Lasso: Least absolute shrinkage and selection operator

Pedro Delicado
Departament d'Estadística i Investigació Operativa
Universitat Politècnica de Catalunya

Statistical Learning, MESIO UPC-UB, 2017-2018 Q2

Introduction

- In the multiple linear regression model (with n observations and p predictors, p possibly greater than n) we consider the penalized least squares coefficients estimator where the penalization is given by the L_1 norm of the estimator.
- This procedure leads to the Lasso method.
- The presentation is based on the following references:
  - Hastie, Tibshirani, and Friedman (2009) *The Elements of Statistical Learning*, Chapter 3 (and particularly Section 3.4).
  - Hastie, Tibshirani, and Wainwright (2015) *Statistical Learning with Sparsity*, Chapters 1 to 5.
  - Tibshirani (2011)
  - Hastie and Qian (2014) *Glmnet vignette*.

In the pathway, we will learn:

- Ridge regression.
- Linear estimators of a regression function.
- Effective number of parameters (or effective degrees of freedom) of a regression estimator.
- Tuning parameters choice based on leave-one-out cross-validation, k-fold cross-validation or generalized cross-validation.
- Efficient computation of leave-one-out cross-validation for linear estimators.
- ...

Multiple linear regression model

- Consider that n pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$ of data, $y_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^p$, are observed from the multiple linear regression model

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i,$$

  where $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. r.v. with zero mean and variance σ^2, and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^{\mathsf{T}} \in \mathbb{R}^{p+1}$ is a vector of unknown coefficients.
- Fitting the model consists in providing estimators for $\boldsymbol{\beta}$ and σ^2.

Ordinary Least Squares (OLS)

- Ordinary Least Squares (OLS) estimator:

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

- In matrix notation: $\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$.
- $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is an unbiased estimator of $\boldsymbol{\beta}$.
- **Gauss–Markov Theorem.** For any $\mathbf{a} \in \mathbb{R}^{p+1}$, the OLS estimator of the linear combination $\mathbf{a}^{\mathsf{T}}\boldsymbol{\beta}$, namely $\mathbf{a}^{\mathsf{T}}\boldsymbol{\beta}_{\text{OLS}}$, is unbiased and it has the lowest variance among the linear unbiased estimates of $\mathbf{a}^{\mathsf{T}}\boldsymbol{\beta}$.
- In particular, following the Bayes rule, the prediction for a new observation $\mathbf{x}$, is $\hat{y} = \mathbf{x}^{\mathsf{T}}\boldsymbol{\beta}$.
- So its best unbiased estimator is $\hat{y}_{\text{OLS}} = \mathbf{x}^{\mathsf{T}}\hat{\boldsymbol{\beta}}_{\text{OLS}}$.

Multicolinearity and bad conditioned matrices

- The computation of $\hat{\beta}_{\text{OLS}}$ is numerically unstable when $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is close to be singular:
- Condition number of a symmetric matrix $\mathbf{A}$: $\kappa(\mathbf{A}) = \sqrt{\frac{\gamma_{\text{max}}}{\gamma_{\text{min}}}}$, where γ_{max} and γ_{min} are, respectively, the largest and lowest eigenvalue absolute values of $\mathbf{A}$.
- $\mathbf{A}$ is not invertible if and only if $\kappa(\mathbf{A}) = \infty$.
- A large value of $\kappa(\mathbf{A})$ (in practice, larger than 30), indicates that numerical problems may appear when inverting $\mathbf{A}$.
- In these cases we say that $\mathbf{A}$ is bad conditioned.
- If $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is bad conditioned then the computation of $\hat{\beta}_{\text{OLS}}$ is numerically unstable.
- A large condition number indicates that $\mathbf{X}$ is close to be singular, that is, close that some columns of $\mathbf{X}$ can be written as linear combinations of the other.
- We talk about multicolinearity between columns of $\mathbf{X}$.

Regularized regression

- Beyond numerical problems, $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ can not be computed when the rank of $\mathbf{X}$ is lower than p (an extreme case of multicolinearity).
- This is the case when $p > n$ (or $p \gg n$, as it can happen in applications with large scale data).
- In practical terms, what happens is that $\mathbf{y}$ can be written as a linear combination of the predictors using infinitely many coefficient vectors, for which the objective OLS objective function is equal to 0, its minimum. So there is no way to select *the best* among those coefficient vectors.
- Shrinkage (or regularized) methods: They add a penalty (depending on $\boldsymbol{\beta}$) to the objective function in such a way that the new optimum is attained at a unique vector $\hat{\boldsymbol{\beta}}$.
- The unbiasedness of OLS estimation is lost, but the new estimators may have lower Mean Square Error (and they are numerically stable).

Ridge regression

- The ridge coefficients minimize a penalized residual sum of squares:

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

$$= \arg\min_{\boldsymbol{\beta}} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{T}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2, \right\}$$

- There is a closed form expression: $\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$.
- Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage of $\boldsymbol{\beta}$ toward zero.

- Alternative expression:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t,$$

for $t \geq 0$. There is a one-to-one decreasing correspondence between parameters $\lambda \in [0, \infty)$ and $t \in (0, \|\hat{\boldsymbol{\beta}}_{\text{OLS}}\|^2]$.

- Observe that changes in scale of the explanatory variables affect the constraint effects (or, equivalently, the effects of penalization term).
- For this reason, from now on we assume that the predictor variables have zero mean and unit variance:

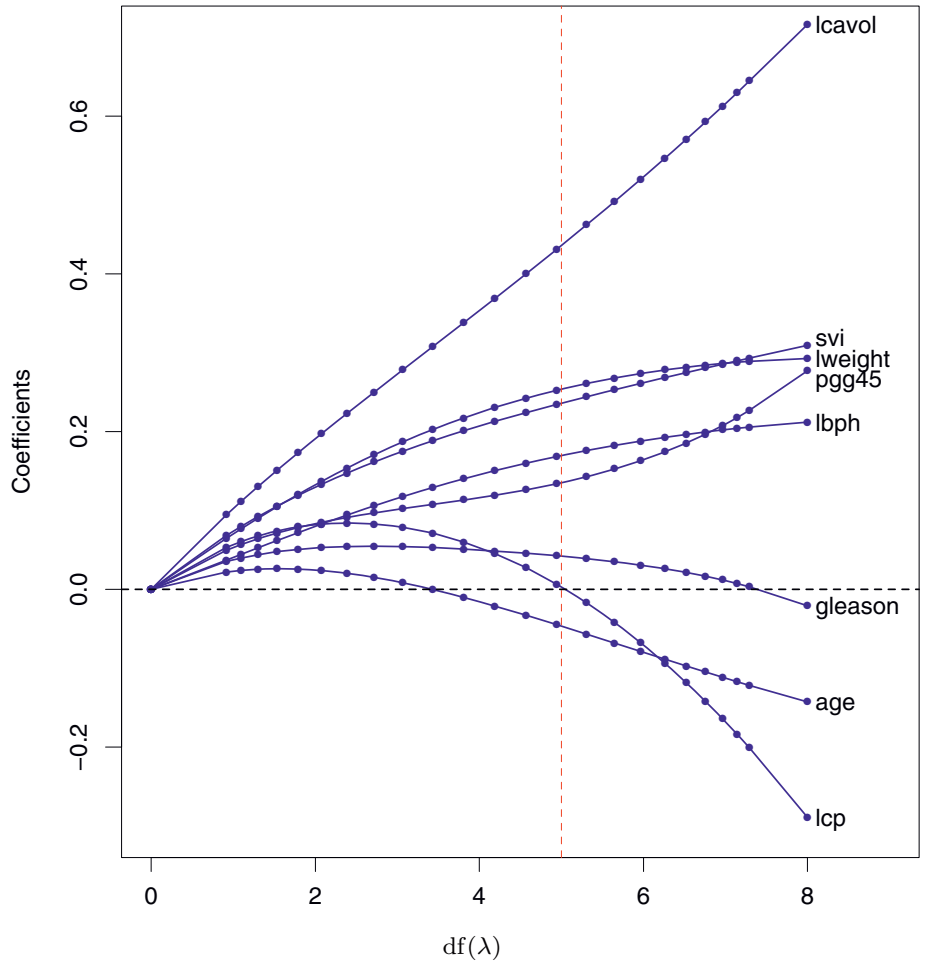
$$\sum_{i=1}^{n} x_{ij} = 0, \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1, \quad j = 1, \ldots, p.$$

- Moreover, the response variable is assumed to have zero mean ($\sum_{i=1}^{n} y_i = 0$), that is, $\beta_0 = 0$.

Prostate cancer example

- Goal: To examine the correlation between the level of log of prostate-specific antigen (lpsa) and a number of clinical measures in 97 men who were about to receive a radical prostatectomy.
- The predictor variables are
  - log cancer volume (lcavol),
  - log prostate weight (lweight),
  - age,
  - log of the amount of benign prostatic hyperplasia (lbph),
  - seminal vesicle invasion (svi),
  - log of capsular penetration (lcp),
  - Gleason score (gleason), and
  - percent of Gleason scores 4 or 5 (pgg45).

Prostate cancer example. Ridge regression



Source: Hastie, Tibshirani, and Friedman (2009)

Explicit solution for the ridge regression

- The ridge regression estimators are the solution of the penalized least squares problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2\right\}$$

that can be expressed as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\Psi(\boldsymbol{\beta})=(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^{\mathsf{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta},$$

that has an explicit solution, as we show now.

