Statistical Learning with Sparsity

Optimization and Statistical Inference

Boen Jiang

Fudan University

April 7, 2025

Lagrangian

Consider general minimization problem

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to} & h_i(x) \leq 0, i = 1, \ldots, m \\ & \ell_j(x) = 0, j = 1, \ldots, r \end{array}$$

Need not be convex, but of course we will pay special attention to convex case.

We define the Lagrangian as

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

New variables $u \in \mathbb{R}^m$, $v \in \mathbb{R}^r$, with $u \geq 0$ (else $L(x, u, v) = -\infty$)

Lagrangian

- Important property: for any $u \geq 0$ and v, $f(x) \geq L(x, u, v)$ at each feasible x.
- Why? For feasible x,

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i \underbrace{h_i(x)}_{\leq 0} + \sum_{j=1}^{r} v_j \underbrace{\ell_j(x)}_{=0} \leq f(x)$$

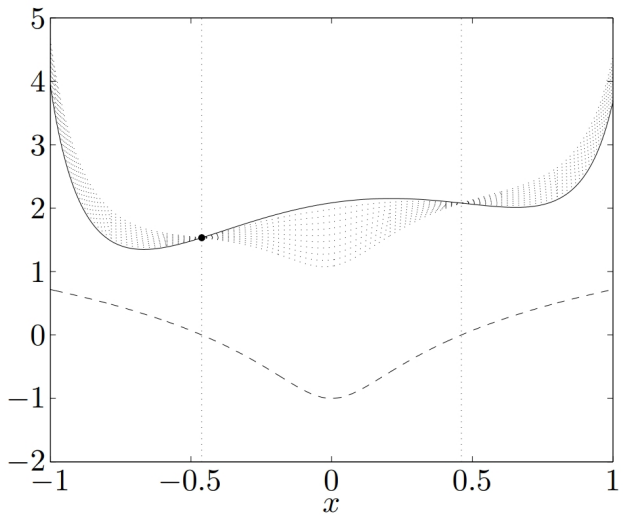


Figure: From Boyd's Convex Optimization, Page 217. f: solid line, h: dashed li
L: dotted line, feasible region $\approx [-0.46, 0.46]$.

Lagrangian dual function

Let C denote primal feasible set, $f^\star$ denote primal optimal value.
Minimizing $L(x, u, v)$ over all x gives a lower bound:

$$f^\star \geq \inf_{x \in C} L(x, u, v) \geq \inf_{x} L(x, u, v) := g(u, v)$$

We call $g(u, v)$ the Lagrange dual function, and it gives a lower bound on $f^\star$ for any $u \geq 0$ and v.
Hence best lower bound: maximize $g(u, v)$ over dual feasible u, v, yielding Lagrange dual problem:

$$\begin{array}{ll} \max\limits_{u,v} & g(u, v) \\ \text{subject to} & u \geq 0 \end{array}$$

Key property, called weak duality: if dual optimal value is $g^\star$, then

$$f^\star \geq g^\star$$

Note that this always holds (even if primal problem is nonconvex). This can be shown by the max-min inequality.

Dual problem is convex

Again, this is always true (even when primal problem is not convex) By definition:

$$g(u,v) = \min_x \left\{ f(x) + \sum_{i=1}^m u_i h_i(x) + \sum_{j=1}^r v_j \ell_j(x) \right\}$$

$$= - \underbrace{\max_x \left\{ -f(x) - \sum_{i=1}^m u_i h_i(x) - \sum_{j=1}^r v_j \ell_j(x) \right\}}_{\text{pointwise maximum of convex (linear) functions in } (u,v)}$$

That is, g is concave in (u, v), and $u \geq 0$ is a convex constraint, so dual problem is a concave maximization problem.

Strong duality

Recall that we always have $f^\star \geq g^\star$ (weak duality). On the other hand, in some problems we have observed that actually

$$f^\star = g^\star$$

which is called strong duality.
Slater's condition: if the primal is a convex problem (i.e., f and $h_1, \ldots, h_m$ are convex, $\ell_1, \ldots, \ell_r$ are affine), and there exists at least one strictly feasible $x \in \mathbb{R}^n$, meaning

$$h_1(x) < 0, \ldots, h_m(x) < 0 \quad \text{and} \quad \ell_1(x) = 0, \ldots, \ell_r(x) = 0$$

then strong duality holds.
Refinement: actually only need strict inequalities for non-affine h_i

(Generalized) KKT Conditions

Given general problem

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to} & h_i(x) \leq 0, i = 1, \ldots, m \\ & \ell_j(x) = 0, j = 1, \ldots, r \end{array}$$

The Karush-Kuhn-Tucker conditions or KKT conditions are:

1. $0 \in \partial_x \left(f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x) \right)$ (stationarity)
2. $u_i \cdot h_i(x) = 0$ for all i (complementary slackness)
3. $h_i(x) \leq 0, \ell_j(x) = 0$ for all i, j (primal feasibility)
4. $u_i \geq 0$ for all i (dual feasibility)

Necessity of KKT conditions

Let $x^\star$ and $u^\star, v^\star$ be primal and dual solutions with zero duality gap (strong duality holds, e.g., under Slater's condition). Then

$$\begin{aligned} f(x^\star) &= g(u^\star, v^\star) \\ &= \min_x f(x) + \sum_{i=1}^m u_i^\star h_i(x) + \sum_{j=1}^r v_j^\star \ell_j(x) \\ &\leq f(x^\star) + \sum_{i=1}^m u_i^\star h_i(x^\star) + \sum_{j=1}^r v_j^\star \ell_j(x^\star) \\ &\leq f(x^\star) \end{aligned}$$

In other words, all these inequalities are actually equalities.

Necessity of KKT conditions

Two things to learn from this:

- The point $x^\star$ minimizes $L(x, u^\star, v^\star)$ over $x \in \mathbb{R}^n$. Hence the subdifferential of $L(x, u^\star, v^\star)$ must contain 0 at $x = x^\star$–this is exactly the stationarity condition.
- We must have $\sum_{i=1}^m u_i^\star h_i(x^\star) = 0$, and since each term here is ≤ 0, this implies $u_i^\star h_i(x^\star) = 0$ for every i-this is exactly complementary slackness.

Primal and dual feasibility hold by virtue of optimality. Therefore:
If $x^\star$ and $u^\star, v^\star$ are primal and dual solutions, with zero duality gap, then $x^\star, u^\star, v^\star$ satisfy the KKT conditions.
Note that this statement assumes nothing a priori about convexity of our problem, i.e., of f, h_i, ℓ_j

Sufficiency of KKT conditions

If there exists $x^\star, u^\star, v^\star$ that satisfy the KKT conditions, then

$$\begin{aligned} g(u^\star, v^\star) &= f(x^\star) + \sum_{i=1}^{m} u_i^\star h_i(x^\star) + \sum_{j=1}^{r} v_j^\star \ell_j(x^\star) \\ &= f(x^\star) \end{aligned}$$

where the first equality holds from stationarity, and the second holds from complementary slackness

Therefore the duality gap is zero (and $x^\star$ and $u^\star, v^\star$ are primal and dual feasible) so $x^\star$ and $u^\star, v^\star$ are primal and dual optimal. Hence, we've shown:

If $x^\star$ and $u^\star, v^\star$ satisfy the KKT conditions, then $x^\star$ and $u^\star, v^\star$ are primal and dual solutions.

Putting it all together

In summary, KKT conditions are equivalent to zero duality gap:

- always sufficient
- necessary under strong duality

Putting it together:
For a problem with strong duality (e.g., assume Slater's condition: convex problem and there exists x strictly satisfying nonaffine inequality contraints),

$$\begin{aligned} & x^\star \text{ and } u^\star, v^\star \text{ are primal and dual solutions} \\ \iff & x^\star \text{ and } u^\star, v^\star \text{ satisfy the KKT conditions.} \end{aligned}$$

(It can also be proved by the Hahn-Banach's theorem.)

Netflix movie-rating challenge

- The Netflix movie-rating challenge has become one of the canonical examples for matrix completion (Bennett and Lanning 2007).
- The Netflix dataset (rating matrix $\boldsymbol{M}$) has n = 17770 movies (columns) and m = 480189 customers (rows).
- Challenges: no single user can have watched all movies, nor can any movie gather ratings from all users.

	Dirty	Meet	Top	The	Catch	The	Con	Big	The	A Few
Customer 1	·	·	·	·	4	·	·	·	·	·
Customer 2	·	·	3	·	·	·	3	·	·	3
Customer 3	·	2	·	4	·	·	·	·	2	·
Customer 4	3	·	·	·	·	·	·	·	·	·
Customer 5	5	5	·	·	4	·	·	·	·	·
Customer 6	·	·	·	·	·	2	4	·	·	·
Customer 7	·	·	5	·	·	·	·	3	·	·
Customer 8	·	·	·	·	·	2	·	·	·	3
Customer 9	3	·	·	·	5	·	·	5	·	·
Customer 10	·	·	·	·	·	·	·	·	·	·

Low rank matrix completion

- Let Ω be the set of subscripts of all known rating values in $\boldsymbol{M}$, then this problem can be preliminarily formulated as constructing a low-rank matrix $\boldsymbol{X}$ such that $\boldsymbol{X}_{ij} = \boldsymbol{M}_{ij}$ for all $(i, j) \in \Omega$.
- Challenge: we have infinitely many possible low-rank matrices $\boldsymbol{X}$ that can satisfy the above condition $\rightsquigarrow$ we need to find the one with the lowest rank.
- A naive approach is

$$\min_{X \in \mathbb{R}^{m \times n}} \operatorname{rank}(X),$$
$$\text{s.t. } X_{ij} = M_{ij}, (i, j) \in \Omega.$$

- However, this is an NP-hard problem (see Candes and Recht 2009).

An example for sparsity other than lasso

- We can use the nuclear norm as a surrogate for the rank function, and solve the following convex optimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_*,$$
$$\text{s.t. } X_{ij} = M_{ij}, (i,j) \in \Omega.$$

- Pros: it can be shown that this optimization problem is a convex problem and under some regular conditions, the two optimization problems are equivalent (see Candes and Recht 2009).
- Furthermore, if there are noises in the data, we can add a regularization term to the optimization problem, and solve the following optimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \mu \|X\|_* + \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2 .$$

SVD decomposition and nuclear norm

Given a matrix $\Theta \in \mathbb{R}^{m \times n}$ (assume $m \leq n$), by SVD decomposition,

$$\Theta = \sum_{j=1}^{m} \sigma_j u_j v_j^\top, \sigma_j \geq 0, u_j \in \mathbb{R}^m, v_j \in \mathbb{R}^n,$$

where σ_j are the singular values of Θ, and u_j, v_j are the left and right singular vectors of Θ. The nuclear norm is the sum of the singular values,

$$\|\Theta\|_* = \sum_{j=1}^{m} \sigma_j.$$

It is also known as the Schatten 1-norm.

Subgradient for nuclear norm

- Watson (1992) completely and thoroughly studied the subdifferential of nuclear matrix norms.
- Generally, the subdifferential (or set of subgradients) of $\|A\|$ is defined by

$$\partial\|A\| = \left\{ G : \mathbb{R}^{m\times n} : \|B\| \geq \|A\| + \operatorname{tr}\left[G^\top (B - A)\right], \text{ for all } B \in \mathbb{R}^{m\times n} \right\}$$

- It is readily established that $G \in \partial\|A\|$ is equivalent to the statements
  1. $\|A\| = \operatorname{tr}\left(G^\top A\right)$
  2. $\|G\|^* \leq 1$ where

$$\|G\|^* = \max_{\|B\|\leq 1} \operatorname{tr}\left[B^\top G\right]$$

and $\|\cdot\|^*$ is the dual norm to $\|\cdot\|$.

Subgradient for nuclear norm

- The subdifferential $\partial\|\Theta\|_*$ of the nuclear norm at Θ consists of all matrices of the form $\mathbf{Z} = \sum_{j=1}^{m} z_j u_j v_j^\top$, where each for $j = 1, \ldots, m$, the scalar $z_j \in \operatorname{sign}(\sigma_j(\Theta))$.
- Note that we need to verify that

$$\left\|\Theta'\right\|_* \geq \|\Theta\|_* + \operatorname{tr}\left[\mathbf{Z}^\top \left(\Theta' - \Theta\right)\right]$$

for all $\Theta' \in \mathbb{R}^{m \times n}$. The inner product for matrix $\mathbf{A}$ and $\mathbf{B}$ is defined by $\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}\left(\mathbf{A}^\top \mathbf{B}\right)$.

Subgradient for nuclear norm

- Since $z_j \in \operatorname{sign}(\sigma_j(\Theta))$, then

$$\operatorname{tr}\left[\mathbf{Z}^\top(\Theta' - \Theta)\right] \leq \operatorname{tr}\left[\boldsymbol{UV}^\top(\Theta' - \Theta)\right]$$

-

$$\begin{aligned}
\|\Theta\|_* + \operatorname{tr}\left[\mathbf{Z}^\top(\Theta' - \Theta)\right] &\leq \sum_{j=1}^{m} \sigma_j + \operatorname{tr}\left[\mathbf{V}\mathbf{U}^\top\left(\Theta' - \mathbf{U}\mathbf{D}\mathbf{V}^\top\right)\right] \\
&= \operatorname{tr}\left[\mathbf{V}\mathbf{U}^\top\Theta'\right] \\
&= \operatorname{tr}\left[\mathbf{V}\mathbf{U}^\top\mathbf{U}_*\mathbf{D}_*\mathbf{V}_*^\top\right] \\
&= \sum_{j=1}^{m} \sigma'_j = \left\|\Theta'\right\|_*,
\end{aligned}$$

since $\mathbf{V}\mathbf{U}^\top\mathbf{U}_*$ can be viewed as the new left singular vectors.

