Lasso linear regression

Sahir Rai Bhatnagar

https://sahirbhatnagar.com/

July 12, 2021

Motivating Example

Lasso Regression

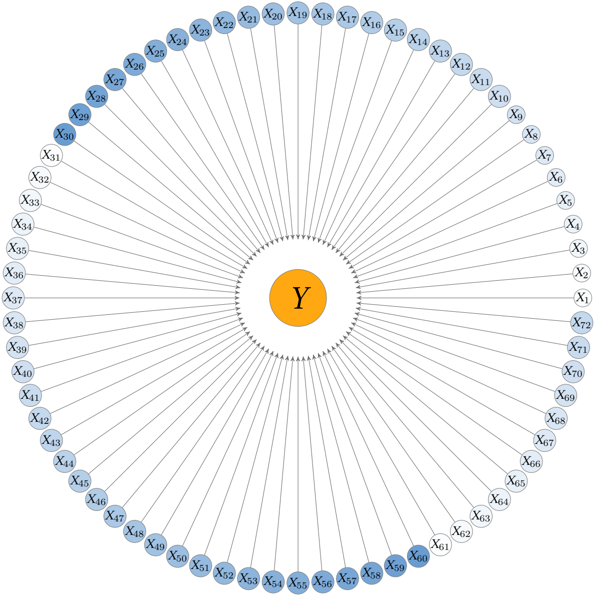
Why does the ℓ_1-norm induce sparsity?

- Analytical point of view
- Geometrical point of view

Algorithms

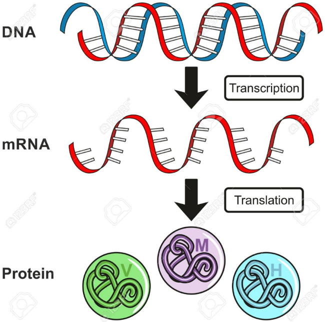
Selecting the tuning parameter λ

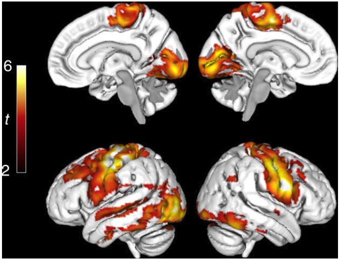
Setting



Y
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
X_9
X_{10}
X_{11}
X_{12}
X_{13}
X_{14}
X_{15}
X_{16}
X_{17}
X_{18}
X_{19}
X_{20}
X_{21}
X_{22}
X_{23}
X_{24}
X_{25}
X_{26}
X_{27}
X_{28}
X_{29}
X_{30}
X_{31}
X_{32}
X_{33}
X_{34}
X_{35}
X_{36}
X_{37}
X_{38}
X_{39}
X_{40}
X_{41}
X_{42}
X_{43}
X_{44}
X_{45}
X_{46}
X_{47}
X_{48}
X_{49}
X_{50}
X_{51}
X_{52}
X_{53}
X_{54}
X_{55}
X_{56}
X_{57}
X_{58}
X_{59}
X_{60}
X_{61}
X_{62}
X_{63}
X_{64}
X_{65}
X_{66}
X_{67}
X_{68}
X_{69}
X_{70}
X_{71}
X_{72}

High-dimensional data ($n << p$)





$$\mathbf{X}_{n\times p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{12} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & x_{np} \end{bmatrix}$$

Motivating Example: The Cancer Genome Atlas (TCGA)

- The response variable in our analysis is **expression of BRCA1**, the first gene identified to increase the risk of early onset breast cancer

Motivating Example: The Cancer Genome Atlas (TCGA)

- The response variable in our analysis is **expression of BRCA1**, the first gene identified to increase the risk of early onset breast cancer
- In the dataset, expression measurements of **17,322 additional genes from 536 patients** are available (and measured on the log scale)

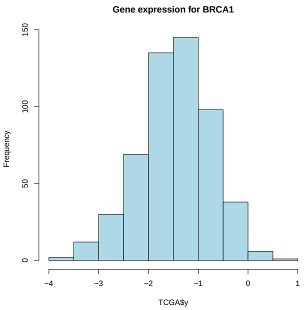
Motivating Example: The Cancer Genome Atlas (TCGA)

- The response variable in our analysis is **expression of BRCA1**, the first gene identified to increase the risk of early onset breast cancer
- In the dataset, expression measurements of **17,322 additional genes from 536 patients** are available (and measured on the log scale)
- Because BRCA1 is likely to interact with many other genes, including tumor suppressors and regulators of the cell division cycle, it is of interest to **find genes with expression levels related to that of BRCA1**

```r
# install.packages("pacman")
pacman::p_load_gh('sahirbhatnagar/mcgillHDA')
library(mcgillHDA)
data(TCGA)
# help(TCGA)
str(TCGA)
```

```
## List of 3
##  $ X    : num [1:536, 1:17322] -1.45 -2.3 -1.94 -2.1 -1.28 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:17322] "15E1.2" "2'-PDE" "7A5" "A1BG" ...
##  $ y    : num [1:536] -1.661 -1.388 -1.925 -1.656 -0.358 ...
##  $ fData:'data.frame':^^I17322 obs. of  2 variables:
##   ..$ chromosome: chr [1:17322] NA NA NA "19" ...
##   ..$ gene_name : chr [1:17322] NA NA NA "alpha-1-B glycoprotein" ...
```

```r
hist(TCGA$y, col = 'lightblue', main = "Gene expression for BRCA1")
```

