Sparse Linear Regression: the LASSO

Andrew Nobel

April, 2021

High Dimensional Linear Regression

Data: Paired observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$

- Centering: assume $\sum_{i=1}^n \mathbf{x}_i = 0$ and $\sum_{i=1}^n y_i = 0$
- Design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and response vector $\mathbf{y} \in \mathbb{R}^n$
- Interested in fitting linear model $\mathbf{y} = \mathbf{X}\,\beta + \varepsilon$

Common situation: More features than variables, that is, $p >> n$

- Common in genomics, biomedicine, climatology
- Requires regularization

Sparsity

Assumption: Only a small number s of the p available features are related to the response; the other features are unimportant

- Sparsity assumption approximately true for many data sets
- Implies true coefficient vector β has only s non-zero components
- Shift in focus: value and *identity* of non-zero coefficients of interest
- Number s referred to as the sparsity of the model

Sparse Linear Regression

Common goals

- Prediction: find sparse $\hat{\beta}$ so that $\mathbf{x}^t\hat{\beta}$ close to y for new pair $(\mathbf{x}, y)$
- Feature selection: identify the "true" features, i.e., $\{j : \beta_j \neq 0\}$

Issue: For OLS and Ridge all estimated coefficients are non-zero

LASSO: Least absolute shrinkage and selection operator

- Replace ridge penalty $\sum_{j=1}^{p} \beta_j^2$ by ℓ_1-penalty $\sum_{j=1}^{p} |\beta_j|$
- The ℓ_1 penalty enforces sparsity but preserves convexity

LASSO Regression

Procedure: Given design matrix $\mathbf{X}$, response vector $\mathbf{y}$, and parameter $\lambda \geq 0$, find coefficient vector $\hat{\beta}_\lambda^{\text{LASSO}}$ minimizing

$$\tilde{R}_{n,\lambda}(\beta) \;=\; \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2 \;+\; \lambda\,||\beta||_1$$

- $||\mathbf{y} - \mathbf{X}\beta||^2$ measures fit of linear model
- $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ measures magnitude of coefficient vector
- Parameter λ controls tradeoff between fit and magnitude

Key fact: The ℓ_1-penalty forces some coefficients $\hat{\beta}_\lambda^{\text{LASSO}}$ to be *exactly* zero

- Increasing λ tends to increase number of zero coefficients in $\hat{\beta}_\lambda^{\text{LASSO}}$

LASSO Estimation as a Convex Program

Fact: For every $\lambda \geq 0$ objective $\tilde{R}_{n,\lambda}(\beta)$ is a convex function of β

Fact: Minimizing $\tilde{R}_{n,\lambda}(\beta)$ is Lagrangian form of the mathematical program

$$\min f(\beta) = ||\mathbf{y} - \mathbf{X}\beta||^2 \text{ subject to } ||\beta||_1 \leq t$$

where t depends on λ. Objective function and constraint set are convex.

Upshot: Zero-ing property follows from *geometry* of the ℓ_1-penalty

Geometry of the L_1 Penalty (from ESL)