- Taking the gradient

$$\nabla\Psi(\boldsymbol{\beta})=-2\mathbf{X}^{\mathsf{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+2\lambda\boldsymbol{\beta},$$

and solving in $\boldsymbol{\beta}$ the equation $\nabla\Psi(\boldsymbol{\beta})=\mathbf{0}$, we obtain

$$\hat{\boldsymbol{\beta}}=\left(\mathbf{X}^{\mathsf{T}}\mathbf{X}+\lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}.$$

- Ridge regression estimator: $\hat{\boldsymbol{\beta}}_{\text{ridge}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y}$.
- Therefore, for any $\mathbf{x} \in \mathbb{R}^p$, the corresponding predicted value is

$$\hat{y} = \mathbf{x}^{\mathsf{T}}\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{x}^{\mathsf{T}}\left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y}.$$

- The vector of fitted values is

$$\hat{\mathbf{y}} = \mathbf{X}\left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y} = \mathbf{H}_\lambda \mathbf{y}.$$

- Compare with the OLS solution: $\hat{\boldsymbol{\beta}}_{\text{OLS}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y}$.

$$\hat{\mathbf{y}}_{\text{OLS}} = \mathbf{X}\left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y} = \mathbf{H}\mathbf{y},$$

  where $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathsf{T}}$ is called the hat matrix.
- $\lim_{\lambda \longrightarrow 0} \hat{\boldsymbol{\beta}}_{\text{ridge}} = \hat{\boldsymbol{\beta}}_{\text{OLS}}$, $\lim_{\lambda \longrightarrow \infty} \hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{0}$.

Practice:

- Prostate data: Ridge regression estimation and *coefficients path*.
- Use the R script `prostate.ridge.regression.R`.

Singular Value Decomposition of $\mathbf{X}$

- Let $\mathbf{X} = \mathbf{UDV}^{\mathsf{T}}$ be the Singular Value Decomposition of $\mathbf{X}$. That is:
  - $\mathbf{U}$, $n \times p$ orthonormal matrix whose columns span the $\mathbf{X}$ column space.
  - $\mathbf{D}$, $p \times p$ diagonal matrix with elements $d_1 \geq \ldots \geq d_p \geq 0$ in the diagonal, that are called singular values of $\mathbf{X}$.
  - $\mathbf{V}$, $p \times p$ orthonormal matrix whose columns span the row space of $\mathbf{X}$.
- Observe that $\mathbf{X}^{\mathsf{T}}\mathbf{X} = \mathbf{VDU}^{\mathsf{T}}\mathbf{UDV}^{\mathsf{T}} = \mathbf{VD}^2\mathbf{V}^{\mathsf{T}}$ and it follows that the eigenvalues of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ are the squared singular values of $\mathbf{X}$:

$$\gamma_j = d_j^2, \ j = 1, \ldots, p.$$

- As we are assuming that the explanatory variables have zero mean, we have that $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is the sample covariance matrix. Then the columns of $\mathbf{V}$ are the principal components of $\mathbf{X}$. Moreover the columns of $\mathbf{U}$ are the scores of the observed data in the principal components.

Numerical stability of ridge regression

- $\hat{\beta}_{\text{ridge}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y}$
- Let us compute the condition number of $\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p$,

$$\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p = \mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathsf{T}} + \lambda \mathbf{V}\mathbf{V}^{\mathsf{T}} = \mathbf{V}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)\mathbf{V}^{\mathsf{T}}.$$

- $\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)$ is a diagonal matrix whose elements in the diagonal are

$$d_j^2 + \lambda = \gamma_j + \lambda, \ j = 1, \ldots, p.$$

- Therefore the condition number of $\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p$ is

$$\kappa\left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right) = \sqrt{\frac{\gamma_1 + \lambda}{\gamma_p + \lambda}}$$

  lower than $\kappa\left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right) = \sqrt{\gamma_1/\gamma_p}$ for all $\lambda > 0$.

- By the way, $\left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} = \mathbf{V}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1}\mathbf{V}^{\mathsf{T}}$, and $\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1} = \text{Diagonal}(1/(d_j^2 + \lambda), j = 1, \ldots, p)$.

Linear estimators of a regression function

- Let $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, be n i.i.d. observs. from the r.v. $(\mathbf{X}, Y)$.
- Let $m(\mathbf{x}) = \mathbb{E}(Y|\mathbf{X} = \mathbf{x})$ be the regression function of Y over $\mathbf{X}$.
- Let $\hat{m}(\mathbf{x})$ an estimator (parametric, non-parametric, ...) of the regression function $m(\mathbf{x})$.
- We say that $\hat{m}(\mathbf{x})$ is a linear estimator when for any fix $\mathbf{x}$, $\hat{m}(\mathbf{x})$ is a linear function of $y_1, \ldots, y_n$:

$$\hat{m}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) y_i,$$

  where in fact $w_i(\mathbf{x}) = w_i(\mathbf{x}; \mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- For the n observed values $\mathbf{x}_i$ of the explanatory variable, let

$$\hat{y}_i = \hat{m}(\mathbf{x}_i) = \sum_{j=1}^{n} w_j(\mathbf{x}_i) y_j$$

  be the fitted values.

- In matrix format,

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{y},$$

where the column vectors $\mathbf{y}$ and $\hat{\mathbf{y}}$ have elements y_i and $\hat{y}_i$, respectively, and the matrix $\mathbf{W}$ has generic (i, j) element

$$w_{ij} = w_j(\mathbf{x}_i).$$

- The matrix $\mathbf{W}$ is analogous to the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$ in OLS estimation of the multiple linear regression:

$$\hat{\mathbf{y}}_{\text{OLS}} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \mathbf{H}\mathbf{y}.$$

- Observe that ridge regression is a linear estimation method:

$$\hat{\mathbf{y}}_{\text{ridge}} = \mathbf{X}\left(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \mathbf{H}_\lambda\mathbf{y}.$$

Effective number of parameters for linear estimators

- Consider the multiple linear regression with p regressors (including the constant term, if it appears in the model):

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon,$$

  $\mathbf{X}$ being a $n \times p$ matrix, $\boldsymbol{\beta} \in \mathbb{R}^p$.

- It is known that

$$\text{Trace}(\mathbf{H}) = \text{Trace}(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top) = \text{Trace}((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}) = \text{Trace}(\mathbf{I}_p) = p,$$

  that is the number of parameters in the model.

- For a linear estimator with matrix $\mathbf{W}$ ($\hat{\mathbf{y}} = \mathbf{W}\mathbf{y}$) we define

$$\nu = \text{Trace}(\mathbf{W}) = \sum_{i=1}^{n} w_{ii},$$

  the sum of diagonal elements of $\mathbf{W}$.

- $\nu = \text{Trace}(\mathbf{W})$ is called the effective number of parameters of the linear estimator corresponding to matrix $\mathbf{W}$.
- In some books (and softwares) ν is called effective degrees of freedom (df) of the regression estimator. This is the terminology used by Hastie, Tibshirani, and Friedman (2009) and Hastie, Tibshirani, and Wainwright (2015), and related packages.
- The interpretation of ν as the effective number of parameters is valid for any linear estimator of the regression function (parametric, nonparametric, ...).
- Then we can compare the degree of complexity of two linear estimators of a regression function just comparing their effective numbers of parameters.
- Usually a good estimator of σ^2, the residual variance, is

$$\hat{\sigma}^2 = \frac{1}{n-\nu}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Effective number of parameters in ridge regression

In the case of ridge regression $\nu = \nu(\lambda) = \mathrm{df}(\lambda)$ has an explicit expression:

$$\mathbf{W} = \mathbf{H}_\lambda = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1}\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T =$$

$$\mathbf{U}\mathbf{D}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1}\mathbf{D}\mathbf{U}^T = \mathbf{U}\left(\mathrm{Diagonal}(dj^2/(d_j^2 + \lambda), j = 1, \ldots, p\right)\mathbf{U}^T$$

$$\Rightarrow \nu(\lambda) = \mathrm{df}(\lambda) = \mathrm{Trace}(\mathbf{H}_\lambda) =$$

$$\mathrm{trace}(\mathbf{U}\left(\mathrm{Diagonal}(dj^2/(d_j^2 + \lambda), j = 1, \ldots, p\right)\mathbf{U}^T) =$$

$$\mathrm{trace}(\left(\mathrm{Diagonal}(dj^2/(d_j^2 + \lambda), j = 1, \ldots, p\right)\mathbf{U}^T\mathbf{U}) =$$

$$\mathrm{trace}(\left(\mathrm{Diagonal}(dj^2/(d_j^2 + \lambda), j = 1, \ldots, p\right)) = \sum_{j=1}^{p} \frac{dj^2}{d_j^2 + \lambda}.$$

$$\nu(\lambda) = \mathrm{df}(\lambda) = \sum_{j=1}^{p} \frac{dj^2}{d_j^2 + \lambda}$$

- $\lim_{\lambda \longrightarrow \infty} \mathrm{df}(\lambda) = 0, \quad \lim_{\lambda \longrightarrow 0} \mathrm{df}(\lambda) = \mathrm{rank}(\mathbf{X})$.
- The effective number of parameters $\nu(\lambda) = \mathrm{df}(\lambda)$ is a decreasing function of penalizing parameter λ:
  - Small values of λ correspond to large numbers ν of effective parameters, close to the number of linearly independent explanatory variables (usually $\min\{n, p\}$), allowing complex and flexible estimators.
  - Large values of λ correspond to small numbers ν of effective parameters, that is, to regression estimators with low complexity and flexibility.