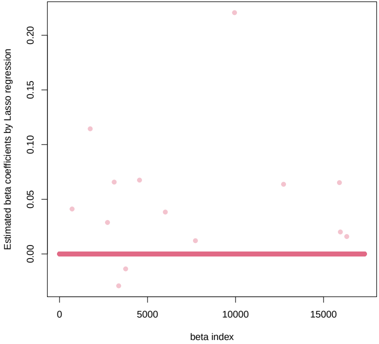


Lasso Regression on TCGA

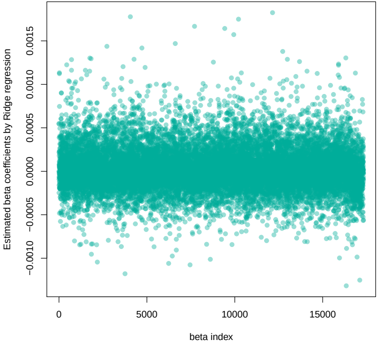
```
set.seed(101) # for reproducibility

# 80% training / 20% testing
sample <- sample.int(n = nrow(TCGA$X), size = floor(.80*nrow(TCGA$X)), replace = F)
X.train <- TCGA$X[sample, ]
X.test  <- TCGA$X[-sample, ]
y.train <- TCGA$y[sample]
y.test <- TCGA$y[-sample]

# fit lasso regression on training
library(glmnet)
fit.lasso <- cv.glmnet(x = X.train, y = y.train, alpha = 1, nfolds = 5, intercept = FALSE)
beta_hat_lasso <- coef(fit.lasso)

# fit ridge regression on training
fit.ridge <- cv.glmnet(x = X.train, y = y.train, alpha = 0, nfolds = 5, intercept = FALSE)
beta_hat_ridge <- coef(fit.ridge)

# predict on test set and MSE
yhat.test.lasso <- predict(fit.lasso, newx = X.test)
(mse.lasso <- mean((yhat.test.lasso - y.test)^2)) # test set mean squared error

## [1] 0.3095205

yhat.test.ridge <- predict(fit.ridge, newx = X.test)
(mse.ridge <- mean((yhat.test.ridge - y.test)^2)) # test set mean squared error

## [1] 0.3182241
```

Estimated Regression Coefficients $\widehat{\beta}^{lasso}$ vs. $\widehat{\beta}^{ridge}$

```r
plot(beta_hat_lasso, pch = 19, ylab = "Estimated beta coefficients by Lasso regression",
 xlab = "beta index", col = alpha("#E16A86",0.4))
plot(beta_hat_ridge, pch = 19, ylab = "Estimated beta coefficients by Ridge regression",
 xlab = "beta index", col = alpha("#00AD9A", 0.4))
```



(a) Lasso



(b) Ridge

Figure: Estimated regression coefficients

La**SS**o: Shrinkage and Selection

- Its name captures the essence of what the lasso penalty accomplishes
  - **Shrinkage**: Like ridge regression, the lasso penalizes large regression coefficients and shrinks estimates towards zero
  - **Selection**: Unlike ridge regression, the lasso produces sparse solutions: some coefficient estimates are exactly zero, effectively removing those predictors from the model

La**SS**o: Shrinkage and Selection

- Its name captures the essence of what the lasso penalty accomplishes
  - **Shrinkage**: Like ridge regression, the lasso penalizes large regression coefficients and shrinks estimates towards zero
  - **Selection**: Unlike ridge regression, the lasso produces sparse solutions: some coefficient estimates are exactly zero, effectively removing those predictors from the model
- Sparsity has two very attractive properties
  - Speed: Algorithms which take advantage of sparsity can scale up very efficiently, offering considerable computational advantages
  - Interpretability: In models with hundreds or thousands of predictors, sparsity offers a helpful simplification of the model by allowing us to focus only on the predictors with nonzero coefficient estimates

The Lasso Objective

- Let $\mathbf{y} = (y_1, \cdots, y_N)$ denote the N-vector of responses, and $\mathbf{X}$ be an $N \times p$ matrix with $x_i \in \mathbb{R}^p$ in its i^{th} row
- Assume we have centered $\mathbf{y}$ and the columns of $\mathbf{X}$ beforehand, and hence the intercept has been omitted.

The Lasso Objective

- Let $\mathbf{y} = (y_1, \cdots, y_N)$ denote the N-vector of responses, and $\mathbf{X}$ be an $N \times p$ matrix with $x_i \in \mathbb{R}^p$ in its i^{th} row
- Assume we have centered $\mathbf{y}$ and the columns of $\mathbf{X}$ beforehand, and hence the intercept has been omitted.
- The lasso finds the solution $\widehat{\boldsymbol{\beta}}_t^{lasso}$ to the optimization problem

$$\begin{aligned} &\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \\ &\text{subject to } \ ||\boldsymbol{\beta}||_1 \leq t. \end{aligned} \tag{1}$$

The Lasso Objective

- Let $\mathbf{y} = (y_1, \cdots, y_N)$ denote the N-vector of responses, and $\mathbf{X}$ be an $N \times p$ matrix with $x_i \in \mathbb{R}^p$ in its i^{th} row
- Assume we have centered $\mathbf{y}$ and the columns of $\mathbf{X}$ beforehand, and hence the intercept has been omitted.
- The lasso finds the solution $\widehat{\boldsymbol{\beta}}_t^{lasso}$ to the optimization problem

$$\begin{aligned} &\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \\ &\text{subject to } \ ||\boldsymbol{\beta}||_1 \leq t. \end{aligned} \tag{1}$$

- By Lagrangian duality, there is a one-to-one correspondence between (1) and the Lagrange version of the problem for some $\lambda \geq 0$:

$$\widehat{\boldsymbol{\beta}}_\lambda^{lasso} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} + \lambda \|\boldsymbol{\beta}\|_1 \tag{2}$$