Coding example

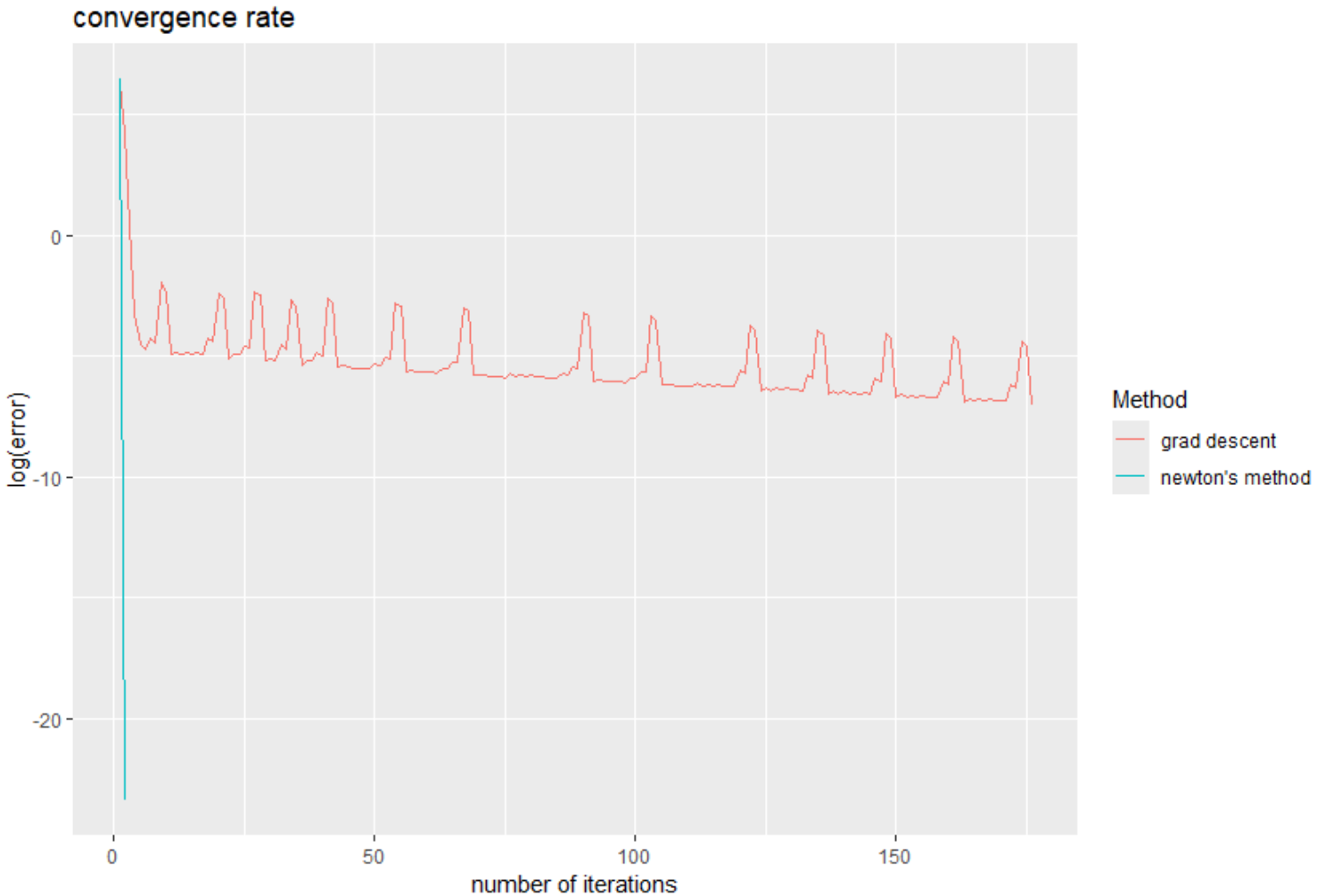
Generally, a standard gradient descent algorithm is given by the following two steps:

- Initial step: we need to initialize the parameters with proper values;
- Iterating until error is small enough.

Within the iteration, we need to compute the following two steps:

- Descent direxion: Δ^t. Taking Newton's method as an example, we have
  `delta = -(solve(hessian(fun, v)) %*% grad(fun, v))`
- Step size: s^t. We can use backtracking line search to find the step size. For example, we can use the following code:
  `s = 1; while (f(beta + s * delta) > f(beta) + 0.3 * s * t(grad(fun, beta)) %*% delta) s = 0.8 * s`

Coding example



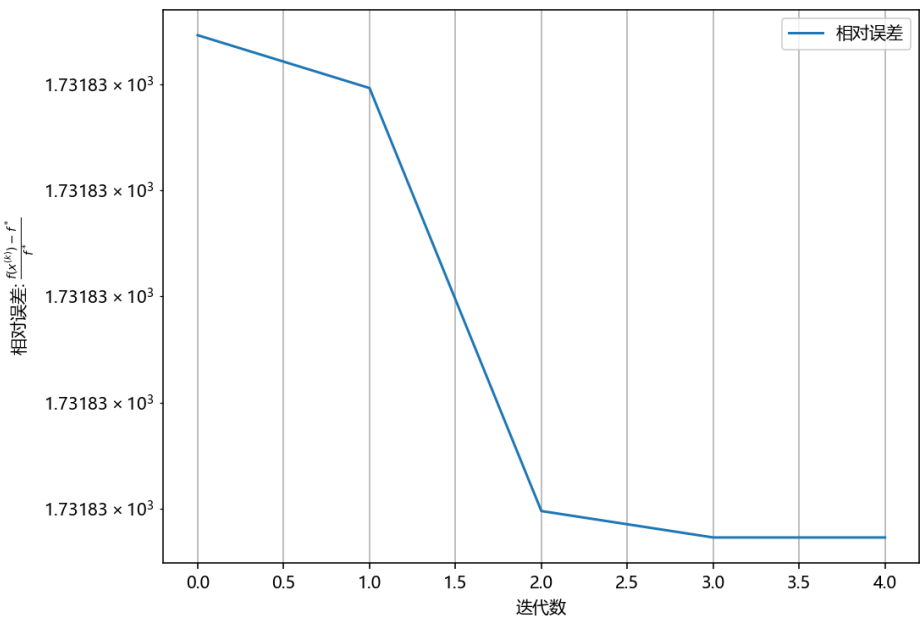
A small tip for backtracking line search

- Backtracking line search is a method to find the step size $s^\top$ at $t-$iteration in the gradient descent algorithm.

$$\beta^{t+1} = \beta^t + s^t \Delta^t.$$

- Given parameters $\alpha \in (0, 0.5)$ and $\gamma \in (0, 1)$, initializing $s = 1$, then set $s \leftarrow \gamma s$ until the following condition is satisfied:

$$f\left(\beta^t + s\Delta^t\right) \leq f\left(\beta^t\right) + \alpha s \left\langle \nabla f\left(\beta^t\right), \Delta^t \right\rangle$$



Gradient descent method with improper backtracking line search parameters.

Proximal operator

- For a convex function h,

$$\mathrm{prox}_h(x) = \underset{u \in \mathrm{dom}\, h}{\arg\min} \left\{ h(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

- Some properties of proximal map can be found in some standard textbooks, hence we omit the details here.
- For nuclear norm, the proximal operator is given by

$$\mathrm{prox}_t(\boldsymbol{B}) = \underset{\boldsymbol{Z}}{\arg\min} \frac{1}{2t}\|\boldsymbol{B} - \boldsymbol{Z}\|_F^2 + \lambda\|\boldsymbol{Z}\|_*$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\boldsymbol{B} - \boldsymbol{Z}\|_F^2 = \sum_{j=1}^{m}\sum_{k=1}^{n}(Z_{jk} - \Theta_{jk})^2$.

Matrix soft-thresholding

Matrix soft-thresholding

$$\text{prox}_t(\boldsymbol{B}) = \arg\min_{\boldsymbol{Z}} \frac{1}{2t}\|\boldsymbol{B} - \boldsymbol{Z}\|_F^2 + \lambda\|\boldsymbol{Z}\|_*,$$

then $\text{prox}_t(\boldsymbol{B}) = S_{\lambda t}(\boldsymbol{B})$, where

$$S_\lambda(\boldsymbol{B}) = \boldsymbol{U}\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}\boldsymbol{V}^\top$$

where $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the singular value decomposition of $\boldsymbol{B}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$ is diagnal with

$$(\boldsymbol{\Sigma}_{\boldsymbol{\lambda}})_{ii} = \max\{\Sigma_{ii} - \lambda, 0\}$$

Convergence analysis

Assumptions

For

$$\min \quad \psi(x) = f(x) + h(x),$$

where f is a differentiable function and h is a convex function, with iteration

$$x^{k+1} = \text{prox}_{t_k h}\left(x^k - t_k \nabla f\left(x^k\right)\right)$$

we assume the following conditions hold:

- The function f is convex and ∇f is Lipschitz continuous with respect to the gradient, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^d$$

- prox operator is well-defined.
- optimal solution $x^\star$ is finite and attainable.

Convergence analysis

For $t > 0$, we define the following "search direction":

$$G_t(x) = \frac{1}{t}\left(x - \operatorname{prox}_{th}(x - t\nabla f(x))\right).$$

Because it is exactly the negative direction of the proximal descent

$$x^{k+1} = \operatorname{prox}_{th}\left(x^k - t\nabla f\left(x^k\right)\right) = x^k - tG_t\left(x^k\right)$$

By second-order Taylor expansion, we have

$$f(y) \leqslant f(x) + \nabla f(x)^{\mathrm{T}}(y - x) + \frac{L}{2}\|y - x\|^2, \quad \forall x, y \in \mathbb{R}^n$$

Then we can obtain the following inequality:

$$f(x - tG_t(x)) \leqslant f(x) - t\nabla f(x)^{\mathrm{T}} G_t(x) + \frac{t^2 L}{2}\|G_t(x)\|^2$$

Convergence analysis

By convexity, we have

$$h(z) \geqslant h\left(x - tG_t(x)\right) + \left(G_t(x) - \nabla f(x)\right)^{\mathrm{T}}\left(z - x + tG_t(x)\right),$$
$$f(z) \geqslant f(x) + \nabla f(x)^{\mathrm{T}}(z - x),$$

Aggregating the above two inequalities, we have

$$\psi\left(x - tG_t(x)\right) \leqslant \psi(z) + G_t(x)^{\mathrm{T}}(x - z) - \frac{t}{2}\left\|G_t(x)\right\|^2, \forall z \in dom(\psi).$$

Denote $\tilde{x} = x - tG_t(x)$, then for optimal solution x^*, we have

$$\begin{aligned}
\psi(\tilde{x}) - \psi^* &\leqslant G_t(x)^{\mathrm{T}}\left(x - x^*\right) - \frac{t}{2}\left\|G_t(x)\right\|^2 \\
&= \frac{1}{2t}\left(\left\|x - x^*\right\|^2 - \left\|x - x^* - tG_t(x)\right\|^2\right) \\
&= \frac{1}{2t}\left(\left\|x - x^*\right\|^2 - \left\|\tilde{x} - x^*\right\|^2\right)
\end{aligned}$$

Convergence analysis

Then for the i-th iteration, we have

$$\begin{aligned}\sum_{i=1}^{k}\left(\psi\left(x^{i}\right)-\psi^{*}\right) &\leqslant \frac{1}{2t}\sum_{i=1}^{k}\left(\left\|x^{i-1}-x^{*}\right\|^{2}-\left\|x^{i}-x^{*}\right\|^{2}\right)\\ &= \frac{1}{2t}\left(\left\|x^{0}-x^{*}\right\|^{2}-\left\|x^{k}-x^{*}\right\|^{2}\right)\\ &\leqslant \frac{1}{2t}\left\|x^{0}-x^{*}\right\|^{2}\end{aligned}$$

Finally, by convexity, we have

$$\psi\left(x^{k}\right)-\psi^{*} \leqslant \frac{1}{k}\sum_{i=1}^{k}\left(\psi\left(x^{i}\right)-\psi^{*}\right) \leqslant \frac{1}{2kt}\left\|x^{0}-x^{*}\right\|^{2}$$

Nesterov's acceleration

- Proximal GD + regular conditions $\rightsquigarrow \mathcal{O}\left(\frac{1}{k}\right)$ convergence rate.
- Nesterov gave three acceleration improvements (1983, 1988, 2005) for convex optimization problems $\rightsquigarrow \mathcal{O}\left(\frac{1}{k^2}\right)$.
- At first, Nesterov's acceleration method was neglected because Newton's method had a better convergence rate.
- However, Newton's method has a high computational cost, and Nesterov's method is more efficient in practice.
- Beck and Teboulle (2008) proposed FISTA, which is a PGD version of Nesterov's (1983) method.

PGD vs FISTA

Proximal Gradient Descent (PGD) algorithm:

1: **Input:** Function $f(x), h(x)$, initial point x^0, initialize $k = 0$
2: **while** not converged **do**
3: $\quad x^{k+1} = \text{prox}_{t_k h}\left(x^k - t_k \nabla f(x^k)\right)$
4: $\quad k \leftarrow k + 1$
5: **end while**

FISTA(look-ahead):

1: **Input:** $x^0 = x^{-1} \in \mathbb{R}^n$, $k \leftarrow 1$
2: **while** not converged **do**
3: $\quad$ Compute $y^k = x^{k-1} + \frac{k-2}{k+1}\left(x^{k-1} - x^{k-2}\right)$
4: $\quad$ Select $t_k = t \in \left(0, \frac{1}{L}\right]$
5: $\quad$ Compute $x^k = \text{prox}_{t_k h}\left(y^k - t_k \nabla f(y^k)\right)$
6: $\quad k \leftarrow k + 1$
7: **end while**

Why FISTA is faster?

- FISTA's convergence rate is $\mathcal{O}\left(\frac{1}{k^2}\right)$ instead of $\mathcal{O}\left(\frac{1}{k}\right)$ for PGD.
- We make the same assumptions as in the previous setup.
- And for simplicity, we make a fixed step size $t_k = \frac{1}{L}$.
- Then

$$\psi\left(x^k\right) - \psi\left(x^*\right) \leqslant \frac{2L}{(k+1)^2}\left\|x^0 - x^*\right\|^2$$

- The key is that FISTA uses the previous two iterations to compute the next iteration, which is a look-ahead step and hence under regular conditions,

$$\frac{t_k}{\gamma_k^2}\left(\psi\left(x^k\right) - \psi^*\right) + \frac{1}{2}\left\|v^k - x^*\right\|^2 \leqslant \frac{t_{k-1}}{\gamma_{k-1}^2}\left(\psi\left(x^{k-1}\right) - \psi^*\right) + \frac{1}{2}\left\|v^{k-1} - x^*\right\|^2$$

General convergence conditions

With the following key assumptions,

- Lipschitz continuity of the gradient of f (i.e., L-smoothness),
- $$f\left(x^k\right) \leqslant f\left(y^k\right) + \left\langle \nabla f\left(y^k\right), x^k - y^k \right\rangle + \frac{1}{2t_k} \left\| x^k - y^k \right\|_2^2$$
- $$\gamma_1 = 1, \quad \frac{\left(1-\gamma_i\right) t_i}{\gamma_i^2} \leqslant \frac{t_{i-1}}{\gamma_{i-1}^2}, \quad i > 1,$$
- $$\frac{\gamma_k^2}{t_k} = \mathcal{O}\left(\frac{1}{k^2}\right)$$

Remarks

- The assumptions are satisfied when $t_k = \frac{1}{L}$ and $\gamma_k = \frac{2}{k+1}$.
- A large class of Nesterov methods can be proved to be $\mathcal{O}\left(\frac{1}{k^2}\right)$.

