SL : RIDGE REGRESSION & LASSO

Least Squares approach to linear regression has several drawbacks

→ With many predictors, the LS solution often exhibits high variance

→ In high dimensional settings ($d \gg n$ – typical in genomics / micro anay studies ; in document classification etc) the LS solution is not unique (matrix X is at most of rank $n < d$, and $X^t X$ is not invertible)

Regularization techniques such as Ridge Regression and the Lasso address these drawbacks, by imposing a penalty on the size of the coefficients.

The Ridge coefficients minimize a penalized sum of squares given by

$$RSS_2(\lambda) := \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij}\right)^2 + \lambda \underbrace{\left(\sum_{j=1}^{d} \beta_j^2\right)}_{L_2\text{-penalty}}$$

TUNING PARAMETER

β_0 = Intercept is not penalized since it is just a measure of average location of the response variable when all predictors are set to 0.

- $\lambda = 0 \Rightarrow$ No penalty $\Rightarrow$ LS solution
- As $\lambda \uparrow \infty \Rightarrow$ Penalize heavily non zero coefficients $\Rightarrow$ All coefficients forced to zero.

$\Rightarrow$ Trade-off between the goodness of fit term and the penalty.

The objective is thus to minimize $\|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$ (2)

$\|\beta\|_2^2$: L_2-norm square of β

where

$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{pmatrix}$ = matrix of observations.

$n \times (d+1)$

(why ?) → standardise : compute for each column j the sample variance $\hat{\sigma}_j^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2$, where $\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$.

Denote by $x_{ij}^{\sigma} := \frac{x_{ij}}{\hat{\sigma}_j}$ the standardized entries.

$$\hat{\beta}^{\sigma} = \underset{\beta^{\sigma}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0^{\sigma} - \sum_{j=1}^{d} x_{ij}^{\sigma} \beta_j^{\sigma}\right)^2 + \lambda \sum_{j=1}^{d} \left(\beta_j^{\sigma}\right)^2 \right\}$$

Introduce $\bar{x}_j^{\sigma}$ = mean of the $x_{ij}^{\sigma} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}^{\sigma}$.

$$= \underset{\beta^{\sigma}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0^{\sigma} - \sum_{j=1}^{d} (x_{ij}^{\sigma} - \bar{x}_j^{\sigma}) \beta_j^{\sigma} - \sum_{j=1}^{d} \bar{x}_j^{\sigma} \beta_j^{\sigma}\right)^2 + \lambda \sum_{j=1}^{d} \left(\beta_j^{\sigma}\right)^2 \right\}$$

$$\hat{\beta}^{s} = \underset{\beta^{s}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0^{s} - \sum_{j=1}^{d} (x_{ij}^{\sigma} - \bar{x}_j^{\sigma}) \beta_j^{s}\right)^2 + \lambda \sum_{j=1}^{d} \left(\beta_j^{s}\right)^2 \right\}$$

where

$$\begin{cases} \beta_0^{s} := \beta_0^{\sigma} - \sum_{j=1}^{d} \bar{x}_j^{\sigma} \beta_j^{\sigma} \\ \beta_1^{s} := \beta_1^{\sigma} \\ \vdots \\ \beta_d^{s} := \beta_d^{\sigma} \end{cases}$$

 # Take-Away Message

Centering inputs does not change coefficients $\beta_1, \dots, \beta_d$.
It only shifts the intercept.

→ Characterization of the solution $\hat{\beta}^s$:

The square error term is:

$$\sum_{i=1}^{n} \left(y_i - \beta_0^s - \sum_{j=1}^{d} (x_{ij}^\sigma - \bar{x}_j^\sigma)\beta_j^s \right)^2$$

$$= \sum_{i=1}^{n} (y_i - \beta_0^s)^2 - 2 \boxed{\sum_{i=1}^{n} (y_i - \beta_0^s) \sum_{j=1}^{d} (x_{ij}^\sigma - \bar{x}_j^\sigma)\beta_j^s} + \sum_{i=1}^{n} \left(\sum_{j=1}^{d} (x_{ij}^\sigma - \bar{x}_j^\sigma)\beta_j^s \right)^2$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{d} y_i (x_{ij}^\sigma - \bar{x}_j^\sigma)\beta_j^s - \beta_0^s \sum_{j=1}^{d} \underbrace{\left(\sum_{i=1}^{n} (x_{ij}^\sigma - \bar{x}_j^\sigma) \right)}_{=0} \beta_j^s$$

since we have centered inputs.

$$= \sum_{i=1}^{n} (y_i - \beta_0^s) + \text{something independent of } \beta_0^s.$$

Minimization with respect to β_0^s occurs at $\hat{\beta}_0^s = \bar{y} = \frac{1}{n}\sum y_i$

 # Take Away Message

After standardizing the input variables, the intercept is estimated as $\bar{y}$. The remaining coefficient estimates are then estimated from a ridge regression without intercept, using standardized inputs.

In summary, it is important to perform a ridge regression according to the following guidelines.

(i) Standardize input variables:

$$X^s = \frac{X - \bar{X}}{\hat{\sigma}}, \qquad \bar{X} = (\bar{X}_1, \dots, \bar{X}_d), \quad \hat{\sigma} = (\hat{\sigma}_1, \dots, \hat{\sigma}_d)$$

$$\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \qquad \hat{\sigma}_j^2 = \frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \bar{X}_j)^2$$

$(n \times d)$ — do not include intercept !!

↳ has entries x_{ij}^s

$$X^s = \begin{pmatrix} | & & | \\ x_1^s & \cdots & x_d^s \\ | & & | \end{pmatrix}, \qquad x_j^s = \frac{x_j - \bar{x}_j}{\hat{\sigma}_j}$$

$$X = \begin{pmatrix} | & | & | \\ x_1 & x_j & x_d \\ | & | & | \end{pmatrix}$$

mean $\bar{x}_j$, std deviation $\hat{\sigma}_j$.

(ii) Put $\hat{\beta}_0^s = \bar{y}$, and consider $y^s = y - \bar{y}$

(iii) Solve (see page 5)

$$(\hat{\beta}_1^s, \dots, \hat{\beta}_d^s) = \underset{\beta_1^s, \dots, \beta_d^s}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i^s - \sum_{j=1}^{d} x_{ij}^s \beta_j^s \right)^2 + \lambda \sum_{j=1}^{d} (\beta_j^s)^2 \right\}$$

$$= \underset{\beta^s}{\operatorname{argmin}} \left\{ \| y^s - X^s \beta^s \|^2 + \lambda \| \beta^s \|^2 \right\}$$

(iv) Express coefficients back on the original scale:

$$\hat{y} = \hat{\beta}_0^s + \hat{\beta}_1^s \left(\frac{x_1 - \bar{x}_1}{\hat{\sigma}_1} \right) + \cdots + \hat{\beta}_d^s \left(\frac{x_d - \bar{x}_d}{\hat{\sigma}_d} \right)$$

($\frac{x_1 - \bar{x}_1}{\hat{\sigma}_1} = x_1^s$, $\frac{x_d - \bar{x}_d}{\hat{\sigma}_d} = x_d^s$)

$\hat{\beta}_j = \frac{\hat{\beta}_j^s}{\hat{\sigma}_j}$

$$= \hat{\beta}_0^s + \hat{\beta}_1 (x_1 - \bar{x}_1) + \cdots + \hat{\beta}_d (x_d - \bar{x}_d)$$

$\hat{\beta}_0 = \hat{\beta}_0^s - \sum_{j=1}^{d} \hat{\beta}_j \bar{x}_j$

$$= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_d x_d$$

■

→ The Ridge Regression (RR) in matrix form is

$$RSS_2(\lambda) = \underset{(n\times 1)}{(y - X\beta)^t} \underset{(n\times d)}{(y - X\beta)} \underset{(d\times 1)}{+ \lambda \beta^t \beta}$$

[We drop the notation X^s, but according to the previous discussion, the matrix of observations X is standardized: columns have 0 mean and unit variance]

$$= y^t y - y^t X\beta - \beta^t X^t y + \beta^t X^t X \beta + \lambda \beta^t \beta$$

$$= y^t y - 2 y^t X \beta + \beta^t X^t X \beta + \lambda \beta^t \beta$$

↔ cf page 4, chapter on Linear Regression

$$\Rightarrow \frac{\partial RSS_2(\lambda)}{\partial \beta} = -2 y^t X + 2 \hat{\beta}^t X^t X + 2\lambda \hat{\beta}^t = 0$$

$$\Rightarrow \hat{\beta}^t (X^t X + \lambda I) = y^t X$$

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

(symmetrical)

$$\Rightarrow \boxed{\hat{\beta}_\lambda = (\lambda I_d + X^t X)^{-1} X^t y} \quad \text{——— } (*)$$

$(d\times 1)$ $(d\times n)(n\times d)$ $(d\times n)(n\times 1)$

Remarks: (i) Centering y as suggested on page 4 is infact not needed:
since X has centered columns, $X^t y = X^t y^s$, (convince yourself why)
so that $\hat{\beta}_\lambda$ remains unchanged, whether y or y^s is used in $(*)$.

