \pi_i)/n$$

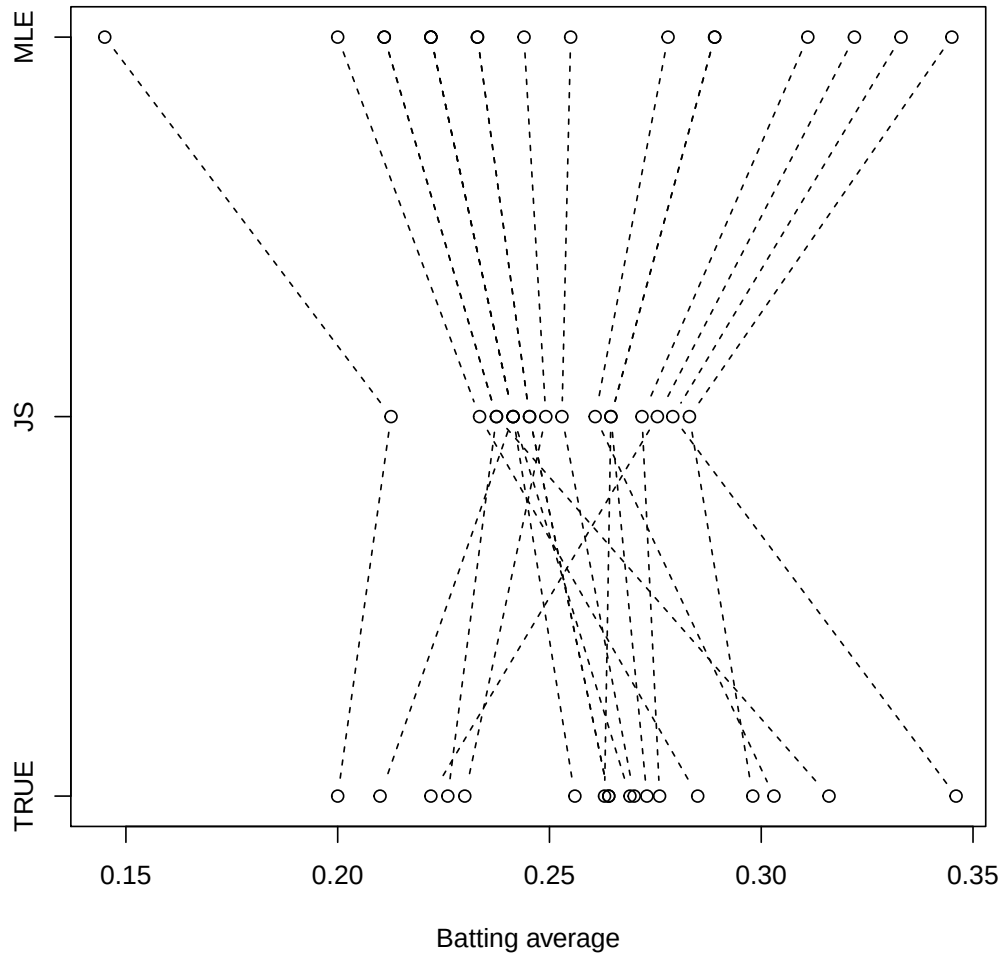
  where π_i is the true average and $n = 90$
- Since batting averages are binomial, we can use the normal approximation

$$Y_i \approx N\left(\pi_i, \frac{\pi_i(1-\pi_i)}{n}\right)$$

  but the variance depends on the mean
- One solution is to make a variance stabilizing transformation

$$X_i = 2\sqrt{n+0.5}\,\arcsin\left(\sqrt{\frac{nY_i + 3/8}{n+3/4}}\right) \approx N(\mu_i, 1)$$

  where $\mu_i = 2\sqrt{n+0.5}\,\arcsin(\sqrt{\frac{n\pi_i+3/8}{90+3/4}})$
- Inverted back $y_i^{JS} = \frac{1}{n}\left[(n+0.75)(\sin(\frac{\hat{\mu}_i^{JS}}{2\sqrt{n+0.5}}))^2 - 0.375\right]$



$$\sum_i (y_i - y_i^{\text{TRUE}})^2 = 0.0425 \quad \sum_i (y_i^{JS} - y_i^{\text{TRUE}})^2 = 0.0205$$

Ridge regression

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Hastie, T. (2020). Ridge regularization: an essential concept in data science. Technometrics, 62(4), 426-433.
- van Wieringen (2015). Lecture notes on ridge regression. arXiv preprint arXiv:1509.09169.

Condition number

- In the linear model, the estimate of β is obtained by solving the normal equations
$$X^\mathsf{T} X\beta = X^\mathsf{T} y$$
- The difficulty of solving this system of linear equations can be described by the *condition number*
$$\kappa(X^\mathsf{T} X) = \frac{d_{\max}}{d_{\min}}$$
the ratio between the largest and smallest singular values of $X^\mathsf{T} X$
- If the condition number is very large, then the matrix is said to be *ill-conditioned* (see Section 2.6 of CASL)

Toy linear model with $n = p = 2$. We set X and β as

$$X = \begin{bmatrix} 10^9 & -1 \\ -1 & 10^{-5} \end{bmatrix} \quad \beta = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

And if we define $y = X\beta$, this gives

$$y = \begin{bmatrix} 10^9 & -1 \\ -1 & 10^{-5} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 10^9 - 1 \\ -0.99999 \end{bmatrix}$$

The reciprocal of condition number, i.e. $1/\kappa(X^\mathsf{T} X) = 9.998e - 29$, is smaller than (my) machine precision, i.e. $2.220446e - 16$

```
X <- matrix(c(10^9, -1, -1, 10^(-5)), 2, 2)
beta <- c(1,1)
y <- X %*% beta

solve( crossprod(X), crossprod(X, y) )

Error in solve.default(crossprod(X)) :
system is computationally singular:
reciprocal condition number = 9.998e-29

.Machine$double.eps
2.220446e-16
```

Ridge regression solution

- Ridge provides a remedy for an *ill-conditioned* X^tX matrix
- If our $n \times p$ design matrix X has column rank less than p (or nearly so in terms of its condition number), then the usual least-squares regression equation is in trouble:

$$\hat{\beta} = (X^tX)^{-1}X^ty$$

- What we do is add a *ridge* on the diagonal - $X^tX + \lambda I_p$ with $\lambda > 0$ - which takes the problem away:

$$\hat{\beta}_\lambda = (X^tX + \lambda I_p)^{-1}X^ty$$

- This is the ridge regression solution proposed by Hoerl and Kennard (1970)

- Ridge regression modifies the normal equations to

$$(X^\mathsf{T} X + \lambda I_p)\beta = X^\mathsf{T} y$$

  and the condition number of $(X^\mathsf{T} X + \lambda I_p)$ is

$$\kappa(X^\mathsf{T} X + \lambda I_p) = \frac{d_{\max} + \lambda}{d_{\min} + \lambda}$$

- Notice that even if $d_{\min} = 0$, the condition number will be finite if $\lambda > 0$
- This technique is known as Tikhonov regularization, after the Russian mathematician Andrey Tikhonov

Penalized (Lagrange) form

- The optimization problem that ridge is solving

$$\min_{\beta} \|y - X\beta\|^2 + \lambda\|\beta\|^2 \tag{1}$$

  where $\|\cdot\|$ is the ℓ_2 Euclidean norm
- The ridge remedy comes with consequences. The ridge estimate is biased toward zero. It also has smaller variance than the OLS estimate.
- Selecting λ amounts to a bias-variance trade-off

Cement data

$n = 13, p = 4$

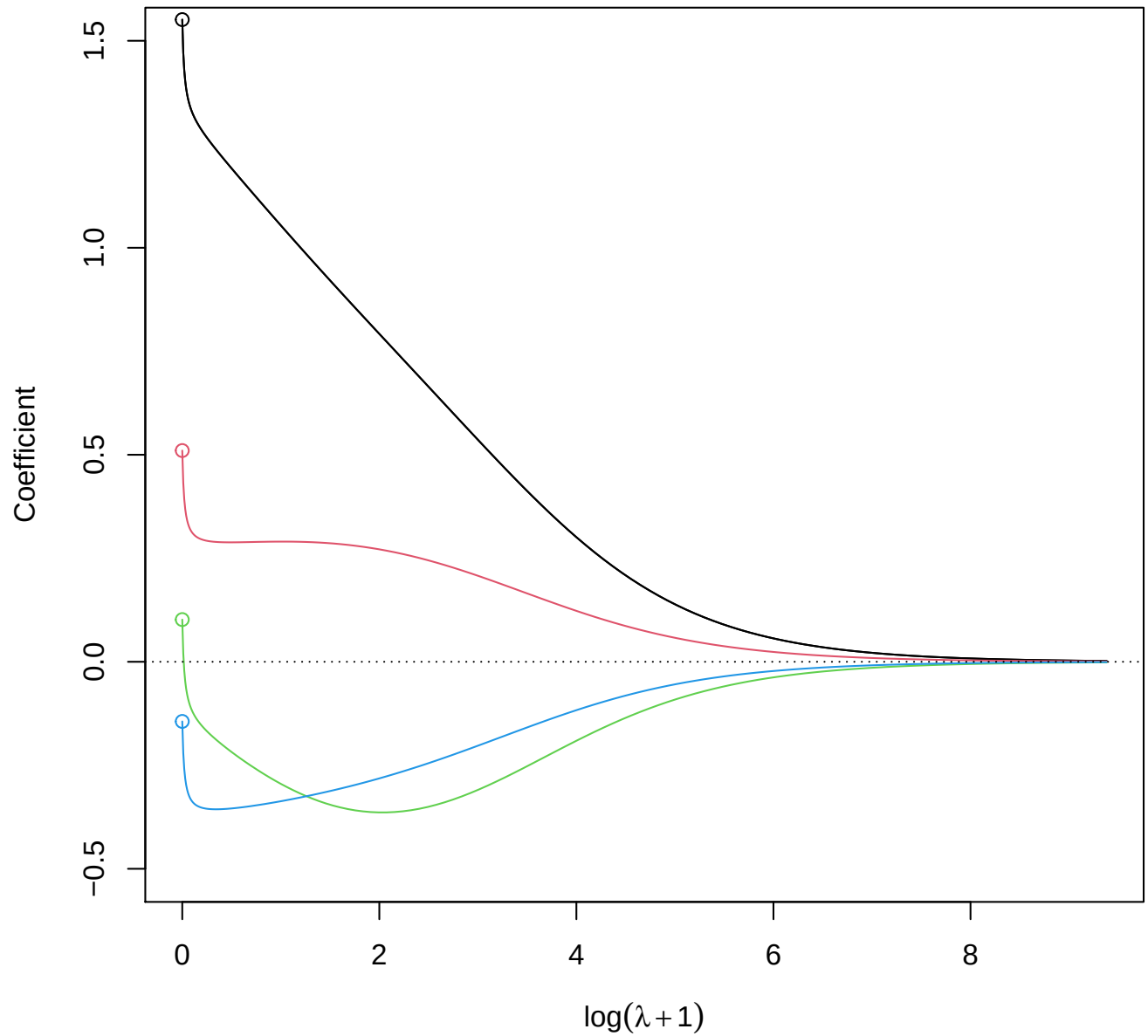
$$R = \begin{bmatrix} 1 & 0.23 & -0.82 & -0.25 \\ 0.23 & 1 & -0.14 & -0.97 \\ -0.82 & -0.14 & 1 & 0.03 \\ -0.25 & -0.97 & 0.03 & 1 \end{bmatrix}$$

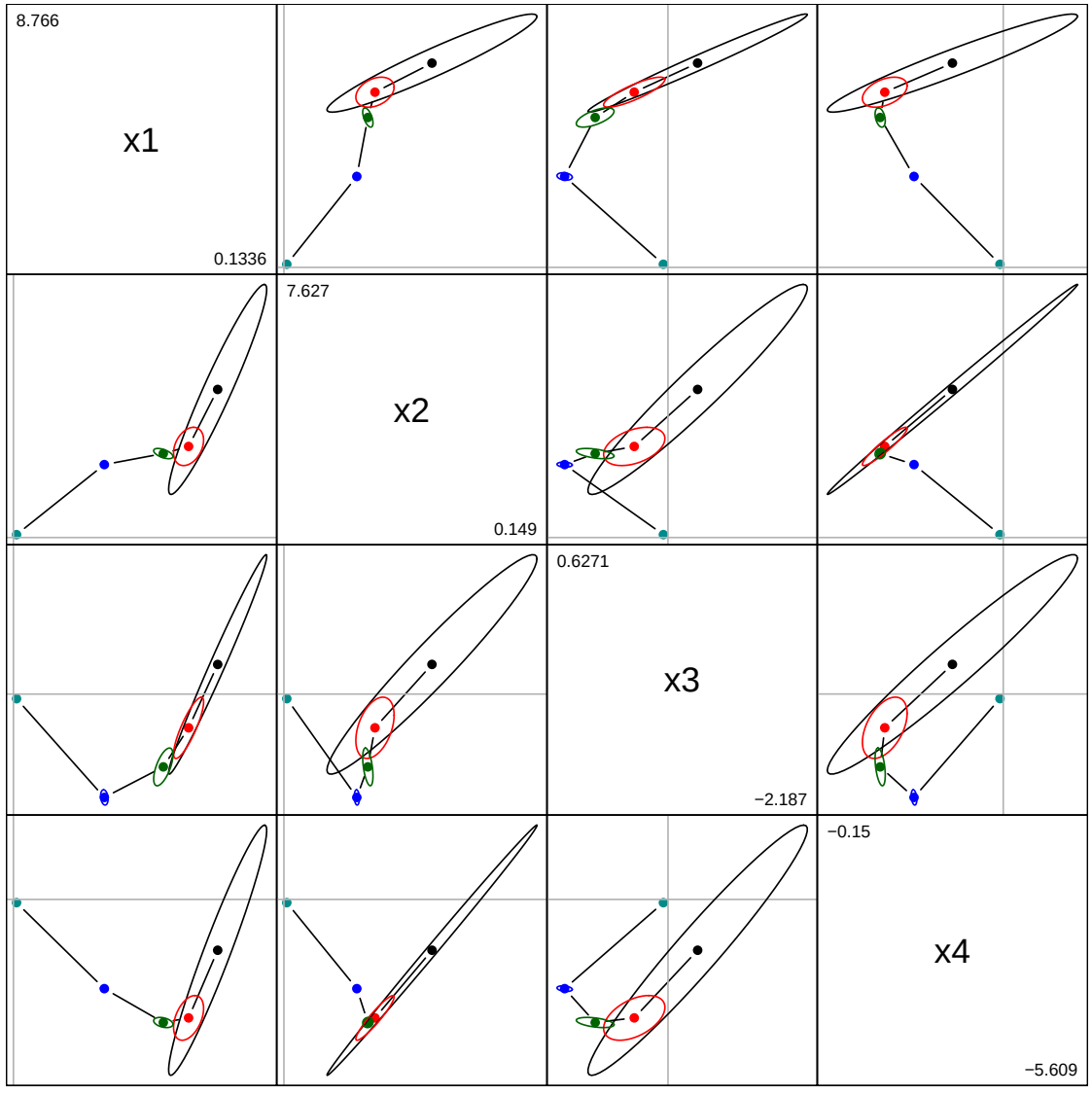
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	62.41	70.07	0.89	0.40
x1	1.55	0.74	2.08	0.07
x2	0.51	0.72	0.70	0.50
x3	0.10	0.75	0.14	0.90
x4	-0.14	0.71	-0.20	0.84

R-squared: 0.9824

	x1	x2	x3	x4
VIF	38.50	254.42	46.87	282.51

Piepel, Redgate (1998) A Mixture Experiment Analysis of the Hald Cement Data, The American Statistician, 52:23–30


Coefficient
1.5
1.0
0.5
0.0
−0.5
0
2
4
6
8
log(λ + 1)



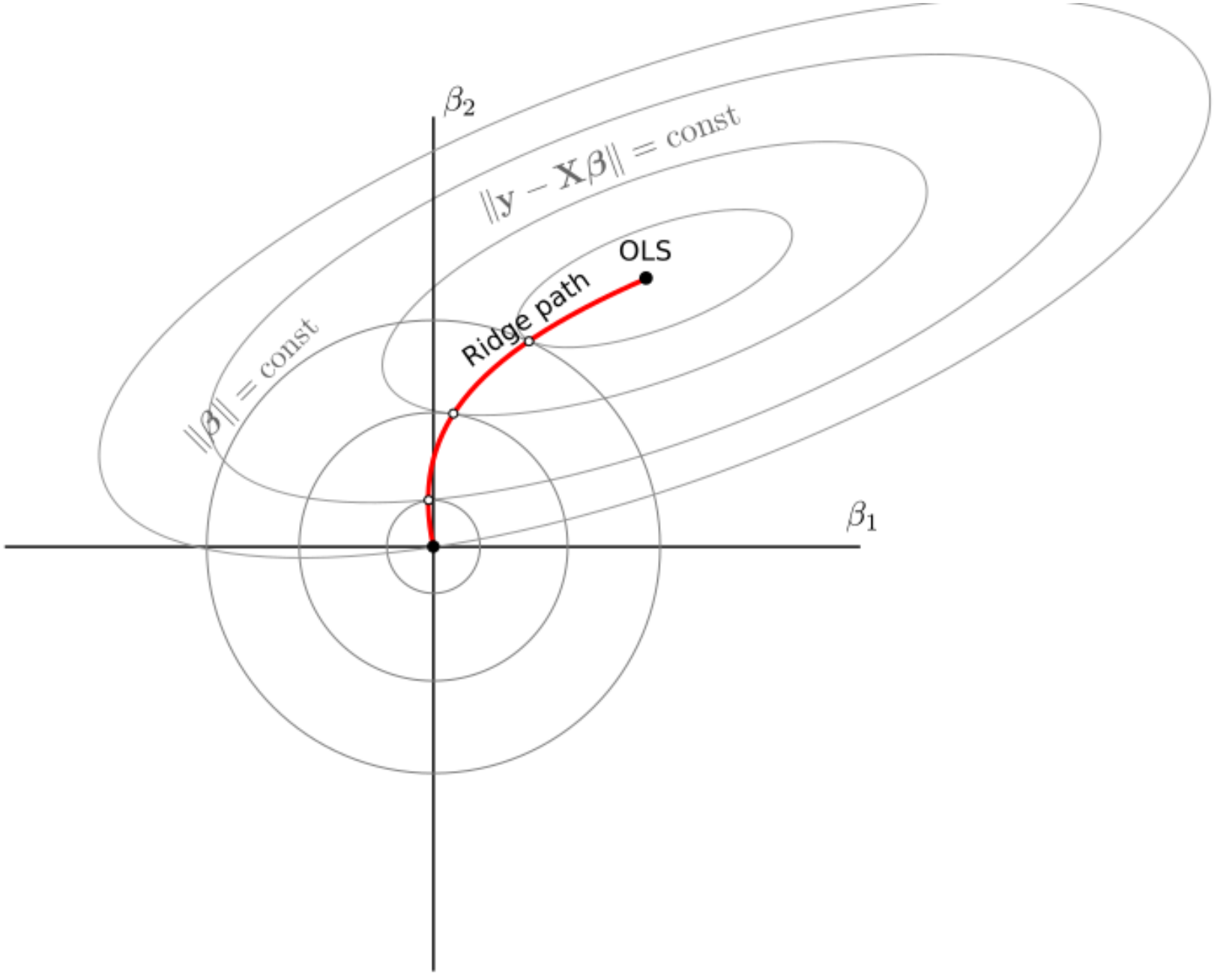
$$\lambda = 0, 0.1, 1, 10, 1000$$

Constrained form

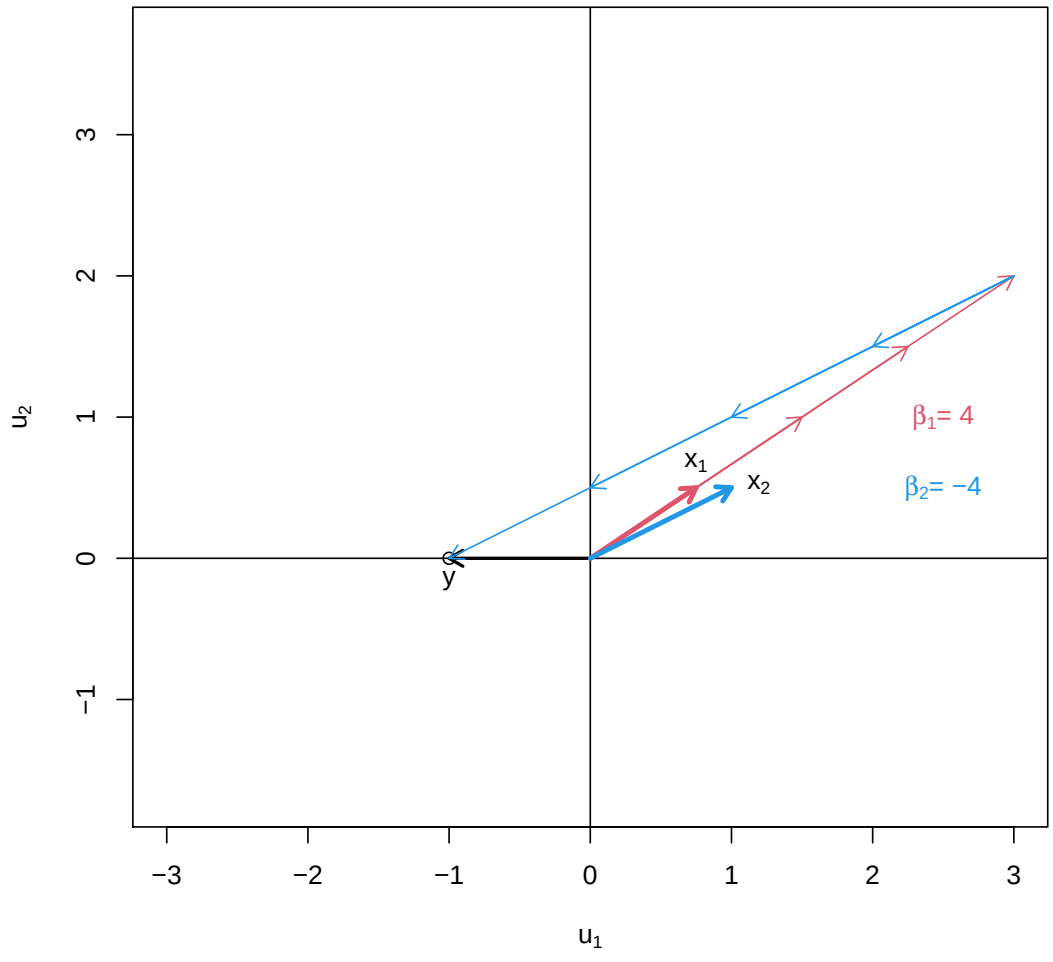
- We can also express the ridge problem as

$$\min_{\beta} \|y - X\beta\|^2 \quad \text{subject to } \|\beta\| \leq c \tag{2}$$