Least squares and maximum likelihood in logistic regression

- Least square: $L_S(y_i, \hat{y}_i) = \frac{1}{2}(y_i - \hat{y}_i)^2$.

$$\hat{\beta}_S := \text{argmin}_{b \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - g^{-1}\left(x_i^T b\right)\right)^2$$

- Likelihood: $L_L(y_i, \hat{y}_i) = y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)$.

$$\hat{\beta}_L := \text{argmin}_{b \in \mathbb{R}^p} - \sum_{i=1}^{n} y_i \log g^{-1}\left(x_i^T b\right) + (1 - y_i) \log\left(1 - g^{-1}\left(x_i^T b\right)\right)$$

Let $h = g^{-1}$, then

$$h(z) = \frac{1}{1 + e^{-z}} \Longrightarrow h'(z) = h(z)(1 - h(z))$$

Maximum likelihood to least squares

For regular logistic regression we have

$$\frac{\partial f_L}{\partial b_j} = -\sum_{i=1}^{n} h'\left(x_i^T b\right) x_{ij} \left(\frac{y_i}{h\left(x_i^T b\right)} - \frac{1-y_i}{1-h\left(x_i^T b\right)} \right)$$

Using $h' = h \cdot (1-h)$ we can simplify this to

$$\frac{\partial f_L}{\partial b_j} = -\sum_{i=1}^{n} x_{ij} \left(y_i \left(1-\hat{y}_i\right) - \left(1-y_i\right)\hat{y}_i\right) = -\sum_{i=1}^{n} x_{ij}\left(y_i - \hat{y}_i\right)$$

so

$$\nabla f_L(b) = -X^T(Y - \hat{Y})$$

Next let's do second derivatives. The Hessian

$$H_L := \frac{\partial^2 f_L}{\partial b_j \partial b_k} = \sum_{i=1}^{n} x_{ij} x_{ik} \hat{y}_i \left(1-\hat{y}_i\right) = X^\top diag\left(\hat{Y}(1-\hat{Y})\right) X \geq 0.$$

Logistics regression and coordinate descent

- Form a quadratic objective function using Taylor expansion about current estimates $\left(\tilde{\beta}_0, \tilde{\beta}\right)$: Idea of Newton method, Iterated Weighted Least Square problem

$$L_Q\left(\beta_0, \beta\right) = -\frac{1}{2N}\sum_{i=1}^{N} w_i\left(z_i - \beta_0 - x_i^T\beta\right)^2 + C\left(\tilde{\beta}_0, \tilde{\beta}\right)$$

- Use Coordinate Descent to solve the problem $\text{minimize}_{(\beta_0,\beta)}\left\{-L_Q\left(\beta_0, \beta\right) + \lambda\|\beta\|_1\right\}$

Logistics regression and coordinate descent(Freidman et al., 2010)

$$\min_{(\beta_{0\ell},\beta_\ell)\in\mathbb{R}^{p+1}} \{-\ell_{Q\ell}(\beta_{0\ell},\beta_\ell) + \lambda P_\alpha(\beta_\ell)\}\,. \tag{26}$$

This amounts to the sequence of nested loops:

outer loop: Decrement λ.

middle loop (outer): Cycle over $\ell \in \{1, 2, \ldots, K, 1, 2 \ldots\}$.

middle loop (inner): Update the quadratic approximation $\ell_{Q\ell}$ using the current parameters $\{\tilde{\beta}_{0k}, \tilde{\beta}_k\}_1^K$.

inner loop: Run the co-ordinate descent algorithm on the penalized weighted-least-squares problem (26).

Alternating direction method of multipliers (ADMM)

- Optimization problem:

$$\text{minimize}_{\beta\in\mathbb{R}^m,\theta\in\mathbb{R}^n}\ f(\beta)+g(\theta) \quad \text{subject to } \mathbf{A}\beta+\mathbf{B}\theta-c=0$$

- Lagrangian:

$$f(\beta)+g(\theta)+\langle\mu,\mathbf{A}\beta+\mathbf{B}\theta-c\rangle$$

- Augmented Lagrangian:

$$L_\rho(\beta,\theta,\mu):=f(\beta)+g(\theta)+\langle\mu,\mathbf{A}\beta+\mathbf{B}\theta-c\rangle+\frac{\rho}{2}\|\mathbf{A}\beta+\mathbf{B}\theta-c\|_2^2$$

- The characteristic of this method is that the objective function can be separated into two parts, but the constraints are coupled.

- $$\min_x \quad f_1(x)+f_2(x) \iff \begin{array}{ll}\min\limits_{x,z} & f_1(x)+f_2(z)\\ \text{s.t.} & x-z=0.\end{array}$$

Variable update

- $$\beta^{t+1} = \arg\min_{\beta \in \mathbb{R}^m} L_\rho\left(\beta, \theta^t, \mu^t\right)$$
$$\theta^{t+1} = \arg\min_{\theta \in \mathbb{R}^m} L_\rho\left(\beta^{t+1}, \theta, \mu^t\right)$$
$$\mu^{t+1} = \mu^t + \rho\left(\mathbf{A}\beta^{t+1} + \mathbf{B}\theta^{t+1} - c\right)$$
where ρ is a stepsize parameter, and it usually takes value at $\left(0, \frac{1+\sqrt{5}}{2}\right]$
- Pros: objective function is not necessarily to be smooth.

ADMM for lasso

- Problem in Lagrangian form

$$\text{minimize}_{\beta\in\mathbb{R}^p,\theta\in\mathbb{R}^p}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2+\lambda\|\theta\|_1\right\} \quad \text{such that } \beta-\theta=0$$

- Augmented Lagrangian

$$L_\rho(\beta,\theta,\mu):=\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2+\lambda\|\theta\|_1\right\}+\langle\mu,\beta-\theta\rangle+\frac{\rho}{2}\|\beta-\theta\|_2^2$$

- Update rules:

$$\begin{aligned}\beta^{t+1}&=\left(\mathbf{X}^T\mathbf{X}+\rho\mathbf{l}\right)^{-1}\left(\mathbf{X}^T\mathbf{y}+\rho\theta^t-\mu^t\right)\\ \theta^{t+1}&=\mathcal{S}_{\lambda/\rho}\left(\beta^{t+1}+\mu^t/\rho\right)\\ \mu^{t+1}&=\mu^t+\rho\left(\beta^{t+1}-\theta^{t+1}\right)\end{aligned}$$

where $\mathcal{S}_{\lambda/\rho}(z)=\text{sign}(z)\left(|z|-\frac{\lambda}{\rho}\right)_+$.

Screening Rule

- $\lambda_{\max} = \max_j |\langle x_j, y \rangle|$, define $\phi = \dfrac{\theta}{\lambda}$, consider the following lasso dual problem

$$\max_{\theta} \frac{1}{2} \left\{ \|y\|_2^2 - \lambda^2 \|y/\lambda - \phi\|_2^2 \right\}, \quad \|X^\top \phi\|_\infty \leq 1.$$

- Assuming that $\lambda_{\max} \geq \lambda' > 0$, lasso dual problem has a solution $\hat{\phi}(\lambda')$. If $\lambda > 0$ and $\lambda \neq \lambda'$ and if

$$\left| \mathbf{x}_j^T \hat{\boldsymbol{\phi}} \left(\lambda' \right) \right| < 1 - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \left| \frac{1}{\lambda'} - \frac{1}{\lambda} \right|,$$

  then $\hat{\beta}_j = 0$.
- We know from the stationarity conditions for the lasso that

$$\left| \mathbf{x}_j^T \hat{\boldsymbol{\phi}}(\lambda) \right| < 1 \Longrightarrow \widehat{\beta}_j(\lambda) = 0$$

Dual polytope projection rule

From the dual, $\hat{\boldsymbol{\phi}}(\lambda)$ is the projection of $\mathbf{y}/\lambda$ into the feasible set $\mathcal{F}_\lambda$. By the projection theorem for closed convex sets, $\hat{\boldsymbol{\phi}}(\lambda)$ is continuous and nonexpansive, which implies

$$\begin{aligned}\left\|\hat{\boldsymbol{\phi}}(\lambda) - \hat{\boldsymbol{\phi}}\left(\lambda'\right)\right\|_2 &\leq \left\|\frac{\mathbf{y}}{\lambda} - \frac{\mathbf{y}}{\lambda'}\right\|_2 \\ &= \|\mathbf{y}\|_2 \left|\frac{1}{\lambda} - \frac{1}{\lambda'}\right|\end{aligned}$$

Then

$$\begin{aligned}\left|\mathbf{x}_j^T \hat{\boldsymbol{\phi}}(\lambda)\right| &\leq \left|\mathbf{x}_j^T \hat{\boldsymbol{\phi}}(\lambda) - \mathbf{x}_j^T \hat{\boldsymbol{\phi}}\left(\lambda'\right)\right| + \left|\mathbf{x}_j^T \hat{\boldsymbol{\phi}}\left(\lambda'\right)\right| \\ &< \|\mathbf{x}_j\|_2 \left\|\hat{\boldsymbol{\phi}}(\lambda) - \hat{\boldsymbol{\phi}}\left(\lambda'\right)\right\|_2 + 1 - \|\mathbf{x}_j\| \, \|_2 \| \, \mathbf{y}\|_2 \left|\frac{1}{\lambda'} - \frac{1}{\lambda}\right| \\ &\leq \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \left|\frac{1}{\lambda'} - \frac{1}{\lambda}\right| + 1 - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \left|\frac{1}{\lambda'} - \frac{1}{\lambda}\right| \quad = 1\end{aligned}$$

Dual polytope projection rule

Global DPP Rule

Suppose we want to calculate a Lasso solution at $\lambda < \lambda_{\max}$. The DPP rule discards the j^{th} variable if

$$\left|\mathbf{x}_j^T \mathbf{y}\right| < \lambda_{\max} - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda}$$

Sequential DPP Rule

Suppose we have the Lasso solution $\hat{\beta}(\lambda')$ at λ' and want to screen variables for solutions at $\lambda < \lambda'$. We discard the j^{th} variable if

$$\left|\mathbf{x}_j^T \left(\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda')\right)\right| < \lambda' - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda}$$

Part II

- Bayesian lasso
- Bootstrap
- Post-selection inference for lasso
- Inference via a debiased lasso

Why should we do inference?

We are trying to answer questions:

- What can we learn about the underlying distribution from the observed set of outcomes of a given sample size?
- How confident can we be of the inference based on a certain number of observations and a certain experiment design?
- The confidence affects the decision-making process

For lasso:

- An attractive feature of l_1-regularized procedures is their ability to combine variable selection with parameter fitting.
- It is sometimes of interest to determine the statistical strength of the included variables, as in " p-value" in traditional models.

Case study: diabetes data

- The diabetes data set is a well-known data set in the statistics community.
- The diabetes dataset consists of 10 physiological variables (age, sex, weight, blood pressure) measure on 442 patients (x), and the response variable (y) is a quantitative measure of disease progression one year after baseline.
- The matrix x2 consists of x plus certain interactions (we ignore this part for simplicity).
- Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) Annals of Statistics, uses this data set to illustrate the lasso and the lars algorithm.

Bayesian lasso

The Bayesian paradigm treats the parameters as random quantities, along with a prior distribution that characterizes our belief in what their values might be:

$$y \mid \beta, \lambda, \sigma \sim N\left(\mathbf{X}\beta, \sigma^2 \mathbf{I}_{N\times N}\right),$$

$$\beta \mid \lambda, \sigma \sim \prod_{j=1}^{p} \frac{\lambda}{2\sigma} e^{-\frac{\lambda}{\sigma}|\beta_j|},$$

using the i.i.d Laplacian prior. Under this model, the negative $\log$ posterior density for $\beta \mid y, \lambda, \sigma$ is given by:

$$\frac{1}{2\sigma^2}\|y - \mathbf{X}\beta\|_2^2 + \frac{\lambda}{\sigma}\|\beta\|_1$$

where we have dropped an additive constant independent of β, and we assume that the columns of X are mean-centered, as is y.

Bayesian lasso

The negative log posterior density for $\beta \mid y, \lambda, \sigma$ is given by:

$$\frac{1}{2\sigma^2}\|y - \mathbf{X}\beta\|_2^2 + \frac{\lambda}{\sigma}\|\beta\|_1.$$

Consequently, for any fixed values of σ and λ, the posterior mode coincides with the lasso estimate.
In classical lasso, we don't assume any distribution of y, thus the distribution of $\hat{\beta}(y, \lambda)$ is unknown, but under the assumptions of Bayesian lasso, we are now able to get the distribution of $\hat{\beta}(y, \lambda)$, and then making inference is possible.

Bayesian inference framework

1. $f(\beta \mid \lambda, \sigma)$ and $f_X(y \mid \beta, \lambda, \sigma)$
2. Get posterior distribution $h_X(\beta \mid y, \lambda, \sigma)$, and $\hat{\beta}(y, \lambda, \sigma) = \operatorname{argmin}_\beta - \log(h_X)$
3. Get posterior distribution $g_x(\hat{\beta} \mid \lambda, \sigma)$ ($\hat{\beta}$ is a function of y , and distribution of $y \mid \lambda, \sigma$ is known.)

Cons

In practice, exact Bayesian calculations are typically intractable, except for the simplest models.

