Regularized estimation of LM and GLM

Part 2. Lasso estimation for LM and GLM

Pedro Delicado

Departament d'Estadística i Investigació Operativa

Universitat Politècnica de Catalunya

1. The Lasso estimation
    - Computation of Lasso
    - Statistical properties of Lasso
    - `glmnet` package in R

2. Lasso estimation in the GLM
    - Preliminaries on MLE and IRWLS
    - Revisiting the IRWLS version of Newton-Raphson for GLM
    - Iterative Re-Weighted Lasso estimation in the GLM

References:

Section 4.4.4 in Hastie, Tibshirani, and Friedman (2009)

Chapters 1, 2 and 3, section 5.4 in Hastie, Tibshirani, and Wainwright (2015)

Section 6.2 3 in James, Witten, Hastie, and Tibshirani (2013)

Tibshirani (2011)

Hastie and Qian (2014)

1. The Lasso estimation
   - Computation of Lasso
   - Statistical properties of Lasso
   - `glmnet` package in R

2. Lasso estimation in the GLM
   - Preliminaries on MLE and IRWLS
   - Revisiting the IRWLS version of Newton-Raphson for GLM
   - Iterative Re-Weighted Lasso estimation in the GLM

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:

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

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

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

Lasso gives sparse solutions

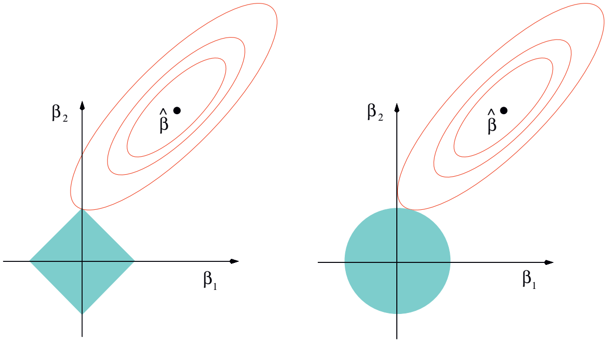
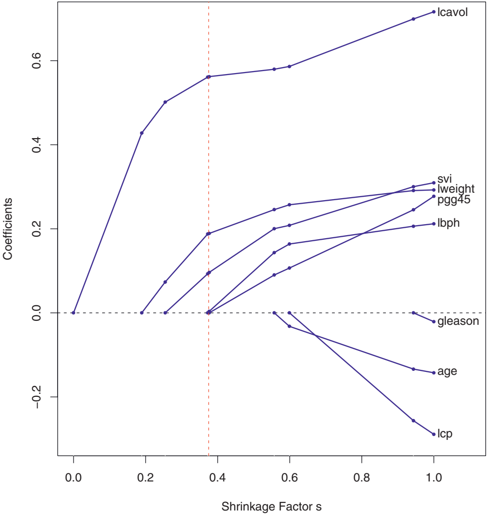


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are 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 estimation depends on the scale of variables, as it happens in ridge regression. Therefore, from now on we assume that the explanatory variables have been previously centered and scaled to have zero mean and unit variance, and that the response variable has been centered. Therefore $\beta_0 = 0$.
- 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)

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

that can be efficiently solved.

Lasso and ℓ_q norms

- For $q > 0$, ℓ_q norm of $\boldsymbol{\beta} \in \mathbb{R}^p$: $\|\boldsymbol{\beta}\|_{\ell_q} = \left(\sum_{j=1}^p |\boldsymbol{\beta}_j|^q\right)^{1/q}$.
- $\|\boldsymbol{\beta}\|_{\ell_\infty} = \lim_{q \longrightarrow \infty} \|\boldsymbol{\beta}\|_{\ell_q} = \max_{j=1,\ldots,p} |\beta_j|$.
- 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 $a \notin \{-1, 0, 1\}$).

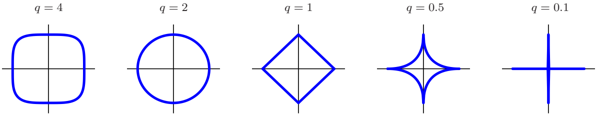


Figure 2.6 *Constraint regions* $\sum_{j=1}^p |\beta_j|^q \leq 1$ *for different values of* q. *For* $q < 1$, *the constraint region is nonconvex.*

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
$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$ s.t. $\lVert\boldsymbol{\beta}\rVert_{\ell_0} \leq t$	$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$ s.t. $\lVert\boldsymbol{\beta}\rVert_{\ell_1} \leq t$	$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^n (y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$ s.t. $\lVert\boldsymbol{\beta}\rVert_{\ell_2} \leq t$

- 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, $q \geq 0$.

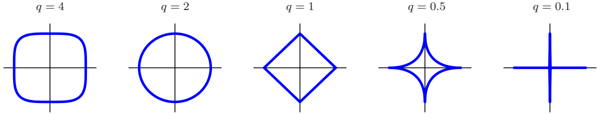


Figure 2.6 *Constraint regions $\sum_{j=1}^p |\beta_j|^q \leq 1$ for different values of q. For $q < 1$, the constraint region is nonconvex.*

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.

