Model Assessment and Selection with Regularization

Yuan Yao

Department of Mathematics
Hong Kong University of Science and Technology

Most of the materials here are from Chapter 5-6 of Introduction to Statistical learning by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.

Spring, 2020

Model Assessment

- Cross Validation
- Bootstrap

Model Selection

- Subset selection
- Ridge Regression
- The Lasso
- Principal Component Regression

Outline

Training error is not sufficient enough

- Training error easily computable with training data.
- Because of possibility of over-fit, it cannot be used to properly assess test error.
- It is possible to "estimate" the test error, by, for example, making adjustments of the training error.
- General purpose method of prediction/test error estimate: validation.

Ideal scenario for performance assessment

- In a "data-rich" scenario, we can afford to separate the data into three parts:
  - training data: used to train various models.
  - validation data: used to assess the models and identify the best.
  - test data: test the results of the best model.
- Usually, people also call validation data or hold-out data as test data.

Cross validation: overcome the drawback of validation set approach

- Our ultimate goal is to produce the best model with best prediction accuracy.
- Validation set approach has a drawback of using ONLY training data to fit model.
- The validation data do not participate in model building but only model assessment.
- A "waste" of data.
- We need more data to participate in model building.

K-fold cross validation

- Divide the data into K subsets, usually of equal or similar sizes (n/K).
- Treat one subset as validation set, the rest together as a training set. Run the model fitting on training set. Calculate the test error estimate on the validation set, denoted as MSE_i, say.
- Repeat the procedures over every subset.
- Average over the above K estimates of the test errors, and obtain

$$\text{CV}_{(K)} = \frac{1}{K}\sum_{i=1}^{K}\text{MSE}_i$$

- Leave-One-Out Cross Validation (LOOCV) is a special case of K-fold cross validation, actually n-fold cross validation.

K-fold cross validation

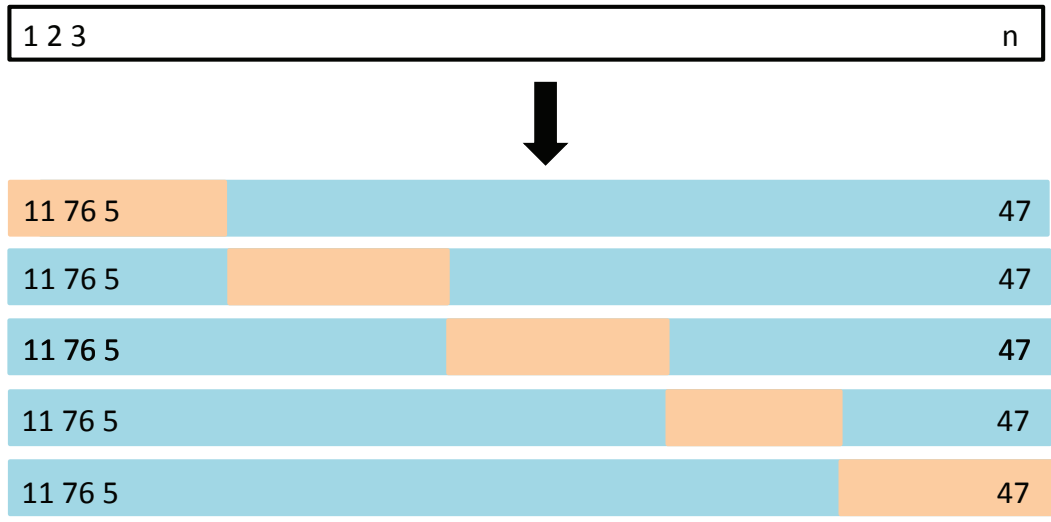

Figure: 5.5. A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.

Inference of Estimate Uncertainty

- Suppose we have data $x_1, ..., x_n$, representing the ages of n randomly selected people in HK.
- Use sample mean $\bar{x}$ to estimate the population mean μ, the avearge age of all residents of HK.
- How to justify the estimation error $\bar{x} - \mu$? Usually by t-confidence interval, test of hypothesis.
- They rely on normality assumption or central limit theorm.
- Is there another reliable way?
- Just bootstrap:

Boostrap as a resampling procedure.

- Take n random sample (with replacement) from $x_1, ..., x_n$.
- calculate the sample mean of the "re-sample", denoted as $\bar{x}_1^*$.
- Repeat the above a large number M times. We have $\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_M^*$.
- Use the distribution of $\bar{x}_1^* - \bar{x}, ..., \bar{x}_M^* - \bar{x}$ to approximate that of $\bar{x} - \mu$.