Bayesian lasso: posteriors

- It can be shown that he posterior distribution of β is given by

$$p(\beta_j \mid \cdot) \propto \exp\left\{-\frac{1}{2\sigma^2}\left[\beta_j^2\left(X_j^T X_j + \lambda\right) - 2\beta_j X_j^T r_j\right]\right\}$$

  hence $\beta_j \mid \cdot \sim N\left(\mu_j, \sigma_j^2\right)$ where

$$\sigma_j^2 = \frac{\sigma^2}{X_j^T X_j + \lambda}, \quad \mu_j = \frac{X_j^T r_j}{X_j^T X_j + \lambda}$$

- The posterior distribution of σ^2 is given by

$$\sigma^2 \mid \cdot \sim \text{ Inverse-Gamma } \left(\frac{\nu}{2}, \frac{s^2}{2}\right)$$

  where

$$\nu = n + 2, \quad s^2 = \|y - X\beta\|^2$$

Bayesian Lasso

I introduced a Gibbs sampler for the Bayesian Lasso model. This algorithm was introduced by Trevor PARK and George CASELLA (2008).
Firstly, we have the following identity:

$$\frac{a}{2}e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}}e^{-z^2/(2s)}\frac{a^2}{2}e^{-a^2 s/2}ds$$

This can be verified by the following identity:

$$\int_0^\infty e^{-\frac{A}{x^2}-Bx^2}dx = \frac{\sqrt{\pi}}{2\sqrt{B}}e^{-2\sqrt{AB}}$$

Hence, Laplace prior can be expressed as a Gaussian distribution with an exponential prior on the variance.

Bayesian Lasso

This suggests the following hierarchical representation of the full model:

$$\begin{aligned}
\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2 &\sim \mathbf{N}_n\left(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n\right), \\
\boldsymbol{\beta} \mid \sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \mathrm{N}_p\left(\mathbf{0}_p, \sigma^2 \mathbf{D}_\tau\right), \\
\mathbf{D}_\tau &= \operatorname{diag}\left(\tau_1^2, \ldots, \tau_p^2\right), \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \sim \pi\left(\sigma^2\right) d\sigma^2 \prod_{j=1}^{p} \frac{\lambda^2}{2} e^{-\lambda^2 \tau_j^2/2} d\tau_j^2, \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &> 0.
\end{aligned}$$

After integrating out $\tau_1^2, \ldots, \tau_p^2$, the conditional prior on $\boldsymbol{\beta}$ has the desired form

$$\pi\left(\boldsymbol{\beta} \mid \sigma^2\right) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}}$$

We use the improper prior density $\pi\left(\sigma^2\right) = 1/\sigma^2$, but any inverse-gamma prior for σ^2 also would maintain conjugacy. (This answers conjugacy.)

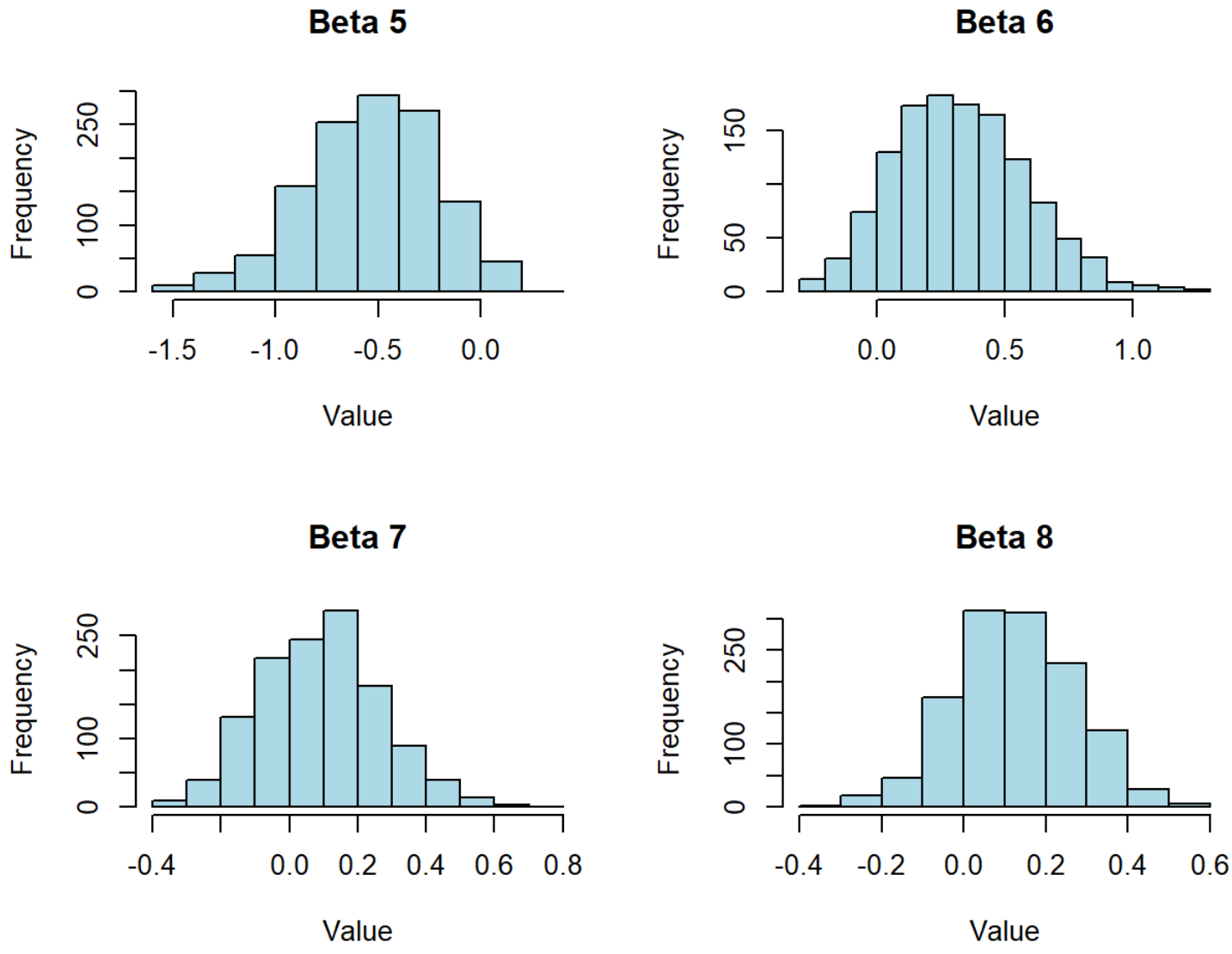
Bayesian Lasso

Element-wise:

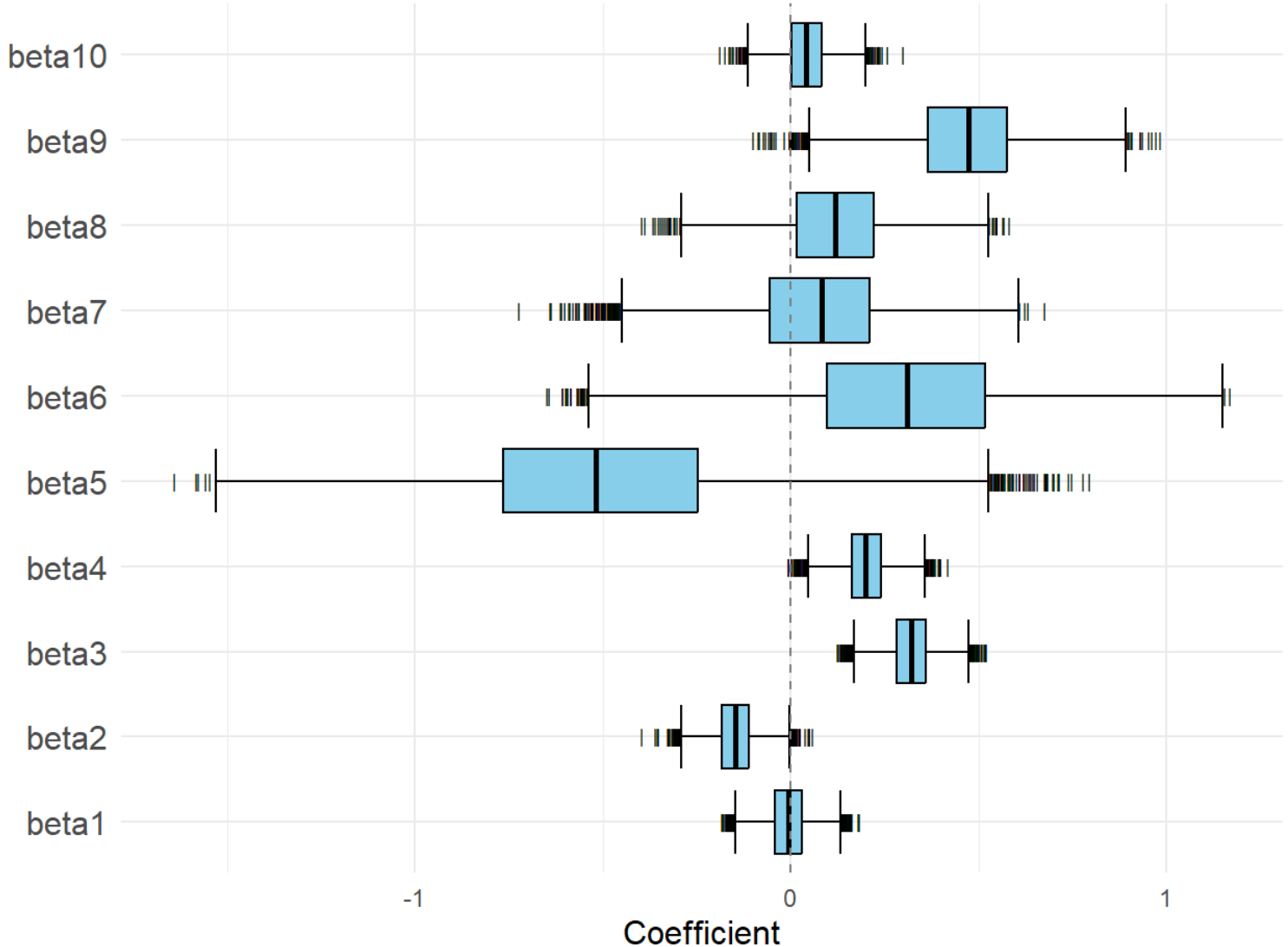
$$\text{Laplace}\left(0, \frac{\sigma}{\lambda}\right) \equiv \int N\left(0, \sigma^2\tau_j^2\right) \cdot \text{Exp}\left(\tau_j^2 \,\middle|\, \frac{\lambda^2}{2}\right) d\tau_j^2$$

$$\begin{aligned} y \mid \beta, \sigma^2 &\sim N\left(X\beta, \sigma^2 I\right), \\ \beta_j \mid \tau_j^2, \sigma^2 &\sim N\left(0, \sigma^2\tau_j^2\right), \\ \tau_j^2 &\sim \text{Exponential}\left(\frac{\lambda^2}{2}\right), \\ \sigma^2 &\sim \text{Inverse-Gamma}\left(a_0, b_0\right) \quad \text{(Inverse Gamma prior).} \end{aligned}$$

MCMC samples



MCMC boxplot



Bayesian Lasso

$$y \mid \beta, \lambda, \sigma \sim N\left(\mathbf{X}\beta, \sigma^2 \mathbf{I}_{N\times N}\right),$$

$$\beta \mid \lambda, \sigma \sim \prod_{j=1}^{p} \frac{\lambda}{2\sigma} e^{-\frac{\lambda}{\sigma}|\beta_j|},$$

posterior

$$p(\beta \mid y, \lambda, \sigma) \propto p(y \mid \beta, \lambda, \sigma) \cdot p(\beta \mid \lambda, \sigma)$$

i.e.,

$$p(\beta \mid y, \lambda, \sigma) \propto \exp\left(-\frac{1}{2\sigma^2}\|y - X\beta\|^2 - \frac{\lambda}{\sigma}\|\beta\|_1\right)$$

My remarks

- They plot boxplot with `outlier.shape = 124, outlier.size = 2, stat_boxplot(geom = "errorbar", linewidth = 0.5, color = "black", size = 0.4)`.
- Something wrong with the left part of figure 6-3, SLS. The β_7's distribution doesn't coincides with figure 6-1, SLS.
- SLS sample 10000 samples with a 1000 burn-in period, which is not enough.
- Figure 6-1 thin the MCMC samples, and the boxplot is not thin.

The Bootstrap: recap

- Setup:
  - $Z_1, \ldots, Z_n \sim_{i.i.d} P$
  - We are interested in some parameter θ of P.
  - We have an estimator $\hat{\theta}_n = g(Z_1, \ldots, Z_n)$ and we want to know the distribution of $\hat{\theta}_n$, so that we can make inference about θ
- Main idea:
  - If we knew P, we could simulate many data sets to obtain the distribution of $\hat{\theta}_n$.
  - Since we don't know P , we simulate from an estimated version.
  - In non-parametric bootstrap we simulate from the empirical distribution $\hat{P}_n$ which places mass $1/n$ on each data point. This amounts to resampling the data with replacement.
  - In parametric bootstrap we first estimate θ by some estimate $\hat{\theta}$, then simulate bootstrap samples from $P_{\hat{\theta}_n}$.