Practice:

- Prostate data: Effective number of parameters in ridge regression.
- Use the R script `prostate.ridge.regression.R`.

Effective degrees of freedom for non-linear estimators

- Let $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$, be n i.i.d.r.v. distributed as $(\mathbf{X}, Y)$.
- Conditioning to $\mathbf{X}_i = \mathbf{x}_i$, $i = 1, \ldots, n$, it is equivalent to say

$$Y_i = m(\mathbf{x}_i) + \varepsilon_i,\ i = 1, \ldots, n,$$

  with $m(\mathbf{x}) = \mathbb{E}(Y|\mathbf{X} = \mathbf{x})$ and $\varepsilon_i = Y_i - m(\mathbf{x}_i)$, having $\mathbb{E}(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma^2$ for all i.
- Let $\hat{m}(\mathbf{x})$ an estimator of the regression function $m(\mathbf{x})$ (that is a random function because it is based on $(Y_1, \ldots, Y_n)$).
- Let $\hat{Y}_i = \hat{m}(\mathbf{x}_i)$.
- The effective degrees of freedom of $\hat{m}(\mathbf{x})$ is defined as

$$\text{df}(\hat{m}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{Cov}(\hat{Y}_i, Y_i).$$

Interpretation:

- A very flexible regression estimator $\hat{m}(\mathbf{x})$ will be able to interpolate the observed data, and then

  $$\hat{Y}_i = Y_i,\ \mathrm{Var}(\hat{Y}_i) = \mathrm{Var}(Y_i) = \sigma^2,\ \mathrm{Cov}(\hat{Y}_i, Y_i)/\sigma^2 = \mathrm{Cor}(\hat{Y}_i, Y_i) = 1,$$

  so $\mathrm{df}(\hat{m}) = n$: $\hat{m}(\mathbf{x})$ has as many degrees of freedom as the number of observed data.
- The constant function equal to the sample mean of $Y_1, \ldots, Y_n$ for all $\mathbf{x}$ has 1 degree of freedom.
- A function that is constant in $\mathbf{x}$ has 0 degrees of freedom if this constant does not depend on the data.

Both definitions of df coincide in linear estimators

Assume that $\hat{m}(\mathbf{x})$ is a linear estimator with matrix $\mathbf{W}$. Assume also that $\mathbb{E}(Y) = 0$. Then

$$\text{df}(\hat{m}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\text{Cov}(\hat{Y}_i, Y_i) = \frac{1}{\sigma^2}\text{Trace}\left(\text{Cov}(\hat{\mathbf{Y}}, \mathbf{Y})\right) =$$

$$\frac{1}{\sigma^2}\text{Trace}\left(\mathbb{E}\left(\hat{\mathbf{Y}}\mathbf{Y}^{\mathsf{T}}\right)\right) = \frac{1}{\sigma^2}\text{Trace}\left(\mathbb{E}\left(\mathbf{W}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\right)\right) =$$

$$\frac{1}{\sigma^2}\text{Trace}\left(\mathbf{W}\mathbb{E}\left(\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\right)\right) = \frac{1}{\sigma^2}\text{Trace}\left(\mathbf{W}\sigma^2\mathbf{I}_p\right) = \text{Trace}(\mathbf{W}).$$

Choosing the tunning parameter λ

- The tunning parameter λ can be chosen by cross-validation (CV), k-fold cross-validation (k-fold CV) or by generalized cross-validation (GCV).
- Given the expression of $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ (linear in $\mathbf{y}$) CV and GCV are not computationally expensive.
- We will first introduce these concepts before talking about efficient computation.

- Predictive Mean Square Error (PMSE). It is the expected squared error made when predicting

$$Y = m(\mathbf{x}) + \varepsilon$$

by $\hat{m}(\mathbf{x})$, where $\mathbf{x}$ is an observation of the random variable $\mathbf{X}$, distributed as the observed explanatory variable, when $\mathbf{X}$ and ε are independent from the sample $\mathcal{Z} = \{(\mathbf{X}_i, Y_i) : i = 1, \ldots, n\}$ used to compute $\hat{m}$:

$$\text{PMSE}(\hat{m}) = E_{\mathcal{Z},\mathbf{X},\varepsilon}\left[(Y - \hat{m}(\mathbf{X}))^2\right].$$

- PMSE is a particular case of expected loss: $E_{\mathcal{Z},\mathbf{X},\varepsilon}\left[L(Y, \hat{m}(\mathbf{X}))\right]$, where $L(y, \hat{y})$ is the loss of predicting the value y by $\hat{y}$.
- Other examples of loss functions: $|y - \hat{y}|$, $\mathbb{I}\{y \neq \hat{y}\}$.

Prediction error in a validation set

- When the number of available data is large (as it usually happens in data mining or in Big Data problems) the sample is randomly divided in three sets:
  - The training set: it is used to fit the model.
  - The validation set: it is used to compute feasible versions of the above criteria for model selection and/or parameter tuning.
  - The test set: it is used to evaluate the generalization (or prediction) error of the final chosen model in independent data.
- Assuming that at least a validation set has been preserved, an estimation of PMSE is the Predictive Mean Squared Error in the validation set:

$$\text{PMSE}_{\text{val}}(\hat{m}) = \frac{1}{n_V}\sum_{i=1}^{n_V}(y_i^V - \hat{m}(\mathbf{x}_i^V))^2,$$

where $(\mathbf{x}_i^V, y_i^V)$, $i = 1, \ldots, n_V$, is the validation set and $\hat{m}(\mathbf{x})$ is the estimator computed using the training set.

Leave-one-out cross-validation

- When the sample size does not allow us to set a validation set aside, leave-one-out cross-validation is an attractive alternative:
  1. Remove the observation $(\mathbf{x}_i, y_i)$ from the sample and fit the regression using the other $(n-1)$ data. Let $\hat{m}_{(i)}(\mathbf{x})$ be the resulting estimator.
  2. Now use $\hat{m}_{(i)}(\mathbf{x}_i)$ to predict y_i.
  3. Repeat the previous steps for $i = 1, \ldots, n$.
  4. Compute

$$\text{PMSE}_{\text{CV}}(\hat{m}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{m}_{(i)}(\mathbf{x}_i))^2.$$

- In ridge regression:

$$\lambda_{CV} = \arg\min_{\lambda \geq 0} \text{PMSE}_{\text{CV}}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^{\mathsf{T}}\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda})^2.$$

Practice:

- Prostate data: Leave-one-out cross-validation in ridge regression.
- Use the R script `prostate.ridge.regression.R`.

k-fold cross validation

- $\text{PMSE}_{\text{CV}}(\hat{m})$ is an approximately unbiased estimator of $\text{PMSE}(\hat{m})$, but has a considerable variance.
- The variance can be reduced doing k-fold cross-validation: The sample is randomly divided in k subsets, each of them is removed by turns from the sample, the model is estimated with the other $(k-1)$ subsamples and the removed subsample is used to compute prediction errors.
- n-fold cross-validation is leave-one-out cross-validation.
- k-fold cross-validation has lower variance than leave-one-out cross-validation but larger bias.
- General recommendation: Use 5-fold or 10-fold cross-validation.

Efficient computation of PMSE_{CV}

- Consider a linear estimator of the regression function with matrix $\mathbf{W} = (w_{ij})_{i,j}$: $\hat{\mathbf{y}} = \mathbf{W}\mathbf{y}$.
- That is

$$\hat{y}_i = \sum_{j=1}^{n} w_{ij} y_j, \ \ i = 1, \ldots, n,$$

  where $w_{ij} = w_j(\mathbf{x}_i) = w_j(\mathbf{x}_i; \mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- In these cases PMSE_{CV} can be calculated avoiding the computational cost of fitting n different regression models.
- For most linear estimators it can be proved that

$$\text{PMSE}_{\text{CV}} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - w_{ii}} \right)^2 .$$

Proof for the ridge regression estimation

- Let $\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda}$ be the estimation of $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ when leaving out the i-th observation:

$$\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{l=1, l\neq i}^{n} \left(y_l - \sum_{j=1}^{p} x_{lj}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\}$$

- Let us define

$$\tilde{y}_l^{(i)} = \begin{cases} y_l & \text{if} \quad l \neq i, \\ \hat{y}_i^{(i)} = \sum_{j=1}^{p} x_{ij} \hat{\beta}^{(i)}_{\text{ridge},\lambda,j} & \text{if} \quad l = i. \end{cases}$$

- It follows that for all $\boldsymbol{\beta} \in \mathbb{R}^p$,

$$\sum_{l=1}^{n} \left(\tilde{y}_l^{(i)} - \sum_{j=1}^{p} x_{lj}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 = \left\{ \sum_{l=1, l\neq i}^{n} \left(y_l - \sum_{j=1}^{p} x_{lj}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\} + \left(\sum_{j=1}^{p} x_{lj} (\hat{\beta}^{(i)}_{\text{ridge},\lambda,j} - \beta_j) \right)^2$$

- Observe that $\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda}$ minimizes both terms in the right hand side. Then it is also

$$\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{l=1}^{n} \left(\tilde{y}_l^{(i)} - \sum_{j=1}^{p} x_{lj}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\}$$

- This is the ridge regression estimator corresponding to a data set with matrix of explanatory variables $\mathbf{X}$ and vector of responses $\tilde{\mathbf{y}}^{(i)} = (\tilde{y}_1^{(i)}, \ldots, \tilde{y}_n^{(i)})^{\mathsf{T}}$.