- Essential idea: Treat the data distribtion (more professionally called empirical distributoin) as a proxy of the population distribution.
- Mimic the data generation from the true population, by trying resampling from the empirical distribution.
- Mimic your statistical procedure (such as computing an estimate $\bar{x}$) on data, by doing the same on the resampled data.
- Evalute your statistical procedure (which may be difficult because it involves randomness and the unknown population distribution) by evaluting your analogue procudres on the re-samples.

Example

- X and Y are two random variables. Then minimizer of $\text{var}(\alpha X + (1-\alpha)Y))$ is

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

- Data: $(X_1, Y_1), ..., (X_n, Y_n)$.
- We can compute sample variances and covariances.
- Estimate α by

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

- How to evalute $\hat{\alpha} - \alpha$, (remember $\hat{\alpha}$ is random and α is unknown).
- Use Bootstrap

Example

- Sample n resamples from $(X_1, Y_1), ..., (X_n, Y_n)$, and compute the sample the sample variance and covariances for this resample. And then compute

$$\hat{\alpha}^* = \frac{(\hat{\sigma}_Y^*)^2 - \hat{\sigma}_{XY}^*}{(\hat{\sigma}_X^*)^2 + (\hat{\sigma}_Y^*)^2 - 2\hat{\sigma}_{XY}^*}$$

- Repeat this procedure, and we have $\hat{\alpha}_1^*, ..., \hat{\alpha}_M^*$ for a large M.
- Use the distribution of $\hat{\alpha}_1^* - \hat{\alpha}, ..., \hat{\alpha}_M^* - \hat{\alpha}$ to approximate the distribution of $\hat{\alpha} - \alpha$.
- For example, we can use

$$\frac{1}{M}\sum_{j=1}^{M}(\hat{\alpha}_j^* - \hat{\alpha})^2$$

  to estimate $E(\hat{\alpha} - \alpha)^2$.
- Use Bootstrap

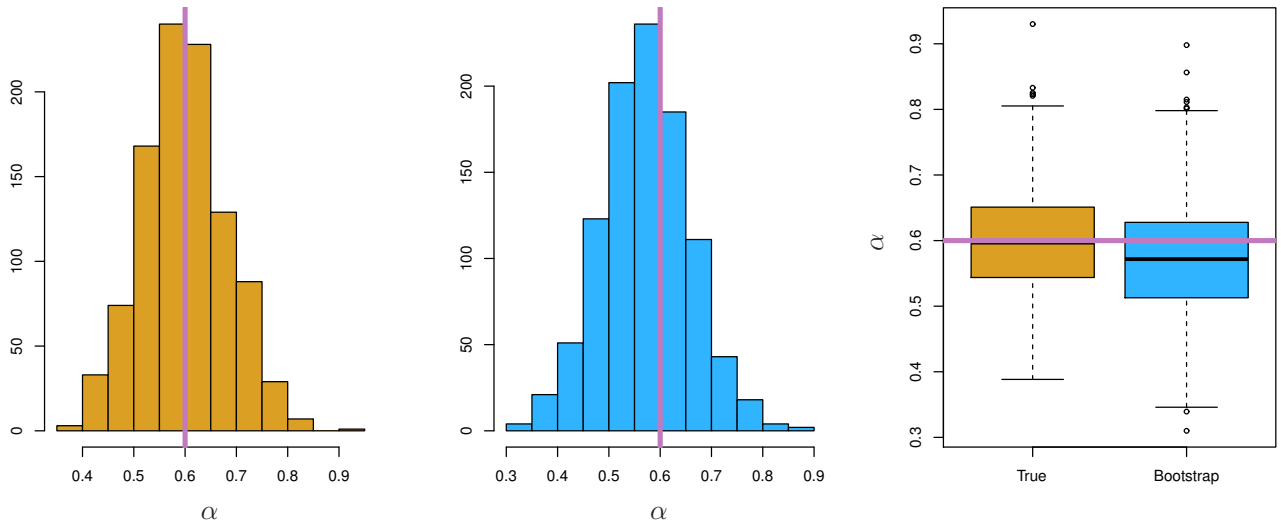

Figure: 5.10. Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population. Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set. Right: The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.

