Lecture 6: Variable Selection - LASSO

Wenbin Lu

Department of Statistics
North Carolina State University

Fall 2019

Outline

Modern Penalization Methods

- L_q penalty, ridge
- LASSO
  - computation, solution path, LARS algorithms
  - parameter tuning, cross validation
  - theoretical properties
- ridge regression

Notations

- data $(\boldsymbol{X}_i, Y_i), i = 1, \cdots, n$
- n: sample size
- d: the number of predictors, $\boldsymbol{X}_i \in R^d$.
- the full index set $\mathcal{S} = \{1, 2, \cdots, d\}$.
- the selected index set given by a procedure is $\hat{\mathcal{A}}$, its size is $|\hat{\mathcal{A}}|$.
- the linear coefficient vector $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^T$.
- the true linear coefficients $\boldsymbol{\beta}_0 = (\beta_{10}, \cdots, \beta_{d0})^T$.
- the true model $\mathcal{A}_0 = \{j : j = 1, \cdots, d, |\beta_{j0}| \neq 0\}$.

Motivations

Shrinkage method for variable selection a continuous process

- more stable, not suffering from high variability.
- one unified selection and estimation framework; easy to make inferences
- advances in theories: oracle properties, valid asymptotic inferences

In general, we standardize the inputs first before applying the methods.

- X's are centered to mean 0 and variance 1
- $\mathbf{y}$ is centered to mean 0

We often fit a model without an intercept.

Ideal Variable Selection

The best results we can expect from a selection procedure:

- able to obtain the correct model structure
  - keep all the important variables in the model
  - filter all the noisy variable out of the model
- has some optimal inferential properties
  - consistent, optimal convergence rate
  - asymptotic normality, most efficient

True model: $y_i = \beta_{00} + \sum_{j=1}^{d} x_{ij}\beta_{j0} + \epsilon_i$,

- important index set: $\mathcal{A}_0 = \{j : \beta_{j0} \neq 0, j = 1, \cdots, d\}$
- unimportant index set: $\mathcal{A}_0^c = \{j : \beta_{j0} = 0, j = 1, \cdots, d\}$

Oracle Properties

An oracle performs as if the true model were known:

1. selection consistency

$$\hat{\beta}_j \neq 0 \quad \text{for} \quad j \in \mathcal{A}_0; \quad \hat{\beta}_j = 0 \quad \text{for } j \in \mathcal{A}_0^c;$$

2. estimation consistency:

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{\mathcal{A}_0} - \boldsymbol{\beta}_{\mathcal{A}_0}) \to_d N(0, \Sigma_I),$$

$\boldsymbol{\beta}_{\mathcal{A}_0} = \{\beta_j, j \in \mathcal{A}_0\}$ and Σ_I is the covariance matrix if knowing the true model.

Penalized Regularization Framework

One revolutionary work is the development of regularization framework:

$$\min_{\beta} L(\beta; \mathbf{y}, \boldsymbol{X}) + \lambda J(\beta)$$

$L(\beta; \mathbf{y}, \boldsymbol{X})$ is the loss function

- For OLS, $L = \sum_{i=1}^{n} [y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij}]^2$
- For MLE methods, $L = \log$ likelihood
- For Cox's PH models, L is the partial likelihood
- In supervised learning, L is the hinge loss function (SVM), or exponential loss (AdaBoost)

L_q Shrinkage Penalty

$J(\beta)$ is the penalty function.

$$J_q(|\beta|) = \lambda \|\beta\|_q^q = \sum_{j=1}^{d} |\beta_j|^q, \quad q \geq 0.$$

- $J_0(|\beta|) = \sum_{j=1}^{d} I(\beta_j \neq 0)$ (L_0; Donoho and Johnstone 1988)
- $J_1(|\beta|) = \sum_{j=1}^{d} |\beta_j|$ (LASSO; Tibshirani 1996)
- $J_2(|\beta|) = \sum_{j=1}^{d} \beta_j^2$ (Ridge; Hoerl and Kennard, 1970)
- $J_\infty(|\beta|) = \max_j |\beta_j|$ (Supnorm penalty; Zhang et al. 2008)