Nonparametric bootstrap

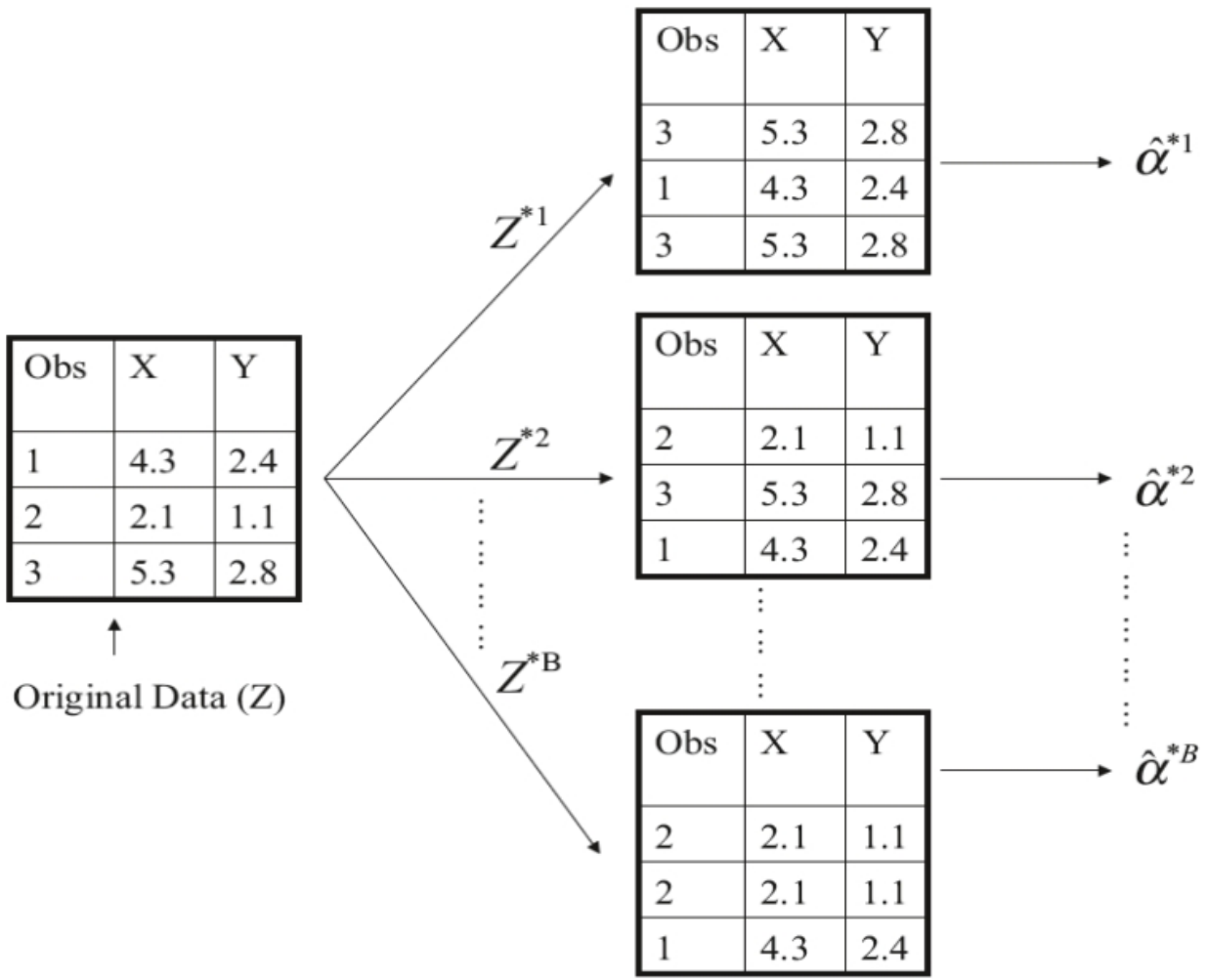


Figure: Nonparametric bootstrap: from ISLR

Bootstrap statistical inference

- The bootstrap is called to be consistent for $\hat{\theta}_n$ if, for all x,

$$\mathbb{P}\left[a_n\left(\hat{\theta}_n - \theta\right) \leq x\right] - \mathbb{P}^*\left[a_n\left(\hat{\theta}_n^* - \hat{\theta}_n\right) \leq x\right] \quad P \quad 0 (n \to \infty).$$

- In classical situations, $a_n = \sqrt{n}$: for example, the maximum-likelihood estimator $\hat{\theta}_n$ satisfies under regularity assumptions

$$\sqrt{n}\left(\hat{\theta}_{n,\mathrm{MLE}} - \theta\right) \quad \xrightarrow{D} \mathcal{N}\left(0, I^{-1}(\theta)\right) (n \to \infty)$$

  where $I(\theta)$ denotes the Fisher information at θ.
- Bootstrap consistency then means

$$\sqrt{n}\left(\hat{\theta}_{n,\mathrm{MLE}}^* - \hat{\theta}_n\right) \xrightarrow{D^*} \mathcal{N}\left(0, I^{-1}(\theta)\right) \text{ in probability } (n \to \infty).$$

- Consistency of the bootstrap typically holds if the limiting distribution of $\hat{\theta}_n$ is Normal, and if the data $Z_1, \ldots, Z_n$ are i.i.d.

10-fold cross-validation: recap

We first obtain an estimate $\hat{\beta}\left(\hat{\lambda}_{CV}\right)$ for a lasso problem according to the following procedures:

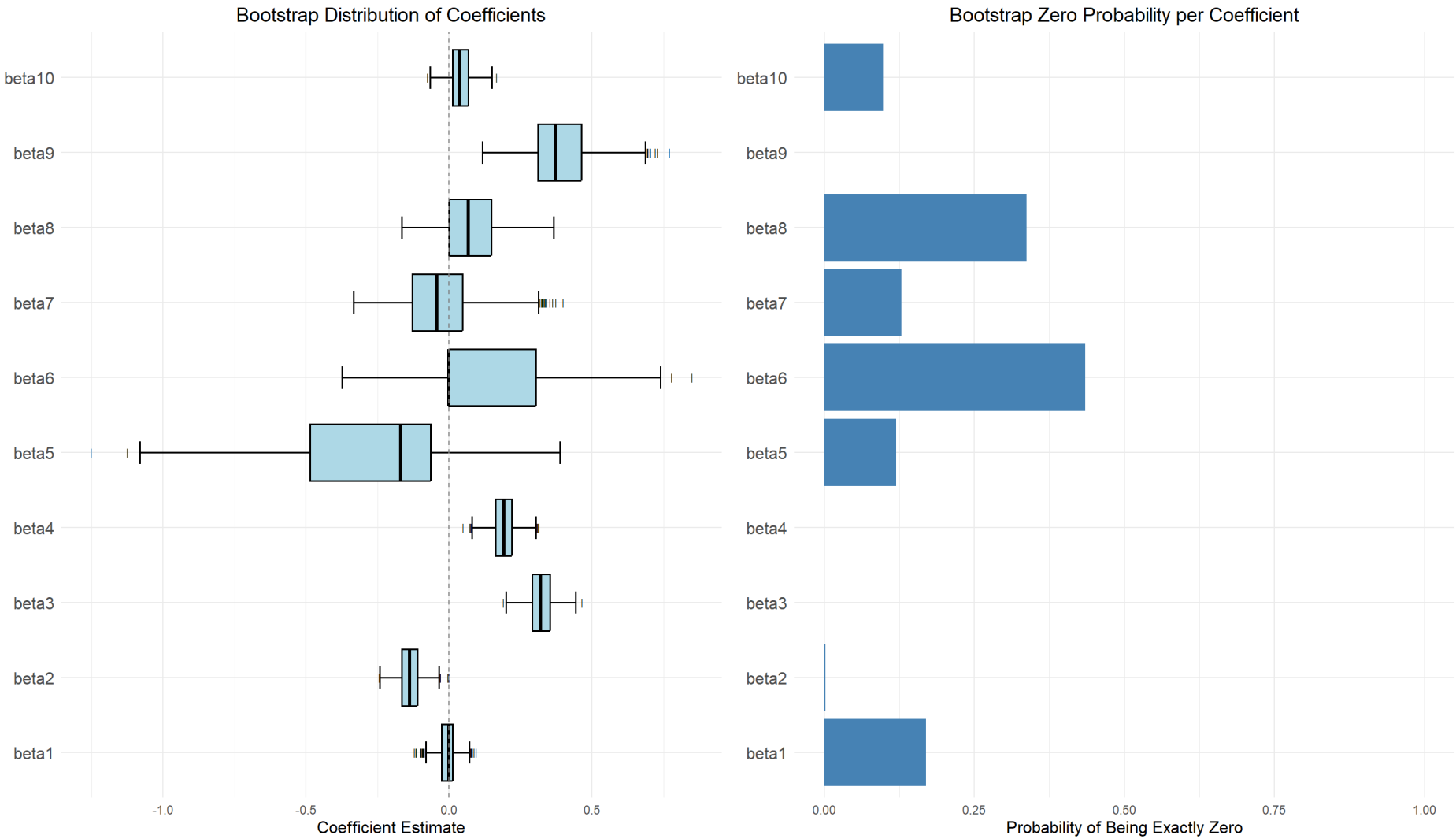
1. Fit a lasso path to (X, y) over a dense grid of values $\Lambda = \{\lambda_l\}_{l=1}^{L}$.
2. Divide the training samples into 10 groups at random.
3. With the k^{th} group left out, fit a lasso path to the remaining $9/10$ths, using the same grid Λ.
4. For each $\lambda \in \Lambda$ compute the mean-squared prediction error for the left-out group.
5. Average these errors to obtain a prediction error curve over the grid $\wedge$.
6. Find the value $\hat{\lambda}_{CV}$ that minimizes this curve, and then return the coefficient vector from our original fit in step (1) at that value of λ.

Bootstrap for lasso inference

Then we approximate the cumulative distribution F of the random pair (X, Y) by the empirical $\mathrm{CDF}\ \hat{F}_N$ defined by the N samples:

- Draw N samples from $\hat{F}_N$ as a bootstrap sample, which amounts to drawing N samples with replacement from the given data set.
- Obtain $\hat{\beta}^* \left(\hat{\lambda}_{CV}\right)$ by repeating steps 1-6 on each bootstrap sample.
- Draw 1000 bootstrap samples, and use the 1000 bootstrap realizations $\hat{\beta}^* \left(\hat{\lambda}_{CV}\right)$ to make inference.

Nonparametric bootstrap



Pairwise plot (nonparametric bootstrap)

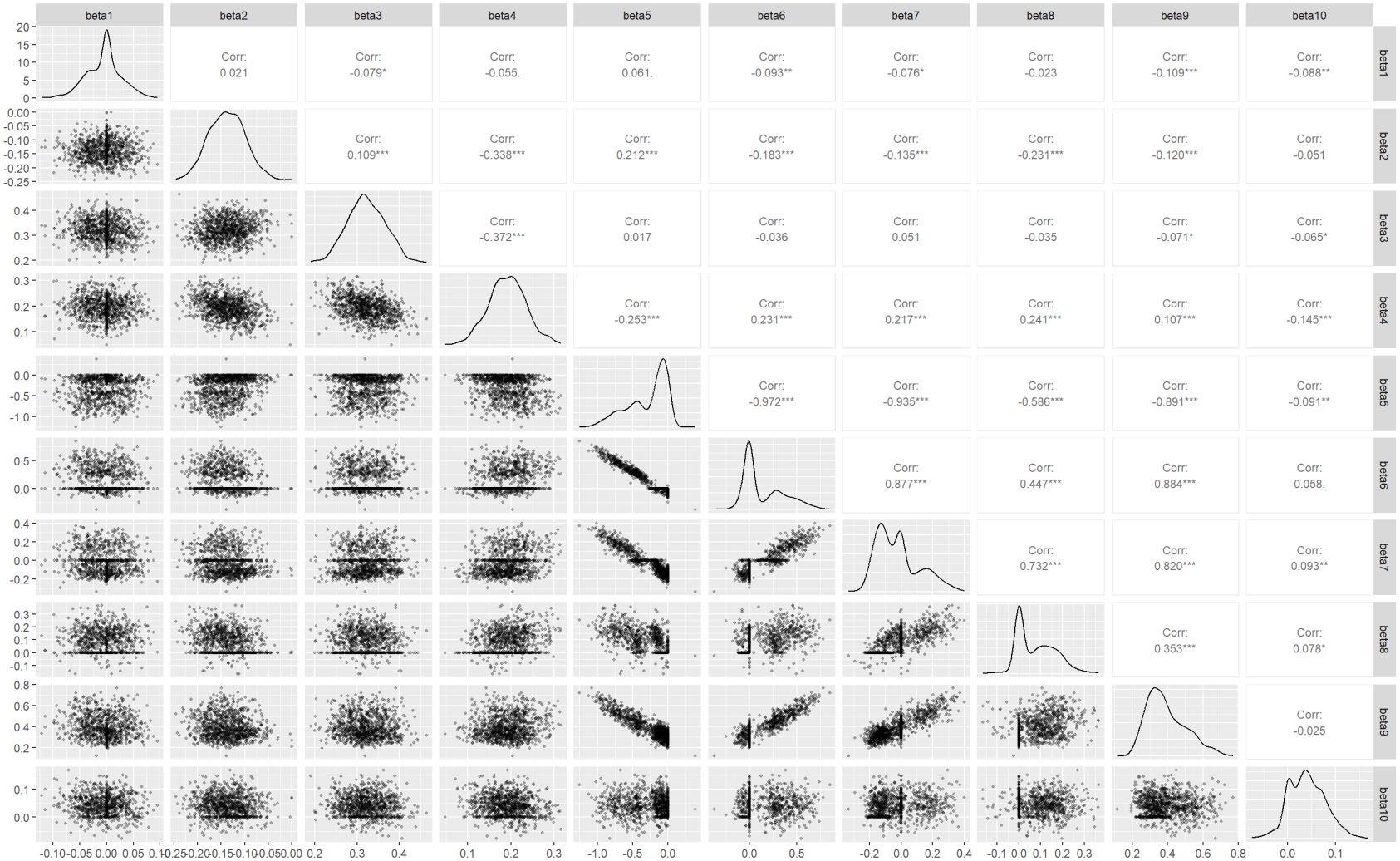


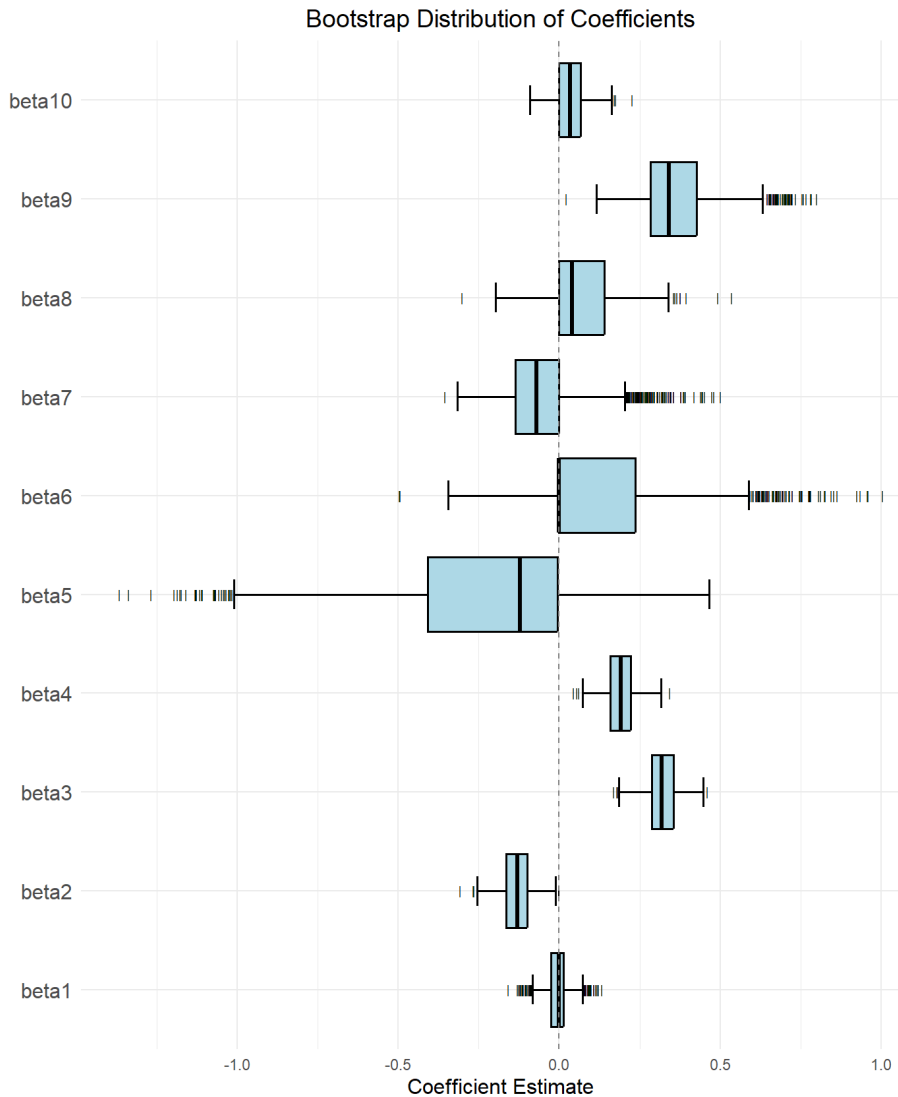
Figure: The samples $\mathbf{x}_5$ and $\mathbf{x}_6$ have high correlation (0.9); we see the corresponding negative correlation in their coefficients, with zero playing a prominent role