- The two problems are of course equivalent: every solution $\hat{\beta}_\lambda$ in (1) is a solution to (2) with $c = \|\hat{\beta}_\lambda\|$


β_2
$\|\mathbf{y} - \mathbf{X}\beta\| = \text{const}$
OLS
Ridge path
$\|\beta\| = \text{const}$
β_1

Overfitting



Large estimates of β are often an indication of overfitting

Bayesian view

- Assume

$$y_i|\beta, X = x_i \sim x_i^t\beta + \epsilon_i$$

  with ϵ_i i.i.d. $N(0, \sigma_\epsilon^2)$. Here we think of β as random as well, and having a prior distribution

$$\beta \sim N(0, \sigma_\beta^2 I_p)$$

- Then the negative log posterior distribution is proportional to (1), with

$$\lambda = \frac{\sigma_\epsilon^2}{\sigma_\beta^2}$$

  and the posterior mean is the ridge estimator
- The smaller the prior variance parameter σ_β^2, the more the posterior mean is shrunk toward zero, the prior mean for β

Important details

- When including an intercept term, we usually leave this coefficient unpenalized, solving

$$\min_{\alpha,\beta} \|y - 1\alpha - X\beta\|^2 + \lambda\|\beta\|^2$$

- Ridge regression is not invariant under scale transformations of the variables, so it is standard practice to centre each column of X (hence making them orthogonal to the intercept term) and then scale them to have Euclidean norm $\sqrt{n}$
- It is straightforward to show that after this standardisation of X, $\hat{\alpha} = \bar{y}$, so we can also centre y and then remove α from our objective function
- Different `R` packages have different defaults, e.g. `glmnet` also standardizes y

- Let $\tilde{y} = (y - 1\bar{y})$ and $\tilde{X} = (X - 1\bar{x}^t)\text{diag}(1/s)$ be the centered y and standardized X, respectively, with
  - $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$,
  - $\bar{x} = (1/n)X'1$,
  - $s = (s_1, \ldots, s_p)^t$ and $s_j^2 = (1/n)\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2$
- Compute the scaled coefficients

$$\tilde{\beta}_\lambda = (\tilde{X}^t\tilde{X} + \lambda I_p)^{-1}\tilde{X}^t\tilde{y}$$

- Transform back to unscaled coefficients

$$\hat{\beta}_\lambda = \text{diag}(1/s)\tilde{\beta}_\lambda \quad \hat{\alpha} = \bar{y} - \bar{x}^t\hat{\beta}_\lambda$$

Ridge computations and the SVD

Tuning parameter

- In many wide-data and other ridge applications, we need to treat λ as a tuning parameter, and select a good value for the problem at hand.
- For this task we have a number of approaches available for selecting λ from a series of candidate values:
  - With a validation dataset separate from the training data, we can evaluate the prediction performance at each value of λ
  - Cross-validation does this effciently using just the training data, and leave-one-out (LOO) CV is especially efficient

SVD

- Whatever the approach, they all require computing a number of solutions $\hat{\beta}_\lambda$ at different values of λ: the *ridge regularization path*
- We can achieve great efficiency via the (full form) Singular Value Decomposition (SVD)

$$X = UDV^t$$

where U $n \times n$ orthogonal, V $p \times p$ orthogonal and D $n \times p$ diagonal, with diagonal entries $d_1 \geq \ldots \geq d_m \geq 0$, where $m = \min(n, p)$

- From the SVD we get

$$\begin{aligned}\hat{\beta}_\lambda &= (VD^tU^tUDV^t + \lambda VV^t)^{-1}VD^tU^ty \\ &= V(D^tD + \lambda I_p)^{-1}D^tU^ty \\ &= \sum_{d_j>0} v_j \frac{d_j}{d_j^2+\lambda}\langle u_j, y\rangle\end{aligned} \tag{3}$$

  where v_j (u_j) is the jth column of V (U), and $\langle a, b\rangle = a^tb$
- Once we have the SVD of X, we have the ridge solution for all values of λ
- When $n > p$ the ridge solution with $\lambda = 0$ is simply the OLS solution for β
- When $p > n$, there are infinitely many least squares solutions for β, all leading to a zero-residual solution. From (3) with $\lambda = 0$ we get a unique solution, the one with minimum Euclidean norm

– Fitted values

$$\begin{aligned}\hat{y}_\lambda &= U\mathrm{diag}\Big(\frac{d_1^2}{d_1^2+\lambda},\ldots,\frac{d_p^2}{d_p^2+\lambda}\Big)U^t y \\ &= \sum_{d_j>0} u_j \frac{d_j^2}{d_j^2+\lambda}\langle u_j, y\rangle\end{aligned}$$

Principal components regression

- Ridge

$$\hat{\beta}_\lambda = V\text{diag}\Big(\frac{d_1}{d_1^2+\lambda}, \ldots, \frac{d_p}{d_p^2+\lambda}\Big) U^t y$$

- Principal components regression with q components

$$\hat{\beta}_q = V\text{diag}\Big(\frac{1}{d_1}, \ldots, \frac{1}{d_q}, 0, \ldots, 0\Big) U^t y$$

- Both operate on the singular values, but where principal component regression thresholds the singular values, ridge regression shrinks them

Ridge and the bias-variance trade-off

Bias

- Assume that the data arise from a linear model $y \sim N(X\beta, \sigma^2 I_n)$, then $\hat{\beta}_\lambda$ will be a biased estimate of β. Throughout this section X is assumed fixed, $n > p$ and X has full column rank
- The ridge estimator can be expressed as

$$\hat{\beta}_\lambda = (X^t X + \lambda I_p)^{-1} X^t X \hat{\beta}$$

- We can get an explicit expression for the bias

$$\begin{aligned} \text{Bias}(\hat{\beta}_\lambda) &= \mathbb{E}(\hat{\beta}_\lambda) - \beta \\ &= V \text{diag}\left(\frac{\lambda}{d_1^2 + \lambda}, \ldots, \frac{\lambda}{d_p^2 + \lambda} \right) V^t \beta \\ &= \sum_{j=1}^{p} v_j \frac{\lambda}{d_j^2 + \lambda} \langle v_j, \beta \rangle \end{aligned}$$

Variance

- Similarly there is a nice expression for the covariance matrix

$$\begin{aligned} \mathrm{Var}(\hat{\beta}_\lambda) &= \sigma^2 V \mathrm{diag}\left(\frac{d_1^2}{(d_1^2+\lambda)^2}, \ldots, \frac{d_p^2}{(d_p^2+\lambda)^2}\right) V^t \\ &= \sigma^2 \sum_{j=1}^{p} \frac{d_j^2}{(d_j^2+\lambda)^2} v_j v_j^t \end{aligned}$$

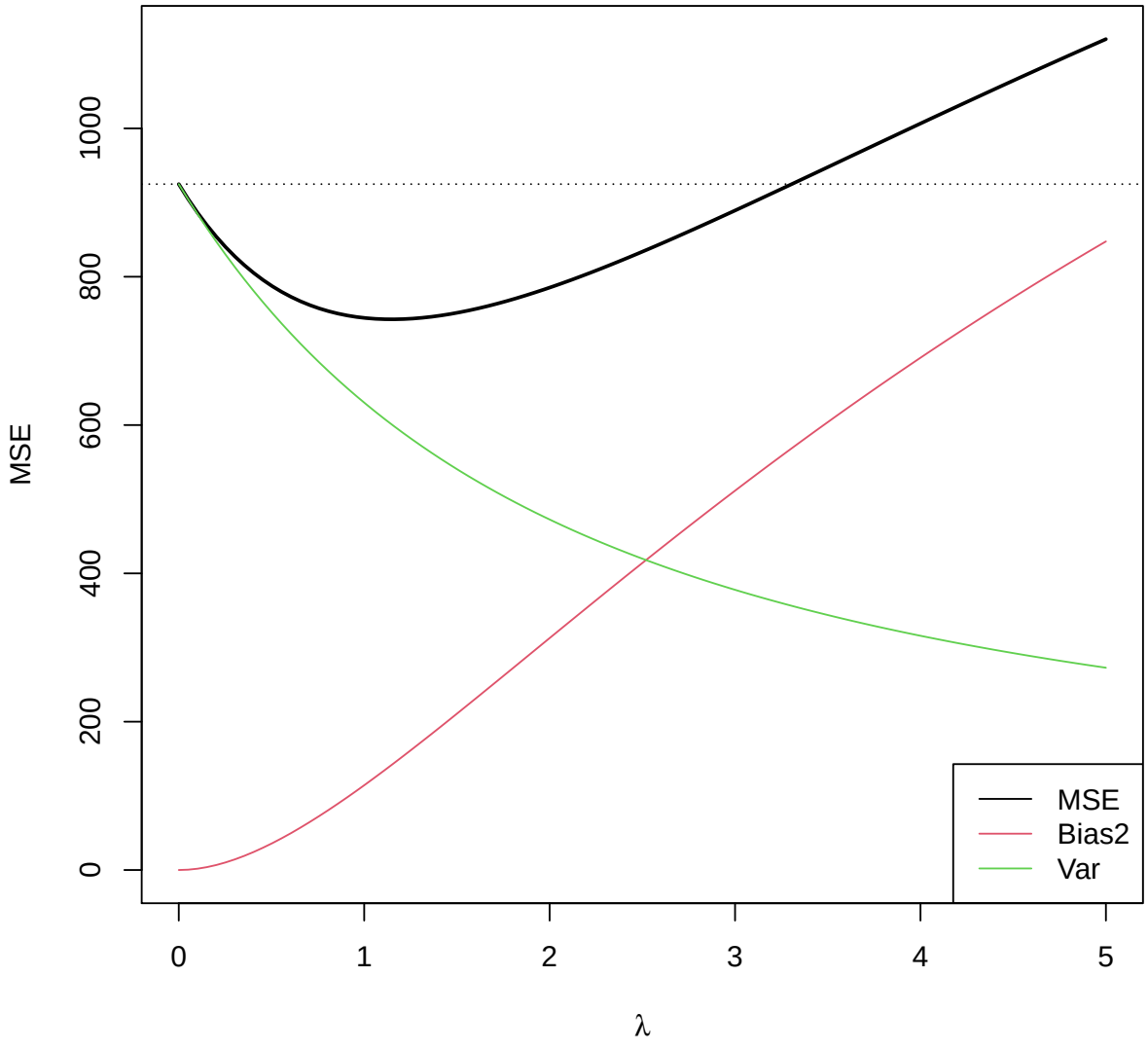
- With $\lambda = 0$, this is $\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^t X)^{-1} \succeq \mathrm{Var}(\hat{\beta}_\lambda)$ for $\lambda > 0$

Mean Squared Error

- MSE of the ridge regression estimator

$$\begin{aligned} \text{MSE}(\hat{\beta}_\lambda) &= \mathbb{E}[(\hat{\beta}_\lambda - \beta)^t(\hat{\beta}_\lambda - \beta)] \\ &= \text{tr}[\text{Var}(\hat{\beta}_\lambda)] + \text{Bias}(\hat{\beta}_\lambda)^t\text{Bias}(\hat{\beta}_\lambda) \end{aligned}$$

- *Theorem (Theobald, 1974)*
  There exists $\lambda > 0$ such that $\text{MSE}(\hat{\beta}_\lambda) < \text{MSE}(\hat{\beta})$.


MSE
0
200
400
600
800
1000
0
1
2
3
4
5
λ
MSE
Bias2
Var

Expected prediction error

- When we make predictions $\hat{y}_i = x_i^t \hat{\beta}_\lambda$ at x_i

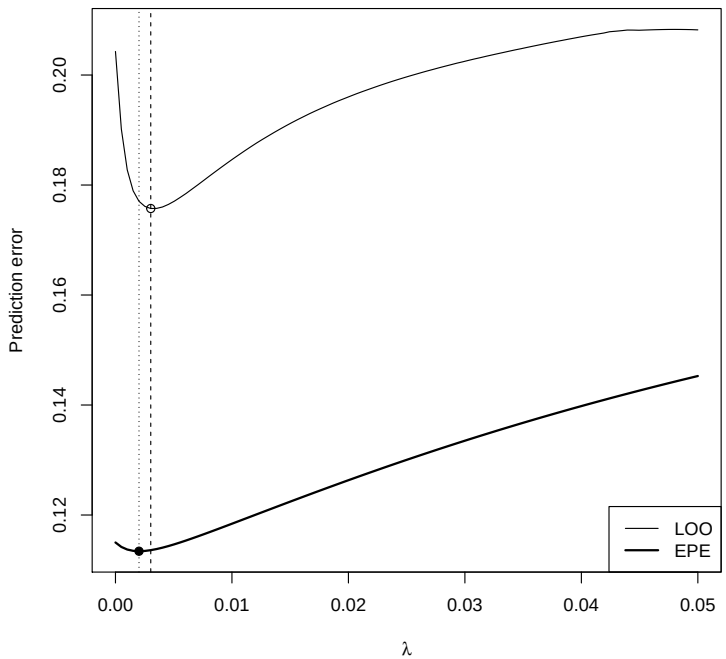
$$\begin{aligned} \text{MSE}(\hat{y}_i) &= \mathbb{E}[(x_i^t \hat{\beta}_\lambda - x_i^t \beta)^2] \\ &= x_i^t \text{Var}(\hat{\beta}_\lambda) x_i + [x_i^t \text{Bias}(\hat{\beta}_\lambda)]^2 \end{aligned}$$

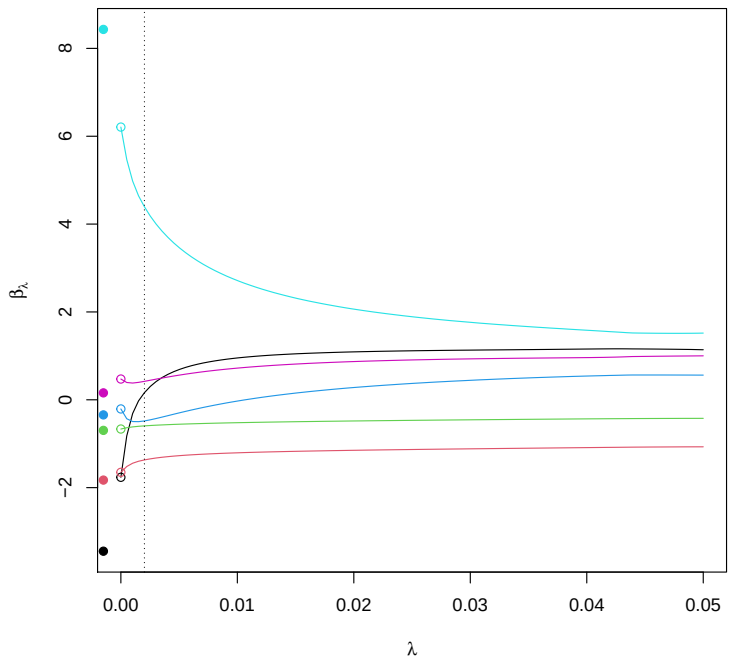
- Expected prediction error

$$\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i^{\text{new}})^2\Big] = \frac{1}{n}\sum_{i=1}^{n}\text{MSE}(\hat{y}_i) + \sigma^2$$

Longley data

$n = 16, p = 6$





Orthonormal design matrix

- Consider an orthonormal design matrix X, i.e. $X^tX = I_p = (X^tX)^{-1}$, e.g.

$$X = \frac{1}{2}\begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

- $\hat{\beta}_\lambda = \frac{1}{(1+\lambda)}\hat{\beta}$
- $\text{Var}(\hat{\beta}_\lambda) = \frac{\sigma^2}{(1+\lambda)^2} I_p$
- $\text{MSE}(\hat{\beta}_\lambda) = \frac{p\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2 \|\beta\|^2}{(1+\lambda)^2}$ with minimum at $\lambda = \frac{p\sigma^2}{\|\beta\|^2}$

Ridge and leave-one-out cross validation

LOO

- For n-fold (LOO) CV, we have another beautiful result for ridge and other linear operators

$$\text{LOO}_\lambda = \frac{1}{n}\sum_{i=1}^n (y_i - x_i^t \hat{\beta}_\lambda^{(-i)})^2 = \frac{1}{n}\sum_{i=1}^n \left(\frac{y_i - x_i^t \hat{\beta}_\lambda}{1 - R_{ii}^\lambda}\right)^2$$

  where $\hat{\beta}_\lambda^{(-i)}$ is the ridge estimate computed using the $(n-1)$ observations with the pair (x_i, y_i) and

$$R^\lambda = X(X^t X + \lambda I)^{-1} X^t$$

- The equation says we can compute all the LOO residuals for ridge from the original residuals, each scaled up by $1/(q - R_{ii}^\lambda)$
- We can obtain R^λ efficiently for all λ via

$$R^\lambda = U \text{diag}\left(\frac{d_1^2}{d_1^2 + \lambda}, \ldots, \frac{d_p^2}{d_p^2 + \lambda}\right) U^t$$

- For each pair (x_i, y_i) left out, we solve

$$\min_{\beta} \sum_{l \neq i} (y_l - x_l^t \beta) + \lambda \|\beta\|^2$$

  with solution $\hat{\beta}_{\lambda}^{(-i)}$.
- Let $y_i^* = x_i^t \hat{\beta}_{\lambda}^{(-i)}$. If we insert the pair (x_i, y_i^*) back into the size $n-1$ dataset, it will not change the solution
- Back at a full n dataset, and using the linearity of the ridge operator, we have

$$y_i^* = \sum_{l \neq i} R_{il}^{\lambda} y_l + R_{ii}^{\lambda} y_i^* = \sum_{l=1}^{n} R_{il}^{\lambda} y_l - R_{ii}^{\lambda} y_i + R_{ii}^{\lambda} y_i^* = \hat{y}_i - R_{ii}^{\lambda} y_i + R_{ii}^{\lambda} y_i^*$$

  from which we see that $(y_i - y_i^*) = (y_i - \hat{y}_i)/(1 - R_{ii}^{\lambda})$

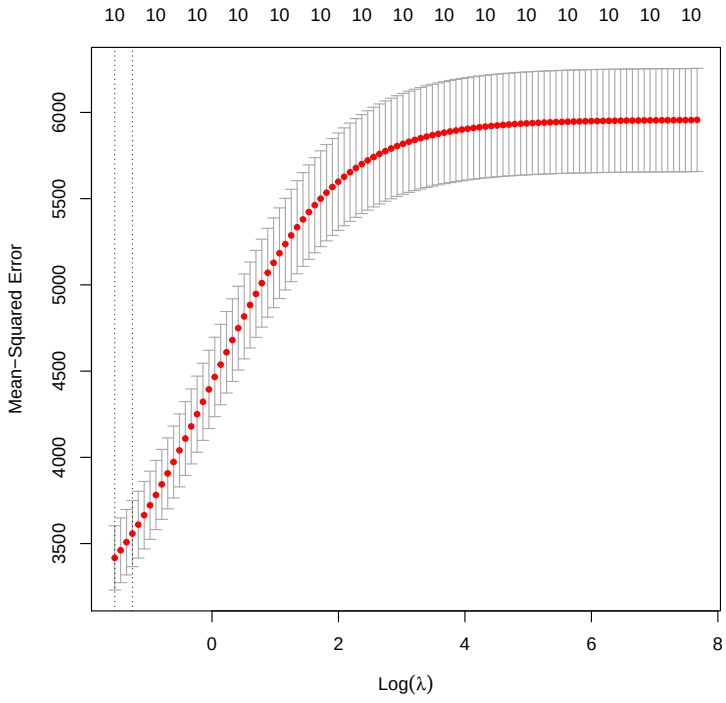
GCV

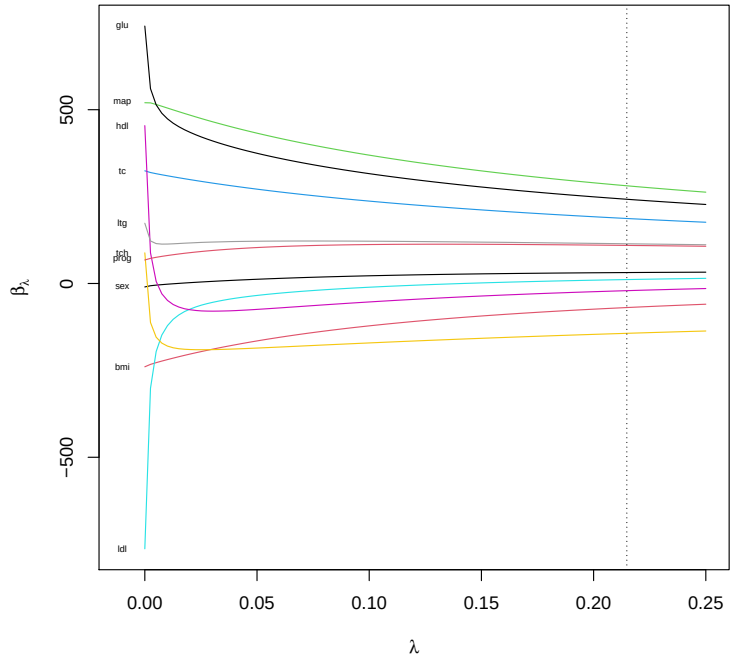
- The identity $\mathrm{tr}(R^\lambda) = \sum_{i=1}^n R_{ii}^\lambda$ suggests $R_{ii}^\lambda \approx \frac{1}{n}\mathrm{tr}(R^\lambda)$
- Generalized cross validation

$$\mathrm{GCV}_\lambda = \frac{1}{n}\sum_{i=1}^n \frac{(y_i - x_i^t\hat{\beta}_\lambda)^2}{(1 - \frac{1}{n}\mathrm{tr}(R^\lambda))^2}$$

Diabetes data

$n = 442, p = 10$





Ridge and the kernel trick

- The fitted values from ridge regression are

$$\hat{y}_\lambda = X(X^tX + \lambda I_p)^{-1}X^ty \tag{4}$$

- An alternative way of writing this is suggested by the following

$$\begin{aligned} X^t(XX^t + \lambda I_n) &= (X^tX + \lambda I_p)X^t \\ (X^tX + \lambda I_p)^{-1}X^t &= X^t(XX^t + \lambda I_n)^{-1} \\ X(X^tX + \lambda I_p)^{-1}X^ty &= XX^t(XX^t + \lambda I_n)^{-1}y \end{aligned}$$

  giving

$$\hat{y}_\lambda = K(K + \lambda I_n)^{-1}y \tag{5}$$

  where $K = XX^t = \{x_i^tx_j\}_{ij}$ is the $n \times n$ *gram* matrix of pairwise inner products, where x_i^t and x_j^t are the ith and jth row of X

- Complexity can be expressed in terms of floating point operations (flops) required to find the solution. (4) requires $O(np^2 + p^3)$ operations, (5) $O(pn^2 + n^3)$ operations

- Suppose we want to add all pairwise interactions

$$
\begin{aligned}
&x_{i1}, x_{i2}, \ldots, x_{ip} \\
&x_{i1}x_{i1}, x_{i1}x_{i2}, \ldots, x_{i1}x_{ip} \\
&\vdots \\
&x_{ip}x_{i1}, x_{ip}x_{i2}, \ldots, x_{ip}x_{ip}
\end{aligned}
$$

  giving $O(p^2)$ columns in the design matrix. Since (5) now requires $O(p^2n^2 + n^3)$ operations, for large p it can be computationally prohibitive
- However, K can be computed directly with

$$K_{ij} = (\frac{1}{2} + x_i^t x_j)^2 - \frac{1}{4} = \sum_k x_{ik}x_{jk} + \sum_{k,l} x_{ik}x_{il}x_{jk}x_{jl}$$

  this amounts to an inner product between vectors of the form

$$(x_{i1}, \ldots, x_{ip}, x_{i1}x_{i1}, \ldots, x_{i1}x_{ip}, x_{i2}x_{i1}, \ldots, x_{i2}x_{ip}, \ldots, x_{ip}x_{ip})$$

  and it requires $O(pn^2)$ operations

Smoothing splines

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Bowman, Evers. Lecture Notes on Nonparametric Smoothing. Section 3
- Eilers, Marx (1996). Flexible smoothing with B-splines and penalties. Statistical science, 11(2), 89–121.

