Statistical Learning

Penalized Linear Regression: Part I

Spring 2024

Shrinkage Methods

Motivation

- Best subset selection
  - Computationally expensive
  - Not feasible when p is large
- Forward/backward selection
  - No guarantee to find the best global submodel
  - The selection process is discrete (“add” or “drop”), often leads to high variance.
- Shrinkage methods
  - A continuous process, does not suffer from high variability

Motivation

- The OLS estimator is a linear function of $\mathbf{y}$, and it is the BLUE.
- But there can be (and often exist) biased estimators with smaller variance
- Recall that the prediction accuracy is

$$\text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

and choosing estimators often involves the bias-variance trade-off.

- Generally, by regularizing (shrinking, penalizing) the estimator in some way, its variance can be reduced; if the corresponding increase in bias is small, we have better prediction accuracy

Shrinkage Methods

- Part I
  - ℓ_2 penalty: Ridge regression
  - ℓ_1 penalty: Lasso
  - Connecting the two: Elastic net; Bridge penalty
- Part II
  - Bias reduction: adaptive Lasso, SCAD, MCP
  - Consistency of penalized methods
  - Penalties for special data structures: grouped lasso, fused lasso

Ridge Regression

A Motivating Example

```
> library(MASS)
> set.seed(1)
> n = 30
>
> # highly correlated variables
> X = mvrnorm(n, c(0, 0), matrix(c(1,0.999, 0.999, 1), 2,2))
> y = rnorm(n, mean=1 + X[,1] + X[,2])
>
> # compare parameter estimates
> summary(lm(y~X))$coef
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)   1.038007  0.1647551  6.300302 9.627026e-07
X1          -11.272638  4.6402098 -2.429338 2.205727e-02
X2           13.265586  4.6315269  2.864193 7.993486e-03
> lm.ridge(y~X, lambda=5)
                 X1        X2
1.1214448 0.8770568 0.9836474
```

Ridge Regression

Penalizing the square of the coefficients

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

- Hoerl and Kennard (1970); Tikhonov (1943)
- $\lambda \geq 0$ is a tuning parameter (penalty level), it controls the amount of shrinkage.
- The coefficients $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ are shrunken towards 0.

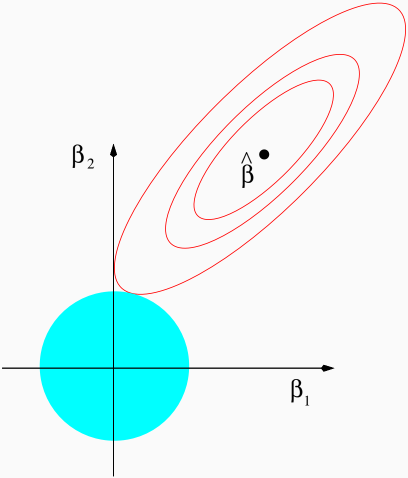
Ridge Regression

An equivalent formulation is given by

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

- There is a one-to-one correspondence between the parameters λ and s
- This is due to the KKT conditions

Ridge Regression



Ridge constrained solution

Ridge Regression

- Ridge regression is mainly used to address multi-collinearity problem in high-dimensional data
- When there are many correlated variables, a wildly large positive coefficient on one variable can be canceled by a similarly large negative coefficient on its correlated cousin.
- Ridge regression alleviate this problem by imposing a size constraint

Ridge Regression

- How to derive the solution $\widehat{\beta}^{\text{ridge}}$
- Degrees of freedom
- Tuning parameter selection
- Connections with other methods

Solution for Ridge Regression

- For a fixed tuning parameter λ, we want to minimize

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}$$

- Take derivative with respect to $\boldsymbol{\beta}$ and set to zero, we have the solution of the ridge regression

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$$

- $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is still a linear estimator