1. The Lasso estimation
   - Computation of Lasso
   - Statistical properties of Lasso
   - `glmnet` package in R
2. Lasso estimation in the GLM
   - Preliminaries on MLE and IRWLS
   - Revisiting the IRWLS version of Newton-Raphson for GLM
   - Iterative Re-Weighted Lasso estimation in the GLM

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 optimization

- Consider the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^q} f(\boldsymbol{x}) \equiv \min_{(x_1,\dots,x_q)\in\mathbb{R}^q} f(x_1,\dots,x_q).$$

- The cyclic coordinate descent algorithm works as follows:
  1. Let $k=0$ and choose an arbitrary initial point $\boldsymbol{x}^0=(x_1^0,\dots,x_q^0)\in\mathbb{R}^q$.
  2. Iterate until convergence:
     - For $i=1,\dots,q$,

       $$x_i^{k+1} = \arg\min_{y\in\mathbb{R}} f(x_1^{k+1},\dots,x_{i-1}^{k+1},y,x_{i+1}^k,\dots,x_q^k).$$

     - STOP if $\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\|$ or $|f(\boldsymbol{x}^{k+1})-f(\boldsymbol{x}^k)|$ are small.
- This algorithm is specially useful when the one-dimensional optimization problems have closed form solution.
- This is the case for the LASSO estimation in regression.

Cyclic coordinate optimization. Properties (I)

Consider the problem $\min_{\boldsymbol{x}\in\mathbb{R}^q} f(\boldsymbol{x})$.

- The cyclic coordinate descent algorithm has the descent property: $f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k)$ far all k.
- Sufficient conditions for the algorithm convergence.
  (i) Assuming that f is twice differentiable, that $\boldsymbol{x}^* \in \mathbb{R}^q$ is a local minimum of f and that the Hessian matrix of f at $\boldsymbol{x}^*$ is positive definite, then the cyclic coordinate descent algorithm converges locally to $\boldsymbol{x}^*$: if $\boldsymbol{x}^0$ is close to $\boldsymbol{x}^*$ then $\lim_k \boldsymbol{x}^k = \boldsymbol{x}^*$.
  (Lange 1999, Section 13.3, page 165)
  (ii) If f is continuously differentiable and strictly convex in each coordinate, then the cyclic coordinate descent algorithm converges to the global minimum of f.

Cyclic coordinate optimization. Properties (II)

- Sufficient conditions for the algorithm convergence (cont.).

(iii) If f has the additive decomposition

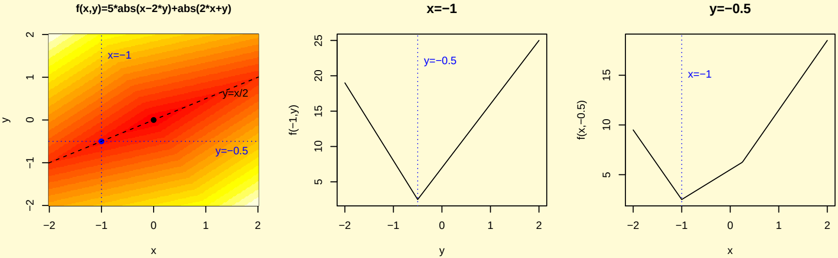
$$f(x_1,\dots,x_q) = g(x_1,\dots,x_q) + \sum_{i=1}^{q} h_i(x_i)$$

where g is differentiable and convex, and the univariate functions h_i are convex (but not necessarily 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)

(iv) The LASSO estimation in regression has this separability property.

Failure of the coordinate descent algorithm

- When the non-differentiable part of $f(x_1,\dots,x_q)$ is not separable, the coordinate descent algorithm may fail to converge.
- Example: $\min_{(x,y)\in\mathbb{R}^2} f(x,y) = 5|x-2y| + |2x+y|$
  The global minimum is $(x,y)=(0,0)$ but any point over the line $y=x/2$ is a fixed point of the algorithm.



Cyclic coordinate descent algorithm for LASSO:

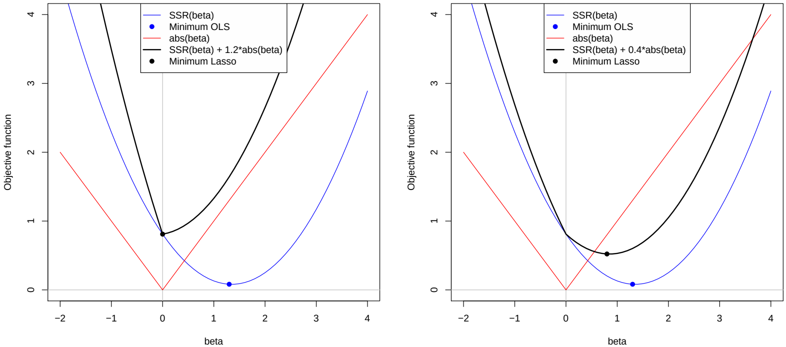
- First we will see that Lasso has a closed form solution when $p = 1$ (single predictor case).
- Then we will give the co-ordinate descent algorithm for a generic p.

Single predictor. Soft thresholding function

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

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

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



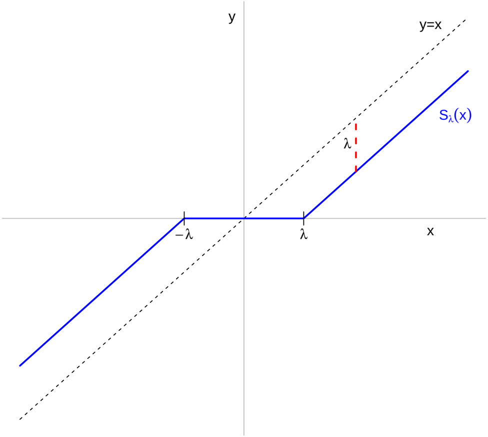
Let $f(\beta)=\frac{1}{2n}\sum_{i=1}^n(y_i-x_i\beta)^2+\lambda|\beta|$. Then,