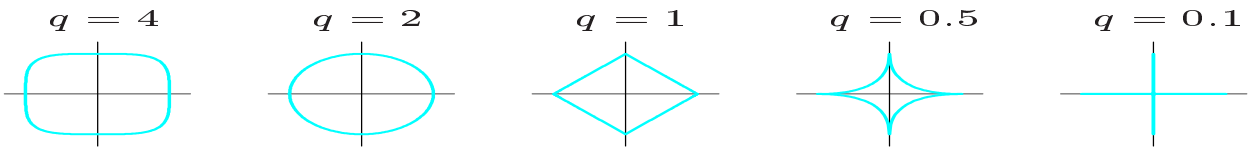

Figure 3.13: *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

The LASSO

Applying the absolute penalty on the coefficients

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{d} |\beta_j|$$

- $\lambda \geq 0$ is a tuning parameter
- λ controls the amount of shrinkage; the larger λ, the greater amount of shrinkage
- What happens if $\lambda \to 0$? (no penalty)
- What happens if $\lambda \to \infty$?

Equivalent formulation for LASSO

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} x_{ij}\beta_j)^2, \quad \text{subject to} \quad \sum_{j=1}^{d} |\beta_j| \leq t.$$

- Explicitly apply the magnitude constraint on the parameters
- Making t sufficiently small will cause some coefficients to be exactly zero.
- The lasso is a kind of continuous subset selection

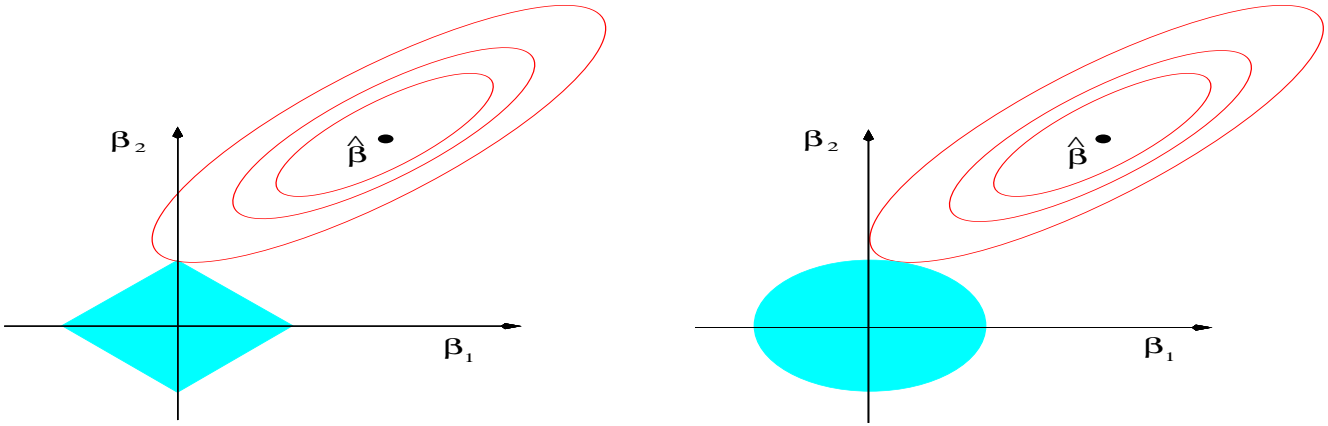

Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

LASSO Solution: soft thresholding

For the orthogonal design case with $X'X = I$, the LASSO method is equivalent to: solve β_j's componentwisely by solving

$$\min_{\beta_j}(\beta_j - \hat{\beta}_j^{\text{ls}})^2 + \lambda|\beta_j|$$

The solution to the above problem is

$$\begin{aligned}
\hat{\beta}_j^{\text{lasso}} &= \text{sign}(\hat{\beta}_j^{\text{ls}})(|\hat{\beta}_j^{\text{ls}}| - \frac{\lambda}{2})_+ \\
&= \begin{cases} \hat{\beta}_j^{\text{ls}} - \frac{\lambda}{2} & \text{if} \quad \hat{\beta}_j^{\text{ls}} > \frac{\lambda}{2} \\ 0 & \text{if} \quad |\hat{\beta}_j^{\text{ls}}| \leq \frac{\lambda}{2} \\ \hat{\beta}_j^{\text{ls}} + \frac{\lambda}{2} & \text{if} \quad \hat{\beta}_j^{\text{ls}} < -\frac{\lambda}{2} \end{cases}
\end{aligned}$$

- shrinks big coefficients by a constant $\frac{\lambda}{2}$ **towards** zero.
- truncates small coefficients to zero **exactly**.