Solution for Ridge Regression

- This is similar to the ordinary least squares solution, but with the addition of a "ridge" down the diagonal
- $\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}$ is always invertible, hence $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is unique
- As $\lambda \to 0$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \widehat{\boldsymbol{\beta}}^{\text{ols}}$
- As $\lambda \to \infty$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \mathbf{0}$

Bias and Variance of Ridge Regression

- When $\widehat{\beta}^{\text{ols}}$ exists, we can also write

$$\begin{aligned}\widehat{\beta}^{\text{ridge}} &= (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \\ &= (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \\ &= (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})\widehat{\beta}^{\text{ols}} \\ &= \mathbf{Z}\widehat{\beta}^{\text{ols}}\end{aligned}$$

  where $\mathbf{Z} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})$.

- How does this shrink $\widehat{\beta}^{\text{ols}}$?

Bias and Variance of Ridge Regression

- The variance of $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is

$$\mathsf{Var}\big(\widehat{\boldsymbol{\beta}}^{\text{ridge}}\big) = \sigma^2(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}$$

- The total variance $\sum_j \mathsf{Var}(\widehat{\beta}_j^{\text{ridge}})$ is a monotone decreasing function of λ.

Bias and Variance of Ridge Regression

- The ridge estimator is biased

$$E(\widehat{\boldsymbol{\beta}}^{\text{ridge}}) = \mathbf{Z}\boldsymbol{\beta}$$

where $\mathbf{Z} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathsf{T}}\mathbf{X})$.

- The total squared bias $\sum_j \text{Bias}^2(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}})$ is a monotone increasing function of λ.

Understanding the Shrinkage

- Suppose we have orthogonal design matrix ($\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{I}$), then

$$
\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\text{ridge}} =& (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})\widehat{\boldsymbol{\beta}}^{\text{ols}} \\
=& (\mathbf{I} + \lambda\mathbf{I})^{-1}\widehat{\boldsymbol{\beta}}^{\text{ols}} \\
=& (1 + \lambda)^{-1}\widehat{\boldsymbol{\beta}}^{\text{ols}},
\end{aligned}
$$

meaning that we just need to shrink $\widehat{\boldsymbol{\beta}}^{\text{ols}}$ by $(1+\lambda)^{-1}$, i.e.,

$$
\widehat{\boldsymbol{\beta}}_j^{\text{ridge}} = \frac{1}{1+\lambda}\widehat{\boldsymbol{\beta}}_j^{\text{ols}}.
$$

Understanding the Shrinkage

- $\text{Var}\big(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}\big) = \frac{1}{(1+\lambda)^2}\text{Var}\big(\widehat{\boldsymbol{\beta}}_j^{\text{ols}}\big)$ (reduced from OLS!)
- $\text{Bias}\big(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}\big) = \frac{-\lambda}{1+\lambda}\boldsymbol{\beta}_j$ (not unbiased!)
- There always exists a λ such that the MSE of $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is smaller than $\widehat{\boldsymbol{\beta}}^{\text{ols}}$

Understanding the Shrinkage

- Let's take a singular value decomposition (SVD) of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$$

  where

  - $\mathbf{U}_{n\times n}$: columns $\mathbf{u}_j$'s form an orthonormal basis for the column space of $\mathbf{X}$, $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}$
  - $\mathbf{V}_{p\times p}$: orthogonal matrix with $\mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I}$
  - $\mathbf{D}_{n\times p}$: matrix with diagonal entries $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$ being the singular values of $\mathbf{X}$
  - Sometimes we can write $\mathbf{X} = \mathbf{F}\mathbf{V}^{\mathsf{T}}$ where each columns of $\mathbf{F}_{n\times p} = \mathbf{U}\mathbf{D}$ is the so-called principal components and each column of $\mathbf{V}$ is a principal direction.