(ii) $(X^t X + \lambda I_d)$ is invertible, provided $\lambda > 0$.

Indeed, $\forall z \in \mathbb{R}^d$, $z \neq 0$,

$$z^t (X^t X + \lambda I_d) z = z^t X^t X z + \lambda z^t z$$

$$= \underbrace{\|Xz\|_2^2}_{\geq 0} + \underbrace{\lambda \|z\|_2^2}_{>0} > 0$$

⇒ Even if X is not full rank, $X^t X + \lambda I_d$ is positive definite.

↳ Very useful in high dimensional settings, for which $d > n$. In this case, $X^t X$ can be at most of rank $n < d$, and thus non invertible (→ Least Square estimate is undefined).
However, $X^t X + \lambda I_d$ is always invertible, and the ridge solution always exist.

(iii) $\frac{\partial^2 RSS_2(\lambda)}{\partial \beta^2} = 2(X^t X + \lambda I_d) \geq 0$

⇒ $\hat{\beta}_\lambda$ is indeed a minimum.

II- ANALYSIS OF THE SOLUTION

↘ Step I: Some pictures might help.

An equivalent way to write the RR problem is

$$\beta^* = \underset{\beta}{\text{argmin}} \|y - X\beta\|_2^2$$
$$\text{subject to } \sum_{i=1}^d \beta_i^2 \leq ?$$

↖ A convex optimization problem

where $\eta \equiv$ budget = how much you are ready to spend on the parameters. (it should be less than the norm of the LS solution)

To understand the relation between η and λ, we solve the constrained optimization problem by first defining the Lagrangian

$$\mathcal{L}(\beta, \mu) = \underbrace{\|y - X\beta\|_2^2}_{RSS} + \mu\left(\sum_{j=1}^{d} \beta_j^2 - \eta\right)$$

(The constraint in 'standard' form is $\sum \beta_j^2 - \eta \leq 0$)

Slater's constraint qualifications (see chapter on SVM) ensures that Strong Duality holds. Since the optimization problem is convex, the optimal solution (β^*, μ^*), where β^* = primal optimal, μ^* = dual optimal, satisfies the Karush-Kuhn-Tucker (KKT) conditions:

① Primal constraint $\quad \sum_{i=1}^{d} \beta_i^2 - \eta \leq 0$

② Dual constraint $\quad \mu \geq 0$

③ Complementary Slackness $\quad \mu\left(\sum \beta_i^2 - \eta\right) = 0$

④ Gradient of the Lagrangian with respect to β vanishes:

$$\nabla_\beta \mathcal{L}(\beta, \mu) = 0$$

↳ The gradient of the Lagrangian is equal to the gradient of $RSS_2(\lambda) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$, but with μ in place of λ.

⇒ Suppose we solved the original problem and obtained $\hat{\beta}_\lambda = \underset{\beta}{\text{argmin}}\, RSS_2(\lambda) = (X^t X + \lambda I_d)^{-1} X^t y$.

Put $\eta = \|\hat{\beta}_\lambda\|_2^2$.

Then $\mu^* = \lambda$ and $\beta^* = \hat{\beta}_\lambda$ satisfy the KKT solutions

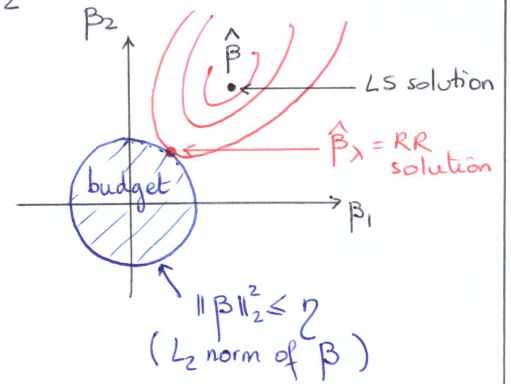
⇒ Both problems have the same solution.

(note that $\mu = 0$ is not solution since the fourth condition yields the LS solution, whose norm is strictly larger than η, violating the primal constraint).

Conversely, solving the constrained optimization problem yields β^* and μ^*, the primal and dual optimals. Setting $\lambda = \mu^*$ in the minimization of the RSS yields the same solution.

Conclusion: the problems are equivalent, with $\eta = \|\hat{\beta}_\lambda\|_2^2$

Picture: with $d = 2$

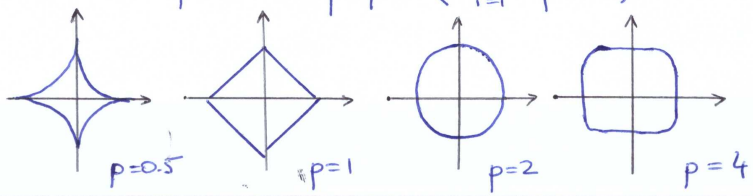


where:

contour lines: $RSS(\lambda)$ take a constant value.

⇒ Other norms may be used instead of the L_2 norm $\|\cdot\|_2$.

Consider the L_p norm $\|\beta\|_p = \left(\sum_{i=1}^{d} |\beta_i|^p\right)^{1/p}$



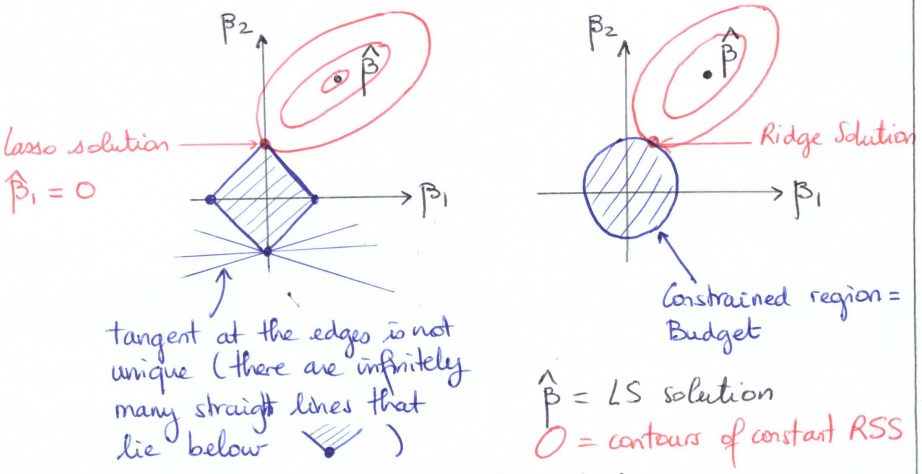
The LASSO uses an L_1 penalty: Tibshirani (1996)

$$RSS_1(\lambda) = \underbrace{\|y - X\beta\|_2^2}_{RSS} + \lambda \underbrace{\|\beta\|_1}_{=\sum_{i=1}^{d}|\beta_i|}$$

Although the lasso shrinks coefficients β to 0 like riddge regression, the L_1 penalty forces some of the coefficients to be exactly 0, when the tuning parameter is sufficiently large:

An equivalent way to write the lasso problem is

$$\min_{\beta} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \sum_{i=1}^{d}|\beta_i| \leq \eta$$



⇒ The lasso sets some coefficients to zero.

⇒ The lasso yields sparse models: it automatically performs variable selection.
(Selecting an appropriate value for λ is essential)
↳ Use Cross-validation.

Remark: Neither the lasso or ridge regression universally dominate the other.

Step II: Case $X^tX = Id$.

In the case of an orthonormal observation matrix X, the ridge regression and lasso problems have explicit solutions, and can easily be compared with the least square solution.

→ LS solution: $RSS = \|y - X\beta\|_2^2 \to \hat{\beta}^{ls} = (X^tX)^{-1}X^ty$, which reduces to $\hat{\beta}^{ls} = X^ty$. [assume we have no intercept]

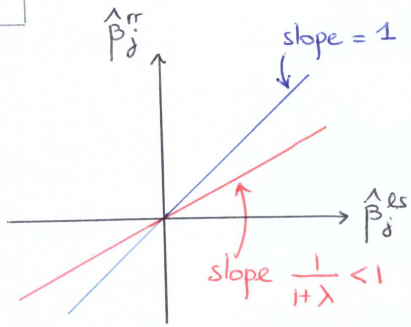
→ RR solution: $RSS_2(\lambda) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$

The solution is $\hat{\beta}^{rr} = (\lambda I_d + \underbrace{X^tX}_{=I_d})^{-1}\underbrace{X^ty}_{=\hat{\beta}^{ls}} = \frac{\hat{\beta}^{ls}}{1+\lambda}$

Thus $\hat{\beta}_j^{rr} = \frac{\hat{\beta}_j^{ls}}{1+\lambda}$

The j-th coefficient

"In RR, estimates are shrunk proportionally towards 0".