Natural cubic spline

- A set of n points (x_i, y_i) can be exactly interpolated using a natural cubic spline with the $x_1 < \ldots < x_n$ as knots. The interpolating natural cubic spline is unique.
- Amongst all functions on $[a, b]$ which are twice continuously differentiable and which interpolate the set of points (x_i, y_i), a natural cubic spline with knots at the x_i yields the smallest roughness penalty

$$\int_a^b (f''(x))^2 dx$$

- $f''(x)$ is the second derivative of f with respect to x - it would be zero if f were linear, so this measures the curvature of f at x.

Smoothing spline

- Smoothing splines circumvent the problem of knot selection by performing regularized regression over the natural spline basis, placing knots at all inputs $x_1, \ldots, x_n$
- With inputs $x_1 < \ldots < x_n$ contained in an interval $[a, b]$, the minimiser of

$$\hat{f} = \arg\min_{f \in C_2} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 dx$$

  amongst all twice continuously differentiable functions on $[a, b]$ is given by a a natural cubic spline with knots in the unique x_i

- The previous result tells us that we can choose natural cubic spline basis $B_1, \ldots, B_n$ with knots $\xi_1 = x_1, \ldots, \xi_n = x_n$ and solve

$$\hat{\beta}_\lambda = \arg\min_\beta \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \beta_j B_j(x_i))^2 + \lambda \int_a^b \Big(\sum_{j=1}^{n} \beta_j B_j''(x) \Big)^2 dx$$

  to obtain the smoothing spline estimate $\hat{f}(x) = \sum_{i=1}^{n} \hat{\beta}_j B_j(x)$
- Rewriting

$$\hat{\beta}_\lambda = \arg\min_\beta \|y - B\beta\|^2 + \lambda \beta^t \Omega \beta$$

  where $B_{ij} = B_j(x_i)$ and $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$, shows the smoothing spline problem to be a type of generalized ridge regression problem with solution

$$\hat{\beta}_\lambda = (B^t B + \lambda \Omega)^{-1} B^t y$$

- Fitted values in Reinsch form

$$\begin{aligned}\hat{y} &= B(B^tB + \lambda\Omega)^{-1}B^ty \\ &= (I_n + \lambda K)^{-1}y\end{aligned}$$

  where $K = (B^t)^{-1}\Omega B^{-1}$ does not depend on λ, and $S = (I_n + \lambda K)^{-1}$ is the $n \times n$ *smoothing matrix*

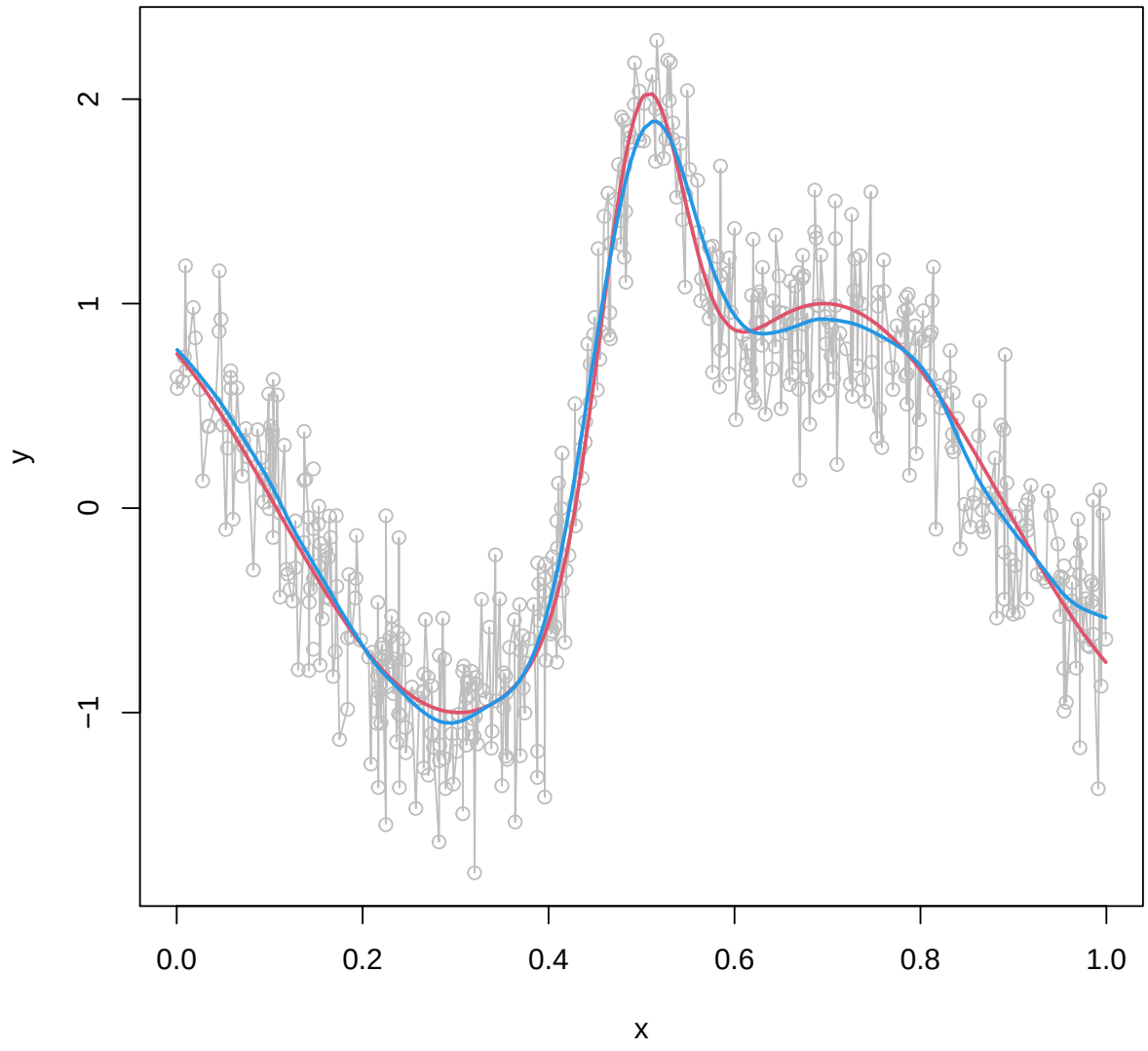
- Leave-one-out cross validation

$$\text{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - S_{ii}}\right)^2$$

- Generalized cross validation

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - \text{tr}(S)/n}\right)^2$$

  where $\text{tr}(S)$ is the effective degrees of freedom



`smooth.spline` result with $\lambda = 0$ and 6.9e-15 by LOO

Reinsch original solution

- The original Reinsch (1967) algorithm solves the constrained optimization problem

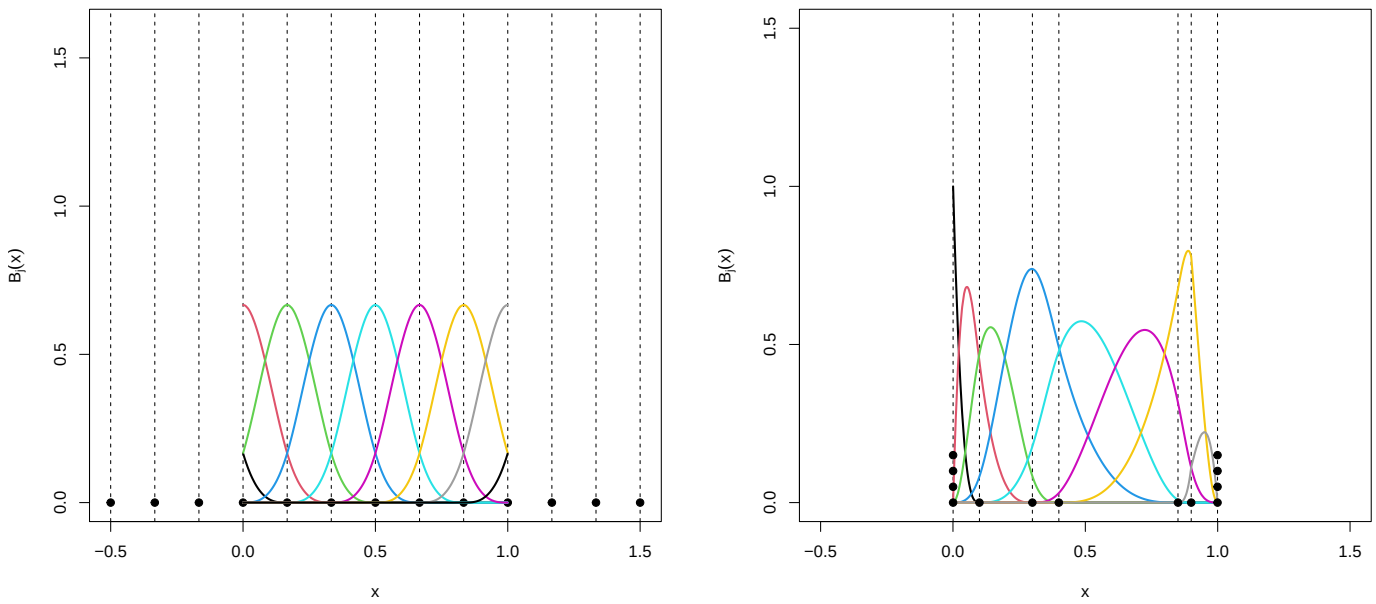
$$\hat{f} = \arg\min_{f \in C_2} \int_a^b (f''(x))^2 dx \text{ such that } \sum_{i=1}^{n} (y_i - f(x_i))^2 \leq c$$

- The previous formulation with a Lagrange parameter on the integral smoothing term instead of the least squares term is equivalent
- See `casl_smspline` implementation in Section 2.6 of CASL

P-splines

B-spline basis

- The truncated power basis suffers from computational issues. The *B*-spline basis is a re-parametrization of the truncated power basis spanning an equivalent space
- The appearance of *B*-splines depends on their knot spacing, e.g.
  - uniform *B*-splines on equidistant knots;
  - non-uniform *B*-splines on unevenly spaced knots and repeated boundary;



Left plot: uniform cubic B-splines with equidistant knots

Right plot: non-uniform cubic B-splines with unevenly spaced knots and duplicated boundary knots

B-spline basis

- B-splines can be computed as differences of truncated power functions
- The general formula for equally-spaced knots is

$$B_j(x) = \frac{(-1)^{M+1}\Delta^{M+1} f_j(x, M)}{h^M M!}$$

satisfying

$$\sum_j B_j(x) = 1$$

where $f_j(x, M) = (x - \xi_j)_+^M$, h is is the distance between knots and Δ^O is the Oth order difference with
$\Delta f_j(x, M) = f_j(x, M) - f_{j-1}(x, M)$,
$\Delta^2 f_j(x, M) = \Delta(\Delta f_j(x, M)) = f_j(x, M) - 2f_{j-1}(x, M) + f_{j-2}(x, M)$

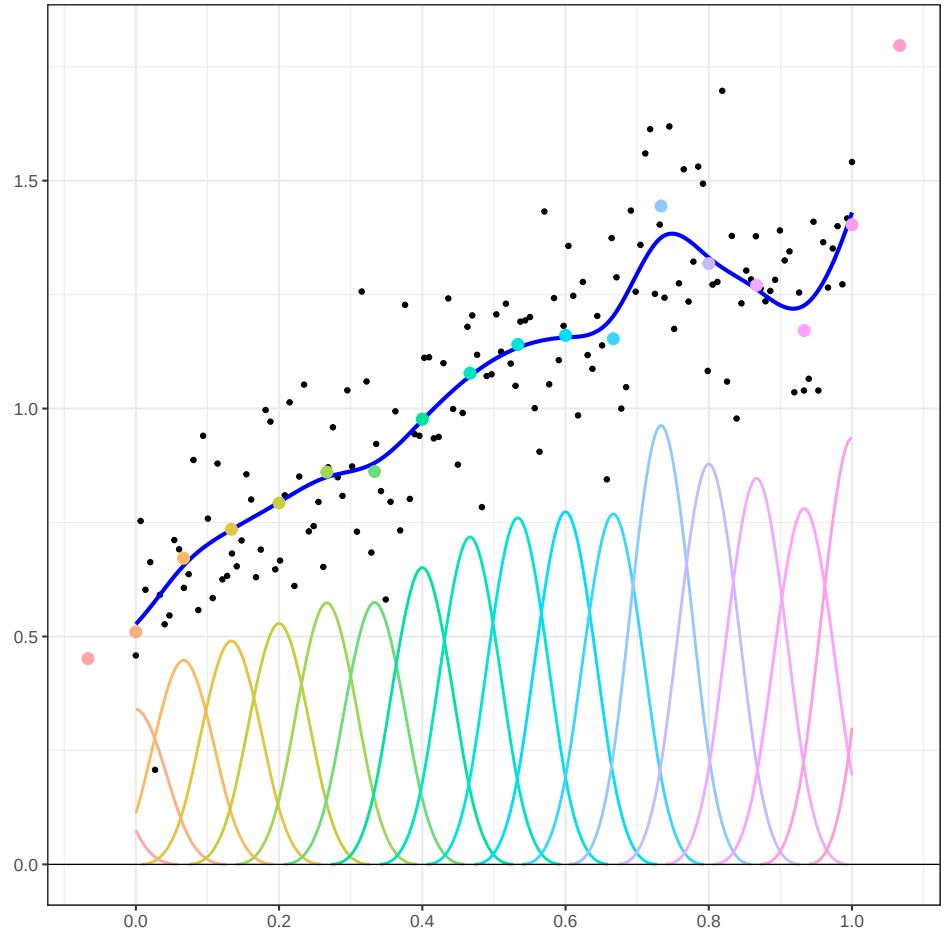
P-splines

- There is an intermediate solution between regression and smoothing splines, proposed more recently by Eilers and Marx (1996)
- P-splines use a basis of (quadratic or cubic) B-splines, B, computed on x and using equally-spaced knots. Minimize

$$\|y - B\beta\|^2 + \lambda\|D\beta\|^2$$

  where $D = \Delta^O$ is the matrix of Oth order differences, with $\Delta\beta_j = \beta_j - \beta_{j-1}$, $\Delta^2\beta_j = \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$ and so on for higher O. Mostly $O = 2$ or $O = 3$ is used.
- Minimization leads to the system of equations

$$(B^tB + \lambda D^tD)\hat{\beta} = B^ty$$



The core idea of *P*-splines: a sum of B-spline basis functions, with gradually changing heights. The blue curve shows the *P*-spline fit, and the large dots the *B*-spline coefficients. R code in `f-ps-show.R`

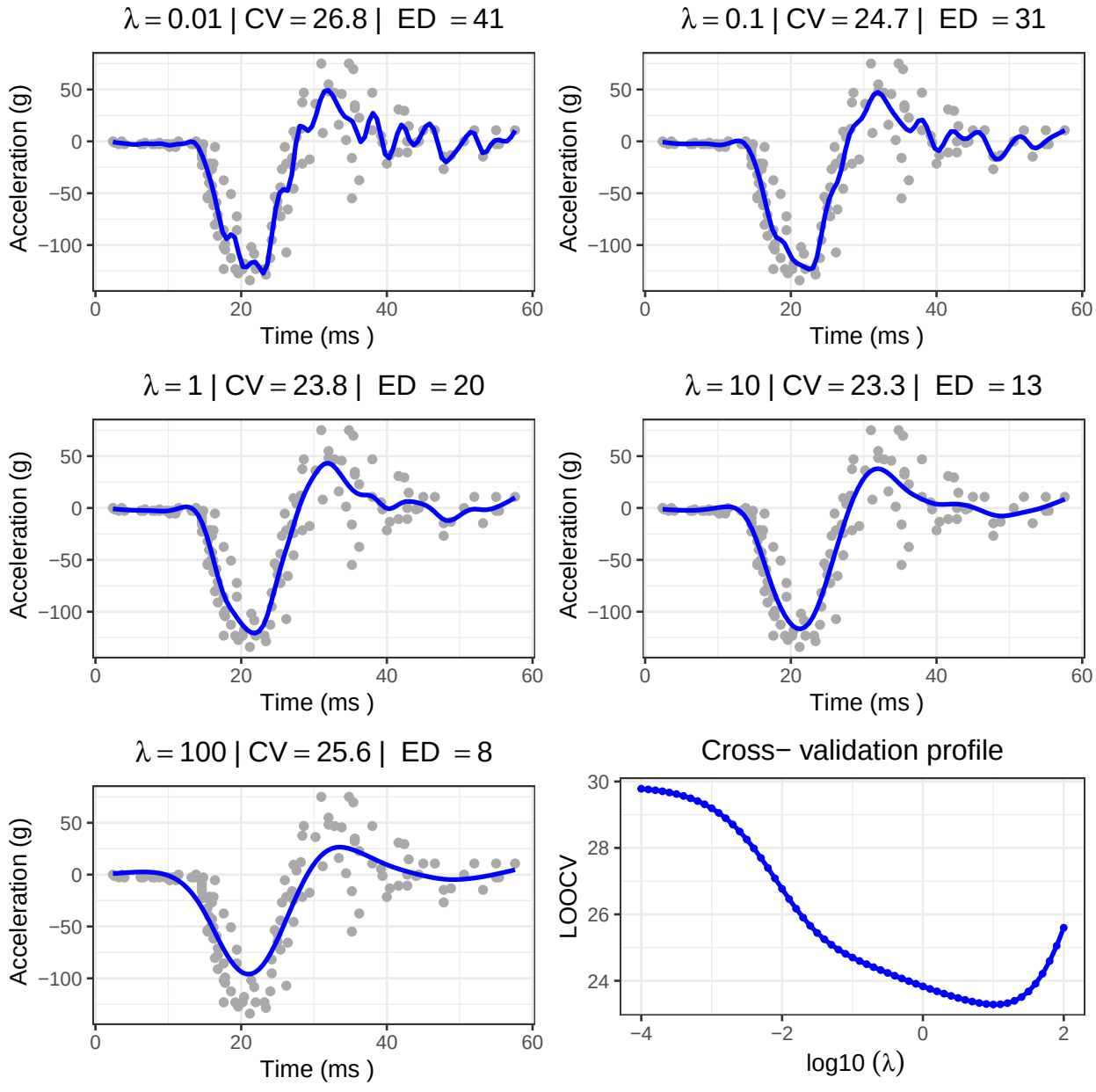
Cross-validation

- We have that $\hat{y} = B(B^tB + \lambda D^tD)^{-1}B^ty = Sy$
- $$\text{LOO} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(-i)})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - S_{ii}}\right)^2$$
- $$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{(1 - \text{tr}(S)/n)^2}$$
- We can compute the trace of R without actually computing its diagonal, using

  $$\text{tr}(S) = \text{tr}((B^tB + P)^{-1}B^tB) = \text{tr}(I_n - (B^tB + P)^{-1}P)$$

  where $P = \lambda D^tD$

mcycle


λ = 0.01 | CV = 26.8 | ED = 41
λ = 0.1 | CV = 24.7 | ED = 31
λ = 1 | CV = 23.8 | ED = 20
λ = 10 | CV = 23.3 | ED = 13
λ = 100 | CV = 25.6 | ED = 8
Cross− validation profile
Acceleration (g)
Time (ms)
LOOCV
log10 (λ)

Sparse Modeling:
Best Subset and the Lasso

Statistical Learning

CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Tibshirani, Wasserman (2017). Sparsity, the Lasso, and Friends. Lecture notes on Statistical Machine Learning

Three norms: ℓ_0, ℓ_1 and ℓ_2

- Let's consider three canonical choices: the ℓ_0, ℓ_1 and ℓ_2 norms:

$$\|\beta\|_0 = \sum_{j=1}^{p} \mathbb{1}\{\beta_j \neq 0\}, \quad \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|, \quad \|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