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

- If $\hat\beta_{\text{OLS}}=\frac{1}{n}\langle \boldsymbol{x},\boldsymbol{y}\rangle\geq 0$ 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 $\hat\beta_{\text{OLS}}=\frac{1}{n}\langle \boldsymbol{x},\boldsymbol{y}\rangle<0$ 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 $\mathcal{S}_\lambda(x) = \text{sign}(x)\left(|x| - \lambda\right)_+$.
- 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 $\boldsymbol{\beta} \in \mathbb{R}^p$, 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 $1, \dots, p$) 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) $\hat\beta_j^{\text{new}} = \mathcal{S}_\lambda \left(\frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{r}^{(j)} \rangle \right)$.
- 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 $\boldsymbol{r}^{(j)} = \boldsymbol{r} + \hat\beta_j \boldsymbol{x}_j$ and $\frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{r}^{(j)} \rangle = \frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{r} \rangle + \hat\beta_j \frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle = \frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{r} \rangle + \hat\beta_j$.
- Then $\hat\beta_j^{\text{new}} = \mathcal{S}_\lambda \left(\hat\beta_j + \frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{r} \rangle \right)$.
- And the new full residuals are $\boldsymbol{r}^{\text{new}} = \boldsymbol{r} - \left(\hat\beta_j^{\text{new}} - \hat\beta_j \right) \boldsymbol{x}_j$.

Practice:

- Prostate data: Lasso estimation for a given λ.
- Use the R script `01_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 giving a non-zero solution is

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

  because for $\lambda > \lambda_0$ the cyclic coordinate descent algorithm has $\boldsymbol{\beta} = \mathbf{0}$ as the only fixed point.
- Warm start: The solution $\hat{\boldsymbol{\beta}}(\lambda_\ell)$ is the initial value (warm start) for the algorithm when looking for the solution $\hat{\boldsymbol{\beta}}(\lambda_{\ell+1})$, $\ell = 1, \ldots, L-1$.
- 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 $\lambda_{\ell+1}$ allows to improve algorithmic efficiency.

Practice:

- Prostate data: Lasso estimation and *coefficients path*.
- Use the R script `01_prostate.lasso.R`.

1 The Lasso estimation
- Computation of Lasso
- Statistical properties of Lasso
- glmnet package in R

2 Lasso estimation in the GLM
- Preliminaries on MLE and IRWLS
- Revisiting the IRWLS version of Newton-Raphson for GLM
- Iterative Re-Weighted Lasso estimation in the GLM

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 $\mathrm{df}(\lambda)$.
- 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 selects 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 $\mathrm{df}(\lambda)$ with no bias.

Advanced statistical properties of Lasso

(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:

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

Four problems have received much attention:

- Prediction and estimation of the regression surface $\boldsymbol{X}\boldsymbol{\beta}$.
- 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 $\boldsymbol{X}$ and mild conditions on ε, it can be proved that

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

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

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

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 estimators.
- 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

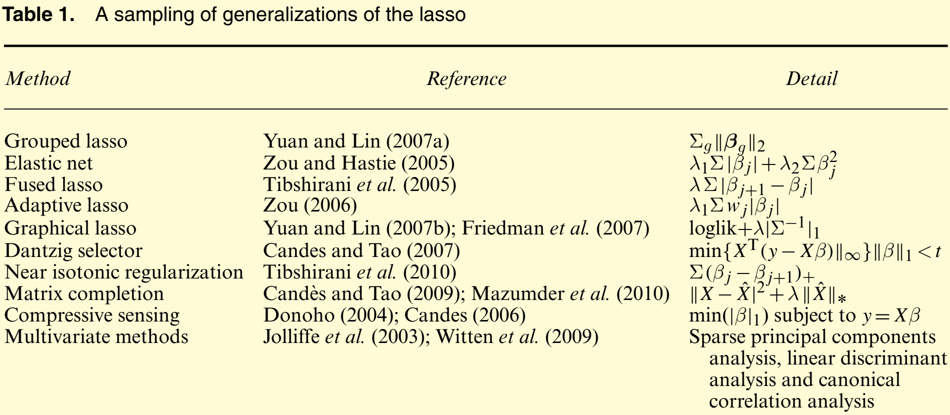


Table 1. A sampling of generalizations of the lasso

Method	*Reference*	*Detail*
Grouped lasso	Yuan and Lin (2007a)	$\Sigma_g \|\boldsymbol{\beta}_g\|_2$
Elastic net	Zou and Hastie (2005)	$\lambda_1 \Sigma \lvert\beta_j\rvert + \lambda_2 \Sigma \beta_j^2$
Fused lasso	Tibshirani *et al.* (2005)	$\lambda \Sigma \lvert\beta_{j+1} - \beta_j\rvert$
Adaptive lasso	Zou (2006)	$\lambda_1 \Sigma w_j \lvert\beta_j\rvert$
Graphical lasso	Yuan and Lin (2007b); Friedman *et al.* (2007)	loglik$+\lambda\lvert\Sigma^{-1}\rvert_1$
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)	$\|X - \hat{X}\rvert^2 + \lambda \|\hat{X}\|_*$
Compressive sensing	Donoho (2004); Candes (2006)	min($\lvert\beta\rvert_1$) 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)

1 The Lasso estimation
- Computation of Lasso
- Statistical properties of Lasso
- glmnet package in R

2 Lasso estimation in the GLM
- Preliminaries on MLE and IRWLS
- Revisiting the IRWLS version of Newton-Raphson for GLM
- Iterative Re-Weighted Lasso estimation in the GLM

`glmnet` package in R (I)

(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 $\boldsymbol{X}$.
- 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

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

  over a grid of values of λ covering the entire range.
- Here $\ell(y, \eta)$ is the 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 `01_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.