Understanding the Shrinkage

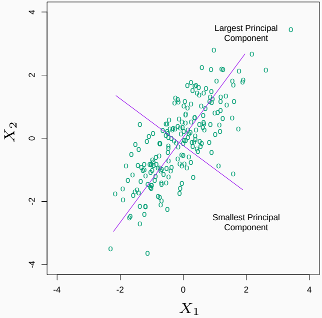


FIGURE 3.9. *Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects* $\mathbf{y}$ *onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.*

Understanding the Shrinkage

- We can view PCA as (assuming $\mathbf{X}$ centered)

$$\widehat{\Sigma} = \frac{1}{n}\mathbf{X}^\mathsf{T}\mathbf{X} = \frac{1}{n}\mathbf{V}\mathbf{D}^2\mathbf{V}^\mathsf{T}$$

  where $\mathbf{D}^2 = \text{diag}(d_1^2, d_2^2, \ldots, d_p^2)$.

- The jth principal component is $\mathbf{z}_j = \mathbf{X}\mathbf{v}_j = d_j\mathbf{u}_j$ with $\text{Var}(\mathbf{z}_j) = d_j^2/n$.

- $\mathbf{u}_j$ is the normalized jth principal component of $\mathbf{X}$

- The ridge estimate $\widehat{\mathbf{y}}^{\text{ ridge}}$ is

$$\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \sum_{j=1}^{p}\mathbf{u}_j\left(\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^\mathsf{T}\mathbf{y}\right)$$

Understanding the Shrinkage

- Hence, ridge regression can be understood as
  1. (1) Perform principle component analysis of $\mathbf{X}$
  2. (2) Project $\mathbf{y}$ onto the principal components: $\mathbf{u}_j^\intercal \mathbf{y}$ for each j
  3. (3) Shrink the projections by the factor $d_j^2/(d_j^2 + \lambda)$
- Directions with smaller eigenvalues d_j^2 get more shrinkage.
- The final ridge estimate of $\mathbf{y}$ is a sum of the p shrunk projections.

Degrees of Freedom for Ridge Regression

- Although $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is p-dimensional, it does not use the full potential of the p covariates due to the shrinkage.
- For example, when $\lambda \to \infty$, all the parameter estimates are shrunk to 0. Intuitively, the d.f. is almost 0.
- If λ is 0, then it reduces to the OLS with d.f. $= p$
- The d.f. of a ridge regression is between 0 and p

Degrees of Freedom for Ridge Regression

- Recall our definition of degrees of freedom (d.f.) in the kNN example:

$$\text{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{Cov}(\widehat{y}_i, y_i) = \frac{1}{\sigma^2} \text{Trace}\Big(\text{Cov}(\widehat{\mathbf{Y}}, \mathbf{Y})\Big)$$

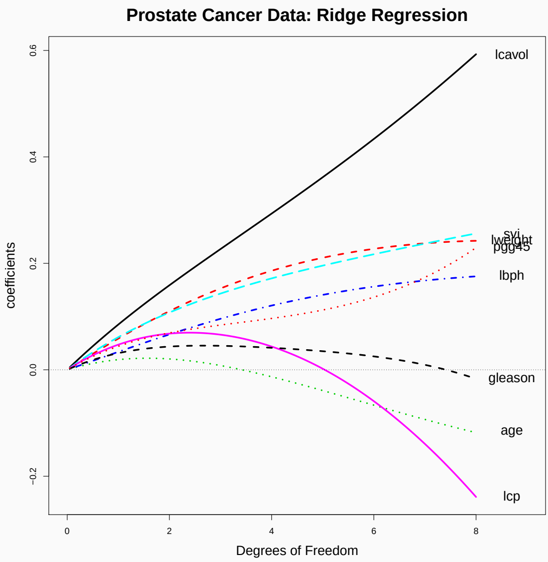
- For ridge regression, we have

$$\widehat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$$

- Then the effective d.f. is

$$\text{df}(\lambda) = \text{Trace}\big(\mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\big) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