The Lasso Objective

- Let $\mathbf{y} = (y_1, \cdots, y_N)$ denote the N-vector of responses, and $\mathbf{X}$ be an $N \times p$ matrix with $x_i \in \mathbb{R}^p$ in its i^{th} row
- Assume we have centered $\mathbf{y}$ and the columns of $\mathbf{X}$ beforehand, and hence the intercept has been omitted.
- The lasso finds the solution $\widehat{\boldsymbol{\beta}}_t^{lasso}$ to the optimization problem

$$\begin{aligned} &\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \\ &\text{subject to } \ ||\boldsymbol{\beta}||_1 \leq t. \end{aligned} \tag{1}$$

- By Lagrangian duality, there is a one-to-one correspondence between (1) and the Lagrange version of the problem for some $\lambda \geq 0$:

$$\widehat{\boldsymbol{\beta}}_\lambda^{lasso} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} + \lambda \|\boldsymbol{\beta}\|_1 \tag{2}$$

- The solution $\widehat{\boldsymbol{\beta}}_\lambda^{lasso}$ in (2) solves the bound problem in (1) with $t = \left\| \widehat{\boldsymbol{\beta}}_\lambda^{lasso} \right\|_1$

Motivating Example

Lasso Regression

Why does the ℓ_1-norm induce sparsity?

- Analytical point of view
- Geometrical point of view

Algorithms

Selecting the tuning parameter λ

Why does the ℓ_1-norm induce sparsity?

Intuition about the sparsity-inducing effect of the ℓ_1-norm may be obtained from several viewpoints:

- Analytical point of view
- Geometrical point of view

Analytical point of view

- Consider a single predictor setting based on the observed data $\{(x_i, y_i)\}_{i=1}^n$. The problem then is to solve

$$\widehat{\beta}^{lasso} = \arg\min_{\beta \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda|\beta| \tag{3}$$

Analytical point of view

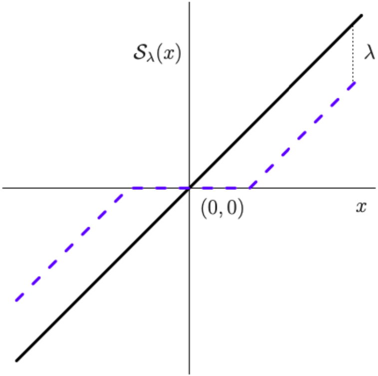
- Consider a single predictor setting based on the observed data $\{(x_i, y_i)\}_{i=1}^n$. The problem then is to solve

$$\widehat{\beta}^{lasso} = \underset{\beta \in \mathbb{R}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda|\beta| \tag{3}$$

- With a **standardized** predictor, the lasso solution (3) is a **soft-thresholded** version of the least-squares (LS) estimate $\widehat{\beta}^{LS}$

$$\begin{aligned}\widehat{\beta}^{lasso} &= S_\lambda\left(\widehat{\beta}^{LS}\right) = \text{sign}\left(\widehat{\beta}^{LS}\right)\left(|\widehat{\beta}^{LS}| - \lambda\right)_+ \\ &= \begin{cases} \widehat{\beta}^{LS} - \lambda, & \widehat{\beta}^{LS} > \lambda \\ 0 & |\widehat{\beta}^{LS}| \leq \lambda \\ \widehat{\beta}^{LS} + \lambda & \widehat{\beta}^{LS} \leq -\lambda \end{cases}\end{aligned}$$

Analytical point of view



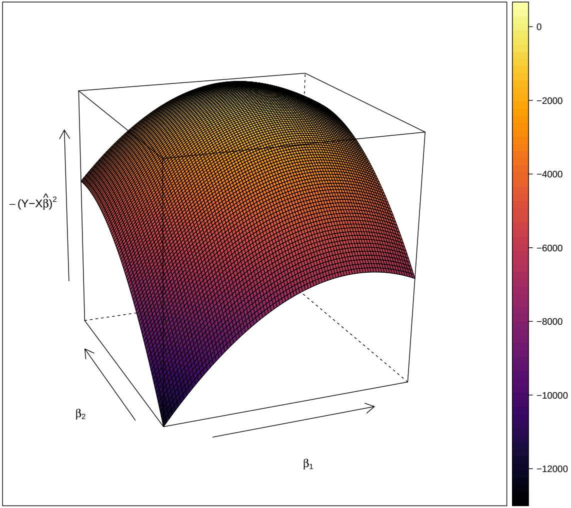
- Soft thresholding function $\mathcal{S}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$ is shown in blue (broken lines), along with the 45° line in black.

[1] Hastie et al. Statistical learning with sparsity: the lasso and generalizations

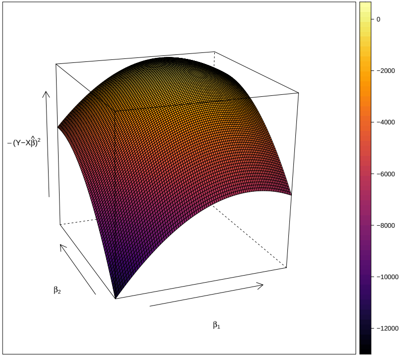
Geometrical point of view

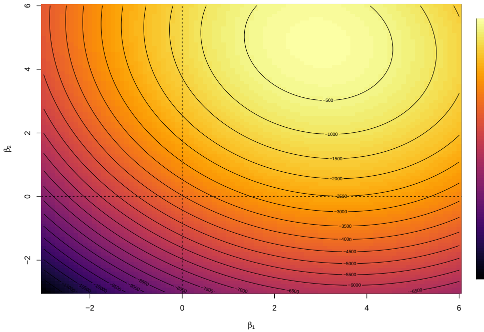
- Consider the following model with two centered predictors ($\mathbf{y}$ is centered)