- Then

$$\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}^{\mathsf{T}}\tilde{\mathbf{y}}^{(i)},$$

$$\hat{\tilde{\mathbf{y}}}^{(i)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(i)}_{\text{ridge},\lambda} = \mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}^{\mathsf{T}}\tilde{\mathbf{y}}^{(i)} = \mathbf{H}_\lambda \tilde{\mathbf{y}}^{(i)}.$$

- Observe that the i-th element of $\hat{\tilde{\mathbf{y}}}^{(i)}$ is just $\hat{y}_i^{(i)} = \sum_{j=1}^{p} x_{ij}\hat{\beta}^{(i)}_{\text{ridge},\lambda,j}$.
- Let $\mathbf{e}_i$ be the n-dimensional vector whose i-th element is 1 and the others are equal to 0.
- Then $\tilde{\mathbf{y}}^{(i)} = \mathbf{y} - (y_i - \hat{y}_i^{(i)})\mathbf{e}_i$ and, consequently,

$$\hat{\tilde{\mathbf{y}}}^{(i)} = \mathbf{H}_\lambda \tilde{\mathbf{y}}^{(i)} = \mathbf{H}_\lambda \left(\mathbf{y} - (y_i - \hat{y}_i^{(i)})\mathbf{e}_i\right) = \mathbf{H}_\lambda \mathbf{y} - (y_i - \hat{y}_i^{(i)})\mathbf{H}_\lambda \mathbf{e}_i = \hat{\mathbf{y}} - (y_i - \hat{y}_i^{(i)})\mathbf{h}_i^\lambda,$$

  where $\mathbf{h}_i^\lambda$ is the i-th column of $\mathbf{H}_\lambda$.
- Looking just at the i-th component, $\hat{y}_i^{(i)} = \hat{y}_i - (y_i - \hat{y}_i^{(i)})h_{ii}^\lambda$, where h_{ii}^λ is the element (i,i) of $\mathbf{H}_\lambda$, or the i-th element in the diagonal of $\mathbf{H}_\lambda$.
- Then $y_i - \hat{y}_i^{(i)} = y_i - \hat{y}_i + (y_i - \hat{y}_i^{(i)})h_{ii}^\lambda$ and we conclude that

$$y_i - \hat{y}_i^{(i)} = \frac{y_i - \hat{y}_i}{1 - h_{ii}^\lambda}.$$

  So the loo-CV errors $(y_i - \hat{y}_i^{(i)})$ can be computed if we know the errors $(y_i - \hat{y}_i)$ when fitting the ridge regression with all the data, and the diagonal of $\mathbf{H}_\lambda$, and the proof concludes.

Practice:

- Prostate data: Efficient computation of PMSE_{CV} in ridge regression.
- Use the R script `prostate.ridge.regression.R`.

Generalized cross-validation

- For linear estimators of the regression function, a modification can be done in the measure of PMSE_{CV}.
- It is known as generalized cross-validation (GCV).
- It consists in replacing in the expression of PMSE_{CV} the values w_{ii}, coming from the diagonal of $\mathbf{W}$, by their average value:

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

  $\nu = \text{Trace}(\mathbf{W}) = \sum_{i=1}^{n} w_{ii}$ is the effective number of parameters.
- In ridge regression, $\lambda_{\text{GCV}} = \arg\min_\lambda \text{PMSE}_{\text{GCV}}(\lambda)$.
- Manipulating the expression of PMSE_{GCV} it follows that

$$\text{PMSE}_{\text{GCV}} = \frac{n\hat{\sigma}^2_\varepsilon}{n-\nu},$$

  where $\hat{\sigma}^2_\varepsilon = \frac{1}{n-\nu}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ estimates the residual variance.

Practice:

- Prostate data: PMSE_{GCV} in ridge regression.
- Use the R script

  `prostate.ridge.regression.R`.

Variance of the ridge regression estimator

Remember that $\hat{\boldsymbol{\beta}}_{\text{ridge}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y}$. Then

$$\begin{aligned}\text{Var}(\hat{\boldsymbol{\beta}}_{\text{ridge}}) &= \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\text{Var}(\mathbf{y})\mathbf{X} \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \\ &= \sigma^2 \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{X} \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1}.\end{aligned}$$

From the s.v.d. of $\mathbf{X}$, $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$, we have deduced that $\mathbf{X}^{\mathsf{T}}\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathsf{T}}$ and that $\left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} = \mathbf{V}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1}\mathbf{V}^{\mathsf{T}}$ Therefore,

$$\begin{aligned}\text{Var}(\hat{\boldsymbol{\beta}}_{\text{ridge}}) &= \sigma^2 \mathbf{V}\left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1} \mathbf{D}^2 \left(\mathbf{D}^2 + \lambda \mathbf{I}_p\right)^{-1} \mathbf{V}^{\mathsf{T}} \\ &= \sigma^2 \mathbf{V}\,\text{Diagonal}\left(d_j^2/(d_j^2 + \lambda)^2, j = 1, \ldots, p\right) \mathbf{V}^{\mathsf{T}}.\end{aligned}$$

The Lasso estimation (Tibshirani 1996)

- Lasso: Least absolute shrinkage and selection operator.
- The Lasso, also a shrinkage method, uses the norm L_1 as penalty term:

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

- Alternative expression:

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

- $t = s\|\hat{\boldsymbol{\beta}}_{OLS}\|_{\ell_1}$, $s \in [0, 1]$. s: shrinkage factor.

Lasso gives sparse solutions

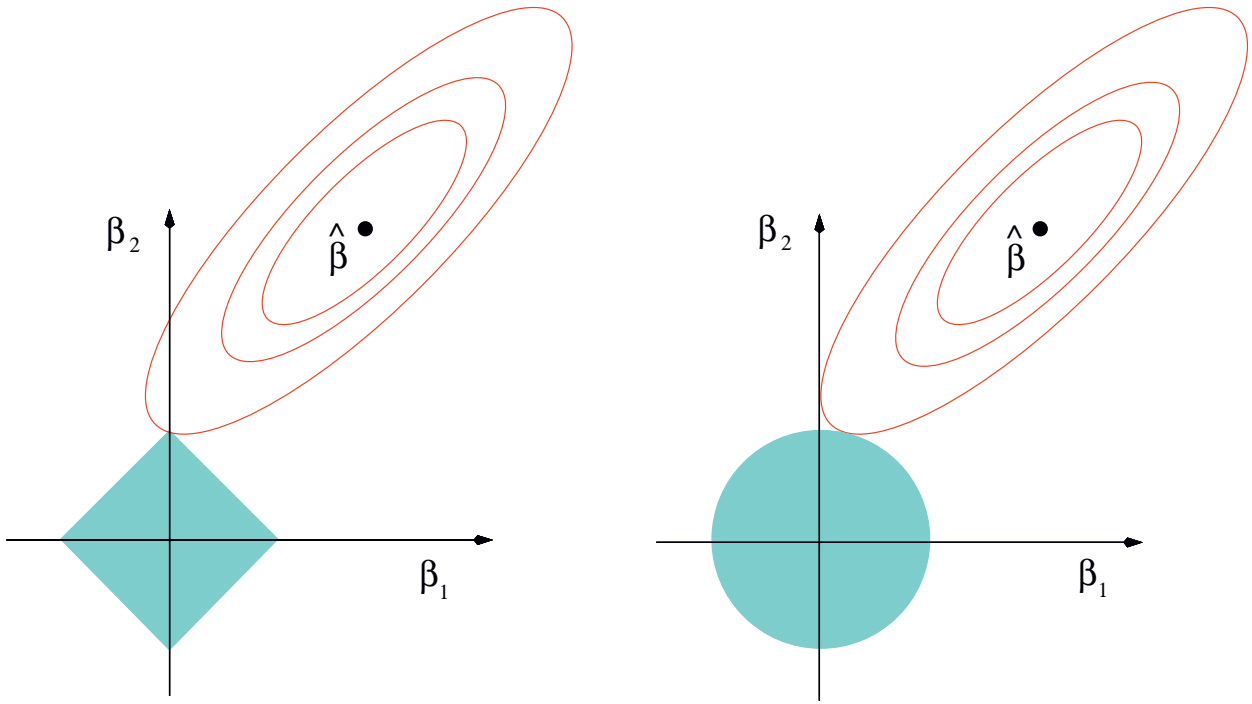
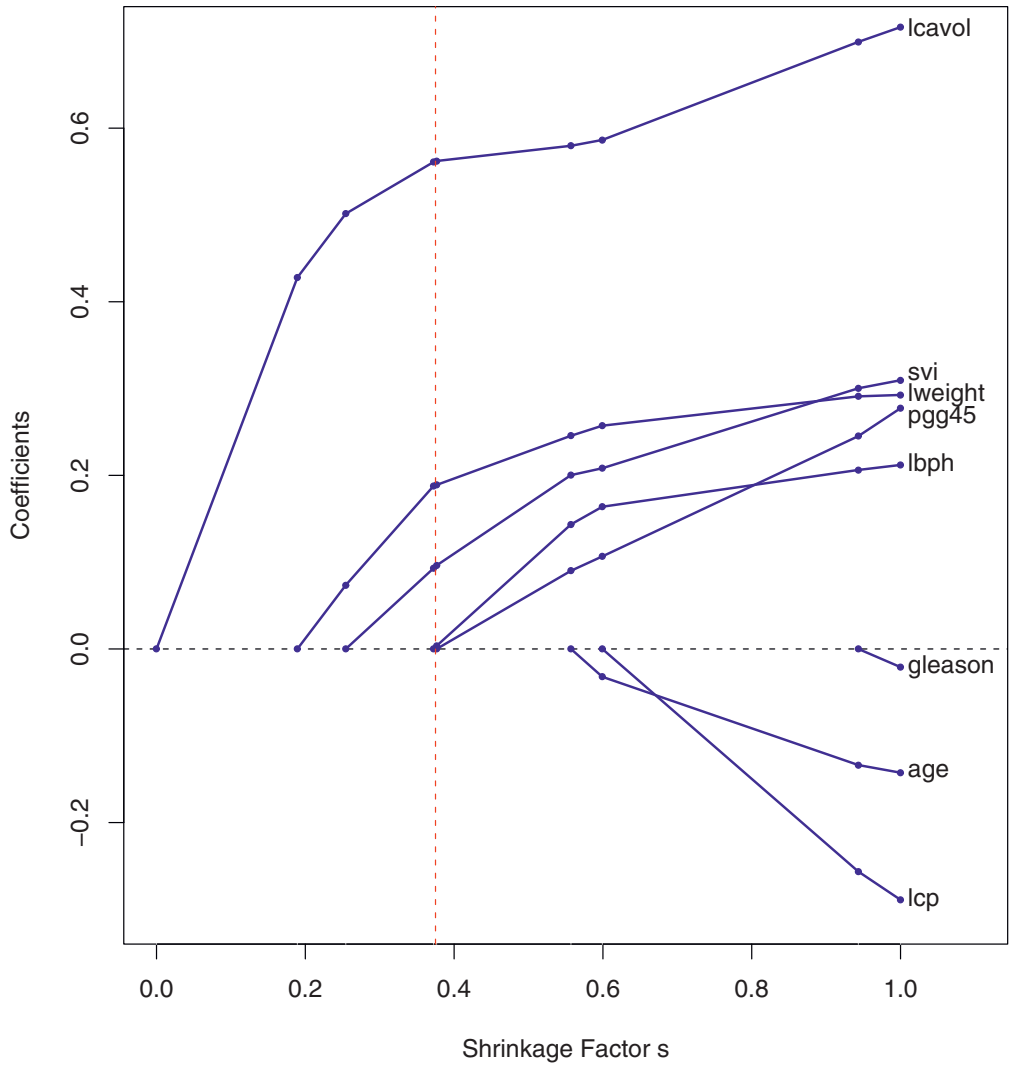