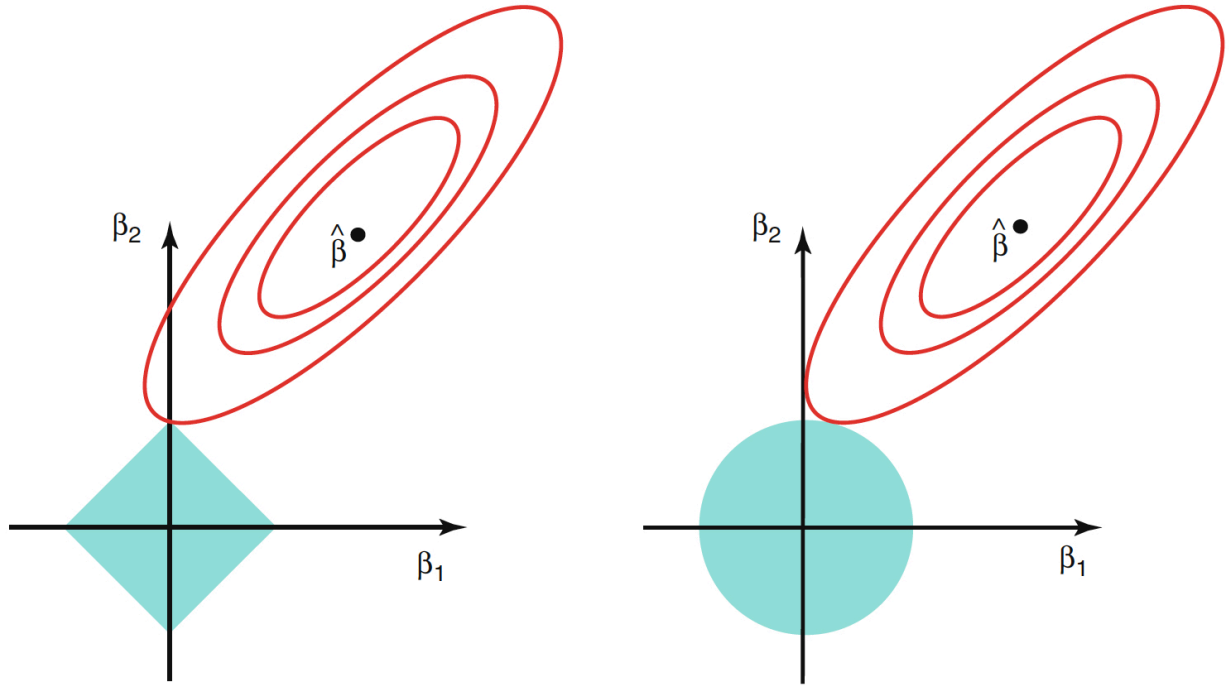
- ℓ_0 is not a proper norm: it does not satisfy positive homogeneity, i.e. $\|a\beta\|_0 \neq |a|\|\beta\|_0$ for $a \in \mathbb{R}$

Constrained form

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_0 \leq c \qquad \text{Best Subset Selection}$$

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_1 \leq c \qquad \text{Lasso Regression}$$

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_2^2 \leq c \qquad \text{Ridge Regression}$$



The “classic” illustration comparing lasso and ridge constraints.
From Chapter 3 of ESL

Sparsity

- *Signal sparsity* is the assumption that only a small number of predictors have an effect, i.e. have $\beta_j \neq 0$
- In this case we would like our estimator $\hat{\beta}$ to be sparse, meaning that $\hat{\beta}_j = 0$ for many components $j \in \{1, \ldots, p\}$
- Sparse estimators are desirable because perform variable selection and improve interpretability of the result
- The best subset selection and the lasso estimators are sparse, the ridge estimator is not sparse

Penalized form

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_0 \qquad \text{Best Subset Selection}$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \qquad \text{Lasso Regression}$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \qquad \text{Ridge Regression}$$

- Suppose that $y \sim N(\mu, 1)$
- ℓ_0 penalty

$$\min_{\mu} \frac{1}{2}(y-\mu)^2 + \lambda \mathbb{1}\{\mu \neq 0\}, \qquad \hat{\mu} = H_{\sqrt{2\lambda}}(y)$$

  where $H_a(y) = y\mathbb{1}\{|y| > a\}$ is the hard-thresholding operator
- ℓ_1 penalty

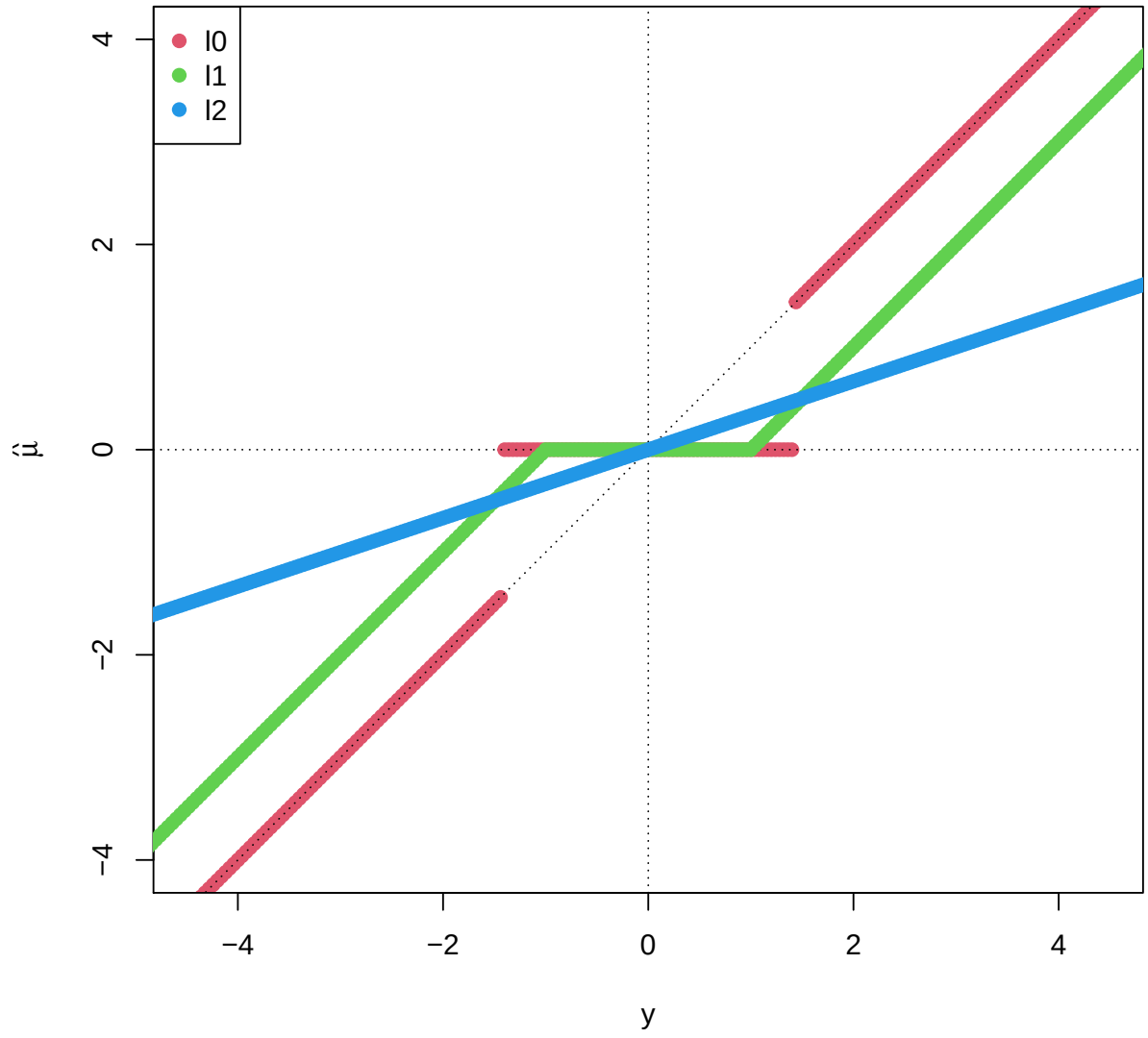
$$\min_{\mu} \frac{1}{2}(y-\mu)^2 + \lambda|\mu|, \qquad \hat{\mu} = S_\lambda(y)$$

  where

$$S_a(y) = \begin{cases} y-a & \text{if } y > a \\ 0 & \text{if } -a \leq y \leq a \\ y+a & \text{if } y < a \end{cases}$$

  is the soft-thresholding operator
- ℓ_2 penalty

$$\min_{\mu} \frac{1}{2}(y-\mu)^2 + \lambda\mu^2, \qquad \hat{\mu} = \left(\frac{1}{1+2\lambda}\right)y$$



$\lambda = 1$

Hard and soft thresholding

- ℓ_0 penalty creates a zone of sparsity but it is discontinuous (hard thresholding)
- ℓ_1 penalty creates a zone of sparsity but it is continuous (soft thresholding)
- ℓ_2 penalty creates a nice smooth estimator but it is never sparse

Orthogonal case

- Suppose $X^tX = I_p$
- OLS estimator

$$\hat{\beta} = X^t y$$

- BSS estimator

$$\hat{\beta} = H_{\sqrt{2\lambda}}(X^t y)$$

- Lasso estimator

$$\hat{\beta} = S_\lambda(X^t y)$$

- Ridge estimator

$$\hat{\beta} = \left(\frac{1}{1+2\lambda}\right) X^t y$$

where $H_a(\cdot)$, $S_a(\cdot)$ are the componentwise hard- and soft-thresholding operators

$$
\begin{aligned}
g(b) &= \frac{1}{2}\|y - Xb\|^2 + \lambda\|b\|_1 \\
&= \frac{1}{2}(y^t y + b^t X^t X b - 2b^t X^t y) + \lambda\|b\|_1 \\
&= \frac{1}{2}y^t y + \frac{1}{2}\sum_{j=1}^{p}\{b_j^2 - 2b_j X^t y + 2\lambda|b_j|\} \\
&= \frac{1}{2}y^t y + \frac{1}{2}\sum_{j=1}^{p} f_j(b_j)
\end{aligned}
$$

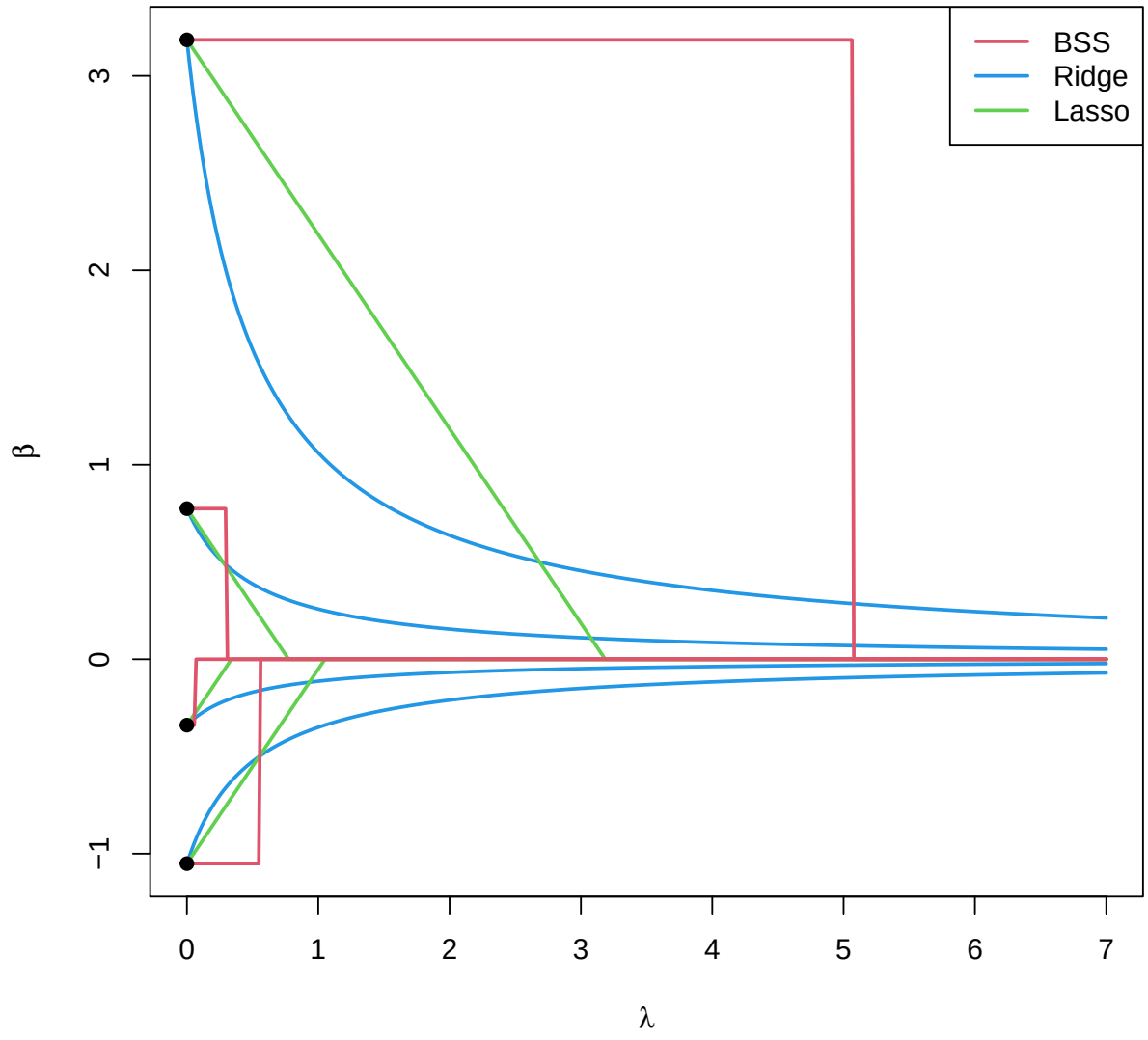
We can minimize each quantity f_j inside the sum independently.

Suppose $b_j \geq 0$ and remove the absolute value:

$$f_j(b_j) = b_j^2 + 2b_j(\lambda - X_j^t y)$$

To minimize this, take the derivative and set it equal to zero:

$$b_j = X_j^t y - \lambda$$



Solution paths of ℓ_0, ℓ_1 and ℓ_2 penalties as a function of λ

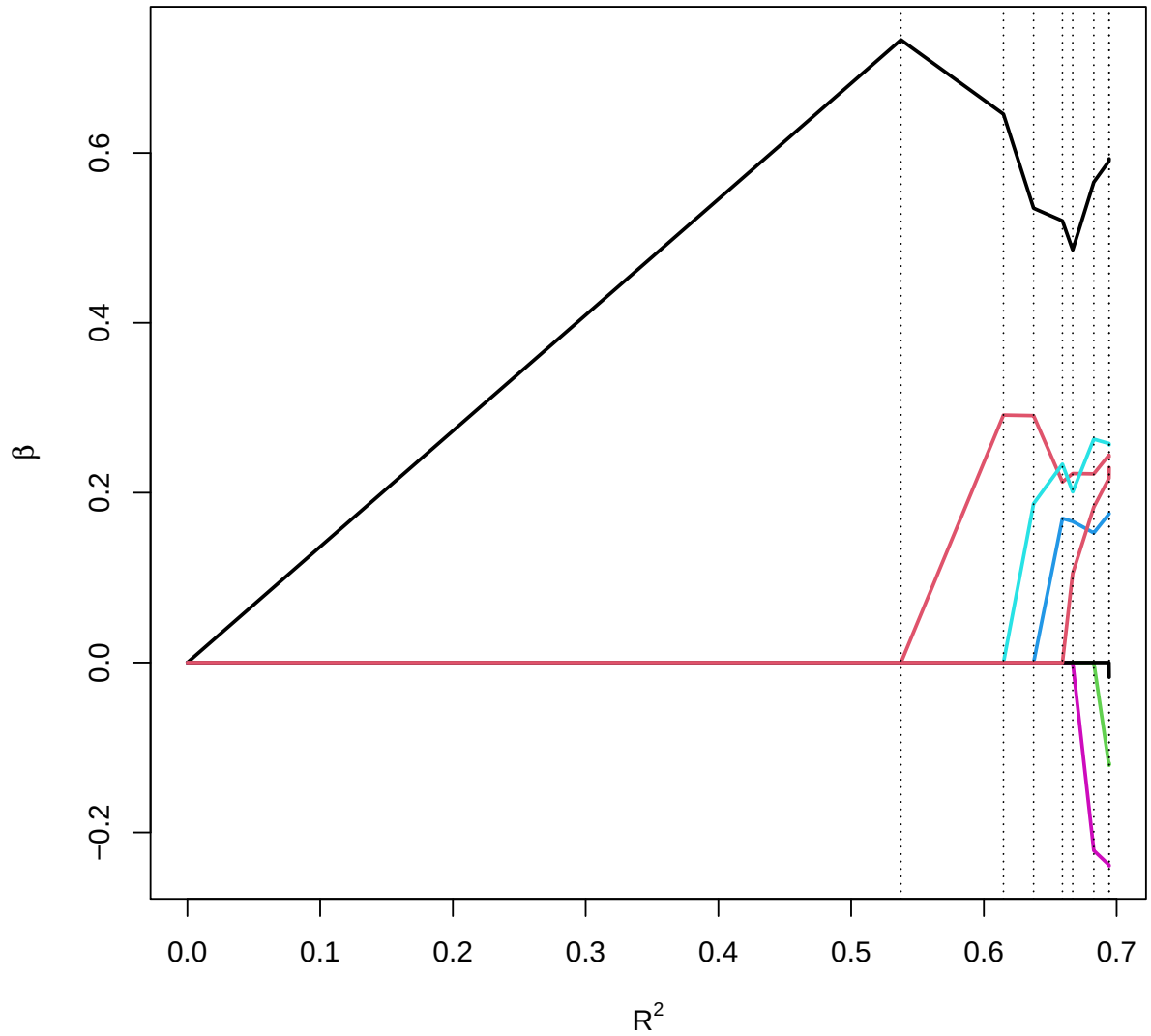
Convexity

- Consider using the norm $\|\beta\|_q = (\sum_{j=1}^{q} |\beta_j|^q)^{1/q}$ as a penalty. Sparsity requires $q \leq 1$ and convexity requires $q \geq 1$. The only norm that gives sparsity and convexity is $q = 1$
- The lasso and ridge regression are *convex optimization problems*, best subset selection is not
- The ridge regression optimization problem is always *strictly convex* for $\lambda > 0$
- The best subset selection optimization problem is N-P-complete because of its combinatorial complexity (there are 2^p subsets), the worst kind of non convex problem

Forward Stepwise Selection

Greedy forward algorithm, sub-optimal but feasible alternative to BSS and applicable when $p > n$

- Set S_0 as the null model (intercept only)
- For $k = 0, \ldots, \min(n-1, p-1)$:
  1. Consider all $p - k$ models that augment the predictors in S_k with one additional predictor
  2. Choose the best among these $p - k$ models and call it S_{k+1}, where best is defined having the smallest RSS
- Select a single best model from among $S_0, S_1, S_2, \ldots$ (e.g. using Cp, BIC, Cross-Validation, validation set, etc.)