1. The Lasso estimation
   - Computation of Lasso
   - Statistical properties of Lasso
   - `glmnet` package in R

2. Lasso estimation in the GLM
   - Preliminaries on MLE and IRWLS
   - Revisiting the IRWLS version of Newton-Raphson for GLM
   - Iterative Re-Weighted Lasso estimation in the GLM

Lasso estimation in the GLM

- Let $(Y_i, \boldsymbol{x}_i)$, $i = 1, \dots, n$, following a GLM corresponding to the parametric model $f(y, \theta_i)$.
- Let $\ell(\theta_i, y_i) = \log f(y_i, \theta_i)$ be log-likelihood contribution for one observation.
- Assume that there is a one-to-one relationship between θ_i and the linear term $\beta_0 + \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta}$:
$$\theta_i = k(\beta_0 + \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta}).$$
- Let $\ell(\beta_0, \boldsymbol{\beta}, y_i, \boldsymbol{x}_i) = \ell(k(\beta_0 + \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta}), y_i)$.
- The Lasso estimation of the GLM solves the problem
$$\min_{\beta_0, \boldsymbol{\beta}} -\frac{2}{n} \sum_{i=1}^{n} \ell(\beta_0, \boldsymbol{\beta}, y_i, \boldsymbol{x}_i) + \lambda ||\boldsymbol{\beta}||_1.$$

1 The Lasso estimation

Computation of Lasso

Statistical properties of Lasso

glmnet package in R

2 Lasso estimation in the GLM

Preliminaries on MLE and IRWLS

Revisiting the IRWLS version of Newton-Raphson for GLM

Iterative Re-Weighted Lasso estimation in the GLM

Preliminaries on maximum likelihood estimation

- Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be i.i.d. distributed as X, a random variable with density (probability) function $f(\boldsymbol{x}|\boldsymbol{\theta})$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k) \in \Theta \subseteq \mathbb{R}^k$.
- Let $\mathcal{X}$ be the sampling space, that is, the set of possible values of $\boldsymbol{X}$.
- The **likelihood function** for $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}$ is defined as

$$\begin{array}{cccl} L(\cdot|\boldsymbol{x}): & \Theta & \longrightarrow & \mathbb{R}^+ \\ & \boldsymbol{\theta} & \longrightarrow & L(\boldsymbol{\theta}|\boldsymbol{x}) = f(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{i=1}^n f(x_i|\boldsymbol{\theta}) \end{array}$$

- **Score vector**, gradient of the log-likelihood: $S(\boldsymbol{x}, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}|\boldsymbol{x})$.
- **Observed Information matrix**, minus the Hessian matrix of the log-likelihood: $O(\boldsymbol{x}, \boldsymbol{\theta}) = -H_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}|\boldsymbol{x})$.
- **Fisher Information matrix**, the expected value of the observed information: $I(\boldsymbol{\theta}) = -\mathbb{E}\left(H_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}|\boldsymbol{X})\right) = nI_1(\boldsymbol{\theta})$, where $I_1(\boldsymbol{\theta}) = -\mathbb{E}\left(H_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}|X)\right)$ corresponds to one observation.

- Under regularity conditions,
  - $\mathbb{E}\left[S(\boldsymbol{X},\boldsymbol{\theta})\right]=\mathbf{0}$.
  - $\mathrm{Var}[S(\boldsymbol{X},\boldsymbol{\theta})]=\mathbb{E}\left[S(\boldsymbol{X},\boldsymbol{\theta})S(\boldsymbol{X},\boldsymbol{\theta})^{\mathsf{T}}\right]=I(\boldsymbol{\theta})$.
- Maximum likelihood estimator (MLE): $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}=\arg\max_{\boldsymbol{\theta}\in\Theta}\log L(\boldsymbol{\theta}|\boldsymbol{x})$.
- **Asymptotic properties of MLE:** Let $\boldsymbol{\theta}_0$ be the true parameter value. Under regularity conditions,
  - $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}\xrightarrow{\mathsf{p}}\boldsymbol{\theta}_0$,
  - $\sqrt{n}\big(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}-\boldsymbol{\theta}_0\big)\ \xrightarrow{d}\ N(0,\ I_1(\boldsymbol{\theta})^{-1})$,

  as n goes to infinity.

  Moreover, the MLE is an asymptotically efficient estimator.

Preliminaries on Newton-Raphson method

- The Newton-Raphson method is an iterative procedure providing a sequence $\{\boldsymbol{x}^k\}_{k\geq 1}$ that, under quite general assumptions, converges to the minimum $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^q} f(\boldsymbol{x})$.
- When starting the step $k+1$ of the algorithm, the last available value is $\boldsymbol{x}^k$.
- In order to obtain the next value $\boldsymbol{x}^{k+1}$, the Newton-Raphson algorithm uses the second order Taylor's approximation of $f(\boldsymbol{x})$ around $\boldsymbol{x}^k$:

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^k) + (\boldsymbol{x}-\boldsymbol{x}^k)^\mathsf{T}\nabla f(\boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^k)^\mathsf{T} Hf(\boldsymbol{x}^k)(\boldsymbol{x}-\boldsymbol{x}^k).$$

- Let $\tilde{f}^k(\boldsymbol{x})$ be the right hand side approximating quadratic function:

$$\tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + (\boldsymbol{x} - \boldsymbol{x}^k)^\mathsf{T} \nabla f(\boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)^\mathsf{T} Hf(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k).$$

- It is possible to minimize $\tilde{f}^k(\boldsymbol{x})$ analytically:
    - Its gradient is $\nabla \tilde{f}^k(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^k) + Hf(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k)$.
    - We solve the equation $\nabla \tilde{f}^k(\boldsymbol{x}) = \mathbf{0}$ and call the solution $\boldsymbol{x}^{k+1}$:

$$\nabla f(\boldsymbol{x}^k) + Hf(\boldsymbol{x}^k)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \mathbf{0} \Rightarrow \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \left(Hf(\boldsymbol{x}^k)\right)^{-1} \nabla f(\boldsymbol{x}^k).$$

- This point $\boldsymbol{x}^{k+1}$ is the minimum of $\tilde{f}^k(\boldsymbol{x})$ if $Hf(\boldsymbol{x}^k)$ is positive definite because

$$H\tilde{f}^k(\boldsymbol{x}^k) = Hf(\boldsymbol{x}^k).$$

- This will be the case when $\boldsymbol{x}^k$ is close to the global minimum of f and this function has continuous second derivatives.