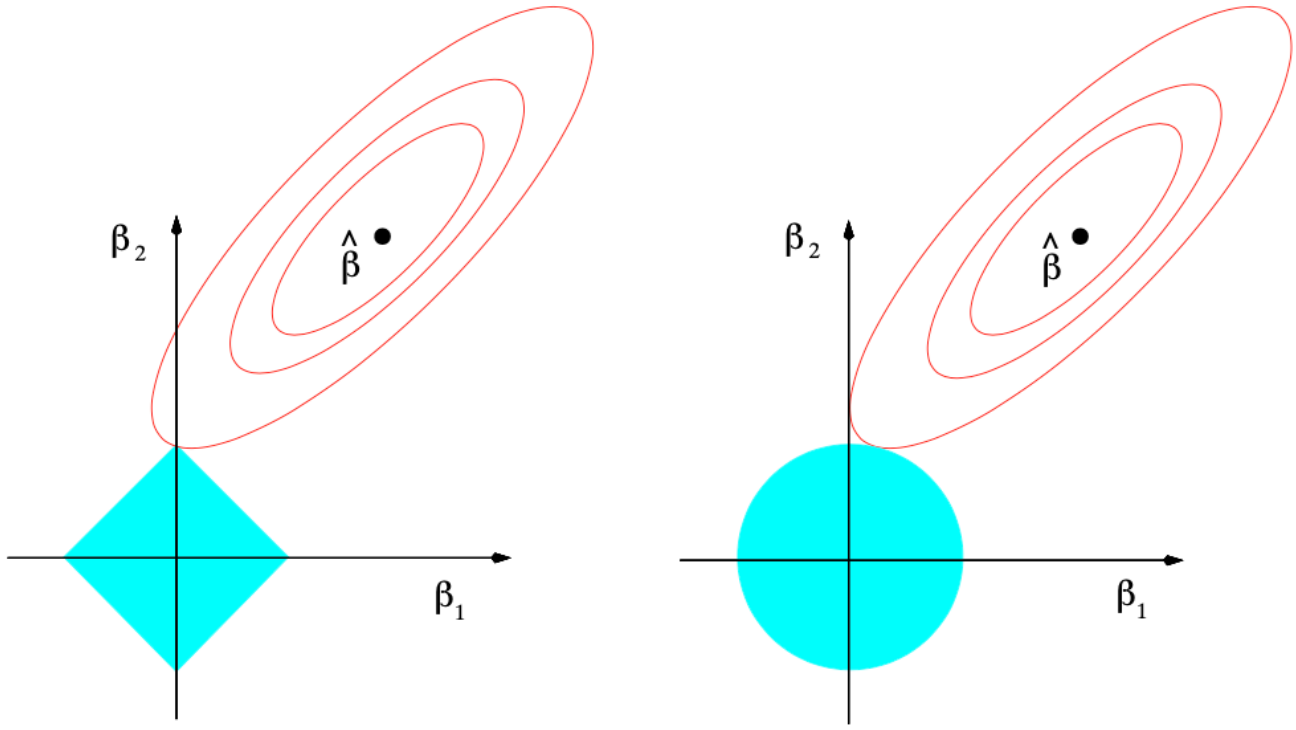


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are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

Selecting Penalty Parameter

Note: Different parameters λ give different solutions $\hat{\beta}_\lambda$. How to choose λ?

- Fix "grid" $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ of parameter values

Approach 1. Independent training set D_n and test set D_m

- Find vectors $\hat{\beta}_{\lambda_1}, \ldots, \hat{\beta}_{\lambda_N}$ using training set D_n
- Select vector $\hat{\beta}_{\lambda_\ell}$ minimizing test error $\hat{R}_m(\beta)$

Approach 2. Cross-validation

- For each $1 \leq \ell \leq N$ evaluate cross-validated risk $\hat{R}^{\text{k-CV}}(\text{LASSO}(\lambda_\ell))$
- Select vector $\hat{\beta}_{\lambda_\ell}$ for which λ_ℓ minimizes cross-validated risk

Theoretical choice of λ

Basic Idea: If $\mathbf{y} = \mathbf{X}\,\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I)$

1. Find a good estimate $\hat{\sigma}^2$ of the noise variance σ^2

2. Choose parameter value

$$\lambda \;=\; \sqrt{\frac{2\hat{\sigma}^2 \log p}{n}}$$

Estimating the Penalty Parameter, cont

Idea: If response vector $\mathbf{y}$ is independent of $\mathbf{X}$ then β should be 0

Procedure: Do the following 20-30 times

1. Randomly permute the components of $\mathbf{y}$ to get a "dummy response" $\tilde{\mathbf{y}}$
2. Apply LASSO procedure to $(\mathbf{X}, \tilde{\mathbf{y}})$ with different values of λ
3. Let $\tilde{\lambda} =$ smallest λ such that $\hat{\beta}_{\lambda}^{\text{LASSO}}(\mathbf{X}, \tilde{\mathbf{y}}) = 0$

Estimate the penalty parameter $\hat{\lambda}$ by median of the $\tilde{\lambda}$'s

Example: B-cell gene expression data

Background: Data from Basso et al. 2005, Affymetrix microarrays

1. Samples: Samples of $n = 211$ normal and tumor tissue
2. Feature vector: Expression measurements of $p = 6,249$ genes
3. Response: Expression of single `ADA` gene

Question: How does the expression of `ADA` depend on the expression of the 6248 other genes?