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Source: Hastie, Tibshirani, and Friedman (2009)

Prostate cancer example. Lasso



Source: Hastie, Tibshirani, and Friedman (2009)

Lasso: Properties

- Lasso provides sparse solutions.
- Lasso enables estimation and variable selection simultaneously in one stage.
- No closed expression for the Lasso estimator.
- Lasso involves a convex optimization problem (convex quadratic objective function, convex feasible region)

$$\begin{aligned} &\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta})^2 \\ &\text{s.t. } \|\boldsymbol{\beta}\|_{\ell_1} \leq t \end{aligned}$$

that can be efficiently solved.

Lasso and ℓ_q norms

- For $q > 0$, ℓ_q norm of $\boldsymbol{\beta} \in \mathbb{R}^p$: $\|\boldsymbol{\beta}\|_{\ell_q} = \left(\sum_{j=1}^p |\beta_j|^q\right)^{1/q}$.
- $\|\boldsymbol{\beta}\|_{\ell_\infty} = \lim_{q \longrightarrow \infty} \|\boldsymbol{\beta}\|_{\ell_q} = \max_{j=1,\ldots,p} |\beta_j|$.
- Defining $0^0 = 0$, $\|\boldsymbol{\beta}\|_{\ell_0} = \sum_{j=1}^p |\beta_j|^0$, the ℓ_0 "norm" of $\boldsymbol{\beta}$ is the number of non-zero entries of $\boldsymbol{\beta}$. This is not a real norm ($\|a\boldsymbol{\beta}\|_{\ell_0} \neq |a|\|\boldsymbol{\beta}\|_{\ell_0}$ for scalars $a \notin \{-1, 0, 1\}$).

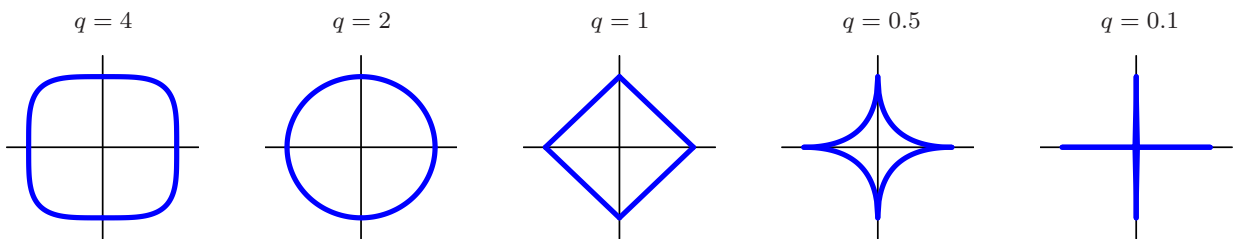


Figure 2.6 *Constraint regions $\sum_{j=1}^p |\beta_j|^q \leq 1$ for different values of q. For $q < 1$, the constraint region is nonconvex.*

Source: Hastie, Tibshirani, and Wainwright (2015)

- Lasso is between the best subset selection (a combinatorial problem) and the ridge regression:

Best subset selection	Lasso	Ridge regression
$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})^2$	$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})^2$	$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})^2$
s.t. $\|\boldsymbol{\beta}\|_{\ell_0} \leq t$	s.t. $\|\boldsymbol{\beta}\|_{\ell_1} \leq t$	s.t. $\|\boldsymbol{\beta}\|_{\ell_2} \leq t$

- The Lasso problem (ℓ_1-penalty) uses the smallest value of q that leads to a convex constraint region.
- In this sense, it is the closest convex relaxation of the best subset selection problem (ℓ_0), among those based on ℓ_q-penalties, $q \geq 0$.

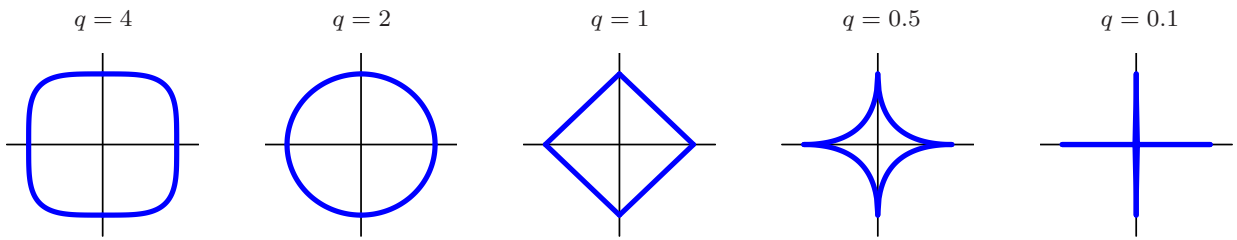


Figure 2.6 *Constraint regions $\sum_{j=1}^p |\beta_j|^q \leq 1$ for different values of q. For $q < 1$, the constraint region is nonconvex.*

Source: Hastie, Tibshirani, and Wainwright (2015)

Lasso: A retrospective (Tibshirani 2011)

- After publication, Tibshirani (1996) did not receive much attention until years later.
- Why? In 2011, Tibshirani's guesses were that
  (a) the computation in 1996 was slow compared with today,
  (b) the algorithms for the Lasso were black boxes and not statistically motivated (until the LARS (least angle regression) algorithm in 2002),
  (c) the statistical and numerical advantages of sparsity were not immediately appreciated (by Tibshirani or the community),
  (d) large data problems (in N, p or both) were rare and
  (e) the community did not have the R language for fast, easy sharing of new software tools.