Newton-Raphson Method

- The recursive formula giving $\boldsymbol{x}^{k+1}$ from $\boldsymbol{x}^k$ is

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \left(Hf(\boldsymbol{x}^k)\right)^{-1} \nabla f(\boldsymbol{x}^k).$$

- The algorithm iterates until convergence: it stops if $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|$ or $|f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^k)|$ are small.
- A sufficient condition for convergence of $\{\boldsymbol{x}^k\}_{k\geq 1}$ to the global minimum $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ is that f is a convex function.
- **Maximum likelihood estimation:**

  Newton-Raphson algorithm: $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + O(\boldsymbol{x}, \boldsymbol{\theta}^k)^{-1} S(\boldsymbol{x}_i, \boldsymbol{\theta}^k)$.

  Fisher Scoring algorithm: $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + I(\boldsymbol{\theta}^k)^{-1} S(\boldsymbol{x}_i, \boldsymbol{\theta}^k)$.
- Both methods coincide for exponential families, because the observed information matrix does not depend on the observed data.

IRWLS for MLE in logistic regression

Let (Y_i, x_i), $i = 1, \ldots, n$, $x_i \in \mathbb{R}$ known constant values and $Y_1, \ldots, Y_n$ independent random variables

$$Y_i \sim \text{Bernoulli}(p_i),\ i = 1, \ldots, n.$$

Assume that for all $i = 1, \ldots, n$

$$p_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \Leftrightarrow \log \frac{p_i}{1 - p_i} = \beta_0 + \beta_1 x_i.$$

Remember that $E(Y_i) = p_i$, $\text{Var}(Y_i) = p_i(1 - p_i)$.

So we have a generalized linear model.

The link function is the logistic function:

$$g(p) = \log \frac{p}{1 - p} \Leftrightarrow g^{-1}(v) = \frac{e^v}{1 + e^v}.$$

Logistic regression: Likelihood function

When (y_i, x_i), $i = 1, \dots, n$, are observed the likelihood function is

$$L(\beta_0, \beta_1) = \Pr(Y_1 = y_1, \dots, Y_n = y_n) = \prod_{i=1}^{n} \Pr(Y_i = y_i) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}$$

with logarithm

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} \left(y_i \log \frac{p_i}{1-p_i} + \log(1-p_i) \right)$$

$$= \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right).$$

Logistic regression: Score function

$$\ell(\beta_0,\beta_1)=\sum_{i=1}^n y_i(\beta_0+\beta_1 x_i)-\sum_{i=1}^n \log\left(1+e^{\beta_0+\beta_1 x_i}\right).$$

$$\frac{\partial\ell(\beta_0,\beta_1)}{\partial\beta_0}=\sum_{i=1}^n y_i-\sum_{i=1}^n\frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}=\sum_{i=1}^n(y_i-p_i)$$

$$\frac{\partial\ell(\beta_0,\beta_1)}{\partial\beta_1}=\sum_{i=1}^n y_i x_i-\sum_{i=1}^n\frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}x_i=\sum_{i=1}^n(y_i-p_i)x_i$$

So,

$$S(\beta_0,\beta_1,\boldsymbol{y})=\nabla\ell(\beta_0,\beta_1)=\boldsymbol{X}^t(\boldsymbol{y}-\boldsymbol{p})$$

where

$$\boldsymbol{X}^t=\begin{pmatrix}1&\cdots&1\\x_1&\cdots&x_n\end{pmatrix},$$

$\boldsymbol{y}$ and $\boldsymbol{p}$ are column vectors.

Logistic regression: Fisher's Information; Fisher Scoring

$$S(\beta_0, \beta_1, \boldsymbol{Y}) = \boldsymbol{X}^t(\boldsymbol{Y} - \boldsymbol{p})$$

$$I(\beta_0, \beta_1) = \mathrm{Var}(S(\beta_0, \beta_1, \boldsymbol{Y})) = \boldsymbol{X}^t \mathrm{Var}(\boldsymbol{Y})\boldsymbol{X} = \boldsymbol{X}^t \boldsymbol{W} \boldsymbol{X}$$

where

$$\boldsymbol{W} = \mathrm{diag}(p_1(1-p_1), \ldots, p_n(1-p_n)).$$

Generic iteration in the Fisher Scoring algorithm:

$$\begin{pmatrix} \beta_0^{m+1} \\ \beta_1^{m+1} \end{pmatrix} = \begin{pmatrix} \beta_0^{m} \\ \beta_1^{m} \end{pmatrix} + I(\beta_0^m, \beta_1^m)^{-1} \nabla \ell(\beta_0^m, \beta_1^m) = \begin{pmatrix} \beta_0^{m} \\ \beta_1^{m} \end{pmatrix} + (\boldsymbol{X}^t \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^t (\boldsymbol{y} - \boldsymbol{p})$$

$$= \begin{pmatrix} \beta_0^{m} \\ \beta_1^{m} \end{pmatrix} + (\boldsymbol{X}^t \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^t \boldsymbol{W} \left(\boldsymbol{W}^{-1} (\boldsymbol{y} - \boldsymbol{p}) \right)$$