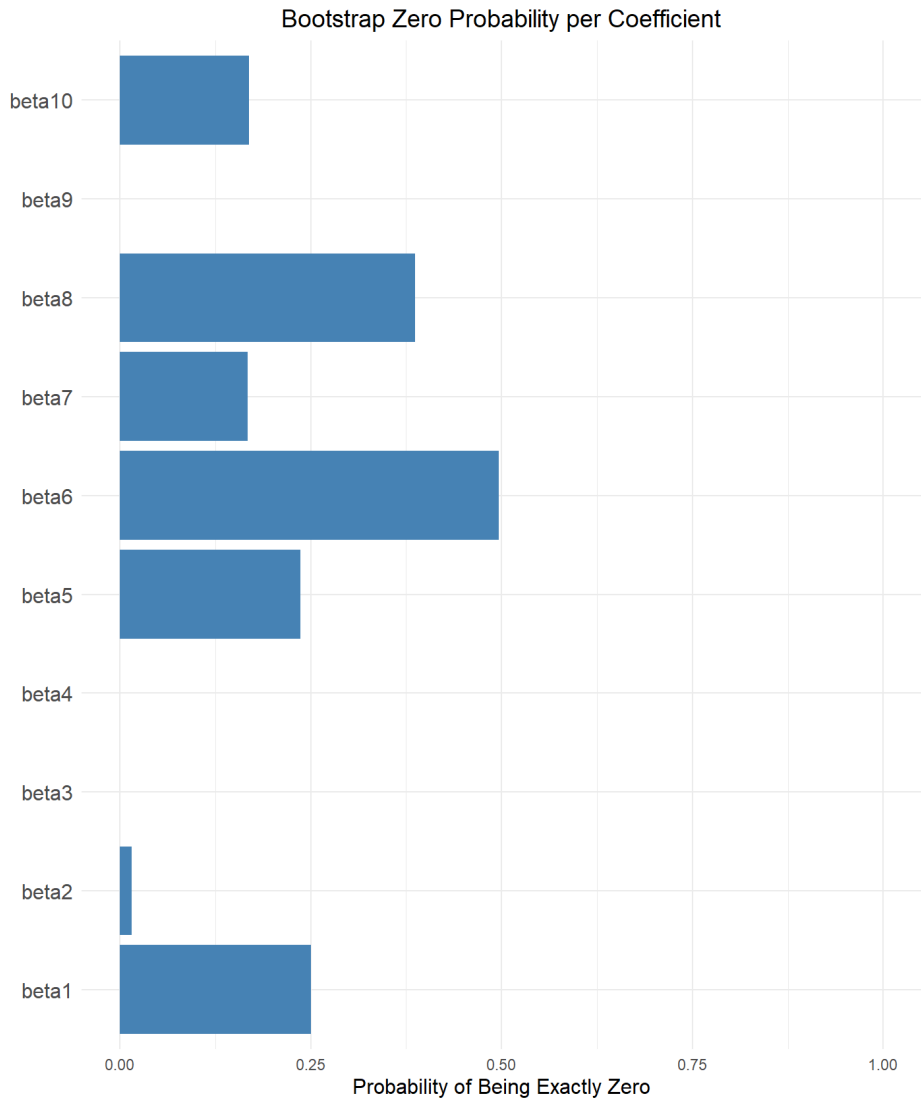
Parametric bootstrap

In contrast to the non-parametric bootstrap, the parametric bootstrap samples from a parametric estimate of F :

- fix X and obtain estimates $\hat{\beta}$ and $\hat{\sigma}^2$ from the full least-squares fit $Y \sim X$.
- then we sample y^* from the Gaussian model (6.1a), that is, $y \mid \beta, \sigma \sim N\left(\mathbf{X}\beta, \sigma^2 \mathbf{l}_{N\times N}\right)$.
- consider (X, y^*) as a new bootstrap sample, then repeat steps $1 \sim 6$ on this sample and get $\hat{\beta}^*\left(\hat{\lambda}_{CV}\right)$.
- draw 1000 bootstrap samples, and use the 1000 bootstrap realizations $\hat{\beta}^*\left(\hat{\lambda}_{CV}\right)$ to make inference.

Parametric bootstrap





Comparison of bootstrap and Bayesian lasso

Table 6.1 *Timings for Bayesian lasso and bootstrapped lasso, for four different problem sizes. The sample size is* $N = 400$.

p	Bayesian Lasso	Lasso/Bootstrap
10	3.3 secs	163.8 secs
50	184.8 secs	374.6 secs
100	28.6 mins	14.7 mins
200	4.5 hours	18.1 mins

- From table 6.1 we know Bayesian lasso is faster for small problems, but its complexity seems to scale as $\mathcal{O}\left(p^2\right)$.
- In contrast, the scaling of the bootstrap seems to be closer to $\mathcal{O}(p)$.
- As we move to GLMs, the Bayesian technical complexities grow, while bootstrap can be applied seamlessly in many situations.

Post-selection inference for lasso

In this section we present some relatively recent ideas on making inference after selection by adaptive methods such as lasso and forward-stepwise regression.

Assumptions

The usual linear regression setup, with an outcome vector $y \in \mathbb{R}^N$ and matrix of predictor variables $X \in \mathbb{R}^{N \times p}$ related by:

$$y = \mathbf{X}\beta + \epsilon, \quad \epsilon \sim N\left(0, \sigma^2 I_{N \times N}\right)$$

where $\beta \in \mathbb{R}^p$ are unknown coefficients to be estimated.

Forward-stepwise regression

Consider forward-stepwise regression. This procedure enters predictors one at a time, choosing the predictors that most decreases the residual sum of squares at each stage:

- Defining RSS_k to be the residual sum of squares for the model containing k predictors;
- we use this change in residual sum of squares to form a test statistic:

$$R_k = \frac{1}{\sigma^2}\left(RSS_{k-1} - RSS_k\right)$$

  with σ assumed to be known now.
- Compare R_k to a $\chi^2(1)$ distribution.

Problems with forward-stepwise regression

Suppose at step k-1, there are n candidates $x_1, \ldots, x_n$ that can be added. For each candidate i , we can calculate:

$$R_k(i) = \frac{1}{\sigma^2}\left(RSS_{k-1} - RSS_k(i)\right)$$

where $RSS_k(i)$ is the RSS after adding the ith candidate. Then $R_k(i) \sim \chi^2(1)$ under null hypothesis $(\beta_i = 0), \forall i \in 1, \ldots, n$.

Problem

However, in forward-stepwise regression, at each step we intentionally choose the candidate with largest $R_k(i)$, that is, $R_k = \max_i R_k(i)$, which is actually not $\chi^2(1)$ distribution anymore!

Simulation study

```
for (i in 1:n_sim) {
  # 生成随机设计矩阵 X
  X_raw <- matrix(rnorm(N * p), N, p)
  X <- qr.Q(qr(X_raw)) # 正交化
  # 生成响应变量 y ~ N(0, 1)
  y <- rnorm(N)
  # 空模型 RSS
  RSS0 <- sum(y^2)
  # 计算每个变量与 y 的投影，从中选择使 RSS 降低最多的
  RSS_drop <- sapply(1:p, function(j) {
    fit <- lm(y ~ X[, j])
    RSS0 - sum(resid(fit)^2)
  })
  # 记录最大的 RSS drop 作为 R1
  R1_vals[i] <- max(RSS_drop)
}
```

Simulation study: qqplot

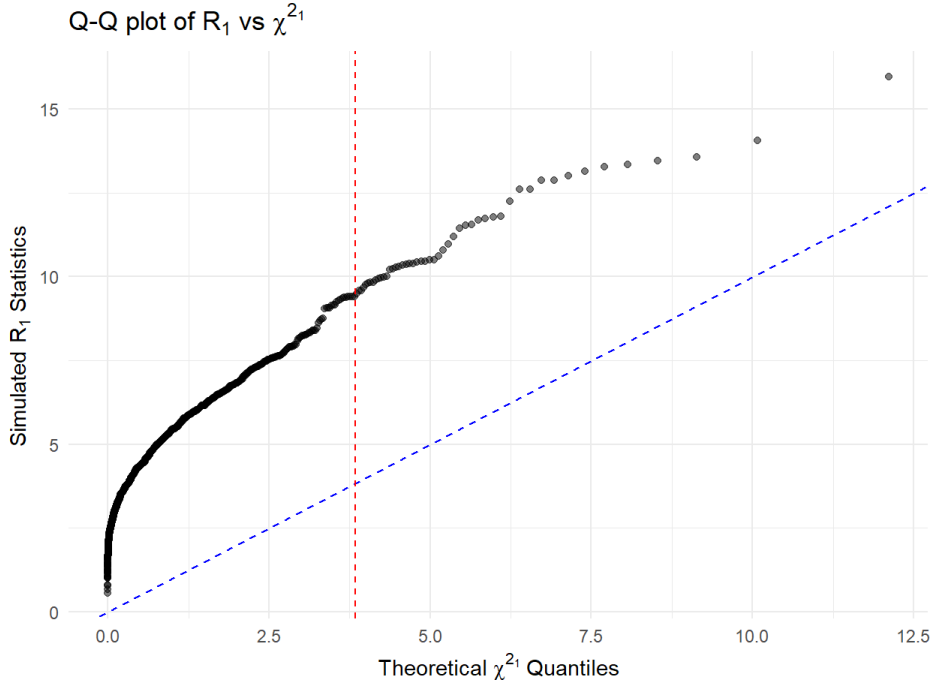


Figure: A simulation example with $N = 100$ observations and $p = 10$ orthogonal predictors and $\beta = 0$. A quantile-quantile plot, constructed over 1000 simulations, of the standard chi-squared statistic R_1, measuring the drop in residual sum-of-squares for the first predictor to enter in forward stepwise regression, versus the χ^2_1 distribution. The dashed vertical line marks the 95% quantile of the χ^2_1 distribution.

Using LAR algorithm to construct tests

Surprisingly, it turns out that for the lasso, a simple test can be derived that properly accounts for the adaptivity:

- Denote the knots returned by the LAR algorithm by $\lambda_1 > \lambda_2 \cdots > \lambda_k$. These are the values of regularization parameter λ where there is a change in the set of active predictors.
- Suppose that we wish to test significance of the predictor entered by LAR at λ_k.
- Let $\mathcal{A}_{k-1}$ be the active set (the predictors with nonzero coefficients) before this predictor was added;
- Let the estimate at the end of this step be $\hat{\beta}(\lambda_{k+1})$
- We refit the lasso, keeping $\lambda = \lambda_{k+1}$ but using just the variables in $\mathcal{A}_{k-1}$. This yields the estimate $\hat{\beta}_{\mathcal{A}_{k-1}}(\lambda_{k+1})$.

The covariance test for lasso

The covariance test statistic is defined by:

$$T_k = \frac{1}{\sigma^2}\left(\left\langle y, \mathbf{X}\hat{\beta}\left(\lambda_{k+1}\right)\right\rangle - \left\langle y, \mathbf{X}\hat{\beta}_{\mathcal{A}_{k-1}}\left(\lambda_{k+1}\right)\right\rangle\right).$$

Remark

- The statistic measures how much of the covariance between the outcome and the fitted model can be attributed to the predictor that has just entered the model.
- Interestingly, for forward-stepwise regression, the corresponding covariance statistic is equal to R_k, however, for the lasso this is not the case.

The covariance test for lasso

The covariance test statistic is defined by:

$$T_k = \frac{1}{\sigma^2}\left(\left\langle y, \mathbf{X}\hat{\beta}\left(\lambda_{k+1}\right)\right\rangle - \left\langle y, \mathbf{X}\hat{\beta}_{\mathcal{A}_{k-1}}\left(\lambda_{k+1}\right)\right\rangle\right).$$

Remark

- Remarkably, under the null hypothesis that all $k-1$ signal variables are in the model, and under general conditions on the model matrix X, for the predictor entered at the next step we have:

$$T_k \xrightarrow{d} \mathrm{Exp}(1), \text{ as } \mathrm{N}, \mathrm{p} \longrightarrow \infty$$

- When σ^2 is unknown, we estimate it using the full model: $\hat{\sigma}^2 = \frac{1}{N-p} RSS_p$. We plug this into T_k, and the exponential test becomes an $F_{2,N-p}$ test.

Simulation study

- Let $\sigma^2 = 1$ for simplicity.
- $\lambda_1 = \max_j \left|X_j^T y\right|, \quad \lambda_2 =$ second-large $\left(\left|X_j^T y\right|\right),$
- $$T_1 = \left(\left\langle y, X\hat{\beta}\left(\lambda_2\right)\right\rangle - \left\langle y, X\hat{\beta}_{\mathcal{A}_0}\left(\lambda_2\right)\right\rangle\right)$$
- $\hat{\beta}_{\mathcal{A}_0}\left(\lambda_2\right)$ is the solution under the null hypothesis that the no variable is in the model $\rightsquigarrow \left\langle y, X\hat{\beta}_{\mathcal{A}_0}\left(\lambda_2\right)\right\rangle = 0.$
- $\left\langle y, X\hat{\beta}\left(\lambda_2\right)\right\rangle = \left\langle y, X_1\hat{\beta}_1\left(\lambda_2\right)\right\rangle$, note that $\hat{\beta}_j(\lambda) = \operatorname{sign}\left(X_j^T y\right)\left(\left|X_j^T y\right| - \lambda\right)_+$, we have

  $$\hat{\beta}_1\left(\lambda_2\right) = \operatorname{sign}\left(X_1^T y\right)\left(\lambda_1 - \lambda_2\right),$$

  with other coefficients being zero.
- T_1 then becomes $T_1 = \lambda_1\left(\lambda_1 - \lambda_2\right)$.