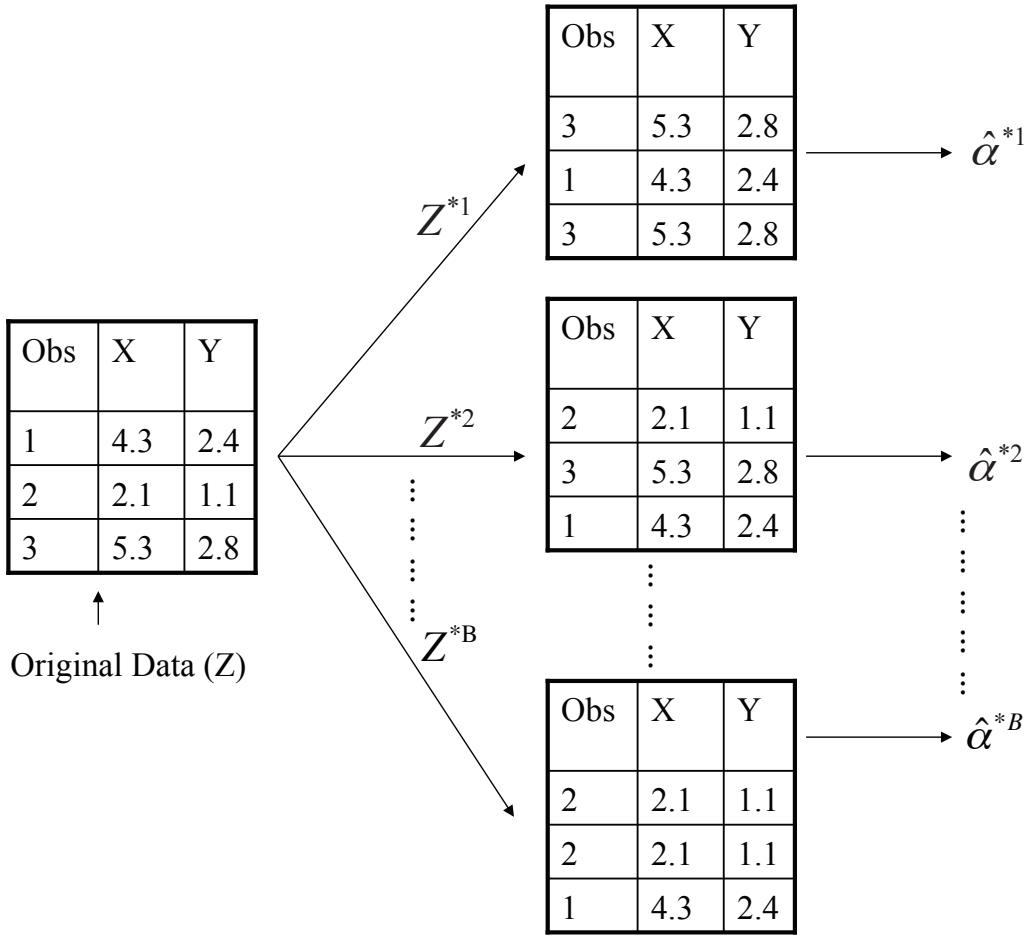
Obs X Y
1 4.3 2.4
2 2.1 1.1
3 5.3 2.8
Original Data (Z)
Z^{*1}
Z^{*2}
Z^{*B}
Obs X Y
3 5.3 2.8
1 4.3 2.4
3 5.3 2.8
Obs X Y
2 2.1 1.1
3 5.3 2.8
1 4.3 2.4
Obs X Y
2 2.1 1.1
2 2.1 1.1
1 4.3 2.4
$\hat{\alpha}^{*1}$
$\hat{\alpha}^{*2}$
$\hat{\alpha}^{*B}$

Figure 5.11. A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.