Comparison: Different Methods Under Orthogonal Design

When X is orthonormal, i.e. $X^T X = I$

- Best subset (of size M): $\hat{\beta}_j^{\text{ls}}$ if rank$(|\hat{\beta}_j^{\text{ls}}|) \leq M$.
  - keeps the largest coefficients
- Ridge regression: $\hat{\beta}_j^{\text{ls}}/(1+\lambda)$
  - does a proportional shrinkage
- Lasso: $\text{sign}(\hat{\beta}_j^{\text{ls}})(|\hat{\beta}_j^{\text{ls}}| - \frac{\lambda}{2})_+$
  - transform each coefficient by a constant factor first, then truncate it at zero with a certain threshold
  - "soft thresholding", used often in wavelet-based smoothing

How to Choose t or λ

The tuning parameter t or λ should be adaptively chosen to minimize the MSE or PE. Common tuning methods include

- the tuning error, cross validation, leave-out-one cross validation (LOOCV), AIC, or BIC.
- If t is chosen larger than $t_0 = \sum_{j=1}^{d} |\hat{\beta}_j^{\text{ls}}|$, then $\hat{\beta}_j^{\text{lasso}} = \hat{\beta}_j^{\text{ls}}$.
- If $t = t_0/2$, the least squares coefficients are shrunk by 50%.

In practice, we sometimes standardize t by $s = t/t_0$ in $[0, 1]$, and choose the best value s from $[0, 1]$.

K-fold Cross Validation

The simplest and most popular way to estimate the tuning error.

1. Randomly split the data into K roughly equal parts
2. For each $k = 1, ..., K$
   - leave the kth portion out, and fit the model using the other $K - 1$ parts. Denote the solution by $\widehat{\beta}^{-k}$
   - calculate prediction errors of $\widehat{\beta}^{-k}$ on the left-out kth portion
3. Average the prediction errors

Define the Index function (allocating memberships)

$$\kappa : \{1, ..., n\} \to \{1, ..., K\}.$$

The K-fold cross-validation estimate of prediction error (PE) is

$$CV = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \widehat{\beta}^{-\kappa(i)} \right)^2$$

Typical choice: $K = 5, 10$

Choose λ

Choose a sequence of λ values.

- For each λ, fit the LASSO and denote the solution by $\widehat{\boldsymbol{\beta}}_{\lambda}^{lasso}$.
- Compute the $CV(\lambda)$ curve as

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{\lambda}^{-\kappa(i)} \right),$$

  which provides an estimate of the test error curve.
- Find the best parameter λ^* which minimizes $CV(\lambda)$.
- Fit the final LASSO model with λ^*. The final solution is denoted as $\widehat{\boldsymbol{\beta}}_{\lambda^*}^{lasso}$

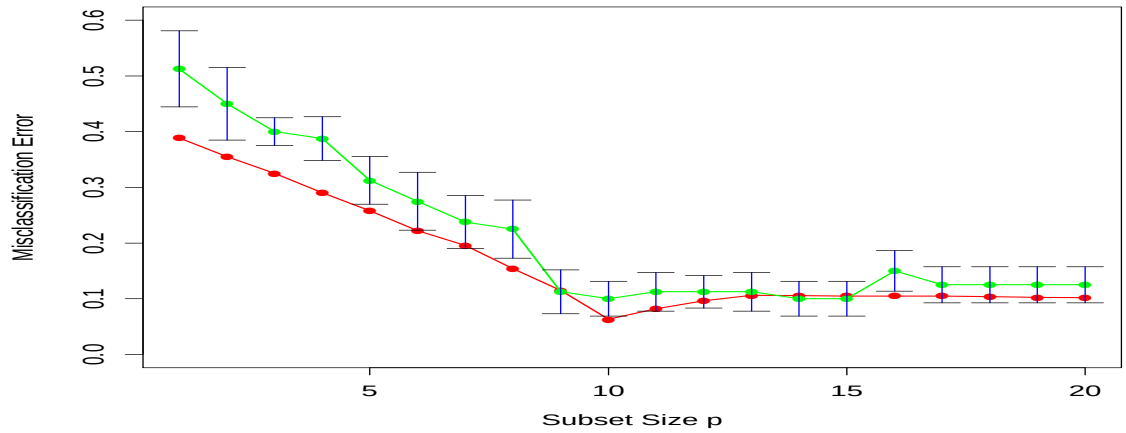

Figure 7.9: *Prediction error (red) and tenfold cross-validation curve (green) estimated from a single training set, from the scenario in the bottom right panel of Figure 7.3.*