→ Lasso solution $RSS_1(\lambda) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$.

$$\min_{\beta}\left(y^ty - y^tX\beta - \beta^tX^ty + \beta^t\underbrace{X^tX}_{=I}\beta + \lambda\sum|\beta_j|\right)$$

$$\Leftrightarrow \min_{\beta}\left(-2\beta^t\hat{\beta}^{ls} + \beta^t\beta + \lambda\sum|\beta_j|\right)$$

$$\Leftrightarrow \min_{\beta}\left(\sum_{j=1}^{d} -2\beta_j\hat{\beta}_j^{ls} + \beta_j^2 + \lambda|\beta_j|\right)$$

↳ separable problem = variables β_j can be optimized individually.

$\Rightarrow$ For each $\beta_j = \beta$, we need to solve

$$\min_{\beta \in \mathbb{R}} \left(-2\beta\hat{\beta}^{ls} + \beta^2 + \lambda|\beta| \right) \quad \text{——— } (*)$$

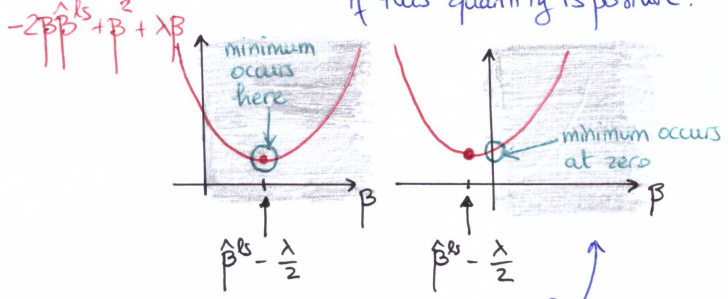
→ If $\hat{\beta}^{ls} > 0$, then necessarily $\hat{\beta}^{lasso}$:= value of β minimizing $(*)$ is ≥ 0 ; since otherwise the criterion could be further minimized by flipping its sign.
On $\{\hat{\beta}^{ls} > 0\}$, the problem is thus equivalent to

$$\min_{\beta \geq 0} \left(-2\beta\hat{\beta}^{ls} + \beta^2 + \lambda\beta \right).$$

$$\searrow \quad \frac{\partial(\dots)}{\partial\beta} = -2\hat{\beta}^{ls} + 2\hat{\beta}^{lasso} + \lambda = 0$$

$$\hat{\beta}^{lasso} = \hat{\beta}^{ls} - \frac{\lambda}{2}.$$

if this quantity is positive:



we are looking for a solution on $\beta \geq 0$.

Denoting $(x)_+ := \max(x, 0)$, provided $\hat{\beta}^{ls} > 0$, the lasso solution is $\hat{\beta}^{lasso} = \left(\hat{\beta}^{ls} - \frac{\lambda}{2}\right)_+$

$$= \text{sign}\left(\hat{\beta}^{ls}\right)\left(|\hat{\beta}^{ls}| - \frac{\lambda}{2}\right)_+$$

this expression will allows us to unify the solution on $\{\hat{\beta}^{ls} < 0\}$.

→ If $\hat{\beta}^{ls} < 0$, then $\hat{\beta}^{lasso} \leq 0$ necessarily.

We need to solve

$$\min_{\beta \leq 0} \left\{ -2\beta\hat{\beta}^{ls} + \beta^2 - \lambda\beta \right\}$$

$$\searrow \quad \frac{\partial(\dots)}{\partial\beta} = -2\hat{\beta}^{ls} + 2\hat{\beta}^{lasso} - \lambda = 0$$

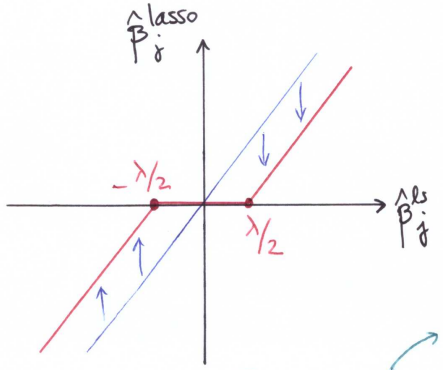
$$\hat{\beta}^{lasso} = \hat{\beta}^{ls} + \frac{\lambda}{2}$$

provided this quantity is negative. If $\hat{\beta}^{ls} + \frac{\lambda}{2} \geq 0$, the value of β minimizing the criterion is zero

Thus $\hat{\beta}^{lasso} = \left(\hat{\beta}^{ls} + \frac{\lambda}{2}\right) 1\left(\hat{\beta}^{ls} + \frac{\lambda}{2} < 0\right)$

$$= -\left(|\hat{\beta}^{ls}| - \frac{\lambda}{2}\right) 1\left(|\hat{\beta}^{ls}| - \frac{\lambda}{2} > 0\right)$$

$$= \text{sign}\left(\hat{\beta}^{ls}\right)\left(|\hat{\beta}^{ls}| - \frac{\lambda}{2}\right)_+$$



When the columns of X are orthogonal, the solution to the lasso problem is

$$\hat{\beta}_j^{lasso} = \text{sign}\left(\hat{\beta}_j^{ls}\right)\left(|\hat{\beta}_j^{ls}| - \frac{\lambda}{2}\right)_+$$

This expression will be useful when discussing the coordinate descent algorithm for the lasso.

In general, as we shall see now, the idea is the same:

↘ RR shrinks coefficients proportionally to zero.

↘ Lasso shrinks some coefficients exactly to zero.

Remark: Analogy with 'best subset selection'.

In best subset selection, for each $k \in \{0, 1, \dots, d\}$, we select the model of size k that gives the lowest RSS:

$$\begin{aligned} &\underset{\beta}{\text{minimize}} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij}\right)^2 \\ &\text{subject to} \quad \sum_{j=1}^{d} 1(\beta_j \neq 0) \leq k \end{aligned}$$

We are looking for a set of coefficient estimates minimizing the RSS, subject to having no more than k non-zero coefficients among $\{\beta_1, \dots, \beta_d\}$.

If X is such that $X^t X = I_d$, the problem reduces to

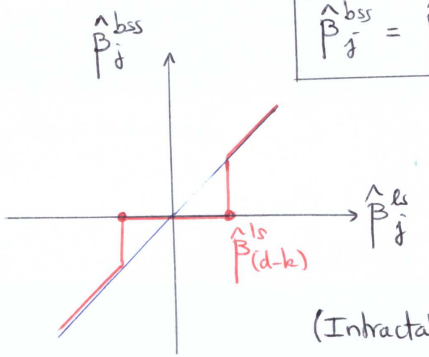
$$\begin{aligned} &\min_{\beta} \quad -2\beta^t \hat{\beta}^{ls} + \beta^t \beta \\ &\text{s.t.} \quad \text{at most } k \text{ coefficients} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\min_{\beta} \quad \sum_{j=1}^{d} -2\beta_j \hat{\beta}_j^{ls} + \beta_j^2 \\ &\text{s.t.} \quad \text{at most } k \text{ coef.} \end{aligned}$$

$-2\beta_j \hat{\beta}_j^{ls} + \beta_j^2$ is quadratic in β_j, and the minimizer is $\hat{\beta}_j^{ls}$. Value at minimum is $-(\hat{\beta}_j^{ls})^2$.

The solution to the minimization problem is $\hat{\beta}_j^{bss} = \hat{\beta}_j^{ls}$, but only for some indices: we only keep the k largest coefficient (in absolute value) to make the sum as small as possible.

$$\hat{\beta}_j^{bss} = \hat{\beta}_j^{ls} \, 1\left(|\hat{\beta}_j^{ls}| > |\hat{\beta}_{(d-k)}^{ls}|\right)$$

($\hat{\beta}_{(d-k)}^{ls}$: ordered coefficients)



"Best subset selection performs a hard-thresholding".

(Intractable if d is 'large' (say $d > 40$))

Step III a. General case (Ridge Regression)

First, recall the SVD decomposition of a $(n \times d)$ matrix X of rank $r \leq d$, (with $d < n$):

There exists an $(n \times r)$ orthonormal matrix U_r ($U_r^t U_r = I_r$)
$(d \times r)$ orthonormal matrix V_r ($V_r^t V_r = I_r$)
$(r \times r)$ diagonal matrix Λ_r, with entries $\lambda_i > 0$

such that $X = U \Lambda V^t$

Notation: $U_r = \underset{(n \times r)}{\begin{pmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{pmatrix}}$ $\quad$ $V_r = \underset{(d \times r)}{\begin{pmatrix} | & & | \\ v_1 & \cdots & v_r \\ | & & | \end{pmatrix}}$

u_1, u_r: $(n \times 1)$ $\quad$ v_1, v_r: $(d \times 1)$

SVD decomposition.