Outline

Interpretability vs. Prediction

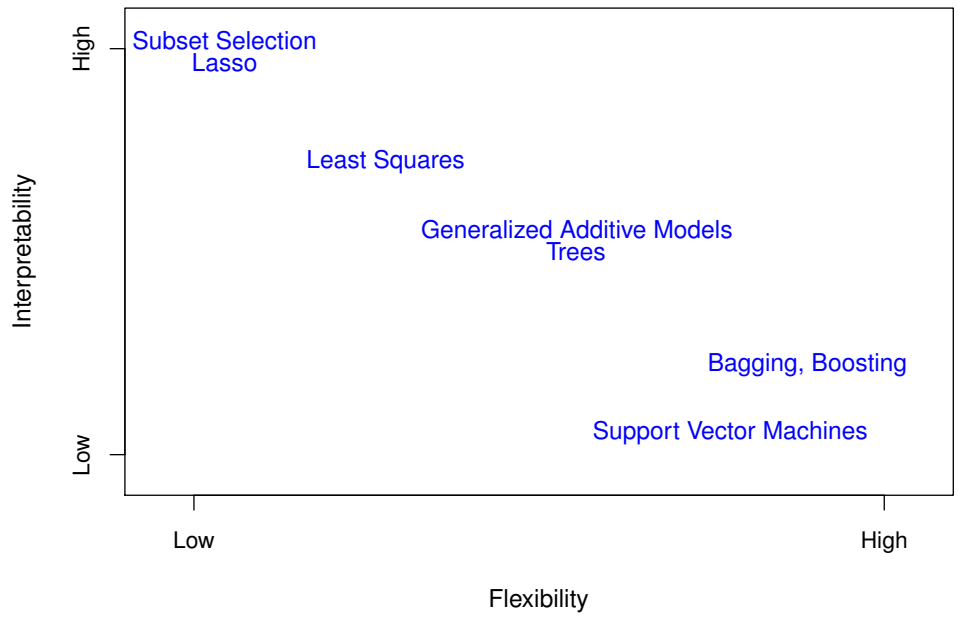

Figure: 2.7. As models become flexible, interpretability drops. **Occam Razor principle**: Everything has to be kept as simple as possible, but not simpler (Albert Einstein).

About this chapter

- Linear model already addressed in detail in Chapter 3.

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon$$

- Model assessment: cross-validation (prediction) error in Chapter 5.
- This chapter is about model selection for linear models.
- The model selection techniques can be extended beyond linear models.
- Details about AIC, BIC, Mallow's C_p mentioned in Chapter 3.

Feature/variable selection

- Not all existing input variables are useful for predicting the output.
- Keeping redundant inputs in model can lead to poor prediction and poor interpretation.
- We consider three ways of variable/model selection:
  1. Subset selection.
  2. Shrinkage/regularization: constraining some regression parameters to 0.
  *3. Dimension reduction: (actually using the "derived inputs" by, for example, principle component approach.)

Best subset selection

- Exhaust all possible combinations of inputs.
- With p variables, there are 2^p many distinct combinations.
- Identify the best model among these models.

Pros and Cons of best subset selection

- Seems straightforward to carry out.
- Conceptually clear.
-
- The search space too large (2^p models), may lead to overfit.
- Computationally infeasible: too many models to run.
- if $p = 20$, there are $2^{20} > 1000,000$ models.

Forward stepwise selection

- Start with the null model.
- Find the best one-variable model.
- With the best one-varialbe model, add one more variable to get the best two-variable model.
- With the best two-varialbe model, add one more variable to get the best three-variable model.
- ....
- Find the best among all these best k-variable models.

Pros and Cons of forward stepwise selection

- Less computation
- Less models ($\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models).
- (if $p = 20$, only 211 models, compared with more than 1 million models for best subset selection).
- No problem for first n-steps if $p > n$.
- Once an input is in, it does not get out.

Backward stepwise selection

- Start with the largest model (all p inputs in).
- Find the best $(p-1)$-variable model, by reducing one from the largest model
- Find the best $(p-2)$-variable model, by reducing one variable from the best $(p-1)$-variable model.
- Find the best $(p-3)$-variable model, by reducing one variable from the best $(p-2)$-variable model.
- ....
- Find the best 1-varible model, by reducing one variable from the best 2-variable model.
- The null model.

Pros and Cons of backward stepwise selection

- Less computation
- Less models ($\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models).
- (if $p = 20$, only 211 models, compared with more than 1 million models for best subset selection).

- Once an input is out, it does not get in.
- No applicable to the case with $p > n$.

Find the best model based on prediction error.

- General approach by Validation/Cross-Validation (addressed in ISLR Chapter 5).
- Model-based approach by Adjusted R^2, AIC, BIC or C_p (ISLR Chapter 3).