Forward Stepwise solution path as a function of training R^2

The Lasso

The name “lasso” was also introduced as an acronym for *Least Absolute Selection and Shrinkage Operator* (Tibshirani, 1996)

The lasso finds the solution $(\hat{\alpha}, \hat{\beta})$ to the optimization problem

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - 1\alpha - X\beta\|_2^2 + \lambda \|\beta\|_1$$

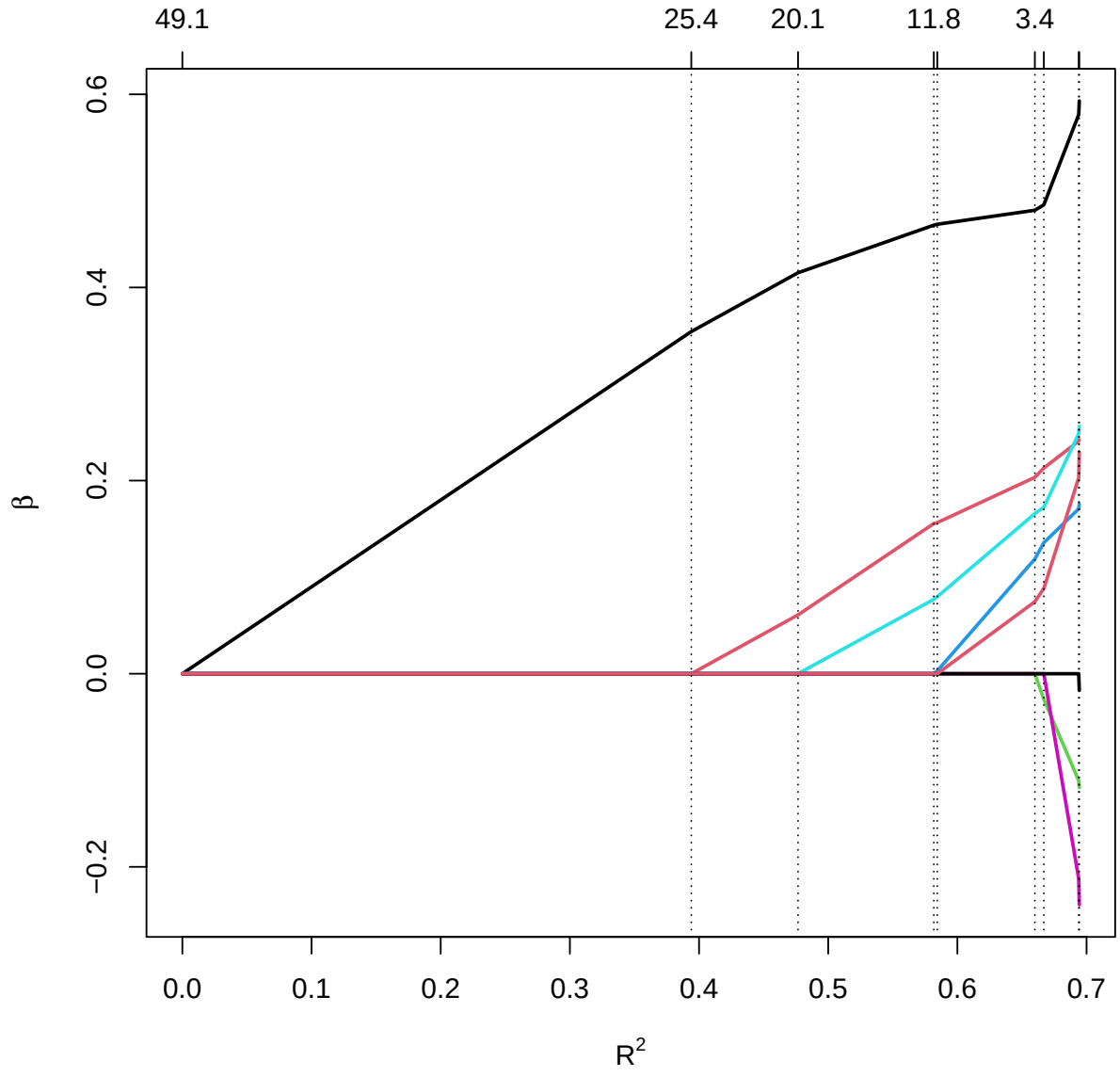
- Typically, we first standardize the predictors X so that each column is centered ($(1/n)\sum_{i=1}^{n} x_{ij} = 0$) and has unit variance ($(1/n)\sum_{i=1}^{n} x_{ij}^2 = 1$)
- Without standardization, the lasso solutions would depend on the units (e.g., feet versus meters) used to measure the predictors. On the other hand, we typically would not standardize if the features were measured in the same units
- For convenience, we also assume that the outcome values y_i have been centered ($(1/n)\sum_{i=1}^{n} y_i = 0$). Centering is convenient, since we can omit the intercept term α in the lasso optimization, and given the solution $\hat{\beta}$

$$\hat{\alpha} = \bar{y} - \sum_{j=1}^{p} \bar{x}_j \hat{\beta}_j$$

- Lagrange form

$$\frac{1}{2n} \min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

- Intercept term omitted (center / scale the columns of X and y)
- The coefficient profiles for the lasso are continuous and piecewise linear over the range of λ, with knots occurring whenever the *active set* changes, or the sign of the coefficients changes



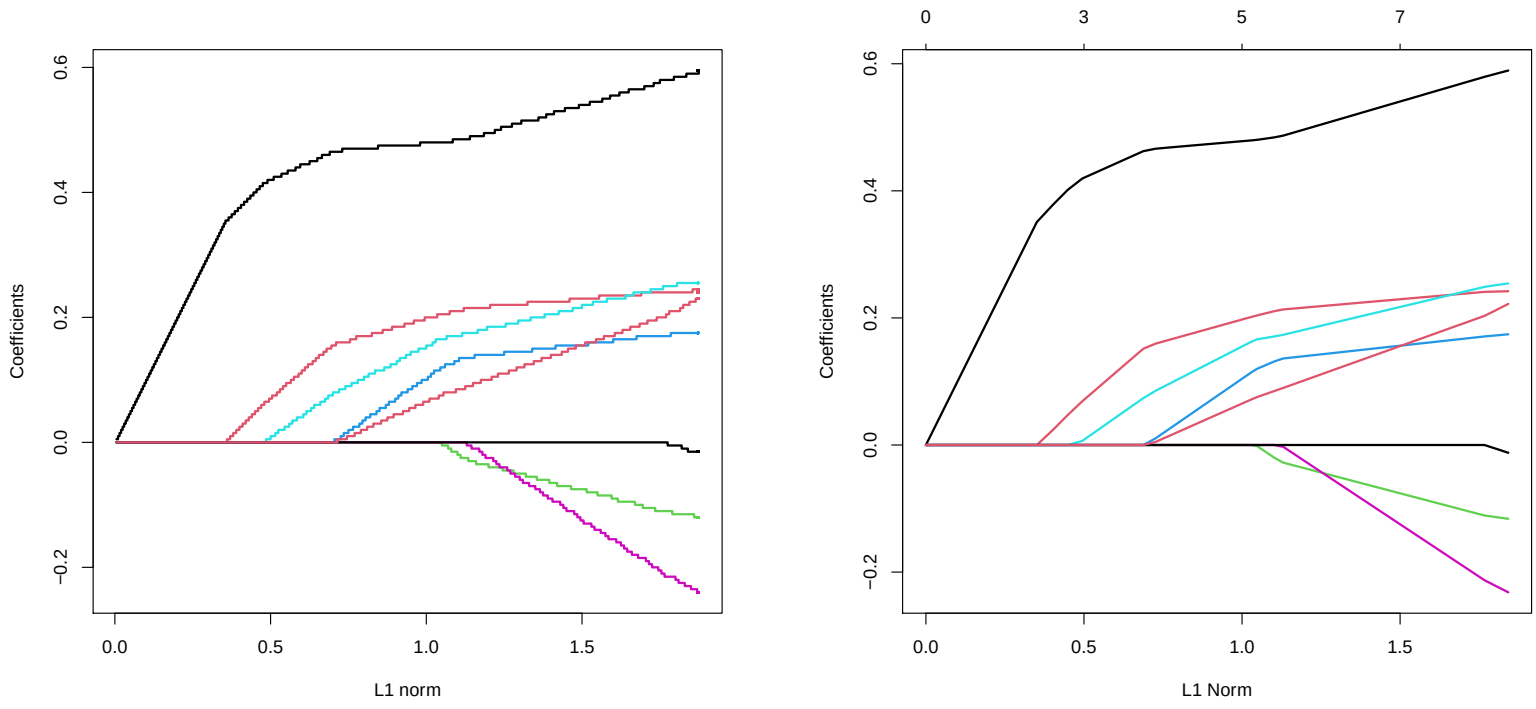
Lasso solution path as a function of training R^2

Boosting with componentwise linear least squares

- Response and predictors are standardized to have mean zero and unit norm
- Initialize $\hat{\beta}^{(0)} = 0$
- For $b = 1, \ldots, B$
  - compute the residuals $r = y - X\hat{\beta}^{(b-1)}$
  - find the predictor X_j most correlated with the residuals r
  - update $\hat{\beta}^{(b-1)}$ to $\hat{\beta}^{(b)}$ with

$$\hat{\beta}_j^{(b)} = \hat{\beta}_j^{(b-1)} + \epsilon \cdot s_j$$

    where s_j is the sign of the correlation
- This is known as *forward stagewise regression* and converges to the least squares solution when $n > p$
- Forward stagewise regression with infinitesimally small step-sizes, i.e. $\epsilon \to 0$, produces a set of solutions which is approximately equivalent to the set of Lasso solutions



Left: forward stagewise regression with $\epsilon = 0.005$; Right: lasso

Convex optimization and the elastic net

Convex function

The objective function of the ℓ_1 penalty, unlike the ℓ_0 penalty, is a continuous function in the regression vector. Not only is the objective function continuous, it is also convex.

We say that a function $f : \mathbb{R}^p \mapsto \mathbb{R}$ *convex* if for any values b_1 and b_2 and quantity $t \in [0, 1]$ we have

$$f(tb_1 + (1 - t)b_2) \leq tf(b_1) + (1 - t)f(b_2)$$

Replacing the $\leq$ with $<$ for $t \in (0, 1)$ yields a definition of *strict* convexity.

The ℓ_1-penalized objective function is in fact convex (but not strictly convex, which makes the solutions non-linear in the y_i, and there is no closed form solution as in ridge).

Convex optimization

A convex function does not have any local minima that are not also global minima. In other words, if the value b_0 minimizes f over a neighborhood of b_0, it must also minimize f over its entire domain.

To see this, assume that b_1 is any point that is not a global minima but set b_2 equal to a global minima of f. Then, for any $t \in [0, 1)$, we have

$$tf(b_1) + (1 - t)f(b_2) < tf(b_1) + (1 - t)f(b_1) = f(b_1)$$

and by convexity this implies that

$$f(tb_1 + (1 - t)b_2) < f(b_1)$$

For any neighborhood around b_1 we can find t close enough to 1 such that $tb_1 + (1 - t)b_2$ is in that neighborhood, and therefore b_1 cannot be a local minimum.

The lack of local optima makes it possible to optimize convex objective functions using e.g. the *coordinate descent algorithm*.

Elastic net

Define the objective function f for some $\lambda > 0$ and $\alpha \in [0, 1]$ as

$$f(\beta; \lambda, \alpha) = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda \left((1-\alpha)\frac{1}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1 \right)$$

and the corresponding *elastic net* estimator as

$$\hat{\beta}_{\lambda,\alpha} = \arg\min_{\beta} f(\beta; \lambda, \alpha)$$

Setting α to 1 yields the Lasso regression and setting it to 0 the ridge regression.

Adding a small ℓ_2-penalty preserves the variable selection and convexity properties of the ℓ_1-penalized regression, while reducing the variance of the model when X contains sets of highly correlated variables.

Coordinate descent

Coordinate descent is a general purpose convex optimization algorithm particularly well-suited to solving the elastic net equation.

Coordinate descent successively minimizes the objective function along each variable. In every step all but one variable is held constant and a value for the variable of interest is chosen to minimize the constrained problem.

This process is applied iteratively over all of the variables until the algorithm converges.

Writing the problem in terms of the individual values of b:

$$f(b) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}b_j)^2 + \lambda\sum_{j=1}^{p}\{\frac{1}{2}(1-\alpha)b_j^2 + \alpha|b_j|\}$$

Let $\tilde{b}$ be a vector of candidate values of the regression vector b and assume $\tilde{b}_l > 0$. Then the derivative of this function with respect to b_l at $b = \tilde{b}$ is

$$\begin{aligned}
\frac{\partial f}{\partial b_l}|_{b=\tilde{b}} &= -\frac{1}{n}\sum_{i=1}^{n} x_{il}(y_i - \sum_{j\neq l}^{p} x_{ij}\tilde{b}_j - x_{il}\tilde{b}_l) + \lambda((1-\alpha)\tilde{b}_l + \alpha) \\
&= -\frac{1}{n}\sum_{i=1}^{n} x_{il}(y_i - \tilde{y}_i^{(l)}) + \frac{1}{n}\sum_{i=1}^{n} x_{il}^2\tilde{b}_l + \lambda(1-\alpha)\tilde{b}_l + \lambda\alpha
\end{aligned}$$

where $\tilde{y}_i^{(l)} = \sum_{j\neq l}^{p} x_{ij}\tilde{b}_j$ is the contribution of all regressors in the model except the lth.

By setting the function to zero and solving for $\tilde{b}_l$ resulting in

$$\tilde{b}_l = \frac{\frac{1}{n}\sum_{i=1}^{n} x_{il}(y_i - \tilde{y}_i^{(l)}) - \lambda\alpha}{\frac{1}{n}\sum_{i=1}^{n} x_{il}^2 + \lambda(1-\alpha)}$$

We can then generalize this equation using the soft-thresholding function as

$$\tilde{b}_l \leftarrow \frac{S_{\lambda\alpha}(\frac{1}{n}\sum_{i=1}^{n} x_{il}(y_i - \tilde{y}_i^{(l)}))}{\frac{1}{n}\sum_{i=1}^{n} x_{il}^2 + \lambda(1-\alpha)}$$

The algorithm updates each of the values of $\tilde{b}$ based on the initial guess of a $p \times 1$ vector of zeros.

KKT conditions

The solution must satisfy the subgradient / Karush-Kuhn-Tuker conditions

$$\begin{aligned} -\frac{1}{n}\langle X_j, y - X\hat{b}\rangle + \lambda s_j &= 0 \\ \sum_{i=1}^{n} x_{ij}(y_i - \sum_{j=1}^{p} x_{ij}\hat{b}_j) &= \lambda s_j \end{aligned}$$

for $j = 1, \ldots, p$, where

$$s_j \in \begin{cases} 1 & \text{if } \hat{b}_j > 0 \\ \text{[-1,1]} & \text{if } \hat{b}_j = 0 \\ -1 & \text{if } \hat{b}_j < 0 \end{cases}$$

which means that if these conditions have not been met, then our $\hat{b}$ vector cannot be optimal.

Cross-validation

Maximum λ

What values of λ should we use when performing cross-validation? Consider the case that $\alpha = 1$. From the coordinate descent updates

$$\tilde{b}_j = \frac{1}{n} X_j^t y - \lambda$$

It follows that if $|X_j^t y| \leq n\lambda$ then $\tilde{b}_j = 0$. The smallest value of λ for which all $\tilde{b}_j$ are zero is therefore:

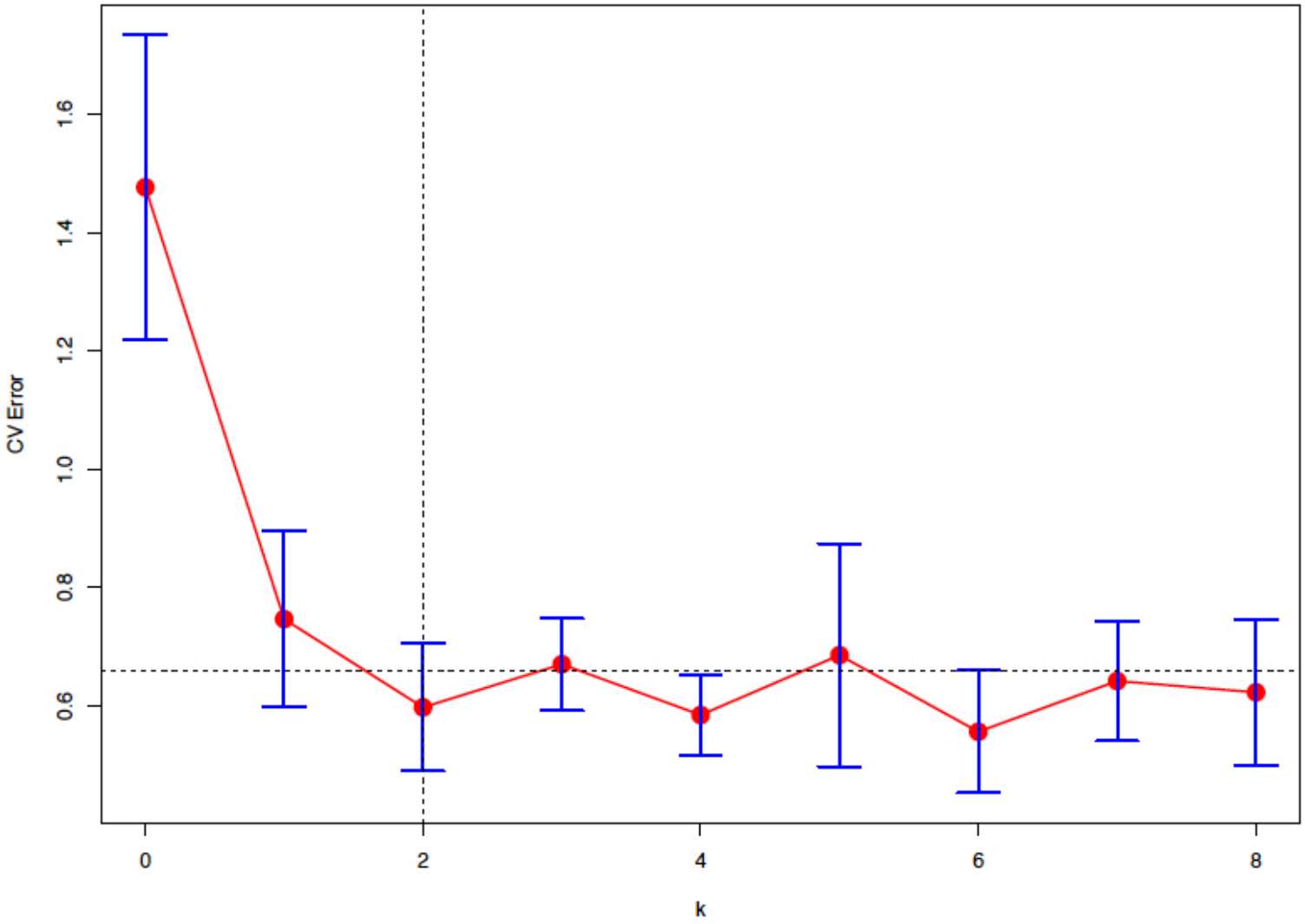
$$\lambda_{max} = \max_{j \in \{1,\ldots,p\}} \left| \frac{X_j^t y}{n} \right|$$

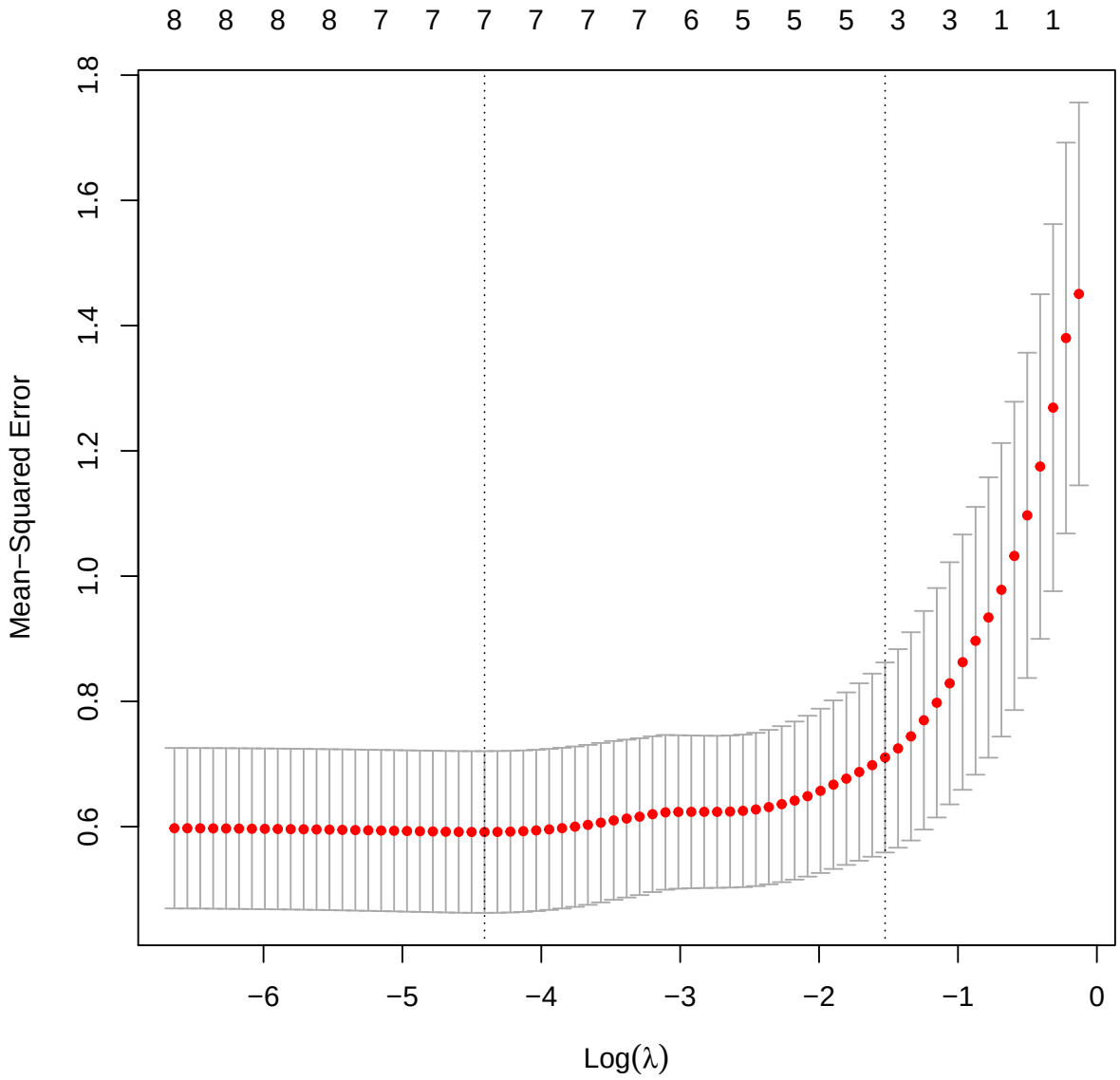
Minimum CV error and the 1se rule

- `lambda.min`: λ that minimize the cross-validation error
- `lambda.1se`: largest value of lambda such that error is within 1 standard error of the minimum (*one standard error rule*). To compute cross-validation "standard errors"

$$\text{se} = \frac{1}{\sqrt{K}} \text{sd}(\text{Err}^{-1}, \ldots, \text{Err}^{-K})$$

where Err^{-k} denotes the error incurred in predicting the observations in the k hold-out fold, $k = 1, \ldots, K$.