Remark: (we drop the subscript for convenience)

- $X^t X = V \Lambda \underbrace{U^t U}_{=I} \Lambda V^t = V \Lambda^2 V^t$

  So that $X^t X v_i = \lambda_i^2 v_i$

  $\Rightarrow$ $X^t X$ has eigenvalue-eigenvector pair (λ_i^2, v_i)

- $X X^t = U \Lambda \underbrace{V^t V}_{=I} \Lambda U^t = U \Lambda^2 U^t$

  So that $X X^t u_i = \lambda_i^2 u_i$

  $\Rightarrow$ $X X^t$ has eigenvalue-eigenvector pair (λ_i^2, u_i)

The ridge solution $\hat{\beta}_\lambda = (X^t X + \lambda I_d)^{-1} X^t y$ can be expressed in terms of the SVD matrices of X:

$$\hat{\beta}_\lambda = (V \Lambda^2 V^t + \lambda I_d)^{-1} V \Lambda U^t y$$

Assume that rank $X = d < n$.

Then V = square matrix and $V^t V = V V^t = I_d$, so that $V^{-1} = V^t$.

(we assume full rank for convenience; but it is not needed. If X is of rank $r < d$, then we can augment the $(d \times r)$ matrix V to make it square (and orthogonal), and associate with these columns zero singular values, so that Λ is square with entries $\lambda_i \geq 0$).

Then $\hat{\beta}_\lambda = (V \Lambda^2 V^t + \lambda V V^t)^{-1} V \Lambda U^t y$

$= \underbrace{(V^t)^{-1}}_{V} (\Lambda^2 + \lambda I_d)^{-1} \underbrace{V^{-1} V}_{I} \Lambda U^t y$

$$\boxed{\hat{\beta}_\lambda = V (\Lambda^2 + \lambda I_d)^{-1} \Lambda U^t y}$$

→ Remark: $\|\hat{\beta}_\lambda\|^2 = \hat{\beta}_\lambda^t \hat{\beta}_\lambda$

$$= \left(\sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i^2 + \lambda} v_i u_i^t y \right)^t \left(\sum_{j=1}^{d} \frac{\lambda_j}{\lambda_j^2 + \lambda} v_j u_j^t y \right)$$

$$= \sum_{i,j} \left(\frac{\lambda_i}{\lambda_i^2 + \lambda} \right) \left(\frac{\lambda_j}{\lambda_j^2 + \lambda} \right) y^t u_i \underbrace{v_i^t v_j}_{\substack{0 \text{ unless} \\ i = j}} u_j^t y$$

$$= \sum_{i=1}^{d} \left(\frac{\lambda_i}{\lambda_i^2 + \lambda} \right)^2 \| u_i^t y \|^2$$

As $\lambda \uparrow \infty$, $\|\hat{\beta}_\lambda\|^2 \to 0$ as expected.

Also, $\hat{\beta} = (X^t X)^{-1} X^t y$ = LS solution has norm

$$\|\hat{\beta}\|^2 = \sum_{i=1}^{d} \frac{1}{\lambda_i^2} \| u_i^t y \|^2 > \|\hat{\beta}_\lambda\|^2 .$$

↑ Note that for the LS solution to be well defined, we need $\lambda_i^2 > 0$; i.e. rank $X = d$. This is not needed for the ridge estimator since $\lambda > 0$.

The ridge estimate of y is

$\hat{y}_\lambda := X \hat{\beta}_\lambda$

$= (U \Lambda V^t) V (\Lambda^2 + \lambda I_d)^{-1} \Lambda U^t y$

$= U \underbrace{\Lambda (\Lambda^2 + \lambda I_d)^{-1} \Lambda}_{\text{diagonal matrix}} U^t y$

$$\Rightarrow \boxed{\hat{y}_\lambda = \sum_{i=1}^{d} u_i \left(\frac{\lambda_i^2}{\lambda_i^2 + \lambda} \right) u_i^t y}$$

Recall that $\text{proj}_u v = \frac{\langle u, v \rangle}{\langle u, u \rangle} u$.

When $\langle u, u \rangle = u^t u = 1$, $\text{proj}_u v = \underbrace{(u^t v)}_{\substack{\text{"coordinate of the projection in} \\ \text{the } u \text{ direction.}}} u$

$\Rightarrow u_i^t y$ = the i-th coordinate of y in the U basis (recall that the columns of U provide an orthonormal basis for the column space of X)

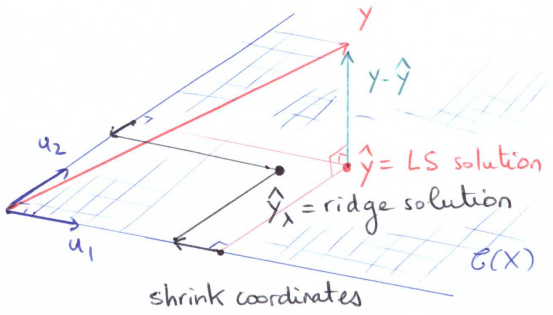
$\Rightarrow$ Ridge regression shrinks these coordinates by a factor $\lambda_i^2 / (\lambda_i^2 + \lambda) < 1$.

↳ When $\lambda = 0$, we recover the LS estimate of y:

$\hat{y} = \sum_{i=1}^{d} (u_i^t y) u_i$: No shrinkage here!

↳ compare this expression with the QR decomposition of $X = QR$, where the columns of Q form an orthonormal basis of $\mathcal{C}(X)$ = column space of X, and the estimate $\hat{y} = \sum_{j=1}^{d} (q_j^t y) q_j$.

[Picture:]



Remark: Coordinates are shrunk by a factor $\frac{\lambda_i^2}{\lambda_i^2+\lambda}$.

⇒ A greater shrinkage is applied to basis vectors with smaller λ_i^2.

↳ coefficient λ_i has an interpretation in terms of Principal Components (PC).

Indeed, recall that for a vector $X \in \mathbb{R}^d$ with covariance matrix Σ, the

- First Principal Component is the linear combination $a_1^t X$ that maximizes $\mathrm{Var}(a_1^t X) = a_1^t \Sigma a_1$, subject to $a_1^t a_1 = 1$ (otherwise $\mathrm{Var}(a_1^t X)$ could be made arbitrarily large).
- Second PC is the linear combination $a_2^t X$ that maximizes $\mathrm{Var}(a_2^t X)$, subject to $a_2^t a_2 = 1$ and $\mathrm{Cov}(a_1^t X, a_2^t X) = a_1^t \Sigma a_2 = 0$ (the second PC is uncorrelated with the first)
- And so on .../...

It turns out that the directions $a_1, a_2, \dots$ correspond to the eigenvectors of Σ.

> The i-th PC of X is $z_i := e_i^t X$, where $e_1, \dots, e_d$ are the eigenvectors of Σ, with associated eigenvalues $\lambda_1, \dots, \lambda_d$ (such that $\lambda_1 \geq \dots \geq \lambda_d > 0$).
> Moreover, $\mathrm{Var}\, z_i = e_i^t \Sigma e_i = \lambda_i$, and $\mathrm{Cov}(z_i, z_j) = 0$ for $i \neq j$.

In practice, the eigenvalues and eigenvectors are computed from the sample covariance matrix $S := \frac{1}{n} X^t X$ ($d \times d$) ($d \times n$)($n \times d$)

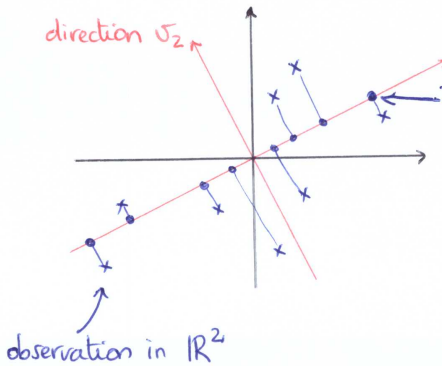
From the SVD decomposition of $X = U \Lambda V^t$, we see that S has eigenvalue-eigenvector pairs $\left(\frac{\lambda_i^2}{n}, v_i\right)$

(recall from page 13 that $X^t X v_i = \lambda_i^2 v_i$)

⇒ The i-th PC is $\underset{(n\times 1)}{z_i} = \underset{(n\times d)(d\times 1)}{X v_i} = \underset{(n\times 1)}{\lambda_i u_i}$

= n realizations of the i-th PC.

The i-th PC is in the v_i direction, and has variance λ_i^2/n.



The first PC lie here since the j-th observation x_j has coordinate $x_j^t v_i = z_{ij}$ in the v_i direction, $j = 1, \dots, n$, and $z_i = (z_{i1}, \dots, z_{in})^t$.