R-squared

- Residue

$$\hat{\epsilon}_i = y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij}$$

- Residual Sum of Squares

$$\text{RSS} = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2$$

- R-squared

$$R^2 = \frac{\text{SS}_{reg}}{\text{SS}_{total}} = 1 - \frac{\text{SS}_{error}}{\text{SS}_{total}}$$

where $\text{SS}_{error} = \text{RSS} = \sum_{i=1}^{n} \hat{\epsilon}_i^2$ and $\text{SS}_{total} = \sum_{i=1}^{n} (y_i - \bar{y})^2$.

Example: Credit data

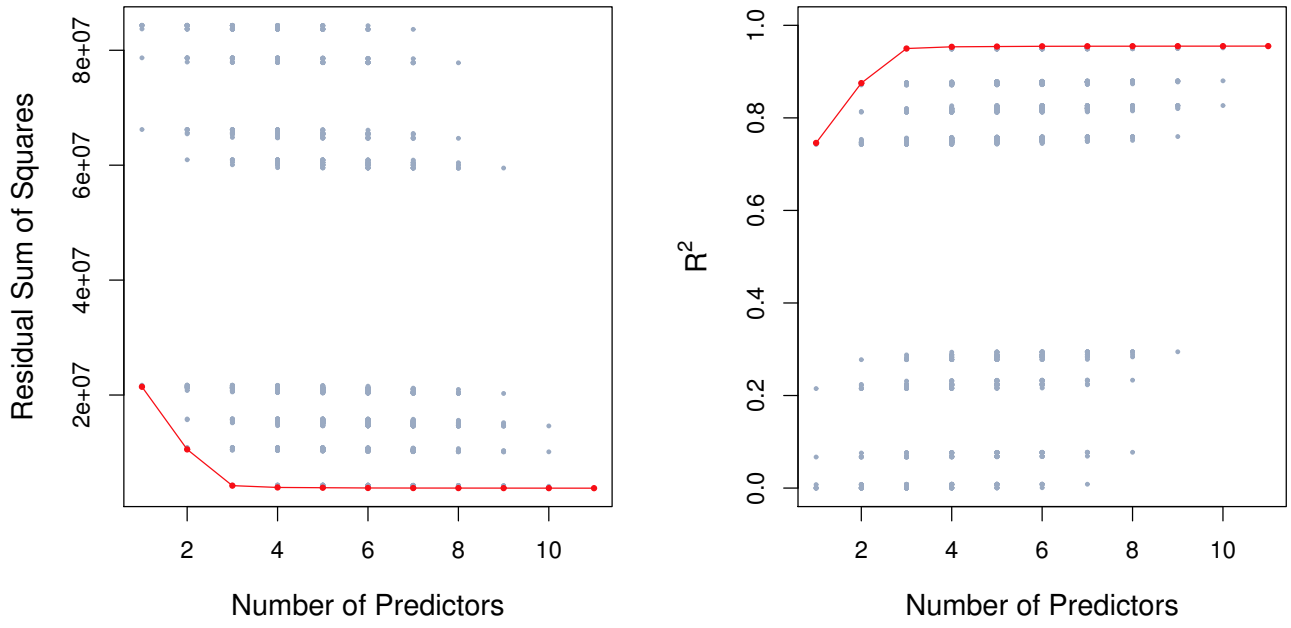

Figure: 6.1. For each possible model containing a subset of the ten predictors in the Credit data set, the RSS and R^2 are displayed. The red frontier tracks the best model for a given number of predictors, according to RSS and R^2. Though the data set contains only ten predictors, the x-axis ranges from 1 to 11, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.

The issues of R-squared

- The R-squared is the percentage of the total variation in response due to the inputs.
- The R-squared reflects the *training error*.
- However, a model with larger R-squared is not necessarily better than another model with smaller R-squared when we consider *test error*!
- If model A has all the inputs of model B, then model A's R-squared will always be greater than or as large as that of model B.
- If model A's additional inputs are entirely uncorrelated with the response, model A contain more noise than model B. As a result, the prediction based on model A would inevitably be poorer or no better.

a) Adjusted R-squared

- The adjusted R-squared, taking into account of the degrees of freedom, is defined as

$$\begin{aligned}
\text{adjusted } R^2 &= 1 - \frac{\text{MS}_{error}}{\text{MS}_{total}} \\
&= 1 - \frac{\text{SS}_{error}/(n-p-1)}{\text{SS}_{total}/(n-1)} \\
&= 1 - \frac{s^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2/(n-1)}
\end{aligned}$$

  With more inputs, the R^2 always increase, but the adjusted R^2 could decrease since more inputs is penalized by the smaller degree of freedom of the residuals.
- The adjusted R-squared is preferred over the R-squared in evaluating models.