CV Error
0.6
0.8
1.0
1.2
1.4
1.6
0
2
4
6
8
k



$\lambda_{\min} = 0.012$ (7 nonzero), $\lambda_{1\text{se}} = 0.21$ (3 nonzero)

Degrees of freedom

- Let $A(\lambda) = \{j \in \{1, \ldots, p\} : \hat{\beta}_j(\lambda) \neq 0\}$ denotes the active set
- The degrees of freedom of the Lasso are the

$$\mathrm{df}(\lambda) = |A(\lambda)|$$

  i.e. the size of the active set

Bayesian interpretation

- A Bayesian viewpoint assumes that β has a double-exponential (Laplace) prior distribution with mean zero and scale parameter a function of λ

$$(1/2\tau)\exp(-\|\beta\|_1/\tau)$$

  with $\tau = 1/\lambda$
- It follows that the posterior mode for β is the lasso solution
- However, the lasso solution is not the posterior mean and, in fact, the posterior mean does not yield a sparse coefficient vector

Extensions of the lasso

Group Lasso

- Suppose we have a partition $G_1, \ldots, G_q$ of $\{1, \ldots, p\}$
- The group Lasso penalty (Yuan and Lin, 2006) is given by

$$\lambda \sum_{k=1}^{q} m_k \|\beta_{G_k}\|_2$$

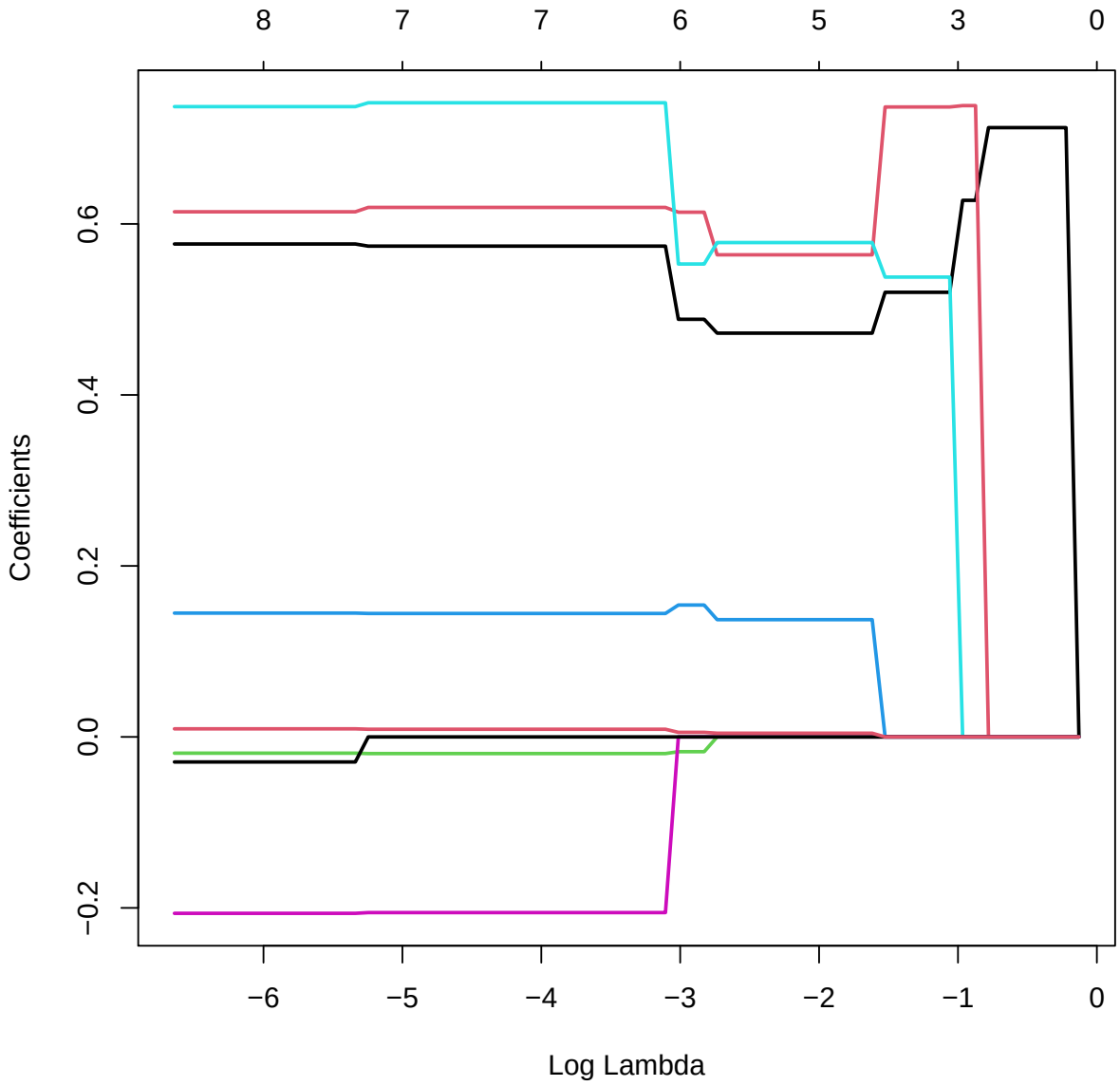
  The multipliers $m_k > 0$ serve to balance cases where the groups are of very different sizes; typically we choose $m_k = \sqrt{|G_k|}$
- This penalty encourages either an entire group G to have $\hat{\beta}_G = 0$ or $\hat{\beta}_j \neq 0$ for all $j \in G$
- Such a property is useful when groups occur through coding for categorical predictors or when expanding predictors using basis functions.

Relaxed Lasso

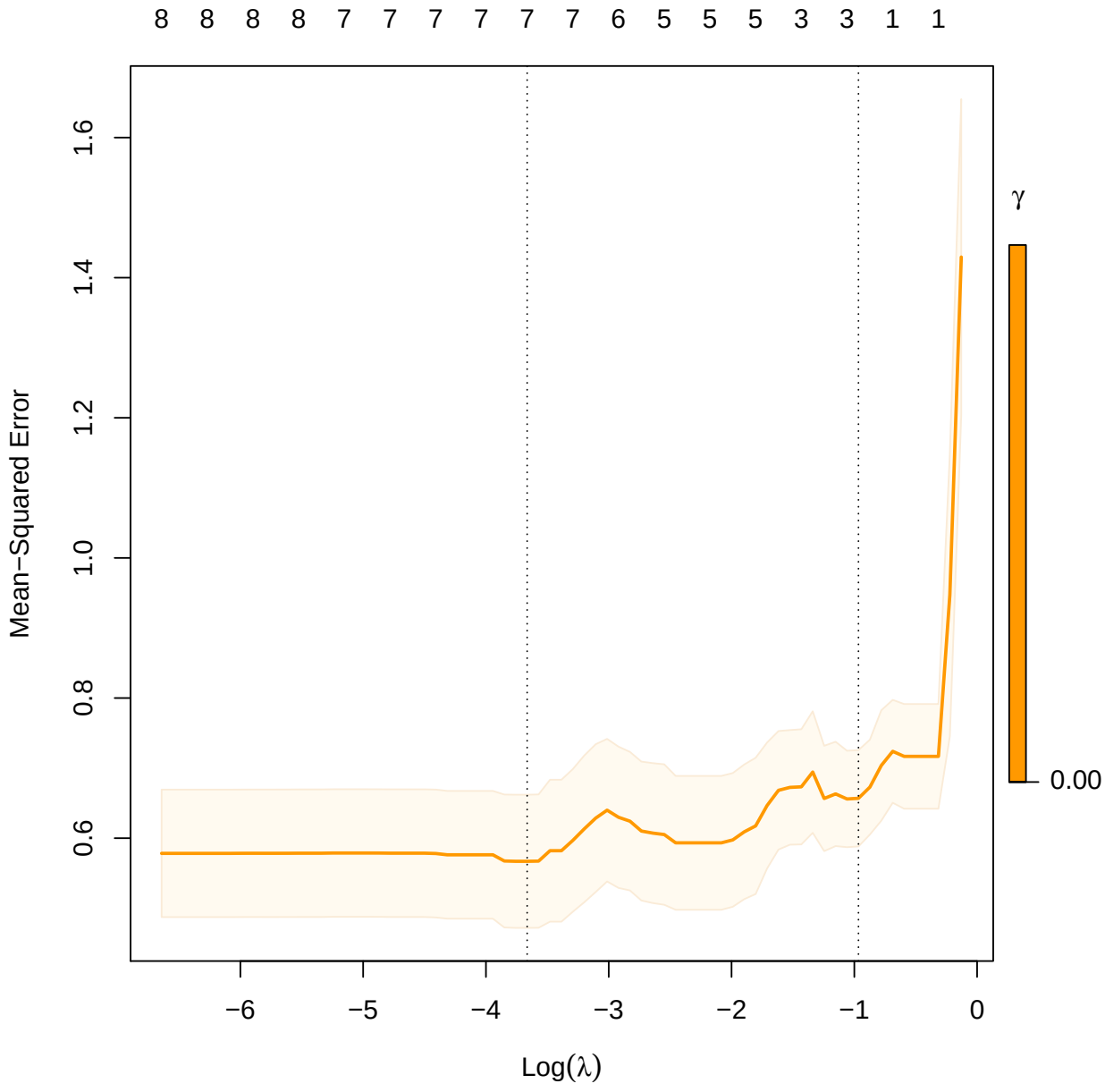
- Originally proposed by Meinshausen (2006). We present a simplified version.
- Suppose $\hat{\beta}_\lambda$ is the lasso solution at λ and let $\hat{A}$ be the active set of indices with nonzero coefficients in $\hat{\beta}_\lambda$
- Let $\hat{\beta}^{\text{LS}}$ be the coeffcients in the least squares fit, using only the variables in $\hat{A}$. Let $\hat{\beta}_\lambda^{\text{LS}}$ be the full-sized version of this coeffcient vector, padded with zeros. $\hat{\beta}_\lambda^{\text{LS}}$ debiases the lasso, while maintaining its sparsity.
- Define the Relaxed Lasso

$$\hat{\beta}_\lambda^{\text{RELAX}} = \gamma\hat{\beta}_\lambda + (1-\gamma)\hat{\beta}_\lambda^{\text{LS}}$$

with $\gamma \in [0, 1]$ is an additional tuning parameter which can be selected by cross-validation



$\gamma = 0$


8 8 8 8 7 7 7 7 7 7 6 5 5 5 3 3 1 1
Mean−Squared Error
0.6
0.8
1.0
1.2
1.4
1.6
γ
0.00
−6
−5
−4
−3
−2
−1
0
Log(λ)

$\gamma = 0$

Data splitting for variable selection

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Dezeure, Buhlmann, Meier, Meinshausen (2015). High dimensional inference: Confidence intervals, p-values and r-software `hdi`. Statistical Science, 533–558

High-dimensional inference

- Consider the gaussian linear model

$$y \sim N_n(1_n\beta_0 + X\beta, \sigma^2 I_n)$$

  with $n \times p$ design matrix X and $p \times 1$ vector of coefficients β
- When $p \geq n$, classical approaches for estimation and inference of β cannot be directly applied
- How to perform inference on β (e.g. confidence intervals and p-values for individual regression parameters β_j, $j = 1, \ldots, p$) in a high-dimensional setting?

Support set

- The *support set* is

$$S = \{j \in \{1, \ldots, p\} : \beta_j \neq 0\}$$

  with cardinality $s = |S|$, and its complement is the *null set*, i.e.

$$N = \{j \in \{1, \ldots, p\} : \beta_j = 0\}$$

- Let $\hat{S} \subseteq \{1, \ldots, p\}$ be an estimator of S. Then

$$|\hat{S} \cap N|$$

  is the number of the wrong selections (type I errors) and

$$|S \setminus \hat{S}|$$

  is the number of wrong deselections (type II errors)

Error rates

- Define the *False Discovery Proportion* (FDP) by

$$\mathrm{FDP}(\hat{S}) = \frac{|\hat{S} \cap N|}{|\hat{S}|}$$

  with $\mathrm{FDP}(\emptyset) = 0$
- *FamilyWise Error Rate* (FWER)

$$\mathrm{P}(\mathrm{FDP}(\hat{S}) > 0) = \mathrm{P}(\hat{S} \cap N \neq \emptyset)$$

- *False Discovery Rate* (FDR)

$$\mathbb{E}(\mathrm{FDP}(\hat{S}))$$

Error control

- We would like to *control* the chosen error rate at level α, i.e.

$$\mathrm{P}(\hat{S} \cap N \neq \emptyset) \leq \alpha \quad \text{or} \quad \mathbb{E}(\mathrm{FDP}(\hat{S})) \leq \alpha$$

  while maximizing some notion of power e.g. the average power

$$\mathrm{AvgPower} = \frac{\sum_{j \in S} \mathrm{P}(\hat{S} \in j)}{|S|}$$

- We are dealing with the trade-off between type I and type II errors, and since FWER is more stringent than FDR, i.e.

$$\mathbb{E}(\mathrm{FDP}(\hat{S})) \leq \mathrm{P}(\hat{S} \cap N \neq \emptyset)$$

  methods that control FWER are less powerful

Simulate data as described in Section 3.1 of Hastie et al. (2020)

Given n (number of observations), p (problem dimensions), s (sparsity level), beta-type (pattern of sparsity), ρ (predictor autocorrelation level), and ν (signal-to-noise ratio (SNR) level)

1. we define coefficients $\beta \in \mathbb{R}^p$ according to s and the beta-type; e.g. beta-type 2: β has its first s components equal to 1, and the rest equal to 0
2. we draw the rows of the predictor matrix $X \in \mathbb{R}^{n \times p}$ i.i.d. from $N_p(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{p \times p}$ has entry (i, j) equal to $\rho^{|i-j|}$ (Toeplitz matrix)
3. we draw the response vector $y \in \mathbb{R}^n$ from $N_n(X\beta, \sigma^2 I_n)$ with σ^2 defined to meet the desired SNR level, i.e. $\sigma^2 = \beta^t \Sigma \beta / \nu$

Lasso active set

Lasso with λ chosen by e.g. the 1-se rule

$$\hat{S} = \{j \in \{1, \ldots, p\} : \hat{\beta}_j \neq 0\}$$

Simulated data with $n = 200$, $p = 1000$, $s = 10$, $\rho = 0$, $\nu = 2.5$:

Size $\lvert\hat{S}\rvert$	# Type I $\lvert\hat{S} \cap N\rvert$	# Type II $\lvert S \setminus \hat{S}\rvert$	FDP $\lvert\hat{S} \cap N\rvert / \lvert\hat{S}\rvert$	Sensitivity $\lvert\hat{S} \cap S\rvert / \lvert S\rvert$
23	13	0	56.5%	100%

100 replications

	1	2	3	4	5	6	7
Size	23	20	13	25	23	21	11
# Type I	13	10	3	15	13	11	4
# Type II	0	0	0	0	0	0	3
FDP	0.57	0.50	0.23	0.60	0.57	0.52	0.36
Sensitivity	1	1	1	1	1	1	0.7

FWER = 99%, FDR = 54.2%, AvgPower = 99.6%

Naïve two-step procedure

1. Perform the lasso in order to obtain the active set

$$\hat{M} = \{j \in \{1, \ldots, p\} : \hat{\beta}_j \neq 0\}$$

2. Use least squares to fit the submodel containing just the variables in $\hat{M}$, i.e. linear regression of the $n \times 1$ response y on the reduced $n \times |\hat{M}|$ submatrix $X_{\hat{M}}$. Obtain

$$\hat{S} = \{j \in \hat{M} : p_j \leq \alpha\}$$

   where p_j is the p-value for testing the null hypothesis $H_j : \beta_j = 0$ in the linear model including only the selected variables

Simulation with $n = 200$, $p = 1000$, $s = 10$, $\rho = 0$, $\nu = 2.5$, $\alpha = 5\%$:

Size $\lvert\hat{S}\rvert$	# Type I $\lvert\hat{S} \cap N\rvert$	# Type II $\lvert S \setminus \hat{S}\rvert$	FDP $\lvert\hat{S} \cap N\rvert / \lvert\hat{S}\rvert$	Sensitivity $\lvert\hat{S} \cap S\rvert / \lvert S\rvert$
15	5	0	33.3%	100%

100 replications

	1	2	3	4	5	6	7
Size	15	18	12	17	18	17	11
# Type I	5	8	2	7	8	7	4
# Type II	0	0	0	0	0	0	3
FDP	0.33	0.44	0.17	0.41	0.44	0.41	0.36
Sensitivity	1	1	1	1	1	1	0.7

FWER = 99%, FDR = 42.1%, AvgPower = 99.6%

j	p_j	Selected
1	0.00	*
2	0.00	*
3	0.00	*
4	0.00	*
5	0.00	*
6	0.00	*
7	0.00	*
8	0.00	*
9	0.00	*
10	0.00	*
37	0.29	
53	0.06	
273	0.00	*
417	0.04	*
427	0.12	
525	0.04	*
577	0.24	
590	0.06	
636	0.16	
673	0.01	*
698	0.31	
721	0.12	
829	0.01	*

- The main problem with the naïve two-step procedure is that it peeks at the data twice: once to select the variables to include in $\hat{M}$, and then again to test hypotheses associated with those variables
- Here $\hat{M}$ is a random variable (it is a function of the data), but inference for linear model assumes it fixed (given a priori)
- A secondary problem is the multiplicity of the tests performed
- A simple idea is to use data-splitting to break up the dependence of variable selection and hypothesis testing (Cox, 1975)

Data-split

The *single-split* approach (Wasserman and Roeder, 2009) splits the data into two parts I and L of equal sizes $n_I = n_L = n/2$:

1. Use variable selection on the L portion (X^L, y^L) to obtain

$$\hat{M}^L \subseteq \{1, \ldots, p\}$$

2. Use the I portion (X^I, y^I) for constructing p-values

$$p_j = \begin{cases} p_j^I & \text{if } j \in \hat{M}^L \\ 1 & \text{if } j \notin \hat{M}^L \end{cases}$$

where p_j^I is the p-value testing $H_j : \beta_j = 0$ in the linear model including only the selected variables, i.e. based on the linear regression of the reduced $n_I \times 1$ response y^I on the reduced $n_I \times |\hat{M}^L|$ matrix $X^I_{\hat{M}^L}$

3. Adjust the p-values for their multiplicity $|\hat{M}^L|$, by e.g. Bonferroni

$$\tilde{p}_j = \min(|\hat{M}^L| \cdot p_j, 1), \quad j = 1, \ldots, p$$

4. Selected variables

$$\tilde{S} = \{j \in \hat{M}^L : \tilde{p}_j \leq \alpha\}$$

j	p_j^L	p_j^I	$\tilde{p}_j^I$	Selected
1	0.00	0.08	1.00	
2	0.00	0.00	0.00	*
3	0.00	0.00	0.00	*
4	0.03	0.01	0.09	
6	0.00	0.00	0.00	*
8	0.00	0.00	0.01	*
9	0.16	0.00	0.00	*
10	0.00	0.00	0.00	*
37	0.03	0.38	1.00	
390	0.15	0.79	1.00	
398	0.01	0.21	1.00	
720	0.24	0.04	0.60	
721	0.02	0.82	1.00	
742	0.04	0.21	1.00	
824	0.02	0.24	1.00	
829	0.01	0.38	1.00	
943	0.15	0.66	1.00	

Theorem

Assume that

1. *the linear model* $y \sim N_n(1\beta_0 + X\beta, \sigma^2 I)$ *holds*
2. *the variable selection procedure satisfies the screening property for the first half of the sample, i.e.*

$$P(\hat{M}^L \supseteq S) \geq 1 - \delta$$

   for some $\delta \in (0, 1)$.

3. *The reduced design matrix for the second half of the sample satisfies* $\text{rank}(X^I_{\hat{M}^L}) = |\hat{M}^L|$.

Then the single-split procedure yields FWER control at α *against inclusion of null predictors up to the additional (small) value* δ*, i.e.*

$$P(\tilde{S} \cap N \neq \emptyset) \leq \alpha + \delta$$

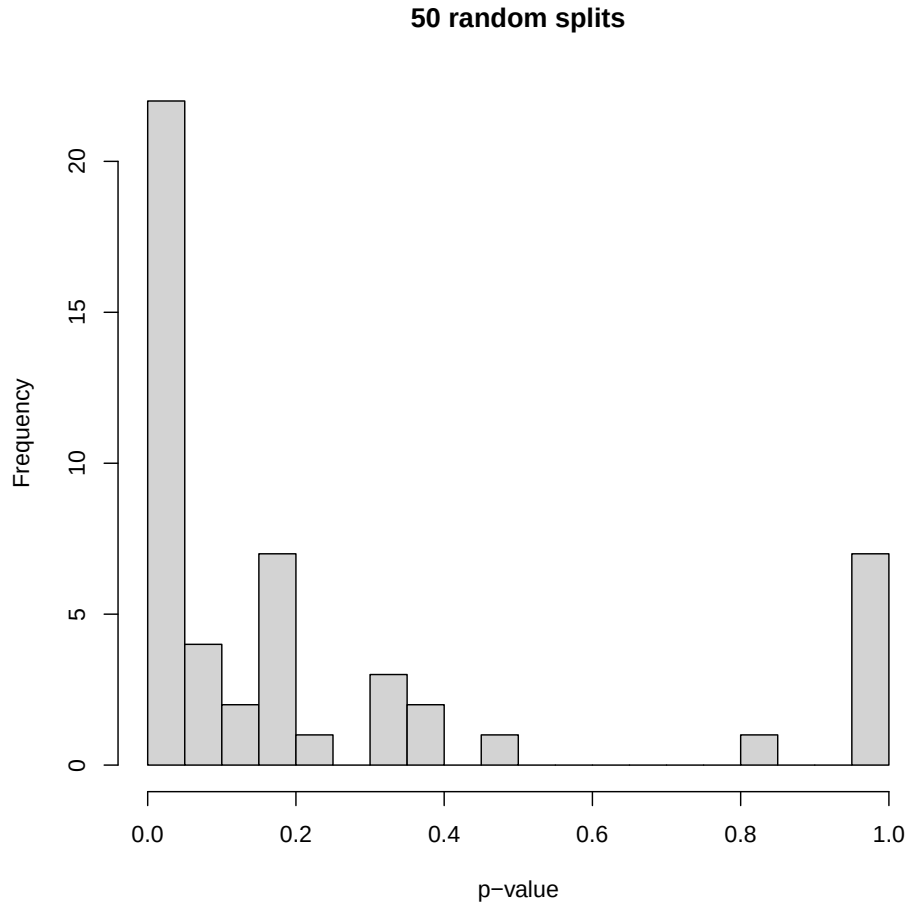
Proof.
Let $E = \{\hat{M}^L \supseteq S\}$ with $\mathrm{P}(E^c) \leq \delta$ by assumption. If E happens, then p_j^I is a valid p-value, i.e. $\mathrm{P}(p_j^I \leq u|E) \leq u$ for $j \in N \cap \hat{M}^L$. We have

$$
\begin{aligned}
& \mathrm{P}(\tilde{S} \cap N \neq \emptyset) = \mathrm{P}\Big(\bigcup_{j \in \hat{M}^L \cap N} \{\tilde{p}_j \leq \alpha\} \Big) \\
= \; & \mathrm{P}\Big(\bigcup_{j \in \hat{M}^L \cap N} \{\tilde{p}_j \leq \alpha\} | E \Big) \mathrm{P}(E) + \mathrm{P}\Big(\bigcup_{j \in \hat{M}^L \cap N} \{\tilde{p}_j \leq \alpha\} | E^c \Big) \mathrm{P}(E^c) \\
\leq \; & \Big[\sum_{j \in \hat{M}^L \cap N} \mathrm{P}\Big(p_j^I \leq \frac{\alpha}{|\hat{M}^L|} | E\Big) \Big] \mathrm{P}(E) + \mathrm{P}\Big(\bigcup_{j \in \hat{M}^L \cap N} \mathbb{1}\{\tilde{p}_j \leq \alpha\} | E^c \Big) \mathrm{P}(E^c) \\
\leq \; & |\hat{M}^L \cap N| \frac{\alpha}{|\hat{M}^L|} \cdot 1 + 1 \cdot \delta \\
\leq \; & \alpha + \delta
\end{aligned}
$$