Looking back at the picture on the top of page 16, we conclude that ridge regression shrinks directions u_i the most corresponding to small λ_i^2 (since factor $\lambda_i^2/(\lambda_i^2+\lambda) \to 1$ as $\lambda_i \uparrow \infty$, and converges to zero as $\lambda_i \to 0$); that is directions in the column space of X having a small variance.

RR protects us against the potentially high variance in the estimation of the parameters along the short directions.

#TakeAwayMessage
RR shrinks directions the most corresponding to the directions in $C(X)$ having small variance

Ex: Simple Linear Regression ($d=1$): $y = \beta_0 + \beta_1 x + \varepsilon$. $\varepsilon \sim \mathcal{N}(0,\sigma^2)$

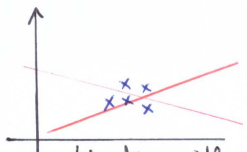
The LS solution is $\hat\beta = (X^tX)^{-1}X^ty$.

After calculations, one can show that

$$\text{Var}\,\hat\beta_0 = \sigma^2\left(\frac{1}{n} + \frac{\bar x}{\sum(x_i-\bar x)^2}\right)$$

$$\text{Var}\,\hat\beta_1 = \frac{\sigma^2}{\sum(x_i-\bar x)^2}$$

The more spread out the input points are, and the less uncertainty there is around the coefficient estimates.



estimation with high uncertainty

estimation is reliable since $\sum(x_i-\bar x)^2$ is large

RR shrinks (potentially large) coefficients in directions of small variance, to reduce the overall variance of the predictor.

Remark: degrees of freedom.

Assuming that $\text{Cov}\,y = \sigma^2 I_n$ = conditionally on x, the output is uncorrelated, with constant variance.

Ridge prediction is $\hat y_\lambda = H_\lambda y$, with $H_\lambda = X(X^tX+\lambda I)^{-1}X^t$

$$\text{Cov}\begin{pmatrix}\hat y\\ y\end{pmatrix} = \text{Cov}\begin{pmatrix}H_\lambda y\\ I_n y\end{pmatrix} = \text{Cov}\begin{pmatrix}H_\lambda\\ I_n\end{pmatrix} y$$

$$= \begin{pmatrix}H_\lambda\\ I_n\end{pmatrix}\text{Cov}\,y\,\begin{pmatrix}H_\lambda^t & I_n\end{pmatrix}$$

$$= \sigma^2\begin{pmatrix}H_\lambda^2 & H_\lambda\\ H_\lambda & I\end{pmatrix}$$

Covariance Matrix of $\begin{pmatrix}\hat y\\ y\end{pmatrix} \in \mathbb{R}^{2n}$

The terms on the diagonal elements of the upper right block of the covariance matrix corresponds to $\text{cov}(y_i, \hat y_i)$.

Thus $\text{Tr}(H_\lambda) = \frac{1}{\sigma^2}\sum_{i=1}^n \text{Cov}(y_i, \hat y_i)$

= Measure of how strongly associated the outputs are with their predicted value.

The higher $\text{Tr}(H_\lambda)$ the more adaptive the estimate

⇒ This motivates the definition of EFFECTIVE DEGREE OF FREEDOM $df(\lambda)$ as

$df(\lambda) = \text{Tr}\,H_\lambda$ = Sum of the eigenvalues of H_λ.

$$df(\lambda) = \sum_{j=1}^{d}\frac{\lambda_j^2}{\lambda_j^2+\lambda}$$

Page 15, we computed

$H_\lambda = U\Lambda(\Lambda^2+\lambda I_d)^{-1}\Lambda U^t$

$\Downarrow$

$H_\lambda u_i = \left(\frac{\lambda_i^2}{\lambda_i^2+\lambda}\right)u_i$.

For $\lambda = 0$, $df(0) = d$ (no regularisation)

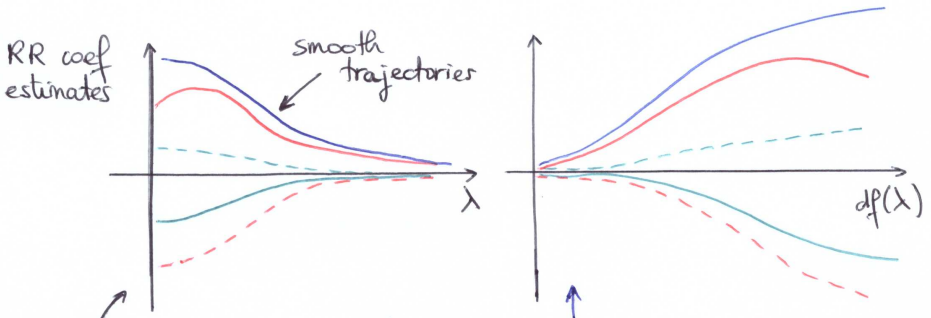
corresponds to the number of parameters to estimate.

↑ As $\lambda \to 0$, $df(\lambda) \to 0$

As we increase the penalty associated with large coefficients, we decrease the effective degree of freedom.

$\Rightarrow$ $df(\lambda)$ gives us a way to quantify the complexity of procedure (model + objective function used).

[Picture]



Each line corresponds to the evolution of one coefficient as the parameter λ is varied.

Alternatively, you may plot the ridge estimates as a function of $\|\hat{\beta}_\lambda\| / \|\hat{\beta}^{LS}\|$ $\Rightarrow$ axis between 0 and 1.

↘ Step III-b. General Case (Lasso)

Recall that the lasso problem is to minimise

$$RSS_1(\lambda) = \|y - X\beta\|_2^2 + \lambda \|\beta\|_1.$$

The solution to the minimization problem may not be unique, but the resulting estimator $\hat{y}_\lambda := X\hat{\beta}_\lambda$ can be shown to be unique.

$\hat{\beta}_\lambda \in \operatorname{argmin} RSS_1(\lambda)$

[Criterion to check whether a solution is unique or not exist]

• Bias and variance of the lasso estimator.

Explicit formulas for the bias and variance of the lasso estimator do not exist. However, their trend is similar to the ridge estimator:

↘ As λ increases, the bias increases

↘ As λ increases, the variance decreases.

Approximate expressions are available, and can be found in Tibshirani (1996) and Osborne et al. (2000). For example, Osborne et al showed that

$$\boxed{Var(\hat{\beta}_\lambda) \approx \sigma^2 (X^tX + W)^{-1} X^tX (X^tX + W)^{-1}}$$

where

$$W = \frac{1}{\|\hat{\beta}_\lambda\|_1 \, \|X^t\hat{\varepsilon}\|_\infty} (X^t\hat{\varepsilon})(X^t\hat{\varepsilon})^t$$

$\hat{\varepsilon} = y - X\hat{\beta}^\lambda$

$\|v\|_\infty = \max_j |v_j|$

• Oracle bounds.

Consider the linear model $y = X\beta^* + \varepsilon$, where the columns of X are standardized. ORACLE BOUNDS for the lasso estimator $\hat{y}^\lambda = X\hat{\beta}_\lambda$ ($\hat{\beta}_\lambda \in \operatorname{argmin} RSS_1(\lambda)$) are available. They involve the so-called COMPATIBILITY CONSTANT $K(\beta)$, defined in terms of X. This constant is a measure of the lack of orthogonality in the columns of X. A simple lower bound on $K(\beta)$ can be given in terms of

- $\theta = \max_{i \neq j} \langle x_i, x_j \rangle$ ← x_i = i-th column of X
- m = number of non zero entries in $\beta \in \mathbb{R}^d$; $m \leq d$.

Then it is possible to show that $K(\beta)^2 \geq 1 - 11\,\theta\, m$

As $\theta \to 0$ ($\perp$ columns), $K(\beta)$ gets closer to 1.

The following bounds are proved in Giraud (2015)

ORACLE BOUND #1 (deterministic bound)

For $\lambda \geq 3 \| X^t \varepsilon \|_\infty$ we have

$$\| X(\hat{\beta}_\lambda - \beta^*) \|_2^2 \leq \inf_{\beta \in \mathbb{R}^d \setminus \{0\}} \left\{ \| X(\beta - \beta^*) \|_2^2 + \frac{\lambda^2}{K(\beta)^2} \|\beta\|_0 \right\}$$

What you can do

The best you can do ↳ the oracle + some error

where $\|\beta\|_0$ = number of non zero entries in β.

Remark: Upper bound get smaller as $K(\beta)$ gets larger, corresponding to orthogonal columns in X.

ORACLE BOUND #2 (probabilistic bound)

For any $L > 0$, the lasso estimator with tuning parameter $\lambda = 3\sigma\sqrt{2 \log d + 2L}$ fullfils with probability at least $1 - e^{-L}$ the risk bound

$$\| X(\hat{\beta}_\lambda - \beta^*) \|_2^2 \leq \inf_{\beta \neq 0} \left\{ \| X(\beta - \beta^*) \|_2^2 + \frac{18\sigma^2}{K(\beta)^2} (L + \log d) \|\beta\|_0 \right\}$$

Such bounds provide guarantees on the performance of the lasso estimator.