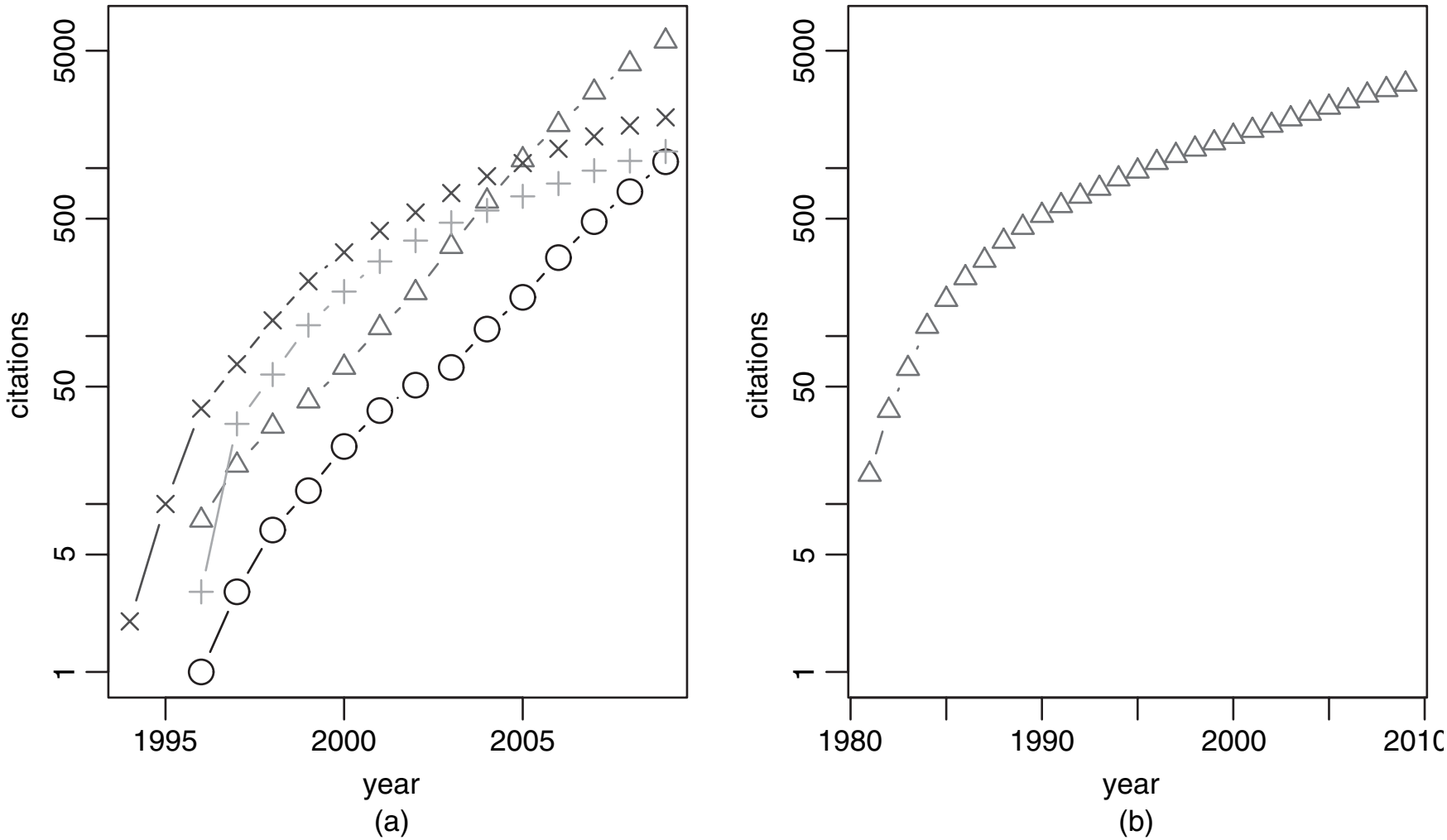


Fig. 2. Cumulative citation counts (on a log-scale) from the Thomson ISI *Web of Knowledge* (the largest abscissa on the x-axis corresponds to August 31st, 2010): (a) the lasso (○) (Tibshirani, 1996), false discovery rate (Δ) (Benjamini and Hochberg, 1995), reversible jump Markov chain Monte Carlo sampling (+) (Green, 1995) and wavelet shrinkage (×) (Donoho and Johnstone, 1994), published between 1994 and 1996; (b) the bootstrap (Δ) (Efron, 1979), published earlier

Source: Peter Bülhmann's comments to Tibshirani (2011)

From Peter Bülhmann's comments to Tibshirani (2011)

[The previous Figure] shows that [Lasso] frequency of citation continues to be in the exponential growth regime, together with the false discovery rate paper from Benjamini and Hochberg (1995): both of these works are crucial for high dimensional statistical inference.

Computation of Lasso

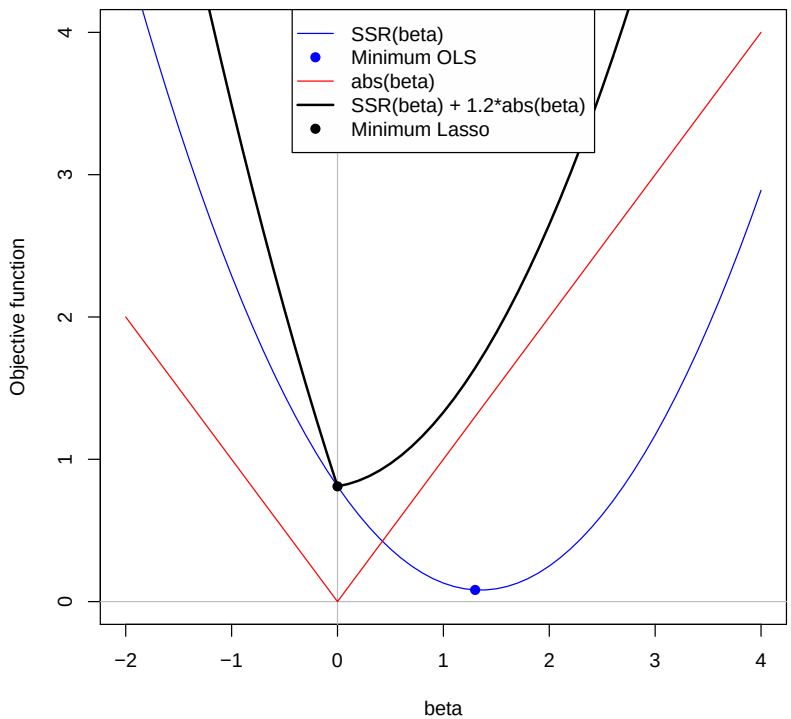
- The original Lasso paper used a standard quadratic program solver.
- This does not scale well and is not transparent.
- The LARS algorithm (Efron, Hastie, Johnstone, Tibshirani, et al. 2004) gives an efficient way of solving the Lasso and connects the Lasso to forward stagewise regression.
- Later on, a cyclic coordinate descent algorithm replaced LARS and, since Friedman, Hastie, and Tibshirani (2010) the `glmnet` R package implements this algorithm.
- Cyclic coordinate descent algorithm:
  - First we will see that Lasso has a closed form solution when $p = 1$ (single predictor case).
  - Them we will give the co-ordinate descent algorithm for a generic p.

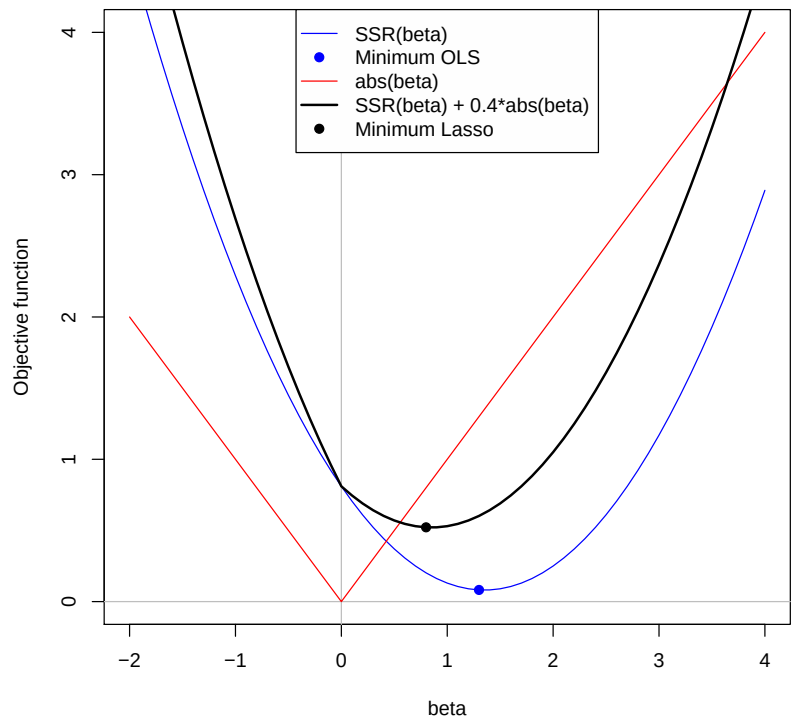
Single predictor. Soft thresholding function

- We observe (x_i, y_i), $i = 1, \ldots, n$, $x_i \in \mathbb{R}$, $y_i \in \mathbb{R}$, and assume

$$\sum_{i=1}^{n} x_i = 0, \ \frac{1}{n}\sum_{i=1}^{n} x_i^2 = 1, \ \sum_{i=1}^{n} y_i = 0 \Rightarrow \hat{\beta}_{\text{OLS}} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i = \frac{1}{n}\langle \mathbf{x}, \mathbf{y} \rangle.$$

- Consider the Lasso problem $\min_{\beta \in \mathbb{R}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda|\beta| \right\}$.