□

P-value lottery

A major problem of the single data-splitting method is that different data splits lead to different p-values



Multi-split

The *multi-split* approach (Meinshausen et al., 2009)

1. For $b = 1, \ldots, B$
   apply the single-split procedure (L^b, I^b) to obtain

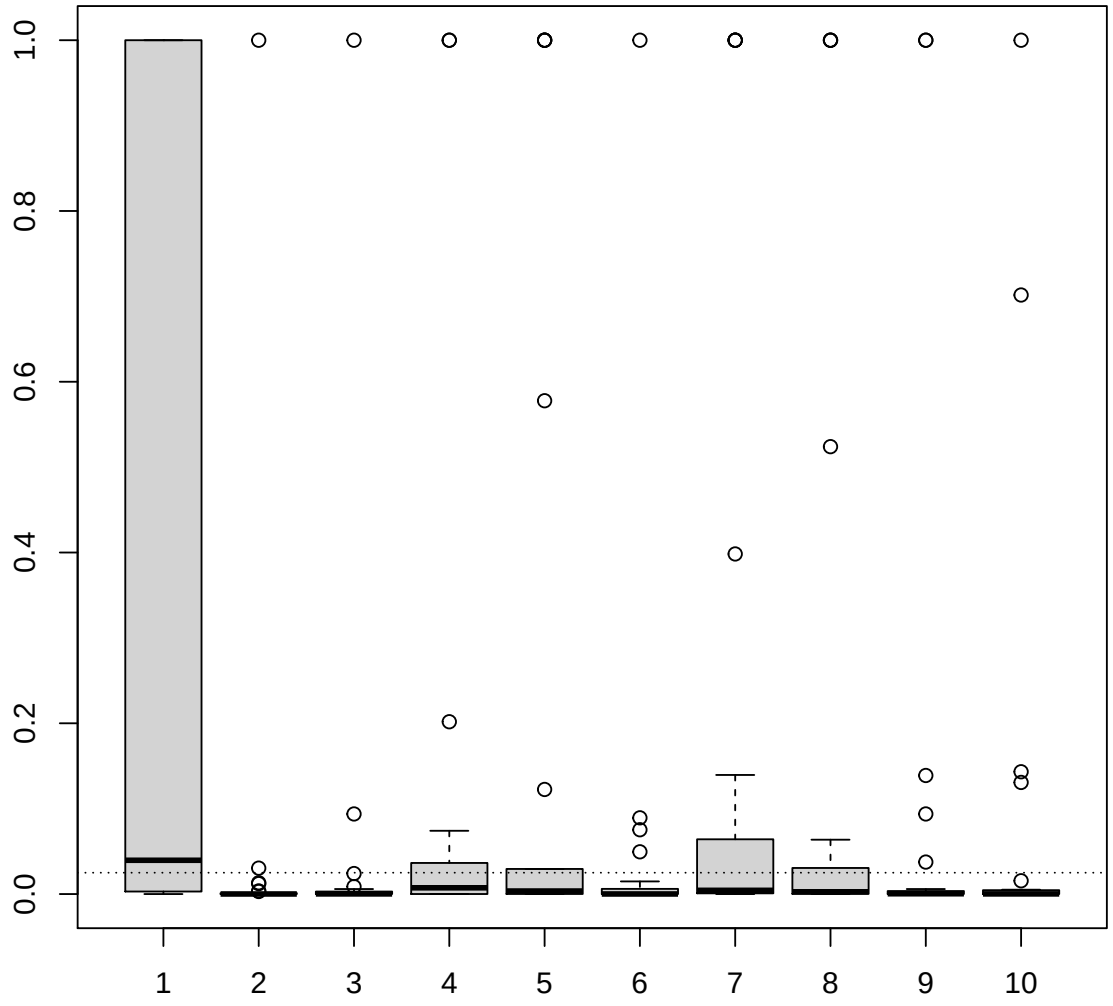
$$\{\tilde{p}_j^b, j = 1, \ldots, p\}$$

2. Aggregate the p-values as

$$\bar{p}_j = 2 \cdot \text{median}(\tilde{p}_j^1, \ldots, \tilde{p}_j^B), \quad j = 1, \ldots, p$$

3. Selected predictors:

$$\bar{S} = \{j \in \{1, \ldots, p\} : \bar{p}_j \leq \alpha\}$$


1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
10

Simultaneous confidence intervals

$$\mathrm{P}(\beta_j \in [\hat{L}_j, \hat{U}_j] \ \forall j \in \{1, \ldots, p\}) \geq 1 - \alpha$$

j	$\hat{L}_j$	$\hat{U}_j$
1	-∞	∞
2	0.69	1.84
3	0.48	1.73
4	0.36	1.49
5	0.47	1.70
6	0.56	1.78
7	0.27	1.57
8	0.40	1.69
9	0.41	1.56
10	0.44	1.56
11	-∞	∞
...		

Stability Selection

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Meinshausen, Buhlmann (2010). Stability selection. JRSS-B, 72:417–473
- Shah, Samworth (2013). Variable selection with error control: another look at stability selection. JRSS-B, 75:55–80.

Stability path

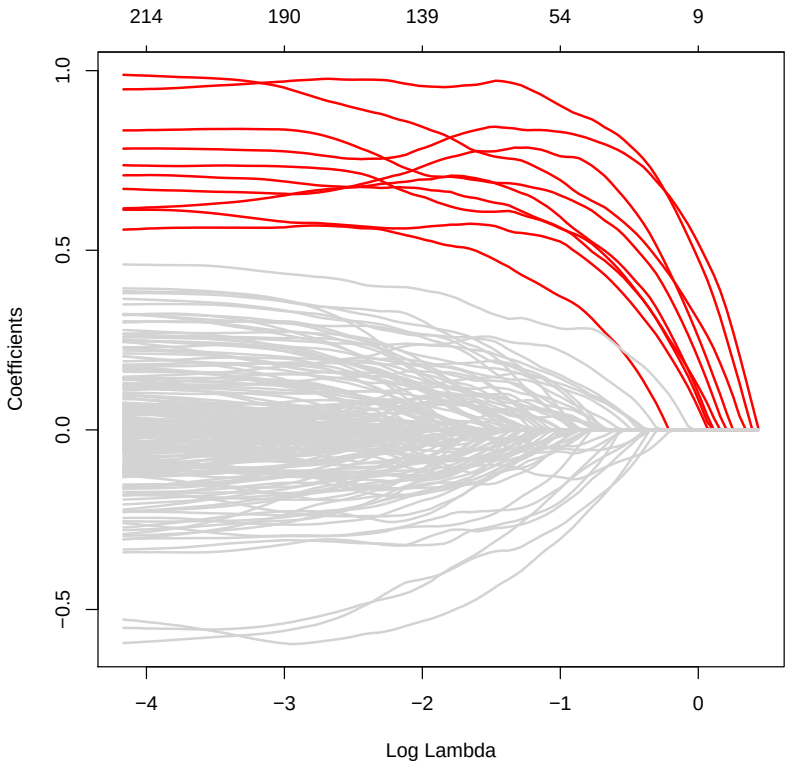
- The *regularisation path* of the lasso is

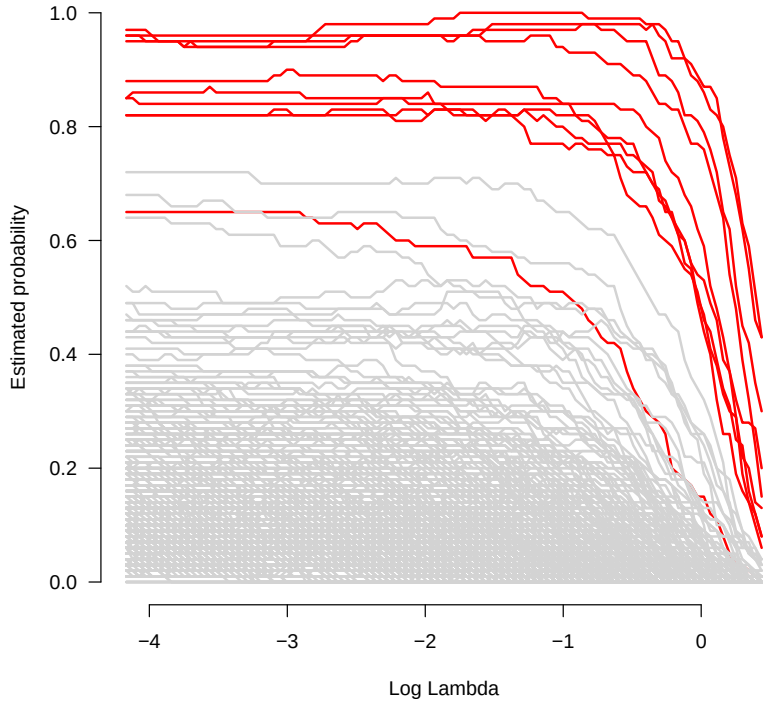
$$\{\hat{\beta}_j(\lambda), j = 1, \ldots, p, \lambda \in \Lambda\}$$

- The *stability path* is

$$\{\hat{\pi}_j(\lambda), j = 1, \ldots, p, \lambda \in \Lambda\}$$

  where $\hat{\pi}_j(\lambda)$ is the estimated probability for the jth predictor to be selected by the lasso(λ) when randomly resampling from the data


214
190
139
54
9
1.0
0.5
0.0
−0.5
Coefficients
−4
−3
−2
−1
0
Log Lambda


1.0
0.8
0.6
0.4
0.2
0.0
Estimated probability
−4
−3
−2
−1
0
Log Lambda

Algorithm 1 Stability Path Algorithm with the Lasso

Require: $B \in \mathbb{N}$, Λ grid, $\tau \in (0.5, 1)$

1: **for** $b = 1, \ldots, B$ **do**

2: Randomly select $n/2$ indices from $\{1, \ldots, n\}$;

3: Perform the lasso on the $n/2$ observations to obtain

$$\hat{S}_{n/2}(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0\} \qquad \forall\ \lambda \in \Lambda$$

4: **end for**

5: Compute the relative selection frequencies:

$$\hat{\pi}_j(\lambda) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\{j \in \hat{S}_{n/2}(\lambda)\} \qquad \forall\ \lambda \in \Lambda$$

6: The set of *stable predictors* is given by

$$\hat{S}_{\text{stab}} = \{j : \max_{\lambda \in \Lambda} \hat{\pi}_j(\lambda) \geq \tau\}$$

Algorithm 2 (Complementary Pairs) Stability Selection

Require: A variable selection procedure $\hat{S}_n$, $B \in \mathbb{N}$, $\tau \in (0.5, 1)$

1: **for** $b = 1, \ldots, B$ **do**

2: Split $\{1, \ldots, n\}$ into (I^{2b-1}, I^{2b}) of size $n/2$, and for each get

$$\hat{S}^{2b-1}_{n/2} \subseteq \{1, \ldots, p\}, \qquad \hat{S}^{2b}_{n/2} \subseteq \{1, \ldots, p\}$$

3: **end for**

4: Compute the relative selection frequencies:

$$\hat{\pi}_j = \frac{1}{2B} \sum_{b=1}^{B} \left(\mathbb{1}\{j \in \hat{S}^{2b-1}_{n/2}\} + \mathbb{1}\{j \in \hat{S}^{2b}_{n/2}\} \right)$$

5: The set of *stable predictors* is given by

$$\hat{S}_{\text{stab}} = \{j : \hat{\pi}_j \geq \tau\}$$

- The relative selection frequency $\hat{\pi}_j$ is an unbiased estimator of

$$\pi_j^{n/2} = \mathrm{P}(j \in \hat{S}_{n/2})$$

  but, in general, a biased estimator of

$$\pi_j^n = \mathrm{P}(j \in \hat{S}_n) = \mathbb{E}(\mathbb{1}\{j \in \hat{S}_n\})$$

- The key idea of stability selection is to improve on the simple estimator $\mathbb{1}\{j \in \hat{S}_n\}$ of π_j^n through subsampling.
- By means of averaging involved in $\hat{S}_{\text{stab}}$, we hope that $\hat{\pi}_j$ will have reduced variance compared to $\mathbb{1}\{j \in \hat{S}_n\}$ and this increased stability will more than compensate for the bias incurred.

Theorem

Assume that

1. $\{\mathbb{1}\{j \in \hat{S}_{n/2}\}, j \in N\}$ *is exchangeable;*
2. *The variable selection procedure is not worse than random guessing, i.e.*

$$\frac{\mathbb{E}(|\hat{S}_{n/2} \cap S|)}{\mathbb{E}(|\hat{S}_{n/2} \cap N|)} \geq \frac{|S|}{|N|}.$$

Then, for $\tau \in (1/2, 1]$

$$\mathbb{E}(|\hat{S}_{\text{stab}} \cap N|) \leq \frac{1}{(2\tau - 1)} \frac{q^2}{p}$$

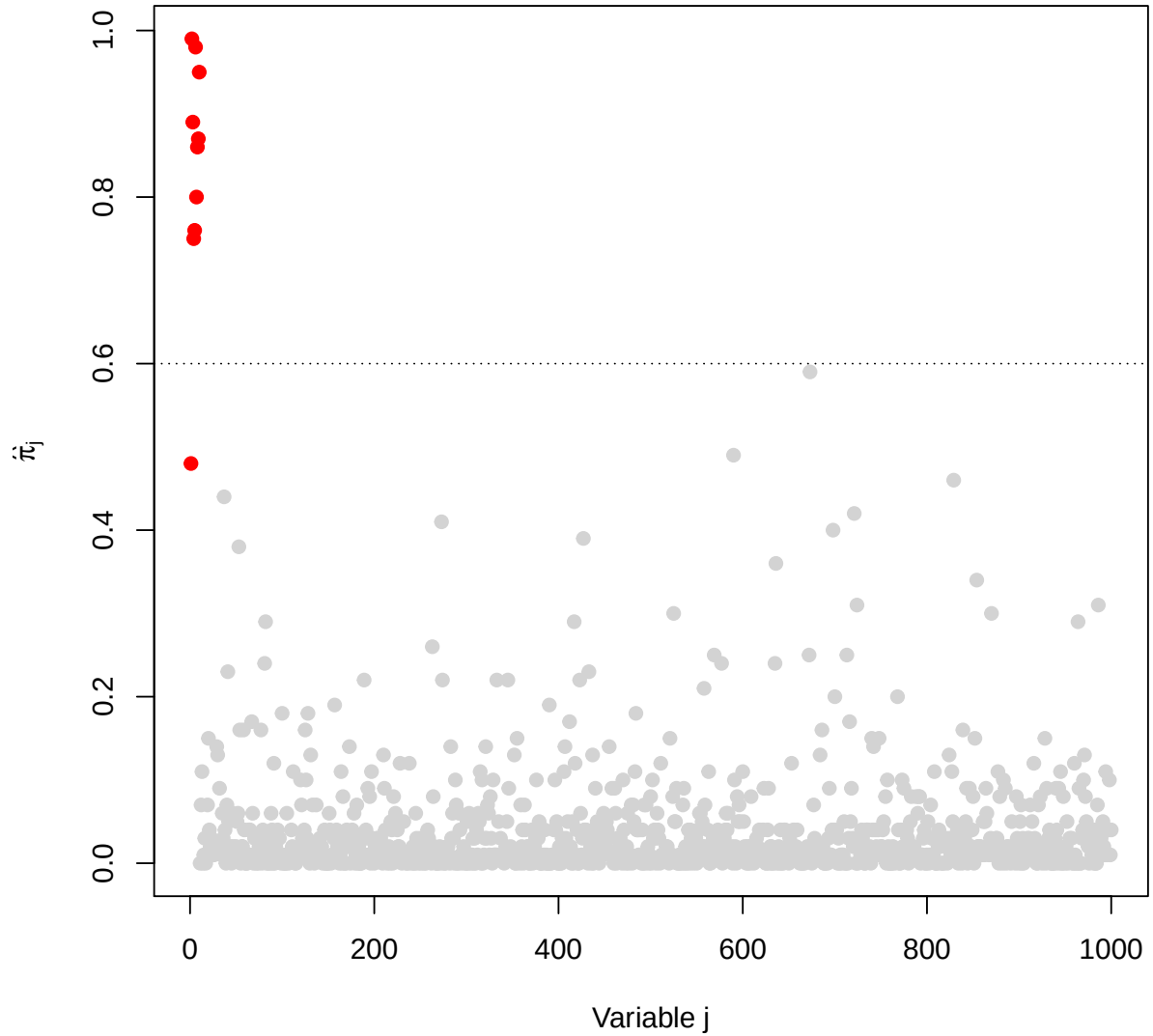
where $q = \mathbb{E}(|\hat{S}_{n/2}|)$

- The choice of the number of subsamples B is of minor importance
- It is possible to fix $q = \mathbb{E}(|\hat{S}_{n/2}|)$ and run the variable selection procedure until it selects q variables. However, if q is too small, one would select only a subset of the signal variables as

$$|\hat{S}_{\text{stab}}| \leq |\hat{S}_{n/2}| = q$$

- For example, with $p = 1000$, $q = 50$ and $\tau = 0.6$ then

$$\mathbb{E}(|\hat{S}_{\text{stab}} \cap N|) \leq 12.5$$


1.0
0.8
0.6
0.4
0.2
0.0
$\hat{\pi}_j$
0
200
400
600
800
1000
Variable j

The knockoff filter

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

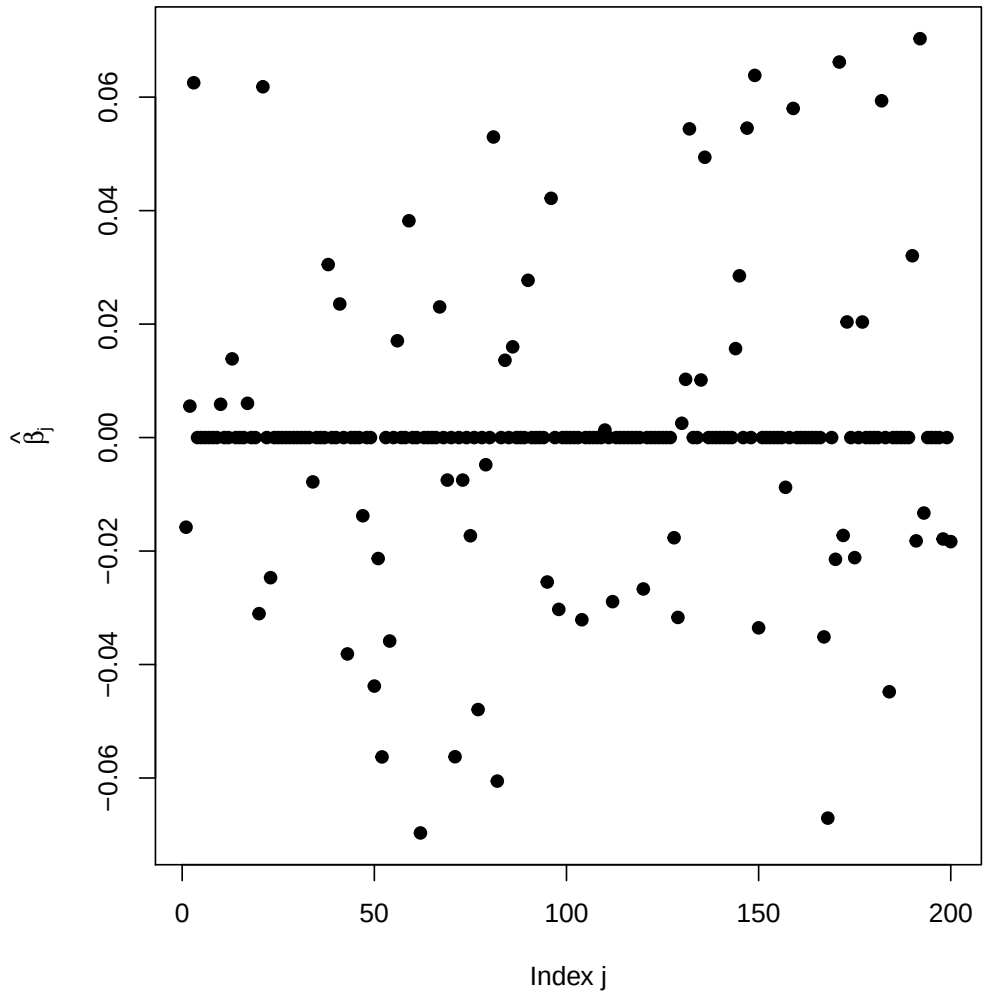
References

- Barber, Candès (2015) Controlling the False Discovery Rate via Knockoffs. Ann. Statist. 43:2504–2537
- Candès, Fan, Janson, Lv (2018). Panning for gold: model-X knockoffs for high dimensional controlled variable selection. JRSS-B 80:551–577.