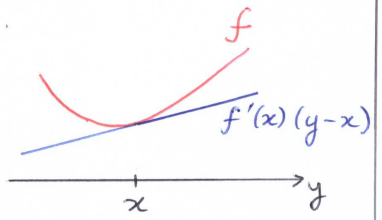
• Further properties of the lasso estimator.

In general, there is no explicit expression for the lasso estimator. However, a lot can be said about the general properties of the minimizer of $RSS_\lambda(\beta) = \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$

Non differentiable with respect to β.

For a differentiable convex function, we have

$$f(y) \geq f(x) + \nabla f(x)^t (y - x) \quad \forall x, y \in \text{dom } f$$



For any convex function, possibly non differentiable, we introduce the SUBDIFFERENTIAL ∂f of f, defined by

$$\partial f(x) = \{ u \in \mathbb{R}^d \mid f(y) \geq f(x) + \langle u, y - x \rangle \ \forall y \in \text{dom } f \}$$

$\in \mathbb{R}^d$

$u \in \partial f(x)$ is called the SUBGRADIENT of f in x.

↘ f is CONVEX $\iff \forall x \in \mathbb{R}^d$, $\partial f(x) \neq \emptyset$

↘ f is CONVEX & DIFFERENTIABLE $\iff \partial f(x) = \{ \nabla f(x) \}$.

Calculus of subgradients.

- Scaling: $\partial(\alpha f) = \alpha\, \partial f$ if $\alpha > 0$
- Addition: $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$ — f_1, f_2 convex

addition of sets

Remark: In many applications, we do not need to derive all elements in ∂f, but only a few.

Example: ℓ_1 norm $\|x\|_1 = \sum_{j=1}^{d} |x_j|$

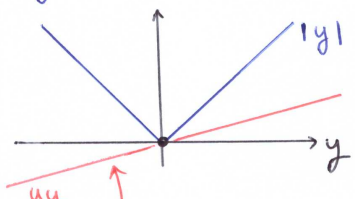
↳ Focus on the case $d=1$.

To derive $\partial|x|$ for $x \in \mathbb{R}$, the only concern is at $x=0$ since $|x|$ is differentiable $\forall x \neq 0$.

At 0, we are looking for the set of $u \in \mathbb{R}$ such that

$\forall y \quad |y| \geq |0| + u(y-0)$

$|y| \geq uy$



stays below $|y|$ if $u \in [-1,1]$

$\text{sign}\, x = 1_{x>0} - 1_{x<0}$

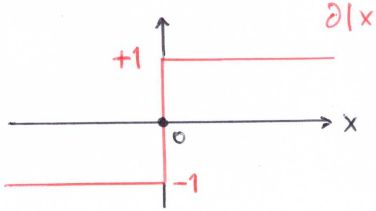
Then

$$\partial|x| = \{ u \in \mathbb{R} \mid \begin{array}{ll} u = \text{sign}\, x & \text{if } x \neq 0 \\ u \in [-1,1] & \text{if } x = 0 \end{array} \}$$

↳ Thus

$$\partial\|x\|_1 = \{ u \in \mathbb{R}^d \mid \begin{array}{ll} u_j = \text{sign}\, x_j & \text{if } x_j \neq 0 \\ u_j \in [-1,1] & \text{if } x_j = 0 \end{array} \}$$

$\in \mathbb{R}^d$



↳ What is the subgradient of $f(x) = x^2 + |x|$? $g(x) = e^x + |x|$?

$$\to \partial f = \{ u \in \mathbb{R} \mid \begin{array}{ll} u = 2x + \text{sign}\, x & \text{if } x \neq 0 \\ u \in [-1,1] & \text{if } x = 0 \end{array} \}$$

$$\to \partial g = \{ u \in \mathbb{R} \mid \begin{array}{ll} u = e^x + \text{sign}\, x & \text{if } x \neq 0 \\ u \in [0,2] & \text{if } x = 0 \end{array} \}$$

Useful fact: The minimizers of a convex function $f : \mathbb{R}^d \to \mathbb{R}$ are characterized by

(◇) $x_* \in \underset{x \in \mathbb{R}^d}{\text{argmin}}\, f(x) \iff 0 \in \partial f(x_*)$

(0 is a subgradient of f in x_*)

Subdifferential of f at x_* is a collection of vectors, and one of them is exactly 0.

Ex: $\partial|\cdot|$ at 0 indeed contains 0.

Indeed, since x_* is a minimizer, $f(y) \geq f(x_*)$ for all y, and $f(y) \geq f(x_*) + \langle 0, y - x \rangle$ so that $0 \in \partial f(x_*)$ indeed.

↳ Back to the lasso problem. We have a characterization of the solution in terms of the subdifferential of

$RSS_\lambda(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$

↑ denote this $\ell(\beta)$, to emphasize the dependence on β.

$$\partial\ell(\beta) = \{ \underbrace{-2X^T(y - X\beta)}_{\text{the differentiable part}} + \lambda z \mid z \in \partial\|\beta\|_1 \}$$

Denoting $\hat{\beta}_\lambda \in \underset{\beta}{\text{argmin}}\, RSS_\lambda(\beta)$ the solution to the lasso problem, we deduce from (◇) the existence of $\hat{z} \in \partial\|\hat{\beta}_\lambda\|_1$ such that $-2X^T(y - X\hat{\beta}_\lambda) + \lambda\hat{z} = 0$, equivalently

(□) $X^T X \hat{\beta}_\lambda = X^T y - \frac{\lambda}{2}\hat{z}$ for $\hat{z} \in \mathbb{R}^d$ such that

$$\hat{z} = \begin{cases} \text{sign}\,(\hat{\beta}_\lambda)_j & \text{if } (\hat{\beta}_\lambda)_j \neq 0 \\ \in [-1,1] & \text{if } (\hat{\beta}_\lambda)_j = 0 \end{cases}$$

KEY RELATION

Writing $\hat{m}_\lambda = \{ j \mid (\hat{\beta}_\lambda)_j \neq 0 \}$

$= $ set of indices for which the associated coefficient is non zero ("support of $\hat{\beta}_\lambda$")

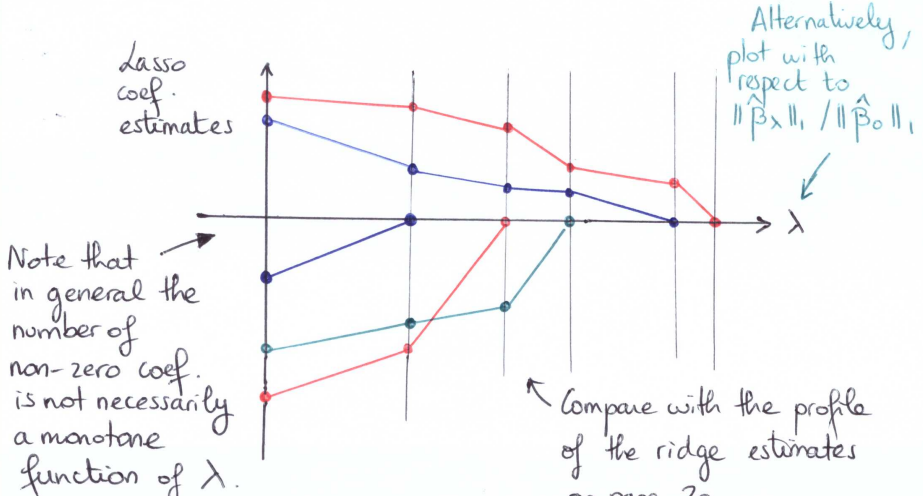
$= \operatorname{supp} \hat{\beta}_\lambda$,

we get from (□) that

$$X_{\hat{m}_\lambda}^t X_{\hat{m}_\lambda} (\hat{\beta}_\lambda)_{\hat{m}_\lambda} = X_{\hat{m}_\lambda}^t y - \frac{\lambda}{2} \operatorname{sign}(\hat{\beta}_\lambda)_{\hat{m}_\lambda}$$

where $X_{\hat{m}_\lambda}$ = version of X, keeping only columns in $\hat{m}_\lambda$, i.e. associated with non zero coefficients.

Consequence: For values of λ for which $\hat{m}_\lambda$ remains constant, the above equation shows that $\hat{\beta}_\lambda$ depends linearly on λ.
In other words, the function $\lambda \mapsto \hat{\beta}_\lambda$ is piecewise constant.



Note that in general the number of non-zero coef. is not necessarily a monotone function of λ.

↖ Compare with the profile of the ridge estimates on page 20.

⇒ Efficient algorithms to compute the full profile exist.

Page 20, we plotted the ridge estimates as a function of $df(\lambda)$ as well, where $df(\lambda)$ was defined to be the trace of $H_\lambda = X(X^tX + \lambda I)^{-1}X^t$.