Simulation study: qqplot

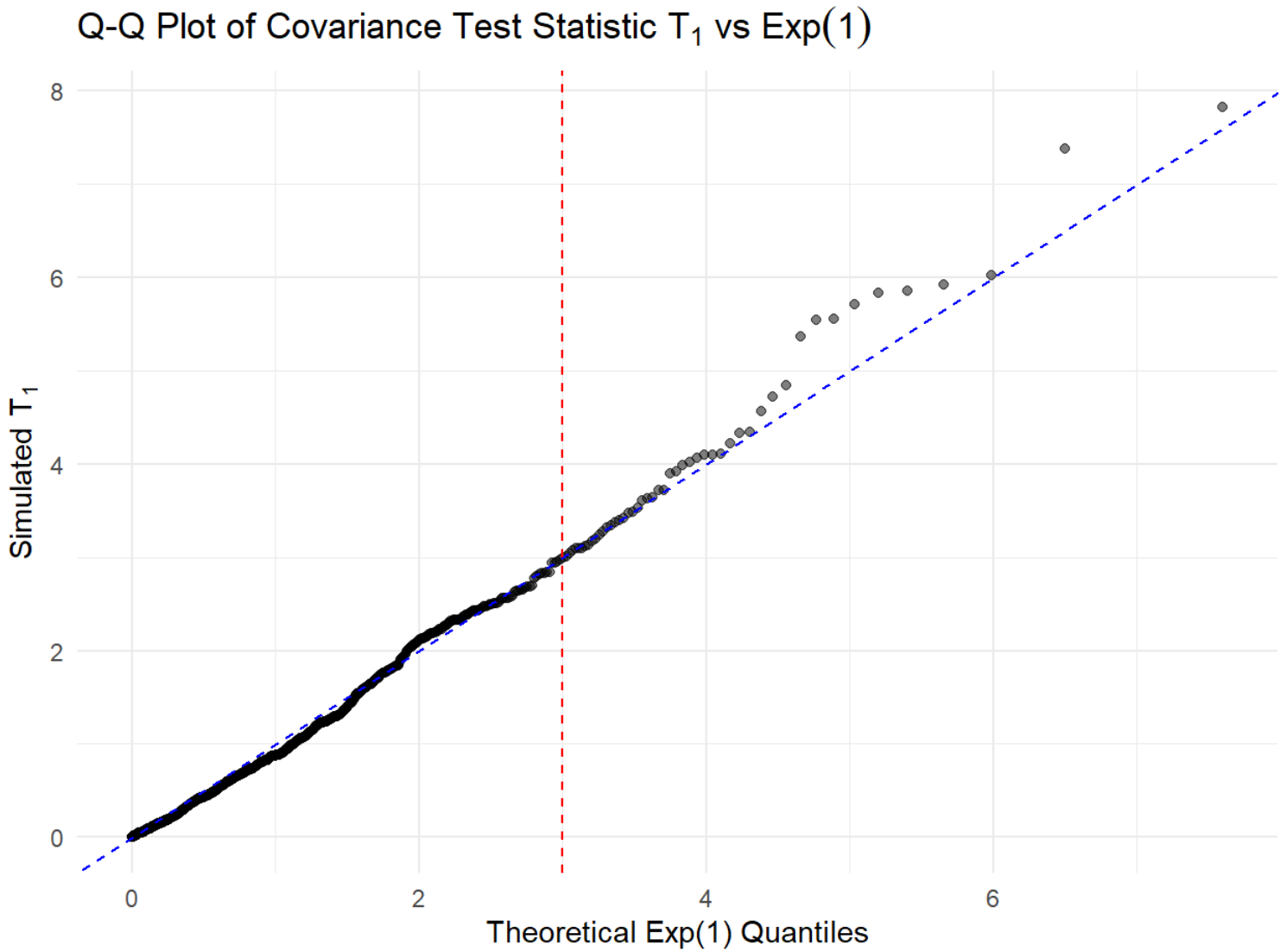
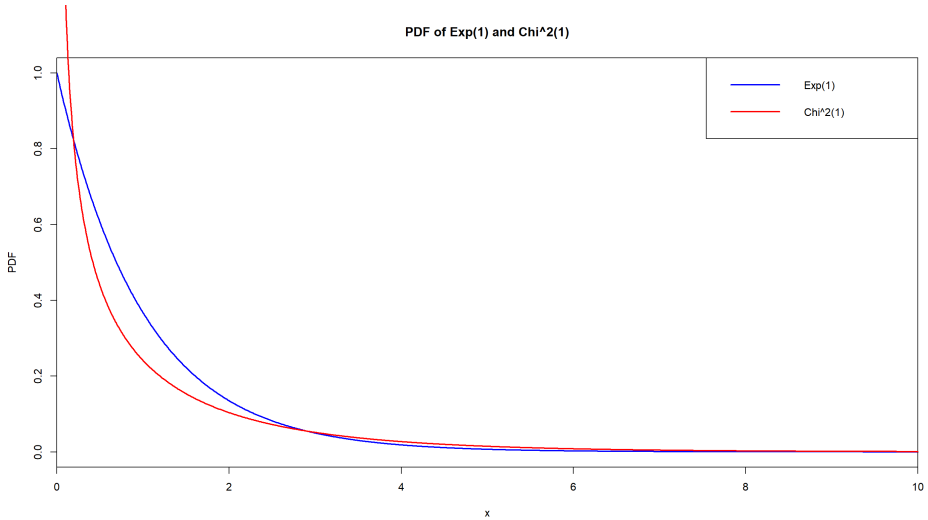


Figure: Quantile-quantile plot for T_1 versus $\mathrm{Exp}(1)$.

The covariance test for lasso



- Why is the mean of the forward-stepwise statistic R_1 much larger than one, while the mean of T_1 is approximately equal to one?
- The reason is shrinkage: the lasso picks the best predictor available at each stage, but does not fit it fully by least squares. It uses shrunken estimates of the coefficients, and this shrinkage compensates exactly for the inflation due to the selection.

The covariance test for lasso

Table 6.2 *Results of forward stepwise regression and LAR/lasso applied to the diabetes data introduced in Chapter 2. Only the first ten steps are shown in each case. The p-values are based on (6.4), (6.5), and (6.11), respectively. Values marked as 0 are* < 0.01.

	Forward Stepwise		LAR/lasso		
Step	Term	p-value	Term	p-value	
				Covariance	Spacing
1	bmi	0	bmi	0	0
2	ltg	0	ltg	0	0
3	map	0	map	0	0.01
4	age:sex	0	hdl	0.02	0.02
5	bmi:map	0	bmi:map	0.27	0.26
6	hdl	0	age:sex	0.72	0.67
7	sex	0	glu^2	0.48	0.13
8	glu^2	0.02	bmi^2	0.97	0.86
9	age^2	0.11	age:map	0.88	0.27
10	tc:tch	0.21	age:glu	0.95	0.44

Remark

The forward-stepwise regression enters 8 terms at level 0.05 , while the covariance test enters only 4.

General scheme for homogeneous setups

We now propose the general scheme for post-selection inference yields exact p-values and confidence intervals in the Gaussian case.

- Claim: Selection events such as LAR, forward-stepwise regression, and the Lasso for fixed λ can be written as $\{\boldsymbol{A}y \leq b\}$ for some matrix $\boldsymbol{A}$ and vector b.

Now suppose that $y \sim \boldsymbol{N}\left(\mu, \sigma^2 \boldsymbol{I}\right)$, and that we want to make inferences conditional on the event $\{\boldsymbol{A}y \leq b\}$. In particular, we wish to make inferences about $\eta^T \mu$, where η might depend on the selection event.

An OLS example: $\theta = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \mu$ is the regression coefficient vector upon which we want to make inference.

Polyhedral lemma (Lee et al. 2013, Taylor et al. 2014)

The previous selection event can be expressed as

$$\{\boldsymbol{A}y \leq b\} = \left\{\boldsymbol{V}^-(y) \leq \eta^T y \leq V^+(y), V^0(y) \geq 0\right\}$$

where, denoting $\alpha = \boldsymbol{A}\eta/\|\eta\|_2^2$,

$$V^-(y) = \max_{j:\alpha_j<0} \frac{b_j - (\boldsymbol{A}y)_j + \alpha_j \eta^T y}{\alpha_j},$$

$$V^+(y) = \min_{j:\alpha_j>0} \frac{b_j - (\boldsymbol{A}y)_j + \alpha_j \eta^T y}{\alpha_j},$$

$$V^0(y) = \min_{j:\alpha_j=0} \left(b_j - (\boldsymbol{A}y)_j\right).$$

Furthermore, $\eta^\top y$ and $\left(V^-(y), V^+(y), V^0(y)\right)$ are statistically independent.

Intuitive graphical interpretation

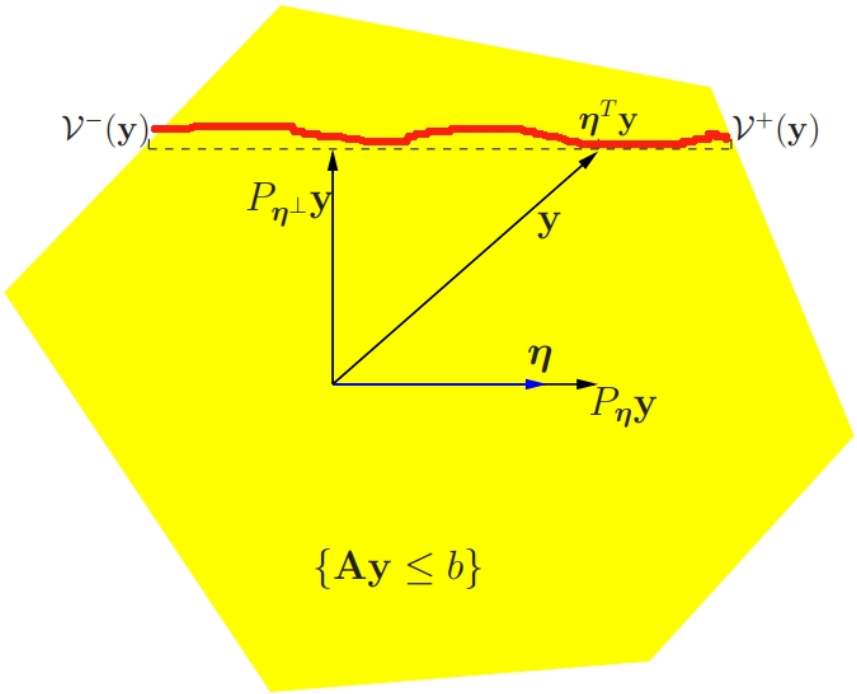


Figure 6.9 *Schematic illustrating the polyhedral lemma (6.7), for the case* $N = 2$ *and* $\|\boldsymbol{\eta}\|_2 = 1$. *The yellow region is the selection event* $\{\mathbf{A}\mathbf{y} \leq b\}$. *We decompose* $\mathbf{y}$ *as the sum of two terms: its projection* $P_{\boldsymbol{\eta}}\mathbf{y}$ *onto* $\boldsymbol{\eta}$ *(with coordinate* $\boldsymbol{\eta}^T\mathbf{y}$*) and its projection onto the* $(N-1)$*-dimensional subspace orthogonal to* $\boldsymbol{\eta}$: $\mathbf{y} = P_{\boldsymbol{\eta}}\mathbf{y} + P_{\boldsymbol{\eta}^\perp}\mathbf{y}$. *Conditioning on* $P_{\boldsymbol{\eta}^\perp}\mathbf{y}$, *we see that the event* $\{\mathbf{A}\mathbf{y} \leq b\}$ *is equivalent to the event* $\{\mathcal{V}^-(\mathbf{y}) \leq \boldsymbol{\eta}^T\mathbf{y} \leq \mathcal{V}^+(\mathbf{y})\}$. *Furthermore* $\mathcal{V}^+(\mathbf{y})$ *and* $\mathcal{V}^-(\mathbf{y})$ *are independent of* $\boldsymbol{\eta}^T\mathbf{y}$ *since they are functions of* $P_{\boldsymbol{\eta}^\perp}\mathbf{y}$ *only, which is independent of* $\mathbf{y}$.

Finding the pivot

Since y is Gaussian, the above lemma suggests that the conditional inference on $\eta^T \mu$ can be made using the truncated distribution of $\eta^T y$, which is a truncated normal distribution. Concretely, denote

$$F^{c,d}_{\mu,\sigma^2}(x) = \frac{\mathbf{\Phi}((x-\mu)/\sigma) - \mathbf{\Phi}((c-\mu)/\sigma)}{\mathbf{\Phi}((d-\mu)/\sigma) - \mathbf{\Phi}((c-\mu)/\sigma)}$$

where $\mathbf{\Phi}$ is the CDF of the standard Gaussian. This is just how the truncated normal distribution with support on $[c, d]$ is defined. It follows that

$$F^{V^-,V^+}_{\eta^\top \mu,\sigma^2\|\eta\|_2^2}\left(\eta^\top y\right) \mid \{\boldsymbol{A}y \leq b\} \sim \boldsymbol{U}(0,1).$$

Spacing test

We now apply the above inference procedure to successive steps of the LAR algorithm. For testing the global null hypothesis, we set $\eta^T y = \lambda_1 = \max_j |\langle x_j, y \rangle|$. We observe that $V^- = \lambda_2$, $V^+ = +\infty$, and hence

$$R_1 = 1 - F_{0,\sigma^2}^{\lambda_2,+\infty}(\lambda_1) \,\Big|\, \{\boldsymbol{A}y \leq b\} = \frac{1 - \boldsymbol{\Phi}(\lambda_1/\sigma)}{1 - \boldsymbol{\Phi}(\lambda_2/\sigma)} \sim \boldsymbol{U}(0,1)$$

This uniform distribution above holds exactly for finite N and p, and for any $\boldsymbol{X}$; and that it is a nonasymptotic version of the covariance test, and is asymptotically equivalent to it (Taylor et al. 2014).

What hypothesis do we test?