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \boldsymbol{\varepsilon}$$

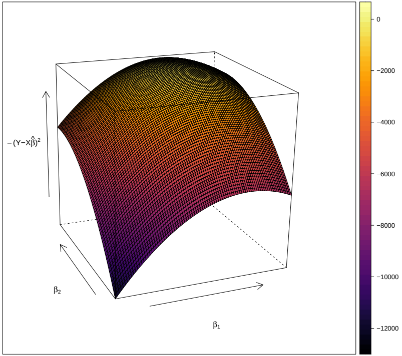


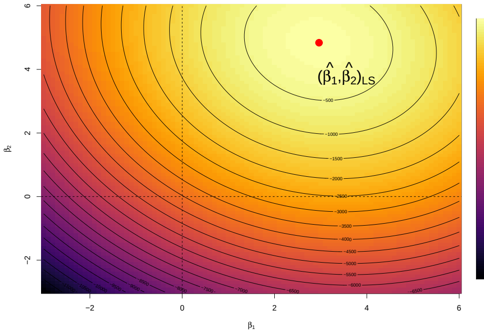
Contours of the least-squares regression surface



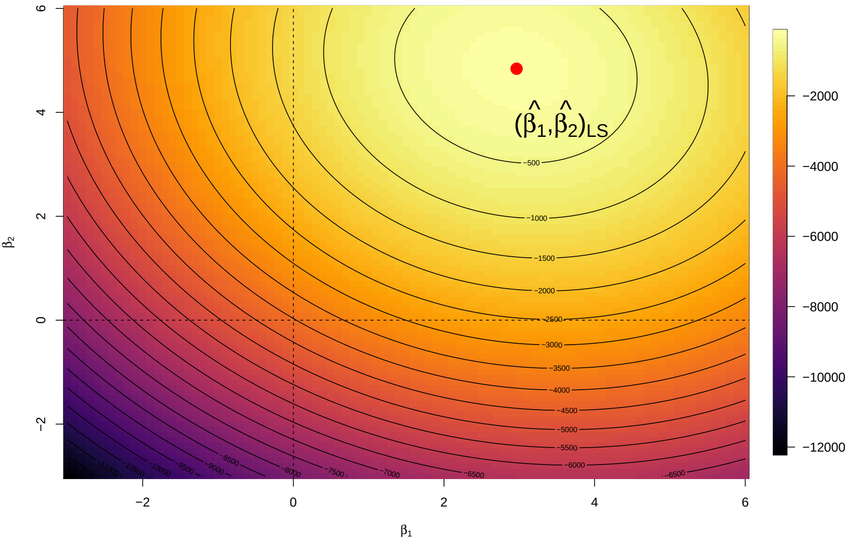


Contours of the least-squares regression surface



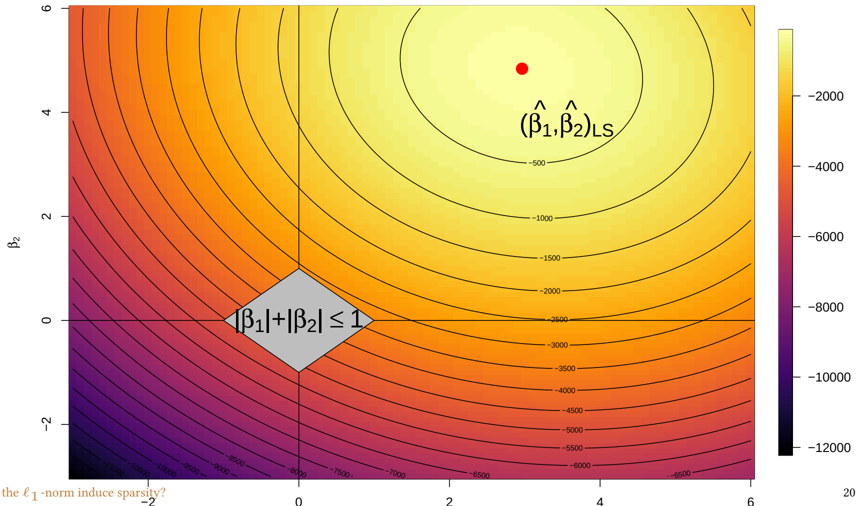


Contours of the least-squares regression surface

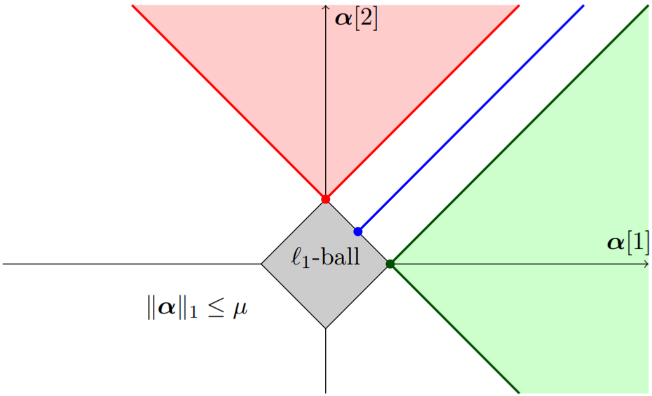


Constraint region of the lasso

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{2N}\|\boldsymbol{y}-\boldsymbol{X\beta}\|_2^2\right\}, \qquad ||\boldsymbol{\beta}||_1 \leq 1.$$