$$
\begin{aligned}
\begin{pmatrix}\beta_0^{m+1}\\ \beta_1^{m+1}\end{pmatrix} &= \begin{pmatrix}\beta_0^{m}\\ \beta_1^{m}\end{pmatrix} + (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\left(\boldsymbol{W}^{-1}(\boldsymbol{y}-\boldsymbol{p})\right)\\
&= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}(\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})\begin{pmatrix}\beta_0^{m}\\ \beta_1^{m}\end{pmatrix} + (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\left(\boldsymbol{W}^{-1}(\boldsymbol{y}-\boldsymbol{p})\right)\\
&= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\left(\boldsymbol{X}\begin{pmatrix}\beta_0^{m}\\ \beta_1^{m}\end{pmatrix} + \left(\boldsymbol{W}^{-1}(\boldsymbol{y}-\boldsymbol{p})\right)\right)\\
&= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{Z},
\end{aligned}
$$

where $\boldsymbol{Z}$ is the $n \times 1$ vector of pseudo-observations, with i-th element

$$
z_i = \beta_0^m + \beta_1^m x_i + \frac{y_i - p_i}{p_i(1-p_i)}.
$$

$$\begin{pmatrix}\beta_0^{m+1}\\ \beta_1^{m+1}\end{pmatrix} = (\boldsymbol{X}^t \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^t \boldsymbol{W} \boldsymbol{Z}.$$

Observe that $\binom{\beta_0^{m+1}}{\beta_1^{m+1}}$ is the weighted least squares (WLS) coefficients estimator in the simple linear regression

with data (x_i, z_i) and weights $p_i(1-p_i)$, $i = 1, \ldots, n$.

Taking into account that these data come from the model

$$Z_i = \beta_0^m + \beta_1^m x_i + \frac{Y_i - p_i}{p_i(1-p_i)},$$

and calling $\varepsilon_i = \frac{Y_i - p_i}{p_i(1-p_i)}$, it follows that

$$E(\varepsilon_i) = 0, \ \mathrm{Var}(\varepsilon_i) = \frac{1}{p_i(1-p_i)}.$$

So the weight of each case (x_i, z_i) is equal to the inverse of the error ε_i variance.

Iteratively re-weighted least squares algorithm (IRWLS) for logistic regression.

- Choose starting values $\boldsymbol{\beta}^0 = (\beta_0^0, \beta_1^0)$ (the choice $\beta_0^0 = \beta_1^0 = 0$ is usually appropriate; choosing the OLS estimates is also possible).
- Set $m = 0$ and iterate the following steps until convergence.

1. Set
$$p_i^m = \frac{e^{\beta_0^m + \beta_1^m x_i}}{1 + e^{\beta_0^m + \beta_1^m x_i}},$$
$$z_i^m = \beta_0^m + \beta_1^m x_i + \frac{y_i - p_i^m}{p_i^m(1 - p_i^m)},\ i = 1, \dots, n.$$
2. Let $(\nu_1^m, \dots, \nu_n^m)$ be the weight vector with $\nu_i^m = p_i^m(1 - p_i^m)$.
3. Fit the linear regression with responses z_i^m and explanatory variable values x_i, (plus the constant term) by weighted least squares using the weights ν_i^m, $i = 1, \dots, n$.
   Let $\boldsymbol{\beta}^{m+1} = (\beta_0^{m+1}, \beta_1^{m+1})$ be the estimated regression coefficients.
4. Set $m = m + 1$ and go back to the step 1.

Logistic regression: Variance of the MLE

We know that

$$\text{Var}\left(\begin{pmatrix}\hat\beta_0^{ML}\\ \hat\beta_1^{ML}\end{pmatrix}\right) \approx I(\hat\beta_0^{ML},\hat\beta_1^{ML})^{-1} = \left(\boldsymbol{X}^t\boldsymbol{W}_{\hat\beta_0^{ML},\hat\beta_1^{ML}}\boldsymbol{X}\right)^{-1}.$$

On the other hand,

$$\text{Var}\left(\begin{pmatrix}\beta_0^{m+1}\\ \beta_1^{m+1}\end{pmatrix}\right) = \text{Var}\left((\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{Z}\right)$$

$$= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\,\text{Var}(\boldsymbol{Z})\,\boldsymbol{W}\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}$$

$$= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{W}^{-1}\boldsymbol{W}\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}$$

$$= (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1} = (\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X})^{-1}.$$

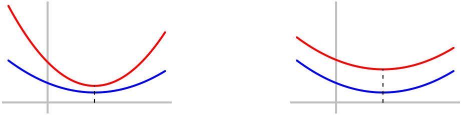
Conclusion: We can estimate the variance of the (β_0,β_1) MLE by the coefficients variance of the last WLS estimator.

Practice:

Follow the R Markdown file `01_IRWLS_logistic.Rmd`.

Revisiting the IRWLS version of Newton-Raphson (or Fisher scoring) for GLM

- The key-point in the IRWLS version of Newton-Raphson (or Fisher scoring) for GLM is that at each step *m* of the algorithm, the following two elements coincide:
    - The point θ_{m+1}^{NR} maximizing the quadratic function $\tilde{\ell}_m(\theta)$ that approximates the log-likelihood function $\ell(\theta)$ around θ_m (or minimizing $-2\tilde{\ell}_m(\theta)$).
    - The point $\theta_{m+1}^{\text{WLS}}$ minimizing the weighted sum of squared residuals in the *m*-th regression problem.
- Do these two functions share just the location of their minimums? Or are they equal up to an additive constant?



We will see that the latter happens.