b) Mallow's C_p

Recall that our linear model (2.1) has p covariates, and $s^2 = RSS/(n-p-1)$ is the unbiased estimator of σ^2.
Assume now more covariates are available. Suppose we use only p of the K covariates with $K \geq p$.
The statistic of Mallow's C_p is defined as

$$C_p = \frac{\text{RSS}(k) + 2(k+1)s^2}{n}$$

where RSS(k) is the residual sum of squares for the linear model with k inputs.
The smaller Mallows' C_p is, the better the model is.
The following AIC is more often used, despite that Mallws' C_p and AIC usually give the same best model.

AIC

AIC stands for Akaike information criterion, which is defined as

$$\text{AIC} = \frac{\text{RSS} + 2(p+1)s^2}{ns^2},$$

for a linear model with p inputs, where $s^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2/(n-p-1)$ is the unbiased estimator of σ^2. AIC aims at maximizing the predictive likelihood. The model with the smallest AIC is preferred. The AIC criterion is try to maximize the expected predictive likelihood. In general, it can be roughly derived in the following. Let θ be a parameter of d dimension. $\hat{\theta}$ is the maximum likelihood estimator of θ based on observations $y_1, ..., y_n$. Let θ_0 be the true (unknown) value of θ, and $\mathcal{I}(\theta_0)$ be the Fisher information.

BIC

- BIC stands for Schwarz's Bayesian information criterion, which is defined as

$$\text{BIC} = \frac{\text{RSS} + (1 + p)\log(n)s^2}{ns^2},$$

  for a linear model with p inputs. Again, the model with the smallest BIC is perferred. The derivation of BIC results from Bayesian statistics and has Bayesian interpretation. It is seen that BIC is formally similar to AIC. The BIC penalizes more heavily the models with more number of inputs.

Penalized log-likelihood

- In general AIC/BIC are penalized maximum likelihood, e.g. BIC aims

$$\textbf{minimize} \; -(\text{log likelihood}) + (1+p)\log(n)/n$$

where, the first term is called deviance. In the case of linear regression with normal errors, the deviance is the same as $\log(s^2)$.

Example: credit dataset

Variables	Best subset	Forward stepwise
one	rating	rating
two	rating, income	rating, income
three	rating, income, student	rating, income, student
four	cards, income, student, limit	rating, income, student, limit

TABLE 6.1. The first four selected models for best subset selection and forward stepwise selection on the Credit data set. The first three models are identical but the fourth models differ.