One-standard Error for CV

One-standard error rule:

- choose the most parsimonious model with error no more than one standard error above the best error.
- used often in model selection (for a smaller model size).

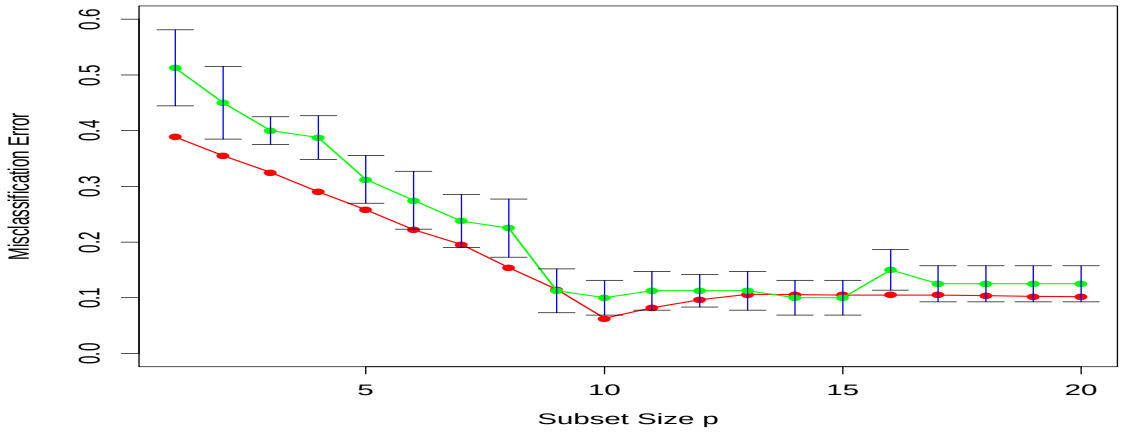

Figure 7.9: *Prediction error (red) and tenfold cross-validation curve (green) estimated from a single training set, from the scenario in the bottom right panel of Figure 7.3.*

Computation of LASSO

- Quadratic Programming
- Shooting Algorithms (Fu 1998; Zhang and Lu, 2007)
- LARS solution path (Efron et al. 2004)
  - the most efficient algorithm for LASSO
  - designed for standard linear models
  - R package "lars"
- Coordinate Descent Algorithm (CDA; Friedman et al. 2010)
  - designed for GLM
  - suitable for ultra-high dimensional problems
  - R package "glmnet"

R package: lars

Provides the entire sequence of coefficients and fits, starting from zero, to the least squares fit.

```
library(lars)
lars(x, y, type = c("lasso", "lar", "forward.stagewise",
      "stepwise"),  trace = FALSE, normalize = TRUE,
      intercept = TRUE, Gram, eps = .Machine$double.eps,
      max.steps, use.Gram = TRUE)
```

- A "lars" object is returned, for which print, plot, predict, coef and summary methods exist.
- Details in Efron et al. (2004).
- With "lasso" option, it computes the complete lasso solution simultaneously for ALL values of the shrinkage parameter (λ or t or s) in the same computational cost as a least squares fit.

Details About LARS function

- x: matrix of predictors
- y: response
- type: One of "lasso", "lar", "forward.stagewise" or "stepwise", all variants of Lasso. Default is "lasso".
- trace: if TRUE, lars prints out its progress
- normalize: if TRUE, each variable is standardized to have unit L_2 norm, otherwise it is left alone. Default is TRUE.
- intercept: if TRUE, an intercept is included (not penalized), otherwise no intercept is included. Default is TRUE.
- Gram: $\boldsymbol{X}^T\boldsymbol{X}$ matrix; useful for repeated runs (bootstrap) where a large $\boldsymbol{X}^T\boldsymbol{X}$ stays the same.
- eps: An effective zero
- max.steps: the number of steps taken. Default is $8\min(d, n - \text{intercept})$, n is the sample size.