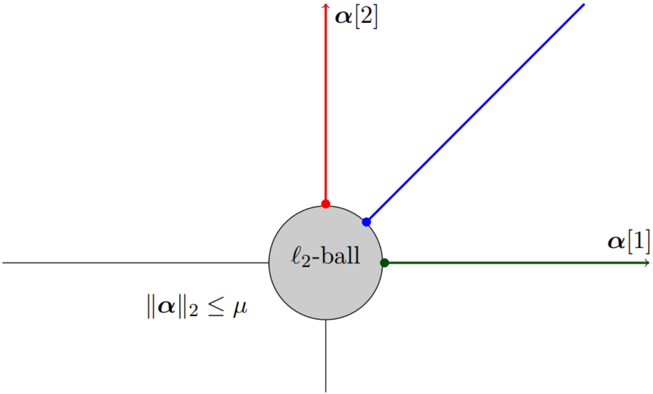


Effect of the Euclidean projection onto the ℓ_1-ball



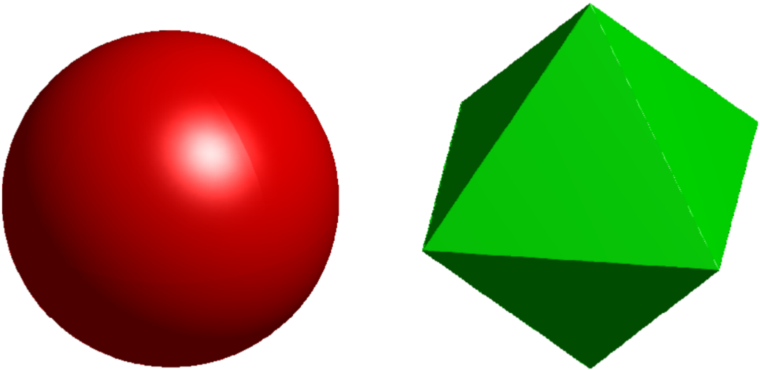
[1] Mairal, Bach and Ponce (2012). Sparse Modeling for Image and Vision Processing.

Effect of the Euclidean projection onto the ℓ_2-ball



[1] Mairal, Bach and Ponce (2012). Sparse Modeling for Image and Vision Processing.

Representation in three dimensions of the ℓ_1- and ℓ_2-balls



(a) ℓ_2-ball in 3D (b) ℓ_1-ball in 3D

[1] Mairal, Bach and Ponce (2012). Sparse Modeling for Image and Vision Processing.

Coordinate descent[1]

- The idea behind coordinate descent is, simply, to optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached

[1] Fu (1998), Friedman et al. (2007), Wu and Lange (2008)

Coordinate descent[1]

- The idea behind coordinate descent is, simply, to optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached

- Coordinate descent is particularly suitable for problems, like the lasso, that have a simple closed form solution in a single dimension but lack one in higher dimensions

[1] Fu (1998), Friedman et al. (2007), Wu and Lange (2008)

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

$$\widetilde{\beta}_j = \arg\min_{\beta_j} \mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = S_\lambda(\tilde{z}_j) = \begin{cases} \tilde{z}_j - \lambda, & \tilde{z}_j > \lambda \\ 0 & |\tilde{z}_j| \leq \lambda \\ \tilde{z}_j + \lambda & \tilde{z}_j < -\lambda \end{cases}$$

Coordinate descent

- Let us consider minimizing $\mathbf{Q}$ with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

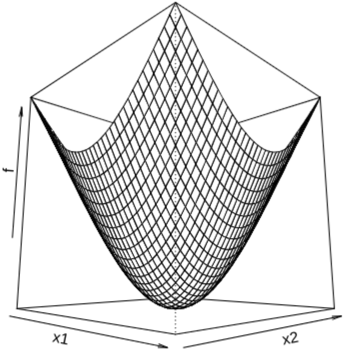
$$\mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j\right)^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k|$$

$$\widetilde{\beta}_j = \arg\min_{\beta_j} \mathbf{Q}(\beta_j|\boldsymbol{\beta}_{-j}) = S_\lambda(\tilde{z}_j) = \begin{cases} \tilde{z}_j - \lambda, & \tilde{z}_j > \lambda \\ 0 & |\tilde{z}_j| \leq \lambda \\ \tilde{z}_j + \lambda & \tilde{z}_j < -\lambda \end{cases}$$

- $\tilde{r}_{ij} = y_i - \sum_{k\neq j} x_{ik}\widetilde{\beta}_k$ $\qquad$ $\tilde{z}_j = n^{-1}\sum_{i=1}^{n} x_{ij}\tilde{r}_{ij}$

- $\{\tilde{r}_{ij}\}_{i=1}^{n}$ are the partial residuals with respect to the j^{th} predictor, and $\tilde{z}_j$ OLS estimator based on $\{\tilde{r}_{ij}, x_{ij}\}_{i=1}^{n}$

Why does this work?



A: Yes! Proof:

$$0 = \nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right)$$

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:

[1] Tseng and Yun (2009). A coordinate gradient descent method for nonsmooth separable minimization.

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:

  - the function $\mathcal{L}(\boldsymbol{\beta})$ is differentiable and
  - the penalty function $P_\lambda(\boldsymbol{\beta})$ is separable $\rightarrow P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$

[1] Tseng and Yun (2009). A coordinate gradient descent method for nonsmooth separable minimization.

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

  has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:
  - the function $\mathcal{L}(\boldsymbol{\beta})$ is differentiable and
  - the penalty function $P_\lambda(\boldsymbol{\beta})$ is separable $\rightarrow P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$
- Lasso-penalized linear regression satisfies both of these criteria

[1] Tseng and Yun (2009). A coordinate gradient descent method for nonsmooth separable minimization.

Convergence

- Numerical analysis of optimization problems of the form

$$\mathbf{Q}(\theta) = \mathcal{L}(\theta) + P(\theta)$$

has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that:
  - the function $\mathcal{L}(\boldsymbol{\beta})$ is differentiable and
  - the penalty function $P_\lambda(\boldsymbol{\beta})$ is separable $\rightarrow P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$
- Lasso-penalized linear regression satisfies both of these criteria
- Furthermore, because the lasso objective is a convex function, the sequence of the objective functions $\left\{ Q\left(\widetilde{\boldsymbol{\beta}}^{(s)}\right)\right\}$ converges to the global minimum

[1] Tseng and Yun (2009). A coordinate gradient descent method for nonsmooth separable minimization.