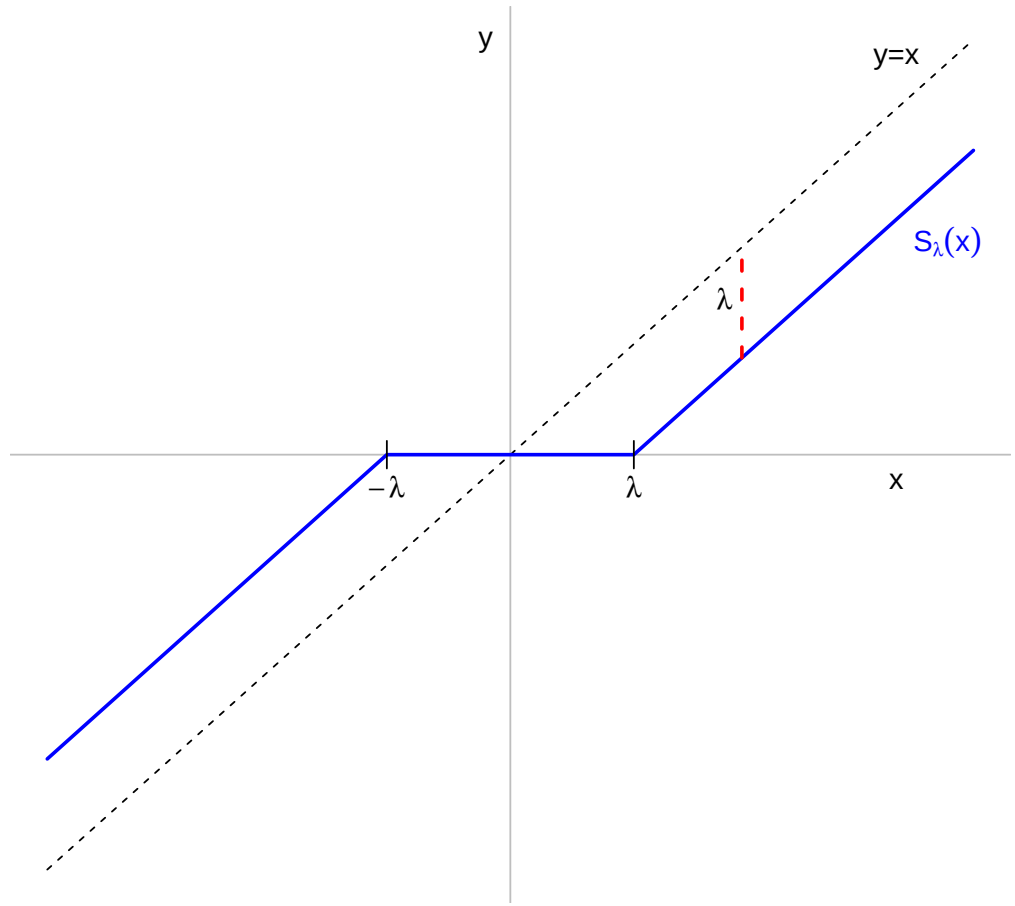
Let $f(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta|$. Then,

$$f'(\beta) = \begin{cases} -\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\beta)x_i + \lambda = -\frac{1}{n}\langle \mathbf{x}, \mathbf{y}\rangle + \beta + \lambda & \text{if} \quad \beta > 0, \\ -\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\beta)x_i - \lambda = -\frac{1}{n}\langle \mathbf{x}, \mathbf{y}\rangle + \beta - \lambda & \text{if} \quad \beta < 0. \end{cases}$$

- If $\hat{\beta}_{\text{OLS}} = \frac{1}{n}\langle \mathbf{x}, \mathbf{y}\rangle \geq 0$ then:
  - $f'(\beta) < 0$ for all $\beta < 0$,
  - $f'(\beta) < 0$ for $\beta \in (0, \max\{0, \hat{\beta}_{\text{OLS}} - \lambda\})$,
  - $f'(\beta) > 0$ for $\beta > \max\{0, \hat{\beta}_{\text{OLS}} - \lambda\}$.
  - Therefore $\hat{\beta}_{\text{Lasso}} = \max\{0, \hat{\beta}_{\text{OLS}} - \lambda\}$.
- If $\hat{\beta}_{\text{OLS}} = \frac{1}{n}\langle \mathbf{x}, \mathbf{y}\rangle < 0$ then: $f'(\beta) > 0$ for all $\beta > 0$.
  - $f'(\beta) > 0$ for all $\beta > 0$,
  - $f'(\beta) > 0$ for $\beta \in (\min\{0, \hat{\beta}_{\text{OLS}} + \lambda\}, 0)$,
  - $f'(\beta) < 0$ for $\beta < \min\{0, \hat{\beta}_{\text{OLS}} + \lambda\}$.
  - Therefore $\hat{\beta}_{\text{Lasso}} = \min\{0, \hat{\beta}_{\text{OLS}} + \lambda\} = -\max\{0, -\hat{\beta}_{\text{OLS}} - \lambda\}$.
- $\hat{\beta}_{\text{Lasso}} = \text{sign}(\hat{\beta}_{\text{OLS}})\max\{0, |\hat{\beta}_{\text{OLS}}| - \lambda\}$.

Soft-thresholding operator

- For $x \in \mathbb{R}$ let $x_+ = \max\{0, x\}$ its positive part.
- For $\lambda > 0$ we define the Soft-thresholding operator $\mathcal{S}_\lambda(x) = \text{sign}(x)\,(|x| - \lambda)_+$.
- Then, in the single predictor case, $\hat{\beta}_{\text{Lasso}} = \mathcal{S}_\lambda(\hat{\beta}_{\text{OLS}})$.



Multiple predictors: Cyclic coordinate descent

- When there are p predictors, the Lasso objective function, to be minimized in $\beta \in \mathbb{R}^p$, is

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

- It has the additive decomposition

$$f(\beta_1,\ldots,\beta_p) = g(\beta_1,\ldots,\beta_p) + \sum_{j=1}^{p} h_j(\beta_j)$$

where g is differentiable and convex, and the univariate functions h_j are convex (but not differentiable), then the cyclic coordinate descent algorithm converges to the global minimum of f. (Hastie, Tibshirani, and Wainwright 2015, Section 5.4.1, for references)

- The cyclic coordinate descent algorithm repeatedly cycle through the predictors in fixed order (say $1, \ldots, p$) the minimization in one coordinate (say the j-th) fixing the others in the last available values for them (say $\hat{\beta}_k$, $k \neq j$):

$$\min_{\beta_j \in \mathbb{R}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k - x_{ij} \beta_j \right)^2 + \lambda \sum_{k \neq j} |\hat{\beta}_k| + \lambda |\beta_j| \right\}.$$

- Define the partial residuals $r_i^{(j)} = y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k$.
- Then, the optimal value for β_j is (with obvious notation) $\hat{\beta}_j^{\text{new}} = \mathcal{S}_\lambda \left(\frac{1}{n} \langle \mathbf{x}_j, \mathbf{r}^{(j)} \rangle \right)$.
- Let $\hat{\beta}_j$ be the last available estimation for β_j before computing $\hat{\beta}_j^{\text{new}}$ and let $r_i = y_i - \sum_{k=1}^{n} x_{ik} \hat{\beta}_k$ be the previous full residuals. Then $\mathbf{r}^{(j)} = \mathbf{r} + \hat{\beta}_j \mathbf{x}_j$ and $\frac{1}{n} \langle \mathbf{x}_j, \mathbf{r}^{(j)} \rangle = \frac{1}{n} \langle \mathbf{x}_j, \mathbf{r} \rangle + \hat{\beta}_j \frac{1}{n} \langle \mathbf{x}_j, \mathbf{x}_j \rangle = \frac{1}{n} \langle \mathbf{x}_j, \mathbf{r} \rangle + \hat{\beta}_j$.
- Then $\hat{\beta}_j^{\text{new}} = \mathcal{S}_\lambda \left(\hat{\beta}_j + \frac{1}{n} \langle \mathbf{x}_j, \mathbf{r} \rangle \right)$.
- And the new full residuals are $\mathbf{r}^{\text{new}} = \mathbf{r} - \left(\hat{\beta}_j^{\text{new}} - \hat{\beta}_j \right) \mathbf{x}_j$.

Practice:

- Prostate data: Lasso estimation for a given λ.
- Use the R script `prostate.lasso.R`.

Solutions path and warm starts

- Typically one want a sequence of Lasso solutions, corresponding to $\lambda_0, \ldots, \lambda_L = 0$.
- The largest value of λ given a non-zeros solution is

$$\lambda_0 = \frac{1}{n} \max_j |\langle \mathbf{y}, \mathbf{x}_j \rangle|,$$

  because for $\lambda > \lambda_0$ the cyclic coordinate descent algorithm has $\boldsymbol{\beta} = \mathbf{0}$ as the only fixed point.
- Warm start: The solution $\hat{\boldsymbol{\beta}}(\lambda_l)$ is the initial value (warm start) for the algorithm when looking for the solution $\hat{\boldsymbol{\beta}}(\lambda_{l+1})$, $l = 1, \ldots, L-1$.
- Usually $L = 100$ is enough and $\lambda_0, \ldots, \lambda_{L-1}$ are evenly spaced.
- Active set for λ: The set of coefficients $\beta_1, \ldots, \beta_p$ that are non-zero for a given value of λ.
- Monitoring the active sets when going from λ_l to λ_{l+1} allows to improve algorithmic efficiency.

Practice:

- Prostate data: Lasso estimation and *coefficients path*.
- Use the R script `prostate.lasso.R`.

Effective degrees of freedom for Lasso (I)

- Lasso is not a linear estimator of the regression function.
- Let $(x_{i1}, \ldots, x_{ip}, Y_i)$, $i = 1, \ldots, n$, be n data following a multiple linear regression model with residual variance σ^2.
- For $\lambda > 0$, let $\hat{Y}_i^\lambda$, $i = 1, \ldots, n$, be the fitted values resulting from the Lasso estimation using penalization parameter λ.
- The effective degrees of freedom of the Lasso estimator when using penalization parameter λ is defined as

$$\mathrm{df}(\lambda) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(\hat{Y}_i^\lambda, Y_i).$$

Effective degrees of freedom for Lasso (II)

- Let $k_\lambda = \|\hat{\beta}^\lambda\|_{\ell_0}$ be the number of non-zero estimated coefficients when using λ.
- Observe that k_λ is a random variable.
- It can be proved that k_λ is an unbiased estimator of $\text{df}(\lambda)$.
- A flexibility trade-off in Lasso:
  - A Lasso estimator with k non-zero coefficients should have more flexibility than a OLS estimator using just k variables fixed in advance, because Lasso select the *best* (in some sense) subset of k variables.
  - But the Lasso estimation of these k coefficient is less flexible than the OLS estimation because the penalization term shrinks the estimated coefficient toward zero, relative to the usual OLS estimates.
  - Both terms compensate each other and, in average, the number of nonzero coefficients estimates $\text{df}(\lambda)$ with no bias.