- Covariance Test (complete null hypothesis): At each stage of LAR, we are testing whether the coefficients of all other predictors not yet in the model are zero.
- Spacing test and Fixed λ test (incremental null hypothesis): At the first step, it tests the global null hypothesis, as does the covariance test. But at subsequent steps, it tests whether the partial correlation of the given predictor entered at that step is zero, adjusting for other variables that are currently in the model.

Debiased lasso

Aim

Directly estimates confidence intervals for the full set of population regression parameters under an assumed linear model, instead of attempting to make inferences about the partial regression coefficients in models derived by LAR or other lasso estimates.

Method

Use debias operation to "invert" the KKT conditions.

Useful reference

Van de Geer, et al. (2014) "On asymptotically optimal confidence regions and tests for high-dimensional models". AOS.

Debiased lasso setup

Consider as before the high dimensional linear model with Gaussian error $\epsilon \sim \boldsymbol{N}(0, \sigma_\epsilon \boldsymbol{I})$:

$$Y = \boldsymbol{X}\beta^0 + \epsilon$$

Define the lasso as:

$$\hat{\beta} = \hat{\beta}(\lambda) := \underset{\beta \in R^p}{\arg\min} \left(\| Y - X\beta \|_2^2 / n + 2\lambda \|\beta\|_1 \right).$$

By the KKT Theorem,

$$\lambda \cdot \hat{\kappa} = \boldsymbol{X}^T (\boldsymbol{Y} - \boldsymbol{X}\hat{\beta})/n$$

where $\|\hat{\kappa}\|_\infty \leq 1$ and $\hat{\kappa}_j = \text{sign}\left(\hat{\beta}_j\right)$ if $\hat{\beta}_j \neq 0$.

Inverting the KKT conditions

Denote $\hat{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X}/n$, thus

$$\hat{\boldsymbol{\Sigma}} \left(\hat{\beta} - \beta^0 \right) + \lambda \hat{\kappa} = \boldsymbol{X}^T \epsilon / n$$

We estimate the population quantity $\hat{\boldsymbol{\Theta}} := \hat{\boldsymbol{\Sigma}}^{-1}$ by nodewise regression on $\boldsymbol{X}$.
For each $j = 1, \ldots, p$, define

$$\hat{\gamma}_j := \underset{\gamma \in \mathbb{R}^{p-1}}{\arg\min} \left(\|X_j - \boldsymbol{X}_{-j}\gamma\|_2^2 / n + 2\lambda_j \|\gamma\|_1 \right).$$

where X_j is the j th column of the design matrix, and $\boldsymbol{X}_{-j}$ denotes the design matrix without the j th column.

Inverting the KKT conditions

Slightly abusing the notation, we see components of

$$\hat{\gamma}_j = \left\{ \hat{\gamma}_{j,k}; k = 1, \ldots, p, k \neq j \right\}.$$

Denote a $p \times p$ matrix $\hat{C}$ as:

$$\hat{\boldsymbol{C}} = \begin{bmatrix} 1 & -\hat{\gamma}_{1,2} & \ldots & -\hat{\gamma}_{1,p} \\ -\hat{\gamma}_{2,1} & 1 & \ldots & -\hat{\gamma}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{\gamma}_{p,1} & -\hat{\gamma}_{p,2} & \ldots & 1 \end{bmatrix}$$

Inverting the KKT conditions

We then define for all $j = 1, \ldots, p$,

$$\hat{\tau}_j^2 := \|X_j - \boldsymbol{X}_{-j}\hat{\gamma}_j\|_2^2 / n + \lambda_j \|\hat{\gamma}_j\|_1 .$$

We further define $\hat{\boldsymbol{T}}^2 := \operatorname{diag}\left(\hat{\tau}_1^2, \ldots, \hat{\tau}_p^2\right)$. Now we are finally ready to define an approximation of $\hat{\boldsymbol{\Sigma}}^{-1}$:

$$\hat{\boldsymbol{\Theta}}_{\text{Lasso}} := \hat{\boldsymbol{T}}^{-2}\hat{\boldsymbol{C}}.$$

We want to use the estimate to construct asymptotic pivots of the Lasso.

Debiasing the lasso

By previous arguments:

$$\hat{\boldsymbol{\Sigma}}\left(\hat{\beta}-\beta^0\right)+\lambda\hat{\kappa}=\boldsymbol{X}^T\epsilon/n$$

Simple calculation yields

$$\hat{\beta}-\beta^0+\hat{\boldsymbol{\Theta}}_{\text{Lasso}}\,\lambda\hat{\kappa}=\hat{\boldsymbol{\Theta}}_{\text{Lasso}}\,\boldsymbol{X}^T\epsilon/n-\boldsymbol{\Delta}/\sqrt{n}$$

where

$$\boldsymbol{\Delta}:=\sqrt{n}\left(\hat{\boldsymbol{\Theta}}_{\text{Lasso}}\,\hat{\boldsymbol{\Sigma}}-\boldsymbol{I}\right)\left(\hat{\beta}-\beta^0\right).$$

Define estimator

$$\hat{b}_{\text{Lasso}}\;:=\hat{\beta}+\hat{\boldsymbol{\Theta}}_{\text{Lasso}}\,\lambda\hat{\kappa}=\hat{\beta}+\hat{\boldsymbol{\Theta}}_{\text{Lasso}}\,\boldsymbol{X}^T(Y-\boldsymbol{X}\hat{\beta})/n.$$

Assumptions and notations

- We assume for now that the rows of $\boldsymbol{X}$ are i.i.d. realizations from a Gaussian distribution whose p-dimensional inner product matrix $\boldsymbol{\Sigma}$ has strictly positive smallest eigenvalue $\Lambda^2_{\min}$ satisfying $1/\Lambda^2_{\min} = \mathcal{O}(1)$. Furthermore, $\max_j \boldsymbol{\Sigma}_{j,j} = \mathcal{O}(1)$.
- We will use the following notation to further assume sparsity w.r.t. rows of $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$:
- Define for all $j = 1, \ldots, p$

$$s_j := \left|\left\{k \neq j : \boldsymbol{\Theta}_{j,k} \neq 0\right\}\right|.$$

- Define the active set of variables $S_0 := \left\{j; \beta^0_j \neq 0\right\}$, and its cardinality $s_0 = |S_0|$.
- We finally denote $\hat{\boldsymbol{\Omega}} := \hat{\boldsymbol{\Theta}}_{\text{Lasso}}\, \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}^T_{\text{Lasso}}$.

Main results

Theorem 2.2 (Van de Geer et al. 2014)

Consider linear model as above with Gaussian error $\epsilon \sim \mathbf{N}\left(0, \sigma_\epsilon^{<} \boldsymbol{I}\right)$ where $\sigma_\epsilon^2 = \mathcal{O}(1)$. Assume the previous assumption holds, and $s_0 = o(\sqrt{n}/\log(p))$ and $\max_j s_j = o(n/\log(p))$. If $\lambda \asymp \sqrt{\log(p)/n}$ for the lasso, and $\lambda_j \asymp \sqrt{\log(p)/n}$ uniformly for the nodewise regression, then

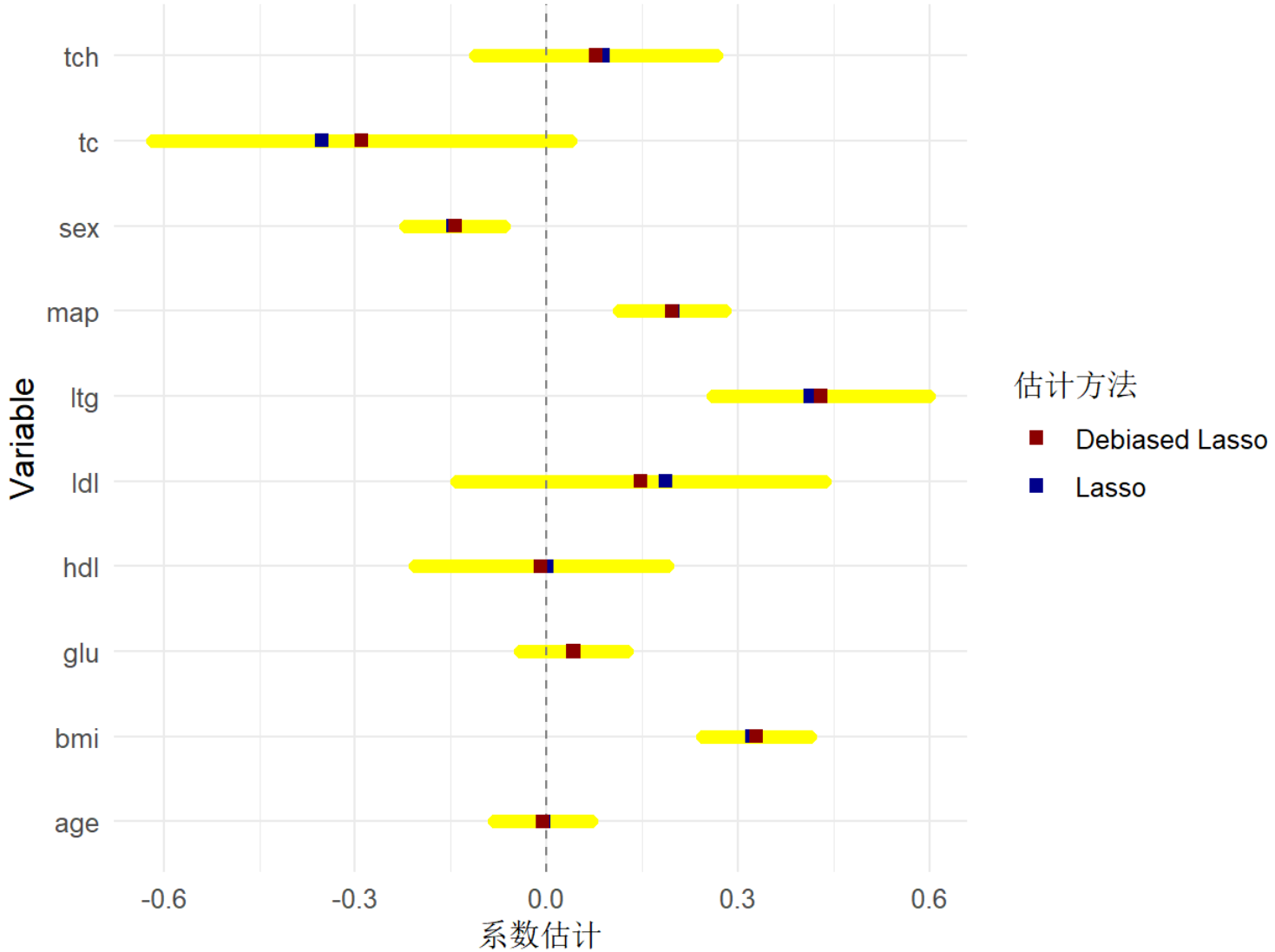
$$\sqrt{n}\left(\hat{b}_{\text{lasso}} - \beta^0\right) = \boldsymbol{W} + \boldsymbol{\Delta}$$

$$\boldsymbol{W} \mid \boldsymbol{X} \sim \boldsymbol{N}\left(0, \sigma_\epsilon^2 \hat{\boldsymbol{\Omega}}\right)$$

$$\|\boldsymbol{\Delta}\|_\infty = o_{\mathbb{P}}(1)$$

$$\left\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Sigma}^{-1}\right\|_\infty = o_{\mathbb{P}}(1)$$

Case study: debiased lasso on diabetes data



PoSI

Denote the index set $M = \{j_1, j_2, \ldots, j_m\} \subset M_F = \{1, \ldots, p\}$, and let $\boldsymbol{X}_M = \{X_{j_1}, \ldots, X_{j_m})$ denote the $n \times m$ submatrix of $\boldsymbol{X}$ indexed by M. We assume that $\boldsymbol{X}_M$ is of full rank. Let $\hat{\beta}_M = \left(\boldsymbol{X}_M^T \boldsymbol{X}_M\right)^{-1} \boldsymbol{X}_M^T Y$ be the unique least square estimate in M , which is an estimator of

$$\beta_M := \mathbb{E}\left(\hat{\beta}_M\right) = \underset{\beta' \in R^m}{\arg\min} \left\|\mu - \boldsymbol{X}_M \beta'\right\|^2.$$

Respectively, we denote multiple regression coefficients for all $j \in M, \hat{\beta}_{j \cdot M} = X_{j \cdot M}^T Y / \left\|X_{j \cdot M}\right\|^2$. We further assume the availability of a valid estimate $\hat{\sigma}^2$ of σ^2 which is independent of all estimates $\hat{\beta}_{j \cdot M}$, and $\hat{\sigma}^2 \sim \sigma^2 \chi_r^2 / r$ for r degrees of freedom.

PoSI

We denote the t-ratio for $\beta_{j\cdot M}$ that uses $\hat{\sigma}^2$ irrespective of M :

$$t_{j\cdot M} := \frac{\hat{\beta}_{j\cdot M} - \beta_{j\cdot M}}{\left(\left(\boldsymbol{X}_M^T \boldsymbol{X}_M\right)^{-1}\right)_{jj}^{1/2} \hat{\sigma}} = \frac{\hat{\beta}_{j\cdot M} - \beta_{j\cdot M}}{\hat{\sigma} \left\| X_{j\cdot M} \right\|} = \frac{(Y-\mu)^T X_{j\cdot M}}{\hat{\sigma} \left\| X_{j\cdot M} \right\|}$$

which has a central t-distribution with r degrees of freedom.

Remark

With such choice of $\hat{\sigma}^2$, the confidence intervals for $\beta_{j\cdot M}$ take the form

$$CI_{j\cdot M}(K) := \left[\hat{\beta}_{j\cdot M} \pm K \left(\left(\boldsymbol{X}_M^T \boldsymbol{X}_M\right)^{-1}\right)_{jj}^{1/2} \hat{\sigma}\right] = \left[\hat{\beta}_{j\cdot M} \pm K\hat{\sigma} / \left\| X_{j\cdot M} \right\|\right]$$

where $K = t_{r,1-\alpha/2}$ to be the $1-\alpha/2$ quantile of the t-distribution of r degrees of freedom, we have marginal $1-\alpha$ coverage guarantee

$$\mathbb{P}\left(\beta_{j\cdot M} \in CI_{j\cdot M}(K)\right) > 1-\alpha$$

References

- ISLR, SLS and ESL,
- 文再文等, 最优化课件,
- Ryan Tibshirani, Convex Optimization slides.
- B& V Convex optimization.
- Course slides for sparse learning in ETH Zurich.
- LLM assisted coding.