Coordinate descent, pathwise optimization, warm starts

- We are typically interested in determining $\widehat{\boldsymbol{\beta}}^{Lasso}$ for a range of values of λ, thereby obtaining the coefficient path

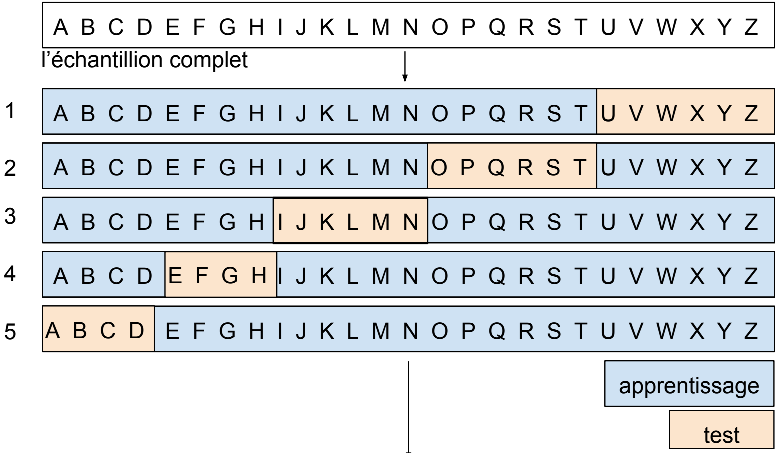
- In applying the coordinate descent algorithm to determine the lasso path, an efficient strategy is to compute solutions for decreasing values of λ, starting at $\lambda_{\max} = \max_{1 \leq j \leq p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$, the point at which all coefficients are 0

- Warm starts $\rightarrow$ By continuing along a decreasing grid of λ values, we can use the solutions $\widehat{\boldsymbol{\beta}}(\lambda_k)$ as initial values when solving for $\widehat{\boldsymbol{\beta}}(\lambda_{k+1})$

Sample Splitting

- As we have discussed, using the observed agreement between fitted values and the data is too optimistic; we require independent data to test predictive accuracy
- One solution we showed earlier, known as sample splitting, is to split the data set into two fractions, a training set and test set, using one portion to estimate $\widehat{\beta}$ (i.e., *train* the model) and the other to evaluate how well $\mathbf{X}_{\mathbf{test}}\widehat{\beta}$ predicts the observations in the second portion (i.e., *test* the model)
- The problem with this solution is that we rarely have so much data that we can freely part with half of it solely for the purpose of choosing λ

Cross-Validation



$$CV(\alpha) = \frac{1}{5}\sum_{v=1}^{5} MSE_v^{(test)}$$

Cross-validation: Details

1. Specify a grid of regularization parameter values $\Lambda = \{\lambda_1, \ldots, \lambda_K\}$
2. Divide the data into V roughly equal parts $D_1, \ldots, D_V$
3. For each $v = 1, \ldots, V$, compute the lasso solution path using the observations in $\{D_u, u \neq v\}$
4. For each $\lambda \in \Lambda$, compute the mean squared prediction error

$$\text{MSPE}_v(\lambda) = \frac{1}{n_v} \sum_{i \in D_v} \left\{ y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{-v}(\lambda) \right\}^2$$

where n_v is the number of observations in D_v, $\widehat{\boldsymbol{\beta}}_{-v}$ are the estimated regression coefficients trained on the observations in $\{D_u, u \neq v\}$, as well as

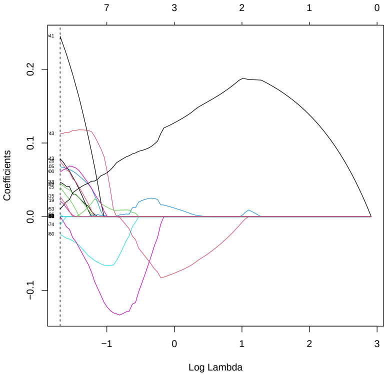
$$\text{CV}(\lambda) = \frac{1}{V} \sum_{v=1}^{V} \text{MSPE}_v(\lambda)$$

Cross-validation: Details

1. $\hat{\lambda}$ is taken to be the value that minimizes $\text{CV}(\lambda)$ and $\widehat{\boldsymbol{\beta}} \equiv \widehat{\boldsymbol{\beta}}(\hat{\lambda})$ the estimator of the regression coefficients
2. Note that
   - $\text{MSPE}_v(\lambda)$ is the mean squared prediction error for the model based on the training data $\{D_u, u \neq v\}$ in predicting the response variables in D_v
   - $\text{CV}(\lambda)$ is an estimate of the expected mean squared prediction error
3. Regardless of the number of cross-validation folds, each observation in the data appears exactly once in a test set

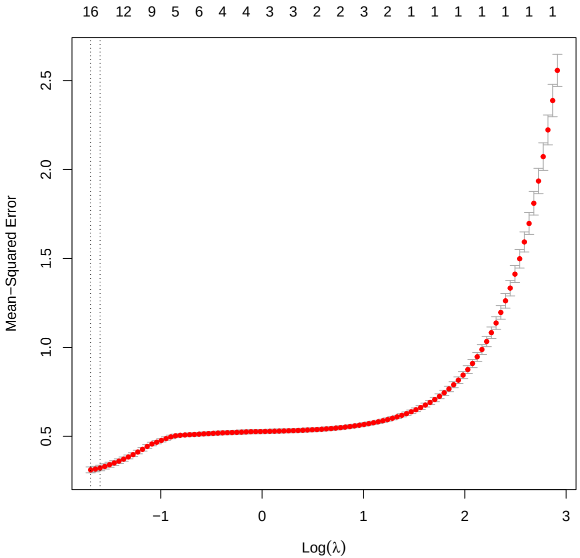
Lasso Solution Path on TCGA

```r
set.seed(101) # for reproducibility
sample <- sample.int(n = nrow(TCGA$X), size = floor(.80*nrow(TCGA$X)), replace = F) # 80% training / 20% testing
X.train <- TCGA$X[sample, ]
X.test  <- TCGA$X[-sample, ]
y.train <- TCGA$y[sample]
y.test <- TCGA$y[-sample]

# fit ridge regression on training
library(glmnet)
cvfit <- cv.glmnet(x = X.train, y = y.train, alpha = 1, nfolds = 5, intercept = FALSE)
fit <- cvfit$glmnet.fit
plot(fit, xvar = "lambda", label = TRUE)
abline(v = log(cvfit$lambda.min), lty = 2)
```