There are two main approaches:

- *Fixed-X knockoffs*
  Requires that X is full rank with $n \geq 2p$
- *Model-X knockoffs*
  Requires assumptions on X but works with $p > n$

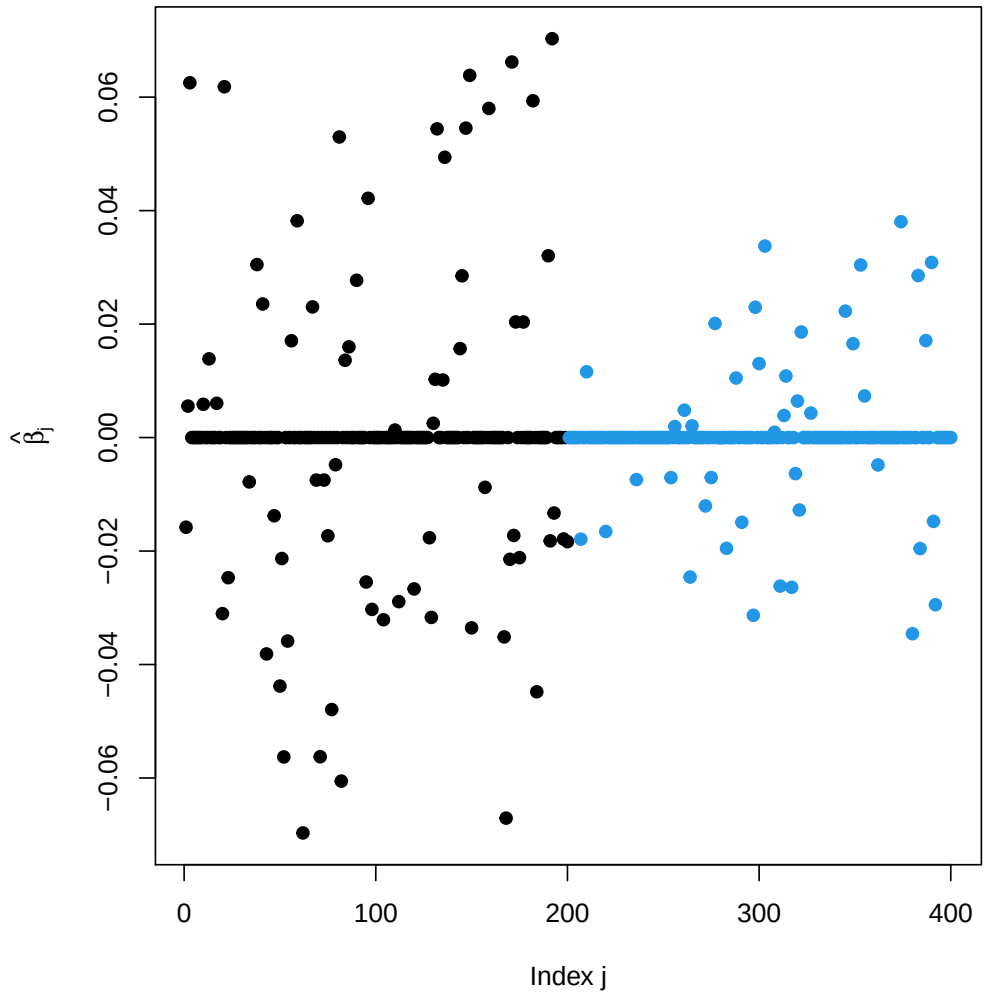
Fixed-X knockoffs



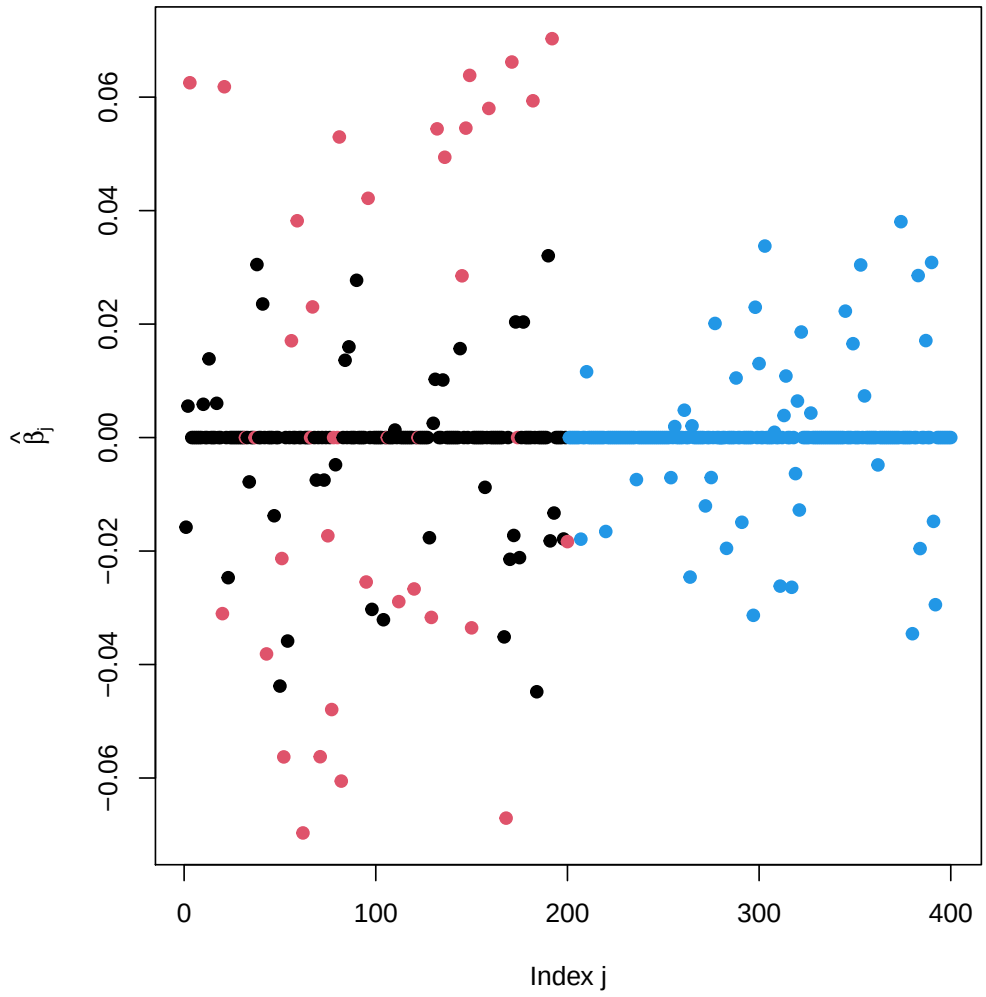
Lasso selects 67 features: $\mathrm{FDP}(\hat{S}) = ?/67$

Main idea

- For each feature X_j, construct a *knockoff* copy $\tilde{X}_j$
- Knockoffs $\tilde{X}_1, \ldots, \tilde{X}_p$ are independent of y and mimic the original variables $X_1, \ldots, X_p$ if they were null



Lasso selects 70 original and 43 knockoff: $\widehat{\mathrm{FDP}}(\hat{S}) = 43/70 \approx 61\%$



$$\text{True FDP}(\hat{S}) = 34/70 \approx 54\%$$

Knockoff construction

- Suppose without loss of generality that the features are centered and scaled such that $\|X_j\|_2^2 = 1$ for all j
- Let $\Sigma = X^t X$ be the correlation matrix of the features
- The method begins by augmenting the design matrix X with a second matrix $\tilde{X} \in \mathbb{R}^{n \times p}$ of knockoff variables, constructed to satisfy

$$
\begin{aligned}
G = [X\ \tilde{X}]^t [X\ \tilde{X}] &= \begin{bmatrix} X^t X & X^t \tilde{X} \\ \tilde{X}^t X & \tilde{X}^t \tilde{X} \end{bmatrix} \\
&= \begin{bmatrix} \Sigma & \Sigma - D \\ \Sigma - D & \Sigma \end{bmatrix}
\end{aligned}
$$

for some diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_p)$ such that G is positive definite

- The knockoffs have the same correlation structure as the original features

$$\tilde{X}^t \tilde{X} = X^t X = \Sigma$$

- The correlation between $\tilde{X}_k$ and X_j is

$$\tilde{X}_j^t X_k = X_j^t X_k \quad \forall\, k \neq j$$

- The correlation between $\tilde{X}_j$ and X_j is

$$\tilde{X}_j^t X_j = 1 - d_j$$

with d_j as close to 1 as possible

Equi-correlated knockoffs

Suppose we require $d_j = d$ for all j. Define

$$\tilde{X} = X(I_p - d\Sigma^{-1}) + UC$$

where

- $U \in \mathrm{R}^{n \times p}$ is an orthonormal matrix such that $U^t X = 0$
- $C \in \mathrm{R}^{p \times p}$ from the Cholesky decomposition of

$$C^t C = 4((d/2)I_p - (d/2)^2 \Sigma^{-1})$$

This approach corresponds to `method="equi"` in the `knockoff` package. A semidefinite programming approach is used to determine the values that minimize $\sum_{j=1}^{p}(1 - d_j)$ subject to the constraints (`method="sdp"`)

The knockoff statistics

- Fit the lasso to the augmented design matrix $[X\ \tilde{X}]$ for $\lambda \in \Lambda$
- Let $[\hat{\beta}(\lambda)\ \tilde{\beta}(\lambda)]$, $\lambda \in \Lambda$ denote the coefficient estimates
- Compute

$$Z_j = \sup\{\lambda \in \Lambda : \hat{\beta}_j(\lambda) \neq 0\} = \text{ first time } X_j \text{ enters the lasso path}$$
$$\tilde{Z}_j = \sup\{\lambda \in \Lambda : \tilde{\beta}_j(\lambda) \neq 0\} = \text{ first time } \tilde{X}_j \text{ enters the lasso path}$$

- Then define the statistics

$$W_j = \max(Z_j, \tilde{Z}_j) \cdot \text{sign}(Z_j - \tilde{Z}_j) = \begin{cases} Z_j & \text{if } X_j \text{ enters first } (Z_j > \tilde{Z}_j) \\ 0 & \text{if } Z_j = \tilde{Z}_j \\ -\tilde{Z}_j & \text{if } \tilde{X}_j \text{ enters first } (Z_j < \tilde{Z}_j) \end{cases}$$

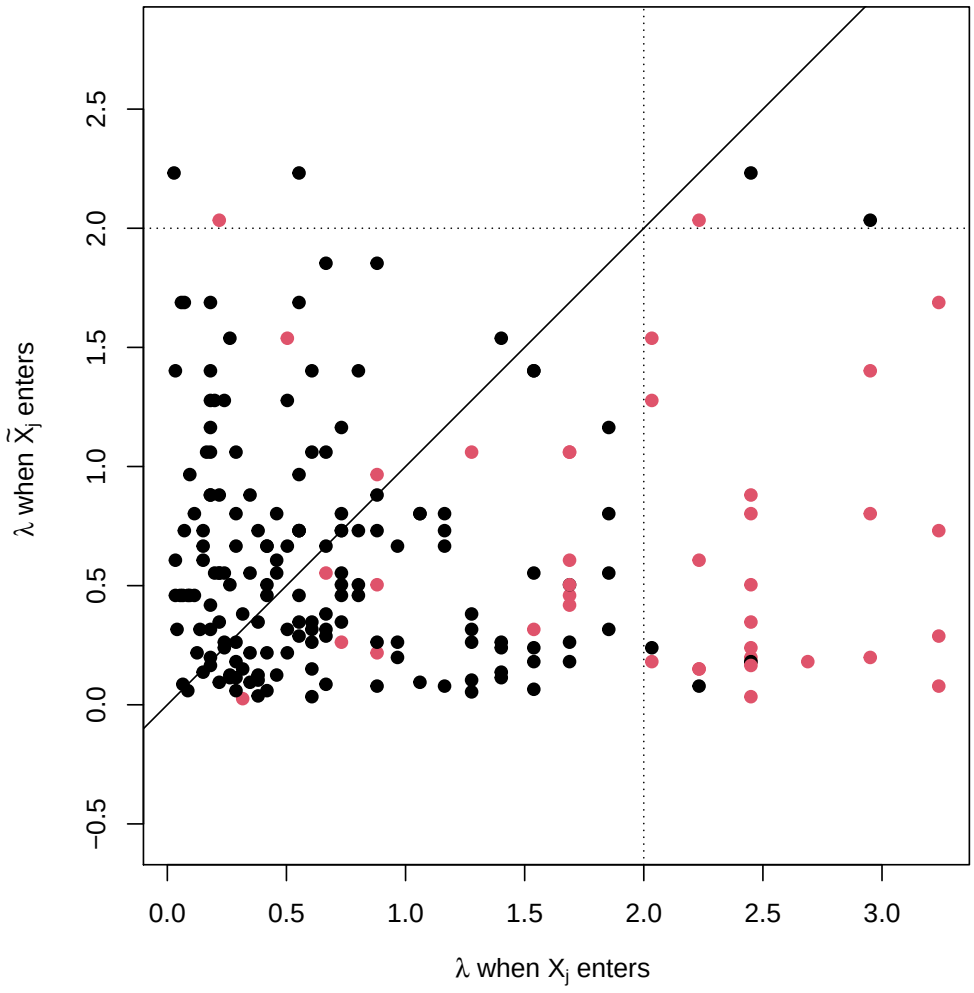
FDP estimate

- For some threshold $\tau \geq 0$, select

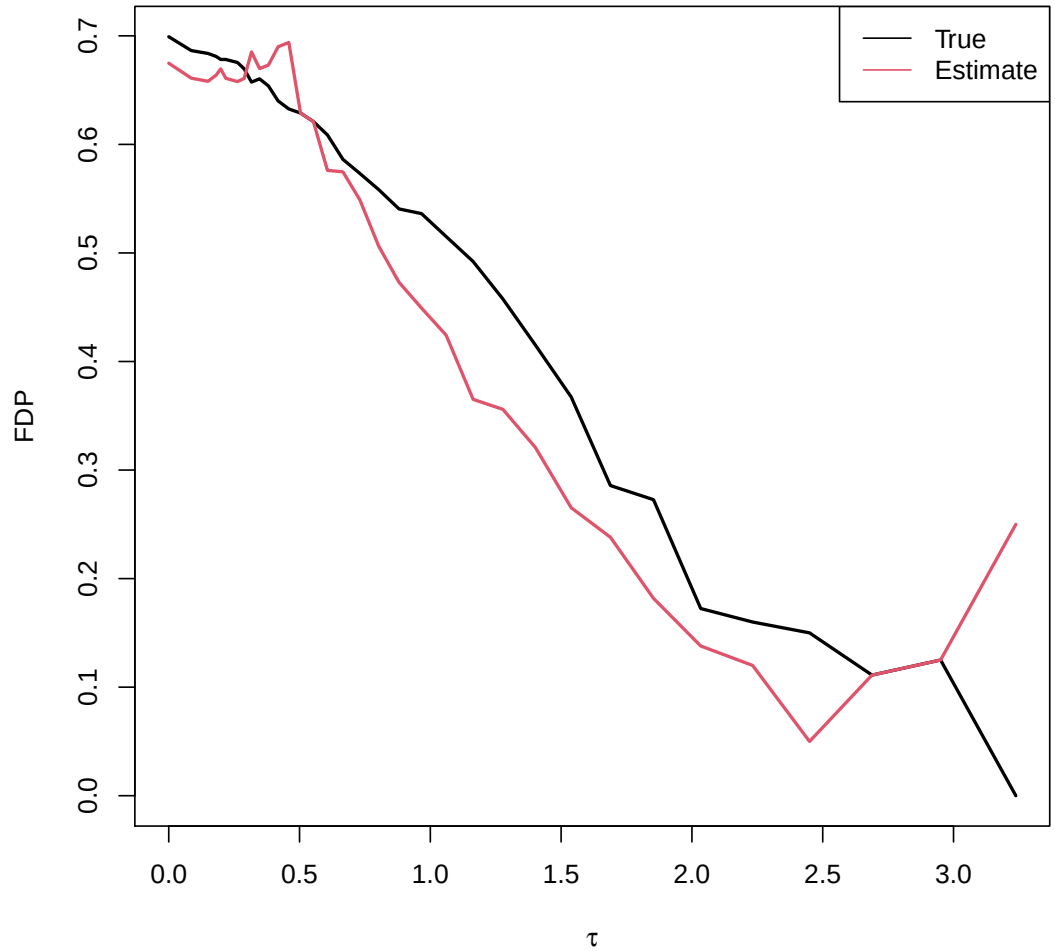
$$\hat{S}_\tau = \{j \in \{1, \ldots, p\} : W_j \geq \tau\}$$

- The knockoff estimate of the FDP is

$$\begin{aligned} \mathrm{FDP}(\hat{S}_\tau) &= \frac{\#\{j \in N : W_j \geq t\}}{\#\{j : W_j \geq t\}} \\ &\approx \frac{\#\{j \in N : W_j \leq -t\}}{\#\{j : W_j \geq t\}} \\ &\leq \frac{1 + \#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\}} = \widehat{\mathrm{FDP}}(\hat{S}_\tau) \end{aligned}$$



For $\tau = 2$, $|\hat{S}_\tau| = 29$ with $\widehat{\mathrm{FDP}}(\hat{S}_\tau) = 4/29$ and $\mathrm{FDP}(\hat{S}_\tau) = 5/29$


True
Estimate
FDP
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
τ

The knockoff procedure chooses a data-dependent threshold

$$\hat{\tau} = \min\left\{\tau > 0 : \widehat{\mathrm{FDP}}(\hat{S}_\tau) \leq \alpha\right\}$$

with $\hat{\tau} = +\infty$ if no such τ exists.

Theorem

For any $\alpha \in (0, 1)$, the knockoff procedure selects

$$\hat{S}_{\hat{\tau}} = \{j \in \{1, \ldots, p\} : W_j \geq \hat{\tau}\}$$

with the guarantee that

$$\mathrm{FDR}(\hat{S}_{\hat{\tau}}) = \mathbb{E}\left(\frac{|N \cap \hat{S}_{\hat{\tau}}|}{|\hat{S}_{\hat{\tau}}|}\right) \leq \alpha$$

where the expectation is taken over ε in the Gaussian linear model $y = X\beta + \varepsilon$ while treating X and $\tilde{X}$ as fixed.

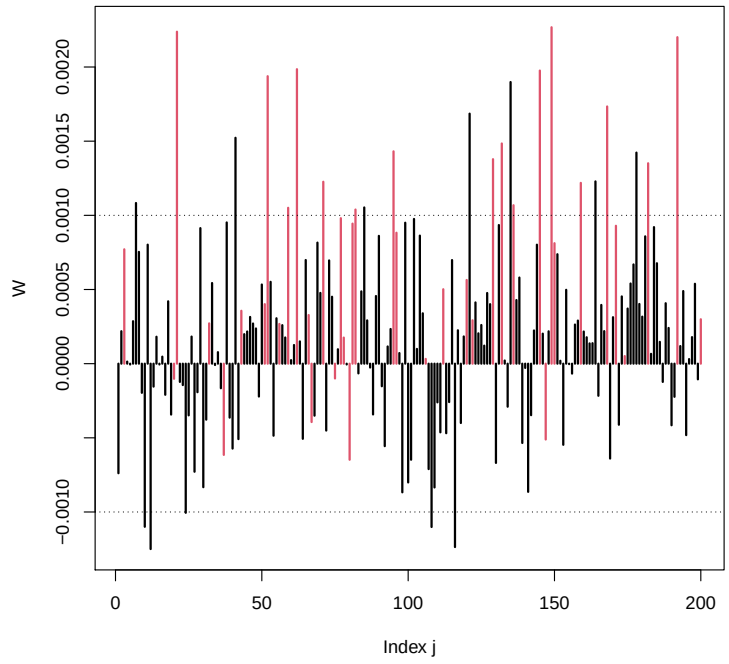
Variable importance statistics

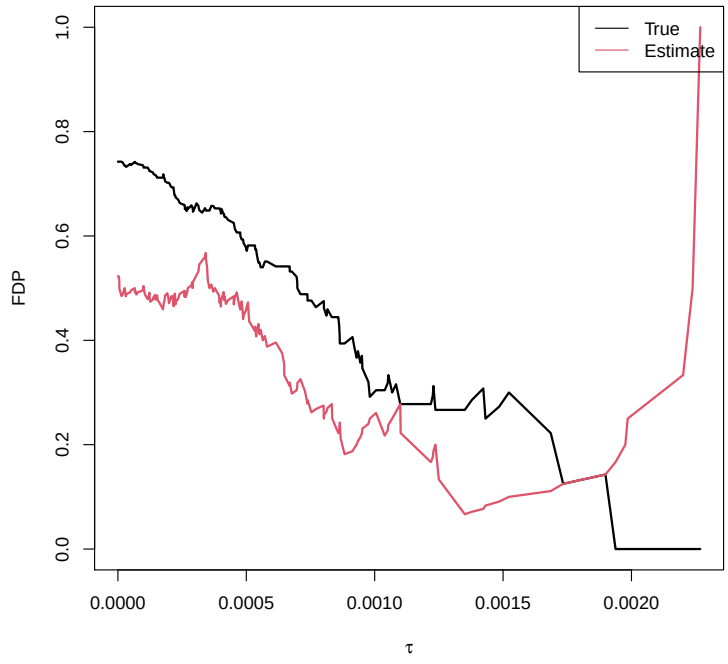
- Fit the Random Forest to the augmented design matrix $[X\ \tilde{X}]$
- Compute

$$Z_j = \text{VariableImportance}(X_j)$$
$$\tilde{Z}_j = \text{VariableImportance}(\tilde{X}_j)$$

  The importance of a variable is measured as the total decrease in node impurities from splitting on that variable, averaged over all trees
- Then define the statistics

$$W_j = \text{abs}(Z_j) - \text{abs}(\tilde{Z}_j)$$





For $\tau = 0.001$, $|\hat{S}_\tau| = 23$ with $\widehat{\mathrm{FDP}}(\hat{S}_\tau) = 4/23$ and $\mathrm{FDP}(\hat{S}_\tau) = 7/23$

Model-X knockoff

Modeling X

- X is treated as a random matrix with i.i.d. rows x_i
- (x_i, y_i), $i = 1, \ldots, n$ are i.i.d. from some unknown distribution
- Assume we know the *marginal distribution* of x_i, e.g.

$$x_i = (x_{i1}, \ldots, x_{ip}) \sim N_p(\mu, \Sigma)$$

- Null features given by *conditional independence*

$$N = \{j \in \{1, \ldots, p\} : y \perp\!\!\!\perp x_j | x_{-j}\}$$

where $x_{-j} = \{x_1, \ldots, x_p\} \setminus \{x_j\}$

Knockoffs in the Gaussian case

- The joint distribution of original features and knockoff copies satisfies

$$[x\ \tilde{x}] \sim N(M, V) \quad \text{with } M = \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \quad V = \begin{bmatrix} \Sigma & \Sigma - D \\ \Sigma - D & \Sigma \end{bmatrix}$$

  where $D = \text{diag}(d_1, \ldots, d_p)$ such that V is positive definite
- Draw a random $\tilde{x}_i$ from the conditional distribution $\tilde{x}_i | x_i$, which is normal with

$$\begin{aligned} \mathbb{E}(\tilde{x}_i | x_i) &= \mu + (\Sigma - D)\Sigma^{-1}(x_i - \mu) \\ \mathbb{V}\text{ar}(\tilde{x}_i | x_i) &= \Sigma - (\Sigma - D)\Sigma^{-1}(\Sigma - D) \end{aligned}$$

- If μ and Σ are unknown, replace by estimates $\hat{\mu}$ and $\hat{\Sigma}$