Example

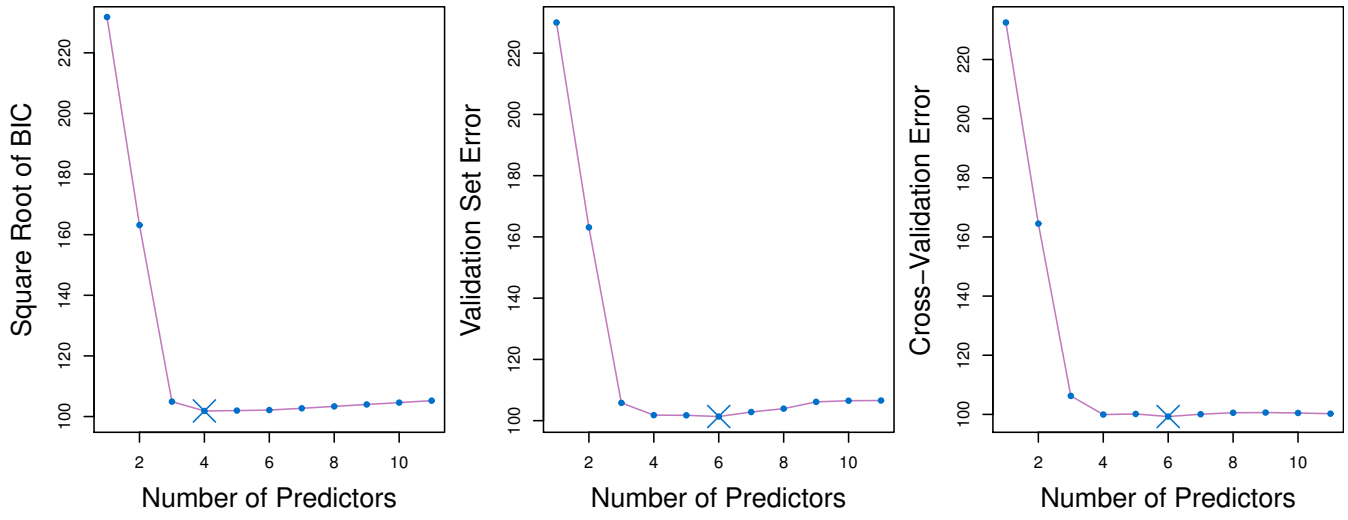

Figure: 6.3. For the Credit data set, three quantities are displayed for the best model containing d predictors, for d ranging from 1 to 11. The overall best model, based on each of these quantities, is shown as a blue cross. Left: Square root of BIC. Center: Validation set errors (75% training data). Right: 10-fold Cross-validation errors.

The one standard deviation rule

- In the above figure, model with 6 inputs do not seem to be much better than model with 4 or 3 inputs.
- Keep in mind the Occam's razor: Choose the simplest model if they are similar by other criterion.

The one standard deviation rule

- Calculate the standard error of the estimated test MSE for each model size,
- Consider the models with estimated test MSE of one standard deviation within the smallest test MSE.
- Among them select the one with the smallest model size.
- (Apply this rule to the Example in Figure 6.3 gives the model with 3 variable.)

Ridge Regression

- The least squares estimator $\hat{\beta}$ is minimizing

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

- The ridge regression $\hat{\beta}_\lambda^R$ is minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

  where $\lambda \geq 0$ is a tuning parameter.
- The first term measuares goodness of fit, the smaller the better.
- The second term $\lambda\sum_{j=1}^{p}\beta_j^2$ is called shrikage penalty, which *shrinks* β_j towards 0.
- The shrinkage reduces variance (at the cost increased bias)!

Tuning parameter λ.

- $\lambda = 0$: no penalty, $\hat{\beta}_0^R = \hat{\beta}^{LS}$.
- $\lambda = \infty$: infinity penalty, $\hat{\beta}_0^R = 0$.
- Large λ: heavy penalty, more shrinkage of the estimator.
- Note that β_0 is not penalized.

Remark.

- If $p > n$, ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance.
- Ridge regression works best in situations where the least squares estimates have high variance.
- Ridge regression also has substantial computational advantages
- $$\hat{\beta}_{\lambda}^{R} = (\mathbf{X}^{T}\mathbf{X}) + \lambda I)^{-1}\mathbf{X}^{T}\mathbf{y}$$

  where I is $p+1$ by $p+1$ diagonal with diagonal elements $(0, 1, 1, ..., 1)$.