Lasso Cross-Validation on TCGA

```
plot(cvfit)
```



Backup Slides

Optimality Conditions

Score functions and penalized score functions

- In classical statistical theory, the derivative of the log-likelihood function $\mathcal{L}(\theta)$ is called the score function, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the likelihood equations (or score equations):

$$0 = \frac{\partial}{\partial \theta} \mathcal{L}(\theta)$$

Score functions and penalized score functions

- In classical statistical theory, the derivative of the log-likelihood function $\mathcal{L}(\theta)$ is called the score function, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the likelihood equations (or score equations):

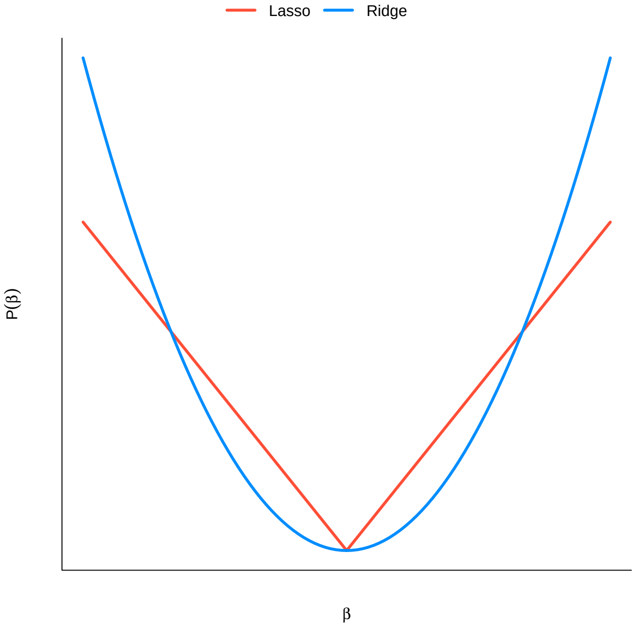
$$0 = \frac{\partial}{\partial \theta} \mathcal{L}(\theta)$$

- Extending this idea to penalized likelihoods involves taking the derivatives of objective functions of the form:

$$\mathbf{Q}(\theta) = \underbrace{\mathcal{L}(\theta)}_{\text{likelihood}} + \underbrace{P(\theta)}_{\text{penalty}}$$

yielding the penalized score function

Ridge vs. Lasso penalty



Penalized likelihood equations

- For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward

$$\widehat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_2^2$$

- For the lasso, the penalized likelihood is not differentiable - specifically, not differentiable at zero - and *subdifferentials* are needed to characterize them

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

http://myweb.uiowa.edu/pbreheny/7240/s19/notes/2-13.pdf

Penalized likelihood equations

- For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward

$$\widehat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_2^2$$

- For the lasso, the penalized likelihood is not differentiable - specifically, not differentiable at zero - and *subdifferentials* are needed to characterize them

$$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Letting $\partial\mathbf{Q}(\theta)$ denote the subdifferential of $\mathbf{Q}$, penalized likelihood equations are:

$$0 \in \partial\mathbf{Q}(\theta)$$

http://myweb.uiowa.edu/pbreheny/7240/s19/notes/2-13.pdf

Karush-Kuhn-Tucker (KKT) Conditions

- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions

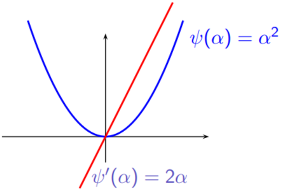
Karush-Kuhn-Tucker (KKT) Conditions

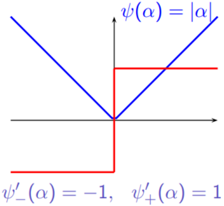
- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions
- For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution

Karush-Kuhn-Tucker (KKT) Conditions

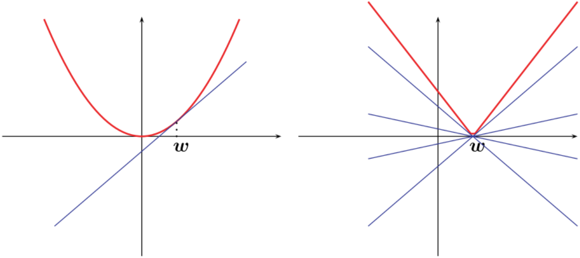
- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions
- For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution
- The idea is simple: to solve for $\widehat{\boldsymbol{\beta}}^{lasso}$, we simply replace the derivative with the subderivative and the likelihood with the penalized likelihood

Subgradients





The gradient of the ℓ_2-penalty vanishes when α get close to 0. On its differentiable part, the norm of the gradient of the ℓ_1-norm is constant.



(a) Smooth case. (b) Non-smooth case.

[1] Mairal, Bach and Ponce (2012). Sparse Modeling for Image and Vision Processing.