OLS Solution

```
R > summary(my_model)

Call:
lm(formula = ADA ~ ., data = gene_expressions)

Residuals:
ALL 211 residuals are 0: no residual degrees of freedom!

Coefficients: (6038 not defined because of singularities)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -412.40983         NA      NA       NA
CDH2             -1.86356         NA      NA       NA
MED6              7.10850         NA      NA       NA
NR2E3            -1.40334         NA      NA       NA
ACOT8             3.48331         NA      NA       NA
ABI1             -5.88529         NA      NA       NA
GNPDA1            0.28055         NA      NA       NA
TANK             -6.02434         NA      NA       NA
HGC6.3           -0.79016         NA      NA       NA
C1orf68          -1.21752         NA      NA       NA
LOC100129361      0.20853         NA      NA       NA
OLFM1                  NA         NA      NA       NA
TIMM17A                NA         NA      NA       NA
N4BP2L2                NA         NA      NA       NA
MCRS1                  NA         NA      NA       NA
 [ reached getOption("max.print") -- omitted 6229 rows ]

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:    NaN
F-statistic:   NaN on 210 and 0 DF,  p-value: NA
```

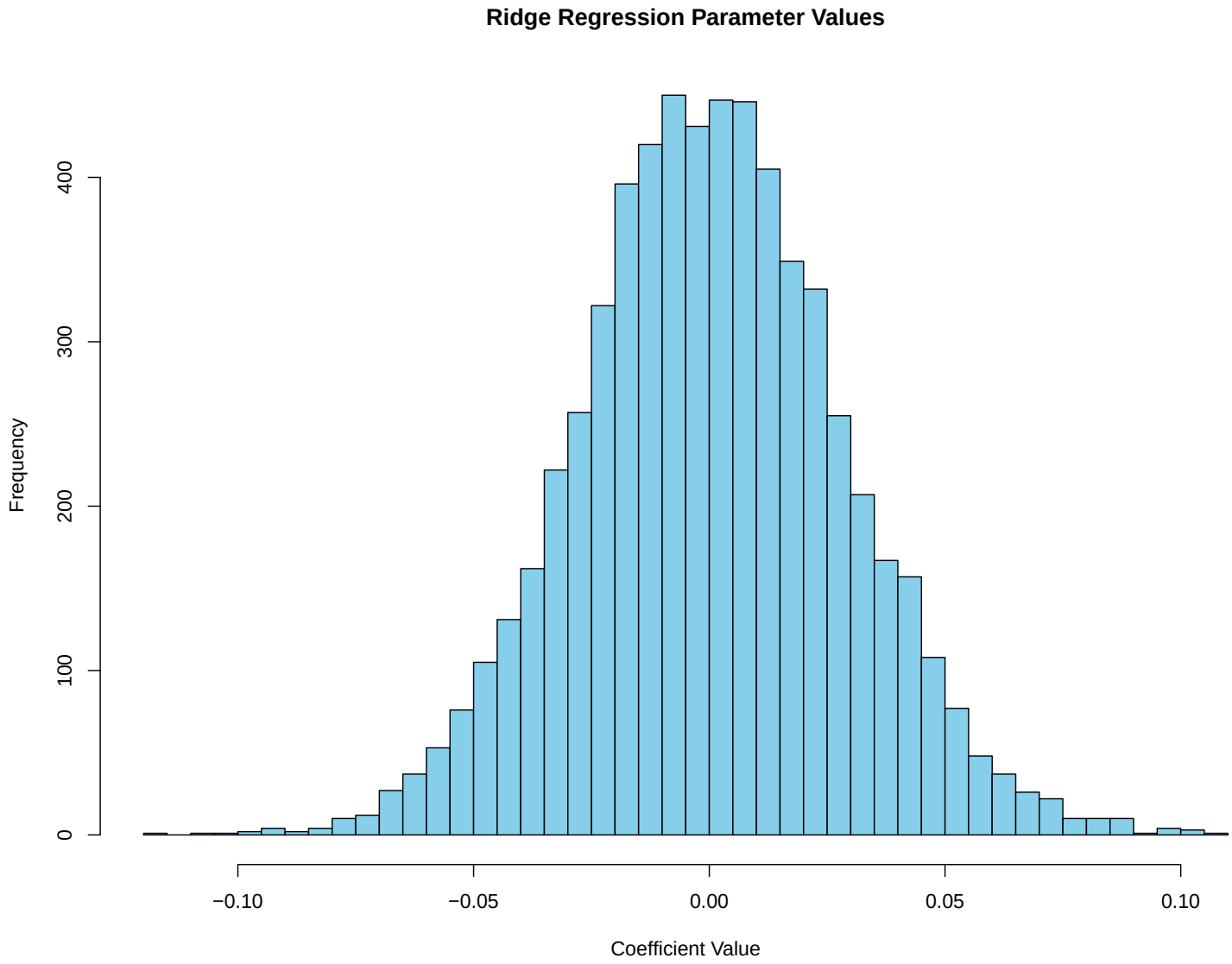
Ridge Solution

- `R` function selects penalty parameter λ based on the variance explained by the first 8 PCs. [1]
- Note: coefficient estimates for every feature are non-zero

```
R > ridge.fit = linearRidge(ADA~., data = gene_expressions)
R > ridge.fit$coef[,"nPCs8"]
          CDH2           MED6          NR2E3          ACOT8
 1.390812e-02 -3.920405e-02  2.380735e-02 -1.577109e-02

  ABI1         GNPDA1           TANK         HGC6.3
 1.902280e-04 -5.952662e-03  1.141530e-02  5.133231e-02

       C1orf68  LOC100129361           CD24          HDAC5
-5.509269e-02 -3.030931e-02 -4.909134e-02 -5.526016e-03

PDCD6         BCL2L11          SH2B3           GNE
1.990365e-02  2.167638e-02 -3.561387e-02 -1.047401e-01
 [ reached getOption("max.print") -- omitted 6232 entries ]

 R > length(which(coef(ridge.fit) == 0))
[1] 0
```

[1] From Cule & De Iorio (2012). A semi-automatic method to guide the choice of ridge parameter in ridge regression

Histogram of Ridge Coefficients



LASSO Solution

- R function selects penalty parameter λ using 10-fold CV