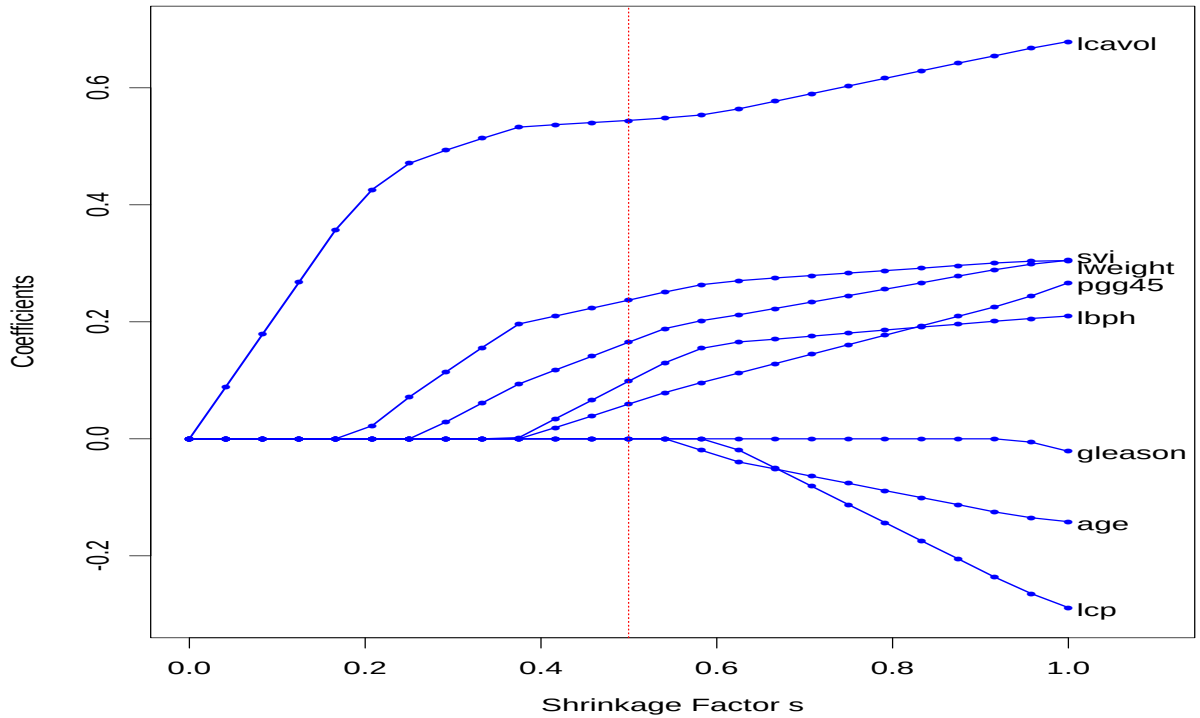

Figure 3.9: *Profiles of lasso coefficients, as tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.5$, the value chosen by cross-validation. Compare Figure 3.7 on page 7; the lasso profiles hit zero, while those for ridge do not.*

Prediction for LARS

```
predict(object, newx, s, type = c("fit", "coefficients"),
    mode = c("step", "fraction", "norm", "lambda"), ...)
coef(object, ...)
```

- object: A fitted lars object
- newx: If type="fit", then newx should be the x values at which the fit is required. If type="coefficients", then newx can be omitted.
- *s*: a value, or vector of values, indexing the path. Its values depends on the *mode* . By default (mode="step"), *s* takes on values between 0 and p.
- type: If type="fit", predict returns the fitted values. If type="coefficients", predict returns the coefficients.