Subdifferential for $|x|$

The subdifferential for $f(x) = |x|$ is:

$$\partial|x| = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

$$\frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| \leq \lambda \qquad \widehat{\beta}_j = 0$$

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\begin{aligned} \frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) &= \lambda \text{sign}(\widehat{\beta}_j) & \widehat{\beta}_j \neq 0 \\ \frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| &\leq \lambda & \widehat{\beta}_j = 0 \end{aligned}$$

- In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model

KKT conditions for the lasso

- $$\widehat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \mathbf{Q}(\theta) = \arg\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\beta||_1$$

- Result: $\widehat{\boldsymbol{\beta}}^{lasso}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions:

$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$
$$\frac{1}{n}|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| \leq \lambda \qquad \widehat{\beta}_j = 0$$

- In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model

- When this correlation is below λ, $\widehat{\beta}_j = 0$

Some remarks

- If we set
$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$$
then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions
- That is, for any $\lambda \ge \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

Some remarks

- If we set
$$\lambda = \lambda_{\max} \equiv \max_{1 \leq j \leq p} \left| \mathbf{x}_j^T \mathbf{y} \right| / n$$
then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions

- That is, for any $\lambda \geq \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

- On the other hand, if we set $\lambda = 0$, the KKT conditions are simple the normal equations for OLS
$$\frac{1}{n} \mathbf{x}_j^\top (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0 \cdot \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

Some remarks

- If we set
$$\lambda = \lambda_{\max} \equiv \max_{1\leq j\leq p} \left|\mathbf{x}_j^T\mathbf{y}\right| / n$$
then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions

- That is, for any $\lambda \geq \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$

- On the other hand, if we set $\lambda = 0$, the KKT conditions are simple the normal equations for OLS
$$\frac{1}{n}\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0 \cdot \text{sign}(\widehat{\beta}_j) \qquad \widehat{\beta}_j \neq 0$$

- Thus, the coefficient path for the lasso starts at $\lambda_{\max}$ and continues until $\lambda = 0$ if $\mathbf{X}$ is full rank; otherwise the solution will fail to be unique for λ values below some point $\lambda_{\min}$

Recall the Lasso Solution in the Orthonormal Design

- When the design matrix $\mathbf{X}$ is orthonormal, i.e., $n^{-1}\mathbf{X}^\top\mathbf{X} = \mathbf{I}$, the lasso estimate is a **soft-thresholded** version of the least-squares (LS) estimate $\widehat{\beta}^{LS}$

$$\begin{aligned}\widehat{\beta}^{lasso} &= S_\lambda\left(\widehat{\beta}^{LS}\right) = \text{sign}\left(\widehat{\beta}^{LS}\right)\left(|\widehat{\beta}^{LS}| - \lambda\right)_+ \\ &= \begin{cases}\widehat{\beta}^{LS} - \lambda, & \widehat{\beta}^{LS} > \lambda \\ 0 & |\widehat{\beta}^{LS}| \leq \lambda \\ \widehat{\beta}^{LS} + \lambda & \widehat{\beta}^{LS} \leq -\lambda\end{cases}\end{aligned}$$

- where $\widehat{\beta}^{LS} = \mathbf{x}_j^\top\mathbf{y}/n$

Probability that $\hat{\beta}_j = 0$

- With soft thresholding, it is clear that the lasso has a positive probability of yielding an estimate of exactly 0 - in other words, of producing a sparse solution

- Specifically, the probability of dropping $\mathbf{x}_j$ from the model is $\mathbb{P}\left(\left|\beta_j^{LS}\right| \leq \lambda\right)$

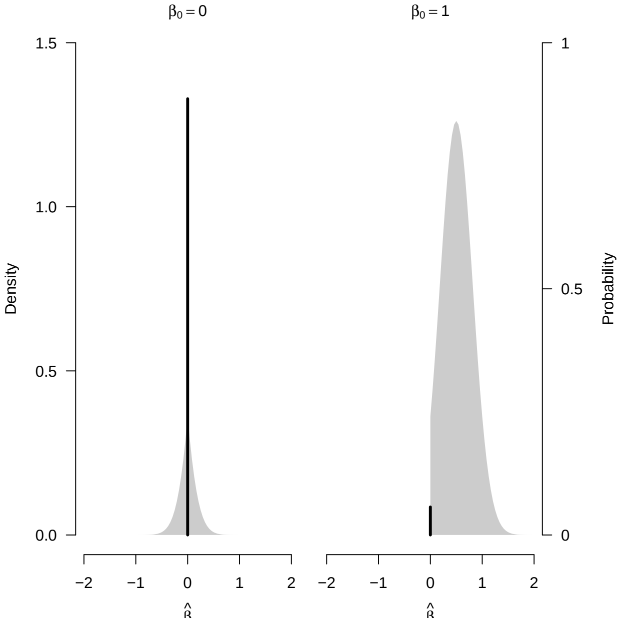
- Under the assumption that $\epsilon_i \overset{\perp\!\!\!\perp}{\sim} \mathrm{N}\left(0, \sigma^2\right)$, we have $\beta_j^{LS} \sim \mathcal{N}(\beta, \sigma^2/n)$ and

$$\mathbb{P}\left(\widehat{\beta}_j(\lambda) = 0\right) = \Phi\left(\frac{\lambda - \beta}{\sigma/\sqrt{n}}\right) - \Phi\left(\frac{-\lambda - \beta}{\sigma/\sqrt{n}}\right)$$

where Φ is the Gaussian CDF

Sampling Distribution

For $\sigma = 1$, $n = 10$, and $\lambda = 1/2$:



Why standard inference is invalid?

- This sampling distribution is very different from that of a classical MLE:
  - The distribution is mixed: a portion is continuously distributed, but there is also a point mass at zero
  - The continuous portion is not normally distributed
  - The distribution is asymmetric (unless $\beta = 0$)
  - The distribution is not centered at the true value of β