```
R > lasso.fit = Lasso(as.matrix(gene_expressions)[,-1], as.matrix(gene_
    expressions)[,1], fix.lambda = FALSE)
R > lasso.fit
$beta0
[1] 11.85041
$beta
  [1]  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
       0.000000000  0.000000000  0.000000000  0.000000000
 [10]  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
       0.000000000 -0.143888739  0.000000000  0.000000000
 [19]  0.113587487  0.000000000  0.000000000  0.000000000  0.000000000
       0.000000000  0.000000000  0.000000000  0.000000000
 [28]  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
       0.000000000  0.000000000  0.000000000  0.000000000
 [ reached getOption("max.print") -- omitted 6212 entries ]
$lambda
[1] 0.09383066
```

LASSO Solution Cont.

- LASSO sets most coefficients to zero. Only 84 are non-zero.

```
R > length(which(lasso.fit$beta != 0))
[1] 84
R > colnames(gene_expressions)[which(lasso.fit$beta != 0)]
 [1] "SH2B3"      "PIGK"       "ACTR2"      "MBNL2"      "POP7"       "RRAGB"
                "RBM14"      "FBLN5"      "RAD51AP1"   "RALBP1"
[11] "GLMN"       "FILIP1L"    "AP2S1"      "CLCN4"      "ZNF384"     "DLG1"
                "AGXT"       "EPHA7"      "F12"        "FABP4"
[21] "FCN1"       "ABCF1"      "TMCC1"      "PDS5B"      "ZHX3"       "SEPT6"
                "RRS1"       "SCFD1"      "MCF2L"      "KHNYN"
[31] "COG4"       "ODZ4"       "GCG"        "PELP1"      "AHDC1"      "RNF115"
                "GNAT2"      "ANGPT2"     "GUCA2A"     "GZMB"
[41] "HBD"        "HLA.DPA1"   "HSD17B1"    "IDH3B"      "ACADS"      "AQP1"
                "ITGA1"      "L1CAM"      "ST20"       "MSMB"
[51] "MYO6"       "NFATC1"     "KRT76"      "FAM8A1"     "PIK3C2B"    "SSH1"
                "ZNF821"     "PSG11"      "PTHLH"      "GATAD1"
[61] "RAD52"      "RGS16"      "BCL9"       "RPS4X"      "RPS27"      "CCL5"
                "SLC4A3"     "SNAPC1"     "BTG1"       "UBE2E1"
[71] "VRK1"       "ZNF23"      "ZNF76"      "DDX39B"     "ACTL6A"     "VNN2"
                "WASF1"      "CD1D"       "MS4A3"      "NRXN1"
[81] "TMPRSS11D"  "POLR1C"    "MDC1"       "TMED10"
```

LASSO Solution Cont.

```
1 R > model <- cv.glmnet(as.matrix(gene_expressions)[,-1], as.matrix(gene_
      expressions)[,1], standardize=TRUE)
2 R > plot(model$glmnet.fit, "lambda", label=TRUE)
```