"Mode" for Prediction in LARS

- If mode="step", then s should be the argument indexes the lars step number, and the coefficients will be returned corresponding to the values corresponding to step s.
- If mode="fraction", then s should be a number between 0 and 1, and it refers to the ratio of the L1 norm of the coefficient vector, relative to the norm at the full LS solution.
- If mode="norm", then s refers to the L_1 norm of the coefficient vector.
- If mode= λ, then s is the lasso regularization parameter

Example: Diabetes Data

```
library(lars)
data(diabetes)
attach(diabetes)

## fit LASSO
object <- lars(x,y)
plot(object)
summary(object)
predict(object,newx=x)

## fit LARS
object2 <- lars(x,y,type="lar")
detach(diabetes)
```

R Code to Compute K-CV Error Curve for Lars

```
cv.lars(x, y, K = 10, index, type = c("lasso",
     "lar", "forward.stagewise", "stepwise"),
      mode=c("fraction", "step"), ...)
```

- x: Input to lars
- y: Input to lars
- K: Number of folds
- index: Abscissa values at which CV curve should be computed. If mode="fraction", this is the fraction of the saturated $|\beta|$. The default value index=seq(from $= 0$, to $= 1$, length $= 100$). If mode="step", this is the number of steps in lars procedure.

R Code: CV.LARS

```
attach(diabetes)
lasso.s<-seq(0,1,length=100)
lasso.cv<-cv.lars(x,y,K=10,index=lasso.s,mode="fraction")
lasso.mcv<-which.min(lasso.cv$cv)

## minimum CV rule
bests1 <- lasso.s[lasso.mcv]
lasso.coef1 <- predict(fit.lasso, s=bests1, type="coef",
   mode="frac")

## one-standard rule
bound<-lasso.cv$cv[lasso.mcv]+lasso.cv$cv.error[lasso.mcv]
bests2<-lasso.s[min(which(lasso.cv$cv<bound))]
lasso.coef2 <- coef(fit.lasso, s=bests2,  mode="frac")
```

Consistency - Definition

Let the true parameters be $\boldsymbol{\beta}_0$ and their estimates by $\widehat{\boldsymbol{\beta}}_n$. The subscript 'n' means the sample size n.

- Estimation consistency

$$\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_0 \to_p \mathbf{0}, \quad \text{as } n \to \infty.$$

- Model selection consistency

$$P\left(\{j : \hat{\beta}_j \neq 0\} = \{j : \beta_{0j} \neq 0\}\right) \to 1, \quad \text{as } n \to \infty.$$

- Sign consistency

$$P\left(\widehat{\boldsymbol{\beta}}_n =_s \boldsymbol{\beta}_0\right) \to 1, \quad \text{as } n \to \infty,$$

where

$$\widehat{\boldsymbol{\beta}}_n =_s \boldsymbol{\beta}_0 \Leftrightarrow \text{sign}(\widehat{\boldsymbol{\beta}}_n) = \text{sign}(\boldsymbol{\beta}_0).$$

Sign consistency is stronger than model selection consistency.

Consistency of LASSO Estimator

Knight and Fu (2000) have shown that

- **Estimation Consistency**: The LASSO solution is of estimation consistency for fixed p. In other words,

$$\widehat{\boldsymbol{\beta}}^{lasso}(\lambda_n) \to_p \boldsymbol{\beta}, \text{as } \lambda_n = o(n).$$

  It is root-n consistent and asymptotically normal.

- **Model Selection Property**: For $\lambda_n \propto n^{\frac{1}{2}}$, as $n \to \infty$, there is a non-vanishing positive probability for lasso to select the true model.

The l_1 penalization approach is called *basis pursuit* in signal processing (Chen, Donoho, and Saunders, 2001).

Model Selection Consistency of LASSO Estimator

Zhao and Yu (2006). On Model Selection Consistency of Lasso. JMLR, 7, 2541–2563.

- If an irrelevant predictor is highly correlated with the predictors in the true model, Lasso may not be able to distinguish it from the true predictors with any amount of data and any amount of regularization.
- Under **Irrepresentable Condition** (IC), the LASSO is **model selection consistent** in both fixed and large p settings.
- The IC condition is represented as

$$|[X_n(1)^T X_n(1)]^{-1} X_n^T(1) X_n(2)| < 1 - \eta,$$

i.e., the regression coefficients of irrevelant covariates $X_n(2)$ on the relevant covariates $X_n(1)$ is constrained. It is almost necessary and sufficient for Lasso to select the true model.

Special Cases for LASSO to be Selection Consistent

Assume the true model size $|\mathcal{A}_0| = q$.