For the lasso, we need the more general definition introduced page 20, namely $df(\lambda) = \frac{1}{\sigma^2} \sum_{i=1}^n \operatorname{cov}(y_i, \hat{y}_i)$.

This definition is given in Efron (2004), under the general assumption that conditionally on x, y has covariance matrix $\sigma^2 I_n$.

It is shown in Zhou et al (2007) that $df(\lambda)$ corresponds to the expected number of non zero coefficients for each λ: $df(\lambda) = \mathbb{E}|\hat{m}_\lambda|$.

↳ it follows that $|\hat{m}_\lambda|$ = # element in m_λ is an unbiased estimate for $df(\lambda)$.
(consistency of the estimate can also be established

- Let's go back to relation (□) on page 25:

$$X^tX\hat{\beta}_\lambda = X^ty - \frac{\lambda}{2}\hat{z}$$

$\times \hat{\beta}_\lambda^t$

$$0 \leqslant \hat{\beta}_\lambda^t X^tX \hat{\beta}_\lambda = \langle \hat{\beta}_\lambda, X^ty - \frac{\lambda}{2}\hat{z} \rangle$$

$$= \sum_{j \in \hat{m}_\lambda} (\hat{\beta}_\lambda)_j \left\{ x_j^t y - \frac{\lambda}{2} \operatorname{sign}(\hat{\beta}_\lambda)_j \right\}$$

x_j = j-th column of X.

Notation: $\|A\|_\infty = \max_j |A_j|$

If $\lambda \geqslant 2 \| X^t y \|_\infty$ and $\hat{m}_\lambda \neq \emptyset$, we have two possibilities.

If $\text{sign}(\hat{\beta}_\lambda)_j = +1$, $x_j^t y - \frac{\lambda}{2} < 0$ from our choice of λ. Thus $\underbrace{(\hat{\beta}_\lambda)_j}_{>0} \{ \underbrace{x_j^t y - \frac{\lambda}{2} \text{sign}(\hat{\beta}_\lambda)_j}_{<0} \} < 0$

If $\text{sign}(\hat{\beta}_\lambda)_j = -1$, $x_j^t y + \frac{\lambda}{2} > 0 \quad \forall j \in \hat{m}_\lambda$.
Thus $\underbrace{(\hat{\beta}_\lambda)_j}_{<0} \{ \underbrace{x_j^t y - \frac{\lambda}{2} \text{sign}(\hat{\beta}_\lambda)_j}_{>0} \} < 0$

$\Rightarrow$ If $\hat{m}_\lambda \neq 0$ for $\lambda \geqslant 2 \| X^t y \|_\infty$, then
$\sum_{j \in \hat{m}_\lambda} (\hat{\beta}_\lambda)_j \{ x_j^t y - \frac{\lambda}{2} \text{sign}(\hat{\beta}_\lambda)_j \} < 0$, a contradiction.

Thus $\hat{m}_\lambda = \emptyset$, that is, all component of $\hat{\beta}_\lambda$ are zero.

Take Away Message

If $\lambda \geqslant 2 \| X^t y \|_\infty$, then $\hat{\beta}_\lambda \equiv 0$. In other words, for λ large enough, the lasso estimator is identically zero.

↳ justifies further the picture on page 26.

III- COMPUTING THE LASSO ESTIMATOR

In general, there is no explicit solution to the lasso problem, and we have to resort to numerical procedures. We discuss two approaches: → LARS algorithm
→ Coordinate Descent algorithm

III.1. LARS algorithm.

The lasso solution is piecewise linear, with respect to the tuning parameter. The LARS algorithm provides a way to compute the entire coefficient path efficiently at low cost.

$\Rightarrow$ The entire solution is available: for all λ.
Very attractive approach: we do not need to recompute the lasso solution on a grid of values of λ.

LARS algorithm was introduced by Efron et al (2004), and can be understood from a modified version of another popular approach for variable selection: Least Angle Regression)

Starting point is relation

$$X_{\hat{m}_\lambda}^t X_{\hat{m}_\lambda} (\hat{\beta}_\lambda)_{\hat{m}_\lambda} = X_{\hat{m}_\lambda}^t y - \frac{\lambda}{2} \text{sign}(\hat{\beta}_\lambda)_{\hat{m}_\lambda},$$

derived on page 26: for the values of λ for which $\hat{m}_\lambda$ remains constant, $\hat{\beta}_\lambda$ depends linearly on λ. The LARS algorithm computes the sequence $\{ \hat{\beta}_{\lambda_1}, \hat{\beta}_{\lambda_2}, \dots \}$ of lasso solutions, for values $\lambda_1, \lambda_2, \dots$ corresponding to the breakpoints of the path $\lambda \to \beta_\lambda$. At a breakpoint, two situations may occur: either one coordinate is removed from $\hat{m}_\lambda$, or one coordinate must be added.

More details about the LARS algorithm can be found in Efron et al (2004)

III.2. Coordinate Descent Algorithm

The Coordinate Descent Algorithm (CDA) optimizes each parameter separately, keeping the others fixed, and cycles around until the coefficients stabilize to some value.

Idea: for a function $f: \mathbb{R}^d \to \mathbb{R}$, if for some x^* the value $f(x^*)$ is such that $f(x^* + \alpha u_i) \geq f(x^*)$ $\forall \alpha \in \mathbb{R}$, $u_i = (0, \dots, 0, \underset{\uparrow i}{1}, 0, \dots, 0)$ (i.e. minimum is located at x^* along each coordinate axis), then it seems reasonable to hope that x^* is the global minimum of f, that is $x^* = \underset{x \in \text{dom} f}{\text{argmin}}\ f(x)$. While this not always holds, there are a wide variety of situations where this is true.

Coordinate Descent Algorithm

Initialize with guess $x = (x_1, \dots, x_d)^t$

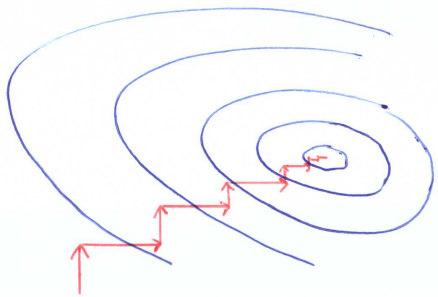
Repeat

  For all $j = 1, \dots, d$ do

    $x_j \leftarrow \underset{x_j}{\text{argmin}}\ f(x_1, \dots, x_d)$

  End

Until Convergence



Can be applied to a wide variety of optimization problems, including in Machine learning (lasso).

Of partialar interest are minimization of criterion of the form

$$\min_{x \in \mathbb{R}^q} \left\{ g(x) + \sum_{j=1}^{d} h_j(x_j) \right\}$$

where – both g and h are convex functions
– g is differentiable.

Typically, g represents the goodness of fit term, and h_j a penalty term. (Tseng, 1988) showed that for functions of this form, the coordinate descent algorithm converges to the solution.

The lasso criterion is of this form: $RSS_1(\lambda) = \|y - X\beta\|_2^2 + \sum_{j=1}^{d} |\beta_j|$

- Derivation of the CDA for the lasso.

We assume that the observation matrix is standardized, so that columns have zero mean and unit variance: $\sum_{i=1}^{n} x_{ij} = 0$ and $\sum_{i=1}^{n} x_{ij}^2 = 1$.

Pages 10/11, we derived the lasso solution explicitely when the columns of X are orthogonal ($X^tX = I$). It is easy to see that a similar expression holds when the lasso is applied to a single predictor. Indeed,

$$\underset{(n \times 1)}{X} = \begin{pmatrix} x_{11} \\ \vdots \\ x_{n1} \end{pmatrix} \Rightarrow X^tX = \sum_{i=1}^{n} x_{i1}^2 = 1$$

Thus $\hat{\beta}_\lambda = \text{sign}(\hat{\beta}) \left(|\hat{\beta}| - \frac{\lambda}{2} \right)_+ =: S(\hat{\beta}, \lambda)$

$:= \underset{\beta \in \mathbb{R}}{\text{argmin}}(RSS_1(\lambda))$ Least Square Solution $\hat{\beta} = X^ty$

In other words,

$$\hat{\beta}_\lambda = S(\hat{\beta}, \lambda) = \begin{cases} \hat{\beta} - \frac{\lambda}{2} & \text{if } \hat{\beta} > 0 \text{ and } |\hat{\beta}| > \frac{\lambda}{2} \\ \hat{\beta} + \frac{\lambda}{2} & \text{if } \hat{\beta} < 0 \text{ and } |\hat{\beta}| > \frac{\lambda}{2} \\ 0 & \text{if } |\hat{\beta}| \leq \frac{\lambda}{2} \end{cases}$$

With more than one (non orthogonal) predictor, we use the CDA which updates coefficients one at a time, and we show that each update applies similarly a soft-thresholding with a "partial residual" as a response variable.