Lasso: Statistical properties

(Based on Bülhmann's comments to Tibshirani 2011. See also Chapters 6 and 11 of Hastie, Tibshirani, and Wainwright 2015 or the book Bühlmann and van de Geer 2011)

Consider a potentially high dimensional linear model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon,\ \mathbf{X}_{n\times p},\ p = p_n \gg n \text{ as } n \longrightarrow \infty.$$

Four problems have received much attention:

- Prediction and estimation of the regression surface $\mathbf{X}\boldsymbol{\beta}$.
- Estimation of parameters $\boldsymbol{\beta}$.
- Variable screening or *Sparsistency*.
- P-values for high-dimensional linear models.

Prediction and estimation of the regression surface

- For fixed design, under no assumptions on $\mathbf{X}$ and mild conditions on ε, it can be proved that

$$\frac{1}{n}\|\mathbf{X}(\hat{\boldsymbol{\beta}}_{\text{Lasso}} - \boldsymbol{\beta})\|_2^2 \leq \|\boldsymbol{\beta}\|_1 O_P(\sqrt{\log p/n}).$$

- Achieving a faster rate of convergence for prediction requires a design condition such as the restricted ℓ_1-eigenvalue assumption:

$$\frac{\frac{1}{n}\nu\mathbf{X}^{\mathsf{T}}\mathbf{X}\nu^{\mathsf{T}}}{\|\nu\|_{\ell_2}^2} \geq \gamma \text{ for all nonzero } \nu \in \mathcal{C}(S_0, 3),$$

for $\gamma > 0$, where $S_0 = \{j : \beta_j \neq 0\}$ is the active variables set and

$$\mathcal{C}(S_0, \alpha) = \{\nu \in \mathbb{R}^p : \|\nu_{S_0^c}\|_{\ell_1} \leq \alpha\|\nu_{S_0}\|_{\ell_1}\}.$$

Estimation of parameters $\boldsymbol{\beta}$

- Active variables set: $S_0 = \{j : \beta_j \neq 0\}$, $s_0 = |S_0|$.
- Under the restricted ℓ_1-eigenvalue assumption, Bühlmann and van de Geer (2011) prove that, with high probability,

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1 \leq O_P(s_0 \sqrt{\log p / n}).$$

- Then $\boldsymbol{\beta}$ is identifiable if $s_0 \leq \sqrt{n / \log p}$, that is, if the true model is sparse.

Variable screening or Sparsistency

- Active variables set: $S_0 = \{j : \beta_j \neq 0\}$. Let $\hat{S} = \{j : \hat{\beta}_j^{\text{Lasso}} \neq 0\}$.
- In order to have asymptotically perfect variable selection,

$$\lim_n \Pr(\hat{S} = S_0) = 1,$$

some restrictive (and rather unlikely to hold in practice!) assumptions must be made, that are sufficient and (essentially) necessary.

- What happens with high probability under no such restrictive conditions is that

$$\lim_n \Pr(\hat{S} \supseteq S_{\text{relev}}) = 1,$$

where S_{relev} is the set of coefficients that are *relevant* in the sense that they are far from 0.

- This result is still valid when the λ (or t) is chosen by CV.

P-values for high-dimensional linear models

- Asymptotic distribution of Lasso estimators has a point mass at zero.
- Standard bootstrap cannot be used.
- Peter Bülhmann and co-authors propose de-sparsifying the Lasso estimator. They prove the asymptotic normality of the de-sparsified estiamtors.
- Finally, Lockhart, Taylor, Tibshirani, Tibshirani, et al. (2014) test the significance of the predictor variable that enters the current Lasso model, in the sequence of models visited along the Lasso solution path.

Lasso: A very active research area

Table 1. A sampling of generalizations of the lasso

Method	*Reference*	*Detail*
Grouped lasso	Yuan and Lin (2007a)	$\Sigma_g \|\boldsymbol{\beta}_g\|_2$
Elastic net	Zou and Hastie (2005)	$\lambda_1 \Sigma \|\beta_j\| + \lambda_2 \Sigma \beta_j^2$
Fused lasso	Tibshirani *et al.* (2005)	$\lambda \Sigma \|\beta_{j+1} - \beta_j\|$
Adaptive lasso	Zou (2006)	$\lambda_1 \Sigma w_j \|\beta_j\|$
Graphical lasso	Yuan and Lin (2007b); Friedman *et al.* (2007)	$\text{loglik} + \lambda \|\Sigma^{-1}\|_1$
Dantzig selector	Candes and Tao (2007)	$\min\{X^{\mathrm{T}}(y - X\beta)\|_\infty\} \|\beta\|_1 < t$
Near isotonic regularization	Tibshirani *et al.* (2010)	$\Sigma(\beta_j - \beta_{j+1})_+$
Matrix completion	Candès and Tao (2009); Mazumder *et al.* (2010)	$\|X - \hat{X}\|^2 + \lambda \|\hat{X}\|_*$
Compressive sensing	Donoho (2004); Candes (2006)	$\min(\|\beta\|_1)$ subject to $y = X\beta$
Multivariate methods	Jolliffe *et al.* (2003); Witten *et al.* (2009)	Sparse principal components analysis, linear discriminant analysis and canonical correlation analysis

Source: Tibshirani (2011)

glmnet package in R

(See the *Glmnet vignette*, Hastie and Qian (2014))

- Glmnet is a package that fits a generalized linear model via penalized maximum likelihood, using the Lasso or elasticnet penalty.
- The authors of glmnet are Jerome Friedman, Trevor Hastie, Rob Tibshirani and Noah Simon.
- The algorithm is extremely fast, and can exploit sparsity in the input matrix $\mathbf{X}$.
- It fits linear, logistic and multinomial, Poisson, and Cox regression models.
- It can also fit multi-response linear regression.

glmnet package in R (II)

- glmnet solves the following problem

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{N}\sum_{i=1}^{N} w_i \ell(y_i, \beta_0 + \boldsymbol{\beta}^T x_i) + \lambda \left[(1-\alpha)||\boldsymbol{\beta}||_2^2/2 + \alpha||\boldsymbol{\beta}||_1\right],$$

  over a grid of values of λ covering the entire range.
- Here $\ell(y, \eta)$ is the negative log-likelihood contribution for observation i; e.g. for the Gaussian case it is $(1/2)(y-\eta)^2$.
- The elastic-net penalty is controlled by α, and bridges the gap between Lasso ($\alpha = 1$, the default) and ridge ($\alpha = 0$).
- The tuning parameter λ controls the overall strength of the penalty.

glmnet package in R (III)

- It is known that the ridge penalty shrinks the coefficients of correlated predictors towards each other while the Lasso tends to pick one of them and discard the others.
- The elastic-net penalty mixes these two; if predictors are correlated in groups, an $\alpha = 0.5$ tends to select the groups in or out together.
- One use of α is for numerical stability; for example, the elastic net with $\alpha = 1 - \epsilon$ for some $\epsilon > 0$ performs much like the Lasso, but removes any degeneracies and wild behavior caused by extreme correlations.

glmnet package in R (IV)

- The glmnet algorithms use cyclical coordinate descent, which successively optimizes the objective function over each parameter with others fixed, and cycles repeatedly until convergence.
- Due to highly efficient updates and techniques such as warm starts, the algorithms can compute the solution path very fast.
- The code can handle sparse input-matrix formats, as well as range constraints on coefficients.
- The core of glmnet is a set of Fortran subroutines, which make for very fast execution.
- The package also includes methods for prediction and plotting, and a function that performs k-fold cross-validation.

Practice:

- Prostate data: Lasso with `glmnet`.
- To scale or not to scale?
- Use the R script `prostate.lasso.R`.
- See the *Glmnet vignette*, Hastie and Qian (2014).

Concluding remarks on Lasso

- Lasso (L_1 penalty) offers a way to simultaneously select variables and estimate the coefficients in generalized linear models (and more).
- Newly developed computational algorithms allow application of these models to large data sets, with both n and p large, particularly when $p \gg n$.
- There is a very active research on the statistical properties of Lasso.
- The package `glmnet` in R is an efficient implementation of Lasso.

Bühlmann, P. and S. van de Geer (2011).
Statistics for High-Dimensional Data: Methodology, Theory and Applications.
Springer.

Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004).
Least angle regression.
The Annals of statistics 32(2), 407–499.

Friedman, J., T. Hastie, and R. Tibshirani (2010).
Regularization paths for generalized linear models via coordinate descent.
Journal of statistical software 33(1), 1.

Hastie, T. and J. Qian (2014).
Glmnet vignette.
Stanford statistics technical report, Department of Statistics, Stanford University,
`http://www.stanford.edu/~hastie/glmnet/glmnet_alpha.html`.
(Version of June 26, 2014).

Hastie, T., R. Tibshirani, and J. Friedman (2009).
The elements of statistical learning (2nd ed.).
Springer.

Hastie, T., R. Tibshirani, and M. Wainwright (2015).
Statistical learning with sparsity: the lasso and generalizations.
CRC Press.

Lockhart, R., J. Taylor, R. J. Tibshirani, R. Tibshirani, et al. (2014).

A significance test for the lasso.
The Annals of Statistics 42(2), 413–468.

Tibshirani, R. (1996).
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society. Series B (Methodological) 58, 267–288.

Tibshirani, R. (2011).
Regression shrinkage and selection via the lasso: a retrospective.
Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73(3), 273–282.
With discussion.