- The underlying model must satisfy a nontrivial condition if the lasso variable selection is consistent (Zou, 2006).
- The lasso is always consistent in model selection under the following special cases:
  - when $p = 2$;
  - when the design is orthgonal;
  - when the covariates have bounded constant correlation, $0 < r_n < \frac{1}{1+cq}$ for some c;
  - when the design has power decay correlation with $\text{corr}(X_i, X_j) = \rho_n^{|i-j|}$, where $|\rho_n| \leq c < 1$.

Ridge Regression ($q = 2$)

Applying the squared penalty on the coefficients

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{d} \beta_j^2$$

- $\lambda \geq 0$ is a tuning parameter
- The solution

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

Note the similarity to the ordinary least squares solution, but with the addition of a "ridge" down the diagonal.

- As $\lambda \to 0$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \widehat{\boldsymbol{\beta}}^{\text{ols}}$.
- As $\lambda \to \infty$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \mathbf{0}$.

Ridge Regression Solution

In the special case of an orthonormal design matrix,

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

This illustrates the essential "shrinkage" feature of ridge regression

- Applying the ridge regression penalty has the effect of shrinking the estimates toward zero
- The penalty introducing bias but reducing the variance of the estimate

Ridge estimator does not threshold, since the shrinkage is smooth (proportional to the original coefficient).

Invertibility

If the design matrix $\boldsymbol{X}$ is not full rank, then

- $\boldsymbol{X}^T\boldsymbol{X}$ is not invertible,
- For OLS estimator,, there is no unique solution for $\widehat{\beta}^{\text{ols}}$.
- This problem does not occur with ridge regression, however.

Theorem: For any design matrix $\boldsymbol{X}$, the quantity $\boldsymbol{X}^T\boldsymbol{X} + \lambda I_d$ is always invertible; thus, there is always a unique solution $\widehat{\beta}^{\text{ridge}}$.

For the ridge estimator, an unbiased estimate for its df is

$$\widehat{df}(\widehat{\beta}^{\text{ridge}}) = \text{Tr}\{\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda I_d)^{-1}\boldsymbol{X}^T\}.$$

Bias and Variance

- The variance of the ridge regression estimate is

$$\text{Var}(\widehat{\boldsymbol{\beta}}^{\text{ridge}}) = \sigma^2 W \boldsymbol{X}^T \boldsymbol{X} W,$$

  where $W = (\boldsymbol{X}^T \boldsymbol{X} + \lambda I_d)^{-1}$.
- The bias of the ridge regression estimate is

$$\text{Bias}(\widehat{\boldsymbol{\beta}}^{\text{ridge}}) = -\lambda W \boldsymbol{\beta}.$$

It can be shown that

- the total variance $\sum_{j=1}^{d} \text{Var}(\hat{\beta}_j^{\text{ridge}})$ is a monotone decreasing sequence with respect to λ
- while the total squared bias $\sum_{j=1}^{d} \text{Bias}^2(\hat{\beta}_j^{\text{ridge}})$ is a monotone increasing sequence with respect to λ.

Mean Squared Error (MSE) of Ridge

Existence Theorem:
There always exists a λ such that the MSE of $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is less than the MSE of $\widehat{\boldsymbol{\beta}}^{\text{ols}}$.

- a rather surprising result with somewhat radical implications: even if the model we fit is exactly correct and follows the exact distribution we specify, we can always obtain a better (in terms of MSE) estimator by shrinking towards zero.

Bayesian Interpretation of Ridge Estimation

Penalized regression can be interpreted in a Bayesian context:

- Suppose $\mathbf{y} = \boldsymbol{X}\beta + \epsilon$, where $\epsilon \sim N(\mathbf{0}, \sigma^2 I_n)$.
- Given the prior $\beta \sim N(\mathbf{0}, \tau^2 I_d)$, then the posterior mean of β given the data is

$$(\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2} I_d)^{-1}\boldsymbol{X}^T\mathbf{y}.$$

R package: lm.ridge

```
library(MASS)
lm.ridge(formula, data, subset, na.action, lambda)
```

- formula: a formula expression as for regression models, of form

$$\text{response} \sim \text{predictors}$$

- data: an optional data frame in which to interpret the variables occurring in formula.
- subset: expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
- na.action a function to filter missing data.
- lambda: A scalar or vector of ridge constants.

If an intercept is present, its coefficient is not penalized.