Revisiting the IRWLS version of Newton-Raphson (or Fisher scoring) for GLM

- The GLM assumes that $f(y,\theta)$ is in the exponential family:

$$f(y,\theta) = h(y)c(\theta)\exp\left(\eta(\theta)t(y)\right).$$

- Let us assume additionally that this family is parameterized in natural form ($\eta(\theta)=\theta$), and that the function $t(y_i)=y_i$ (that is, the sample mean is a sufficient statistic):

$$f(y,\theta) = h(y)c(\theta)\exp\left(\theta y\right),\ \ell(\theta,y) = \log h(y) + \log c(\theta) + \theta y.$$

- The score function is

$$S(\theta,Y) = \frac{\partial \ell(\theta,y)}{\partial\theta} = \frac{\partial}{\partial\theta}\log c(\theta) + Y$$

- As $E_\theta(S(\theta,Y))=0$, it follows that $\mu = E_\theta(Y) = -\frac{\partial}{\partial\theta}\log c(\theta)$.
- Then $S(\theta,Y) = Y-\mu$.
- Then $V_\theta(Y) = V_\theta(S(\theta,Y)) = -\frac{\partial^2}{\partial\theta^2}\log c(\theta)$.

- Consider again $(Y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, from this GLM model, with parameters $\theta_i = k(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta)$, respectively, and $\ell(\beta_0, \beta, y_i, x_i) = \ell(k(\beta_0 + x_i^{\mathsf{T}}\beta), y_i)$.
- Then, applying the chain rule,

$$\nabla_{\beta_0,\beta}\ell(\beta_0, \beta, y_i, x_i) = \left.\frac{\partial \ell(\theta, y_i)}{\partial \theta}\right|_{\theta=\theta_i} \nabla_{\beta_0,\beta} k(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta) =$$

$$S(\theta_i, y_i) k'(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta)\tilde{\boldsymbol{x}}_i = (y_i - \mu_i) k'(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta)\tilde{\boldsymbol{x}}_i,$$

  where $\tilde{\boldsymbol{x}}_i^{\mathsf{T}} = (1, \boldsymbol{x}_i^{\mathsf{T}})$.
- Let $\ell(\beta_0, \beta, \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^n \ell(\beta_0, \beta, y_i, x_i)$ be the full likelihood. Then

$$\nabla_{\beta_0,\beta}\ell(\beta_0, \beta, \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^n (y_i - \mu_i) k'(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta)\tilde{\boldsymbol{x}}_i = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{y} - \boldsymbol{\mu}).$$

  where $\boldsymbol{K} = \mathrm{Diag}(k'(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\beta), i = 1, \ldots, n)$, and $\boldsymbol{X}$ is the matrix with rows $\tilde{\boldsymbol{x}}_i^{\mathsf{T}}$.

- The Fisher's Information matrix is

$$I(\beta_0, \beta) = V(\nabla_{\beta_0,\beta}\ell(\beta_0, \beta, \boldsymbol{y}, \boldsymbol{X})) = V(\boldsymbol{X}^\mathsf{T} K(\boldsymbol{y} - \boldsymbol{\mu})) = \boldsymbol{X}^\mathsf{T}\boldsymbol{K}\, V(\boldsymbol{y})\boldsymbol{K}\boldsymbol{X}.$$

- Observe that $V(\boldsymbol{y}) = \boldsymbol{D} = \text{Diag}\left(- \left.\frac{\partial^2 \log c(\theta)}{\partial\theta^2}\right|_{\theta=\theta_i}, i = 1, \ldots, n \right)$.
- Then $I(\beta_0, \beta) = \boldsymbol{X}^\mathsf{T}\boldsymbol{W}\boldsymbol{X}$ with

$$\boldsymbol{W} = \boldsymbol{K}\boldsymbol{D}\boldsymbol{K} = \text{Diag}\left(- k'(\beta_0 + \boldsymbol{x}_i^\mathsf{T}\beta)^2 \left.\frac{\partial^2 \log c(\theta)}{\partial\theta^2}\right|_{\theta=\theta_i}, i = 1, \ldots, n \right).$$

- Remember that, in exponential families, the Hessian matrix of $\ell(\beta_0, \beta, \boldsymbol{y}, \boldsymbol{X})$ with respect to (β_0, β) is $-I(\beta_0, \beta)$.

- Let $\boldsymbol{\beta}$ be the column vector with components β_0 and β.
- In the iteration m of the Newton-Raphson algorithm to minimize $-2\ell(\boldsymbol{\beta})$, the second order Taylor expansion around $\boldsymbol{\beta}^m$ is

$$-2\tilde{\ell}_m(\boldsymbol{\beta}) = -2\ell(\boldsymbol{\beta}^m) - 2(\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{y}-\boldsymbol{\mu}) + (\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}(\boldsymbol{\beta}-\boldsymbol{\beta}^m) =$$

$$-2\ell(\boldsymbol{\beta}^m) - 2(\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\,\boldsymbol{W}^{-1}\boldsymbol{K}(\boldsymbol{y}-\boldsymbol{\mu}) + (\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}(\boldsymbol{\beta}-\boldsymbol{\beta}^m) =$$

$$-2\ell(\boldsymbol{\beta}^m) - 2(\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{\varepsilon} + (\boldsymbol{\beta}-\boldsymbol{\beta}^m)^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}(\boldsymbol{\beta}-\boldsymbol{\beta}^m) =$$

$$\gamma - 2\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{Z} + \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}\boldsymbol{\beta},$$

where γ is a constant term that does not depend on $\boldsymbol{\beta}$,

$$\boldsymbol{\varepsilon} = \boldsymbol{W}^{-1}\boldsymbol{K}(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{K}^{-1}\boldsymbol{D}^{-1}\boldsymbol{K}^{-1}\boldsymbol{K}(\boldsymbol{y}-\boldsymbol{\mu}) = \boldsymbol{K}^{-1}\boldsymbol{D}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$$

and $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{\beta}^m + \boldsymbol{\varepsilon}$.