Update on the j-th coordinate, keeping the remaining $(d-1)$ fixed
↳ our current estimates: $\beta_1, \dots, \beta_{j-1}, \beta_{j+1}, \dots, \beta_d$.

Goal: minimize

$$f(\beta_j) = \sum_{i=1}^n \Big(y_i - \sum_{k=1}^d \beta_k x_{ik} \Big)^2 + \lambda \sum_{k=1}^d |\beta_k|$$

$$= \sum_{i=1}^n \Big(y_i - \underbrace{\sum_{k \neq j}^d \beta_k x_{ik}}_{\text{fixed}} - \beta_j x_{ij} \Big)^2 + \lambda \underbrace{\sum_{k \neq j}^d |\beta_k|}_{\text{fixed}} + \lambda |\beta_j| .$$

$y_i^{(j)}$:=

$$= \sum_{i=1}^n \Big(\underbrace{y_i - y_i^{(j)}}_{\text{"partial residual"}} - \beta_j x_{ij} \Big)^2 + \lambda |\beta_j| + \text{something independent of } \beta_j$$

⇒ Minimizing $f(\beta_j)$ with respect to β_j only reduces to solving the lasso problem with a single predictor, except that the response variable is now $y_i - y_i^{(j)}$.

⇒ $\beta_j \leftarrow S\Big(\sum_{i=1}^n x_{ij} (y_i - y_i^{(j)}) \, , \, \lambda \Big)$

since the least squares estimate of β_j is

$$x_j^t \underbrace{(y - \tilde{y}^{(j)})}_{\text{partial residual}} = \sum_{i=1}^n x_{ij} (y_i - y_i^{(j)})$$

, with $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ $\tilde{y}^{(j)} := \begin{pmatrix} y_1^{(j)} \\ \vdots \\ y_n^{(j)} \end{pmatrix}$

Denoting $R_j := x_j^t \Big(y - \sum_{k \neq j} \beta_k x_k \Big)$, the algorithm reads:

Coordinate Descent Algorithm for the Lasso

Initialization $\beta = \beta_0 \in \mathbb{R}^d$

Repeat until convergence of β, the loop

For $j = 1, \dots, d$

$$R_j = x_j^t \Big(y - \sum_{k \neq j} \beta_k x_k \Big)$$

$$\beta_j \leftarrow \text{sign}(R_j) \Big(|R_j| - \frac{\lambda}{2} \Big)_+$$

Output β

Start the CDA for a large value λ_{max} of the tuning parameter, and repeat the procedure for each value of λ on a grid, down to some value λ_{min}. Each solution is used as a warm start for the next problem.

⇒ The LARS algorithm computes the entire solution, while the CDA solution is on a grid of values of λ.

The coordinate descent algorithm for the lasso, introduced by Friedman et al (2007) is shown in Friedman et al (2010) to be faster than many competitors (up to a factor 10/100). Competitors implemented in R include:

→ lars (for a square error loss)

→ lilogreg (for the lasso and penalized logistic regression). Algorithm is based on interior points for convex optimization problems by Koh et al (2007)

The CDA for the lasso (square error) and logistic regression (K-class classification, $K \geq 2$) is implemented in glmnet.

↳ The numerical study was carried out in low ($n > p$) and high dimension ($p \gg n$), for dense and sparse features (lots of zeros in the observation matrix). The improvement is considerable in particular in high dimensions.

↖ One reason why the CDA is very fast is that updates are trivial.

IV - THE ELASTIC NET.

The elastic net was introduced by Zou & Hastie (2005) as a new regularization and variable selection method. The criterion to minimize is

$$RSS(\lambda, \alpha) = \|y - X\beta\|_2^2 + \underbrace{\lambda(1-\alpha)\|\beta\|_1}_{:=\lambda_1} + \underbrace{\lambda\alpha}_{:=\lambda_2}\|\beta\|_2^2$$

where $\lambda > 0$ and $0 < \alpha < 1$.

↑ α controls the mix between the ℓ_1 and ℓ_2 penalty.

Idea: Perform variable selection like the lasso, and shrink coefficients like ridge regression.

- Analysis of the solution when $X^tX = I$.

We proceed as on pages 10/11 for the lasso:

$$\min_\beta RSS(\lambda_1, \lambda_2)$$

$$= \min_\beta \|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2$$

$$= \min_\beta \left(y^ty - 2\beta^t \underbrace{X^ty}_{=\hat\beta^{ls}} + \beta^t \underbrace{X^tX}_{=I}\beta + \lambda_1 \sum|\beta_j| + \lambda_2 \beta^t\beta \right)$$

$$\Leftrightarrow \min_\beta \sum_{j=1}^d \left\{ -2\beta_j\hat\beta_j^{ls} + (1+\lambda_2)\beta_j^2 + \lambda_1|\beta_j| \right\}$$

↳ minimization can be performed for each variable β_j separately

$$\Leftrightarrow \min_\beta \left\{ -2\beta\hat\beta^{ls} + (1+\lambda_2)\beta^2 + \lambda_1|\beta| \right\}$$

↖ Proceed as for the lasso:

- On $\{\hat\beta^{ls} > 0\}$, observe that necessarily the solution $\hat\beta^{\text{elastic net}}$ is positive. The criterion to minimize is thus $-2\beta\hat\beta^{ls} + (1+\lambda_2)\beta^2 + \lambda_1\beta$, on the positive half line. The minimum is attained at $\frac{1}{1+\lambda_2}\left(\hat\beta^{ls} - \frac{\lambda_1}{2}\right)$, provided this quantity is positive, and at 0 otherwise.

• Proceed similarly for $\{\hat{\beta}^{ls} < 0\}$.

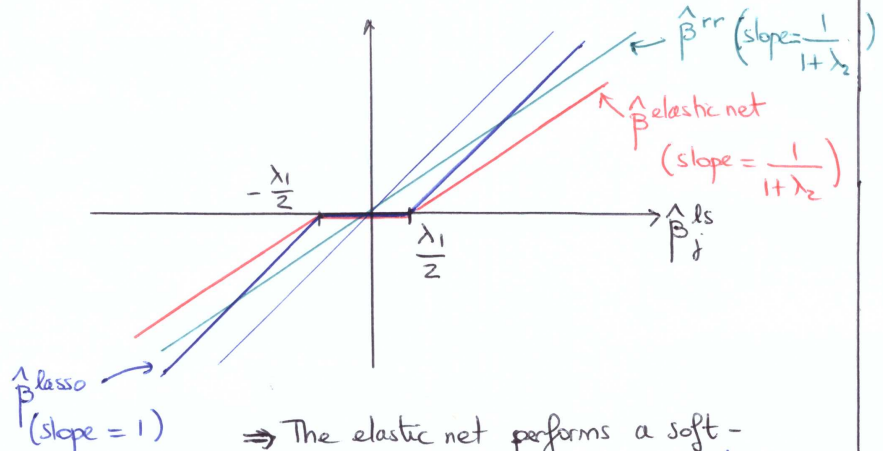
Putting things together, we conclude that

$$\hat{\beta}_j^{\text{elastic net}} = \frac{1}{1+\lambda_2}\,\text{sign}(\hat{\beta}_j^{ls})\left(|\hat{\beta}_j^{ls}| - \frac{\lambda_1}{2}\right)_+$$

$$= \frac{S(\hat{\beta}_j^{ls}, \lambda_1/2)}{1+\lambda_2}$$

↖ where $S(\cdot,\cdot)$ is defined bottom of page 31.

[Picture]



⇒ The elastic net performs a soft-thresholding (since coefficients $|\hat{\beta}_j^{ls}|$ less than $\lambda_1/2$ are set to zero), combined with a proportional shrinkage (slope is $\frac{1}{1+\lambda_2} < 1$, and not equal to one).

• Computing the elastic net solution.

The coordinate descent procedure proves useful here as well to compute the elastic net solution. Proceeding as for the lasso (pages 32/33), we obtain:

Coordinate Descent Algorithm for the elastic net.

Initialization $\beta = \beta_0 \in \mathbb{R}^d$

Repeat until convergence of β, the loop

For $j = 1, \dots, d$

$$R_j = x_j^t\left(y - \sum_{k \neq j} \beta_k x_k\right)$$

$$\beta_j \leftarrow \frac{1}{1+\lambda_2}\,\text{sign}(R_j)\left(|R_j| - \frac{\lambda_1}{2}\right)_+$$

Output β

Remark: Advantages of the elastic net over the lasso.

With (highly) correlated variables, the lasso tends to select one variable from the group, and does not care which one is selected; while the ridge solution tends to share similar coefficients amongst correlated variables.

⇒ With α close to 0, the elastic net tends to return a sparse solution better 'behaved' → it can select groups of correlated variables.

See Zou & Hastie (2005) for more information.