Prostate Cancer Example



Selecting the Tuning Parameter λ

- The R command lm.ridge (from MASS package) returns GCV, which can be used to select λ.
- glmnet can also fit ridge regression by setting $\alpha = 0$
- The leave-one-out cross-validation (CV) error? In the context of linear regression
  1. Hold the ith sample (x_i, y_i) as a test sample, fit a regression model based on the remaining $(n-1)$ observations, and denote the coefficient as $\widehat{\boldsymbol{\beta}}_{[-i]}$
  2. Calculate the prediction error on the holdout sample $(y_i - x_i^\mathsf{T}\widehat{\boldsymbol{\beta}}_{[-i]})^2$
  3. Repeat for every sample and

$$\text{CV} = \sum_{i=1}^{n} \left(y_i - x_i^\mathsf{T}\widehat{\boldsymbol{\beta}}_{[-i]}\right)^2$$

Bayesian Interpretation

- The ridge regression solution can be viewed from a Bayesian perspective, where we give a prior distribution $\boldsymbol{\beta} \sim \mathcal{N}(0, \sigma^2/\lambda)$.
- Then the posterior distribution of $\boldsymbol{\beta}$ is normal, with posterior mean

$$\left(\mathbf{X}^\intercal \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^\intercal \mathbf{y}.$$

Notes on the scale of predictors

The solution is not invariant with respect to the scale of the predictors!

We normalize the columns of the design matrix $\mathbf{X}$ such that they have unit sample variance. We further center the data, that is, both y and the columns of $\mathbf{X}$ have mean zero. Then, we can fit a linear regression model without an intercept (we don't penalize the intercept). The parameters on the original scale can be reversely solved.

Some packages (e.g., "glmnet") in R handles the centering and scaling automatically: it will do the transformation before running the algorithm, and then will transform the obtained results back to the original scale.

Lasso: Least Absolute Shrinkage and Selection Operator

Motivation

- The ridge regression shrinks the coefficients towards 0, however, they are not exactly zero. Hence, we haven't achieved any "selection" of variables.
- Parsimony: we would like to select a small subset of predictions. Forward/backword/subset does not provide global solution and can be myopic at each step.
- Lasso provides a continuous process. We will discuss:
    - The formulation, the solution when $\mathbf{X}$ is orthogonal
    - Computation methods and solution path

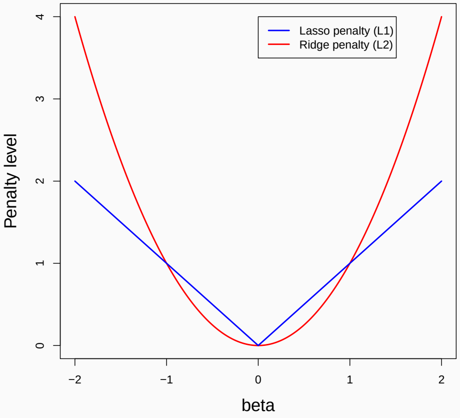
Lasso

Least absolute shrinkage and selection operator (Tibshirani 1996)

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Shrinkage of the ℓ_1 norm of the parameters
- Selection of parameters, some will be exactly 0

Lasso



Lasso penalty (L1)
Ridge penalty (L2)
Penalty level
beta
0
1
2
3
4
−2
−1
0
1
2

Lasso

Lasso Under Orthogonal Design

Again, it will be helpful to view Lasso assuming orthogonal design, i.e., $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{I}_p$. Then

$$\begin{aligned}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} + \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2 + \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2\end{aligned}$$

where the cross product term

$$2(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}})^\mathsf{T}(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 2\mathbf{r}^\mathsf{T}(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 0,$$

since the second term is in the column space of $\mathbf{X}$, while $\mathbf{r}$ is orthogonal to that space.

Lasso Under Orthogonal Design

- Since $\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2$ is not a function of $\boldsymbol{\beta}$, we minimize

$$\|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Then, we have

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\text{lasso}} &= \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{p} (\widehat{\beta}_j^{\text{ols}} - \beta_j)^2 + \lambda|\beta_j|.
\end{aligned}$$

- This means we can solve the lasso estimators individually from the OLS estimator.