Output for lm.ridge

Returns a list with components

- coef: matrix of coefficients, one row for each value of lambda. Note that these are not on the original scale and are for use by the coef method.
- scales: scalings used on the X matrix.
- Inter: was intercept included?
- lambda: vector of lambda values
- ym: mean of y
- xm: column means of x matrix
- GCV: vector of GCV values

Example for Ridge Regression

```
library(MASS)
longley
names(longley)[1] <- "y"
lm.ridge(y ~ ., longley)
plot(lm.ridge(y ~ ., longley,
              lambda = seq(0,0.1,0.001)))
```

(demo)

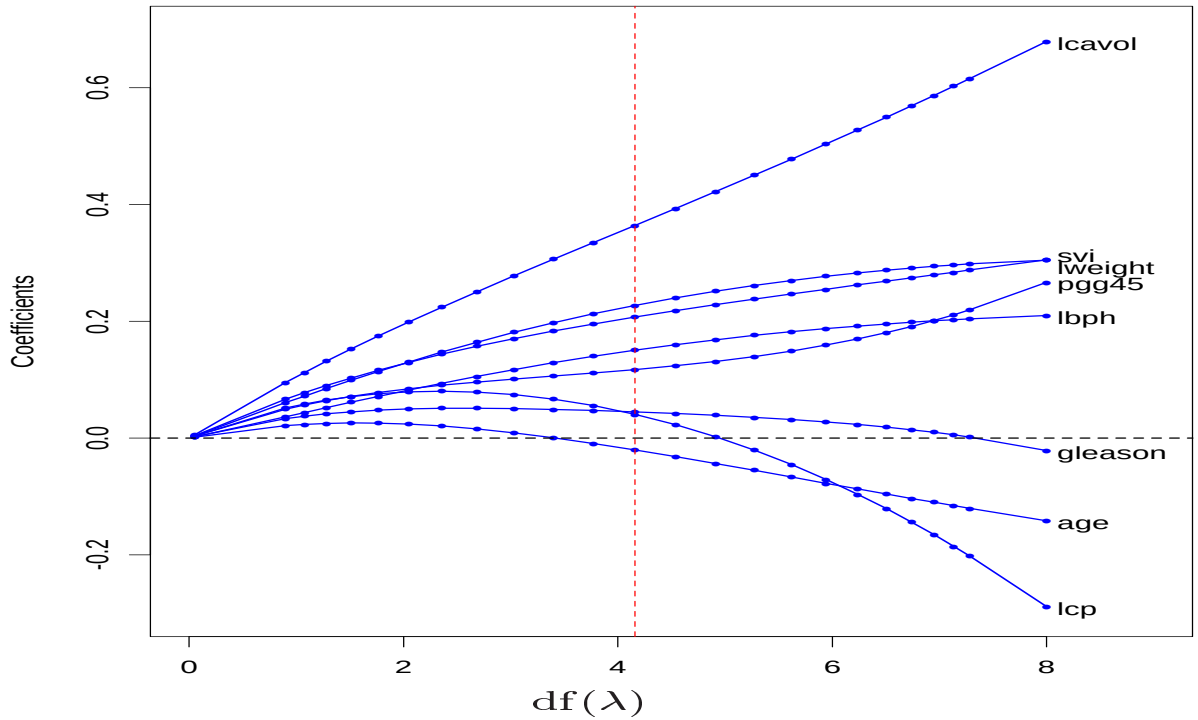

Figure 3.7: *Profiles of ridge coefficients for the prostate cancer example, as tuning parameter* λ *is varied. Coefficients are plotted versus* $\mathrm{df}(\lambda)$*, the effective degrees of freedom. A vertical line is drawn at* $\mathrm{df} = 4.16$*, the value chosen by cross-validation.*

Lasso vs Ridge

Ridge regression:

- is a continuous shrinkage method; achieves better prediction performance than OLS through a biase-variance trade-off.
- cannot produce a parsimonious model, for it always keeps all the predictors in the model.

The lasso

- does both continuous shrinkage and automatic variable selection simultaneously.

Empirical Comparison

Tibshirani (1996) and Fu (1998) compared the prediction performance of the lasso, ridge and bridge regression (Frank and Friedman, 1993) and found

- None of them uniformly dominates the other two.
- However, as variable selection becomes increasingly important in modern data analysis, the lasso is much more appealing owing to its sparse representation.