- Consider now the Weighted Least Square (WLS) problem where the response variable is contained in the vector $\boldsymbol{Z}$ defined before, the matrix of explanatory variables is $\boldsymbol{X}$ and the weights are given by the diagonal matrix $\boldsymbol{W}$.
- The objective function to be minimized in the WLS estimation is

$$Q_m(\beta) = (\boldsymbol{Z} - \boldsymbol{X}\beta)^{\mathsf{T}}\boldsymbol{W}(\boldsymbol{Z} - \boldsymbol{X}\beta) = \boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z} - 2\beta^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{Z} + \beta^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}\beta.$$

- Observe that $\kappa = -2\tilde{\ell}_m(\boldsymbol{\beta}) - Q_m(\boldsymbol{\beta})$ is constant in $\boldsymbol{\beta}$. So the value $\boldsymbol{\beta}^{m+1}$ minimizing $Q_m(\boldsymbol{\beta})$ also minimizes $-2\tilde{\ell}_m(\boldsymbol{\beta})$.
- This is the key point in the IRWLS version of the Newton-Raphson algorithm.
- When we introduced the IRWLS for the first time, we only shown that the value $\boldsymbol{\beta}^{m+1}$ minimizing $Q_m(\boldsymbol{\beta})$ also minimizes $-2\tilde{\ell}_m(\boldsymbol{\beta})$.
- Now we have seen that the functions $-2\tilde{\ell}_m(\boldsymbol{\beta})$ and $Q_m(\boldsymbol{\beta})$ are equal up to an additive constant.
- This fact will be crucial for the Lasso estimation of the GLM.

IRWLS version of Newton-Raphson for GLM

Problem: $\min_{\boldsymbol\beta} -2\ell(\boldsymbol\beta)$

Step 0: Take $\boldsymbol\beta^0$ arbitrarily (at random, with all components equal to zero, as the OLS estimator, ...). Let $m = 0$.

Step 1: Approximate the objective function using the second order Taylor approximation of $\ell(\boldsymbol\beta)$ around $\boldsymbol\beta^m$ and express the approximation in terms of a weighted sum of squares regression errors:

$$-2\ell(\boldsymbol\beta) \approx -2\tilde\ell_m(\boldsymbol\beta) = \kappa + Q_m(\boldsymbol\beta).$$

Step 2: Solve the problem $\min_{\boldsymbol\beta} Q_m(\boldsymbol\beta)$ by WLS. Let $\boldsymbol\beta^{m+1}$ be the optimum.

Step 3: Stop if $\|\boldsymbol\beta^{m+1} - \boldsymbol\beta^m\|$ or $|\ell(\boldsymbol\beta^{m+1}) - \ell(\boldsymbol\beta^m)|$ are small, or if the maximum number of iterations is reached.
Otherwise let $m = m + 1$ and go to Step 1.

1 The Lasso estimation
   - Computation of Lasso
   - Statistical properties of Lasso
   - `glmnet` package in R

2 Lasso estimation in the GLM
   - Preliminaries on MLE and IRWLS
   - Revisiting the IRWLS version of Newton-Raphson for GLM
   - Iterative Re-Weighted Lasso estimation in the GLM

Lasso estimation in the GLM

- The Lasso estimation of the GLM solves the problem

$$\min_{\beta_0,\beta} \frac{1}{n}\sum_{i=1}^{n} -2\ell(\beta_0,\beta,y_i,x_i) + \lambda||\beta||_1 \equiv \min_{\boldsymbol{\beta}} -\frac{2}{n}\ell(\boldsymbol{\beta}) + \lambda\|\beta\|_1$$

  where $\boldsymbol{\beta}^{\mathsf{T}} = (\beta_0, \beta^{\mathsf{T}})$.

- The way this problem is solved in the R library `glmnet` (see Hastie, Tibshirani, and Wainwright 2015, Chapter 5) is a modified version of the IRWLS.
- The same approximation of $-2\ell(\boldsymbol{\beta})$ by a quadratic function is done.
- Strictly speaking, the proposal is not a Newton-Raphson algorithm.

Iterative Re-Weighted Lasso estimation in the GLM

Problem: $\min_{\boldsymbol\beta} -\frac{2}{n}\ell(\boldsymbol\beta) + \lambda\|\beta\|_1$

Step 0: Take $\boldsymbol\beta^0$ arbitrarily (at random, with all components equal to zero, as the OLS estimator, ...). Let $m = 0$.

Step 1: Approximate the objective function using the second order Taylor approximation of $\ell(\boldsymbol\beta)$ around $\boldsymbol\beta^m$ and express the approximation in terms of a weighted sum of squares regression errors:

$$-\frac{2}{n}\ell(\boldsymbol\beta) + \lambda\|\beta\|_1 \approx -\frac{2}{n}\tilde\ell_m(\boldsymbol\beta) + \lambda\|\beta\|_1 = \frac{\kappa}{n} + \frac{1}{n}Q_m(\boldsymbol\beta) + \lambda\|\beta\|_1.$$

Step 2: Solve the problem $\min_{\boldsymbol\beta} \frac{1}{n}Q_m(\boldsymbol\beta) + \lambda\|\beta\|_1$ by weighted Lasso (standard coordinate descent). Let $\boldsymbol\beta^{m+1}$ be the optimum.

Step 3: Stop if $\|\boldsymbol\beta^{m+1} - \boldsymbol\beta^m\|$ or $|\ell(\boldsymbol\beta^{m+1}) - \ell(\boldsymbol\beta^m)|$ are small, or if the maximum number of iterations is reached.
Otherwise let $m = m + 1$ and go to Step 1.

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.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013).
An Introduction to Statistical Learning with Applications in R.
Springer.

Lange, K. (1999).
Statistics and Computing.
Springer.

IRW Lasso

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.