Lasso Under Orthogonal Design

- Each of the β_j's is essentially solving for

$$\arg\min_{x} \ (x-a)^2 + \lambda|x|, \quad \lambda > 0$$

- The solution is simply

$$\begin{aligned}
\widehat{\beta}_j^{\text{lasso}} &= \begin{cases} \widehat{\beta}_j^{\text{ols}} - \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} > \lambda/2 \\ 0 & \text{if} \quad |\widehat{\beta}_j^{\text{ols}}| \leq \lambda/2 \\ \widehat{\beta}_j^{\text{ols}} + \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} < -\lambda/2 \end{cases} \\
&= \text{sign}(\widehat{\beta}_j^{\text{ols}}) \Big(|\widehat{\beta}_j^{\text{ols}}| - \lambda/2 \Big)_+ \\
&\doteq \text{SoftTH}(\beta_j^{\text{ols}}, \lambda)
\end{aligned}$$

- A large λ will shrink some of the coefficients to exactly zero, which achieves "variable selection".

Equivalent Formulation

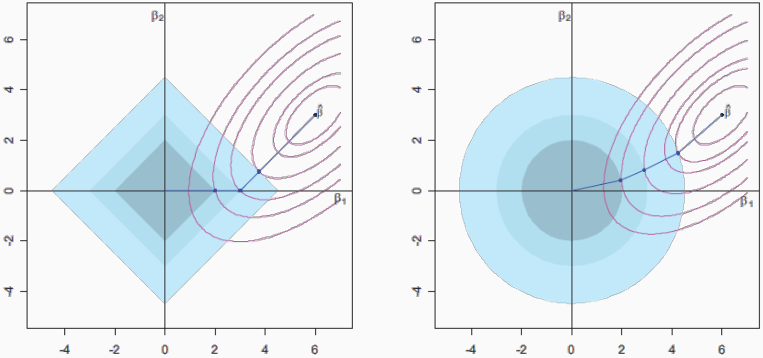
- The Lasso optimization problem is equivalent to

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

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

- Each value of λ corresponds to an unique value of s.
- Compare Ridge and Lasso?

Linear Regression



Comparing Lasso and Ridge solutions

Computation of Lasso Solution

- Shooting algorithm (Fu 1998): sequentially and iteratively update each parameter estimate (coordinate descent algorithm).
- Least angle regression (Efron et al. 2004)
  - The path of solutions is piecewise linear in λ
  - Cost is approximately one least-squares calculation $\mathcal{O}(np^2)$
  - Connection with stagewise regression
- Coordinate descent (Friedman et al. 2010): The most popular implementation, glmnet package; $\mathcal{O}(np)$
  - Also provides the solution path for the entire sequence of λ, starting with the largest one
  - Use the previous estimation of $\boldsymbol{\beta}$ as a warm start for smaller λ

ℓ_q Penalties

- Ridge is ℓ_2 penalty
- Lasso is ℓ_1 penalty
- Best subset is ℓ_0 penalty
- Bridge penalty is ℓ_q normal

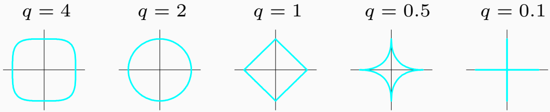


FIGURE 3.12. *Contours of constant value of $\sum_j |\beta_j|^q$ for given values of q.*

- Elastic-net is a hybrid of ℓ_1 and ℓ_2:

$$\lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

R Functions

- Use R help and R manuals
- Linear models: function lm
- QR decomposition qr; Cholesky decomposition chol; PCA princomp, prcomp; SVD svd.
- Ridge regression:
  - package MASS; function lm.ridge
  - package glmnet; function glmnet and cv.glmnet with `alpha` = 0
- Lasso:
  - package lars; function lars
  - package glmnet; function glmnet and cv.glmnet with `alpha` = 1