Example: Ridge Regularization Path in Credit data

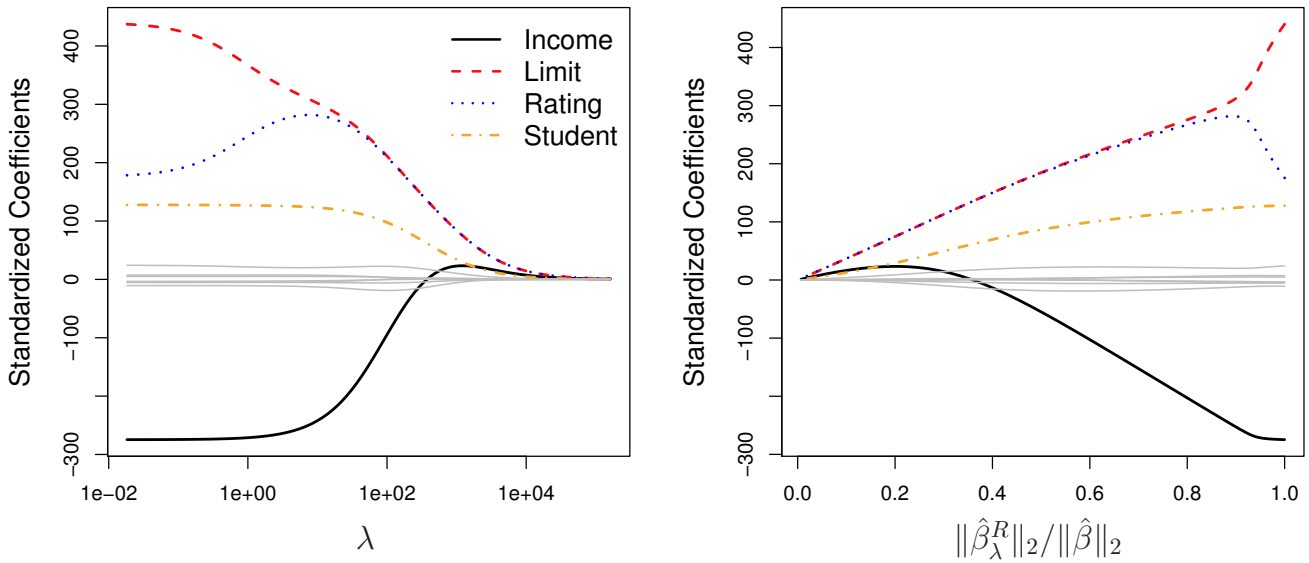

Figure: 6.4. The standardized ridge regression coefficients are displayed for the Credit data set, as a function of λ and $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. Here $\|\mathbf{a}\|_2 = \sqrt{\sum_{j=1}^{p} a_j^2}$.

The Lasso

- Lasso stands for Least Absolute Shrinkage and Selection Operator.
- The Lasso estimator $\hat{\beta}_\lambda^L$ is the minimizer of

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|$$

- We may use $\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$, which is the l_1 norm.
- LASSO often shrinks coefficients to be identically 0. (This is not the case for ridge)
- Hence it performs variable selection, and yields sparse models.

Example: Lasso Path in Credit data.

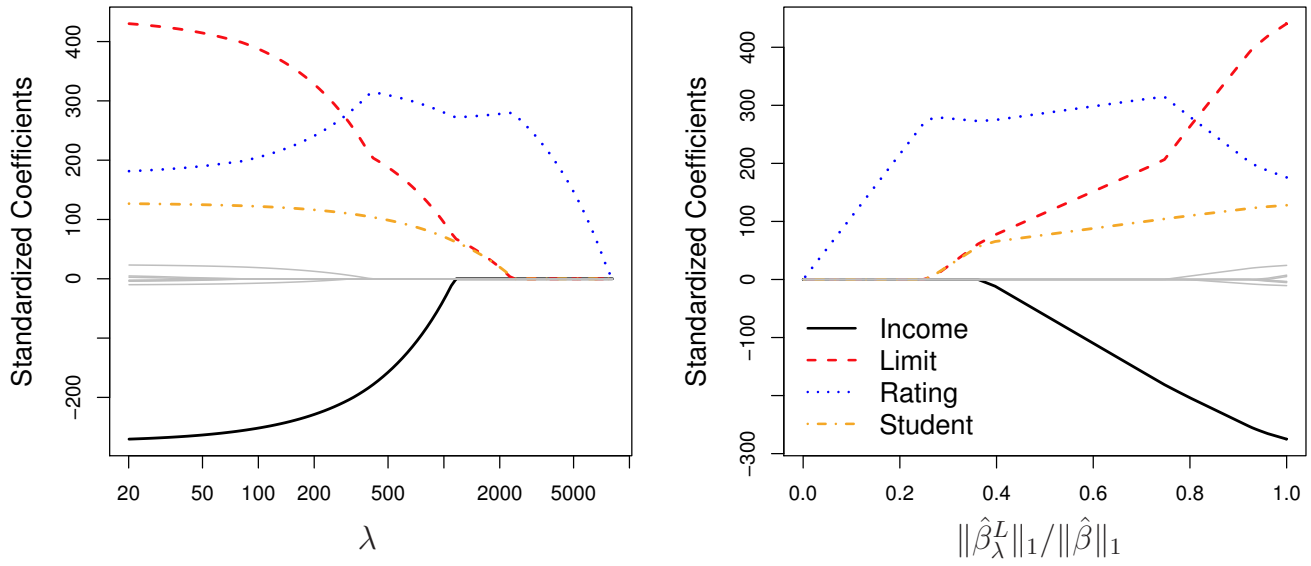

Figure: 6.6. The standardized lasso coefficients on the Credit data set are shown as a function of λ and $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

Another formulation

- For Lasso: Minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq s$$

- For Ridge: Minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 \text{ subject to } \sum_{j=1}^{p}\beta_j^2 \leq s$$

- For l_0: Minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 \text{ subject to } \sum_{j=1}^{p} I(\beta \neq 0) \leq s$$

  l_0 method penalizes number of non-zero coefficients. A difficult (NP-hard) problem for optimization.

Variable selection property for Lasso

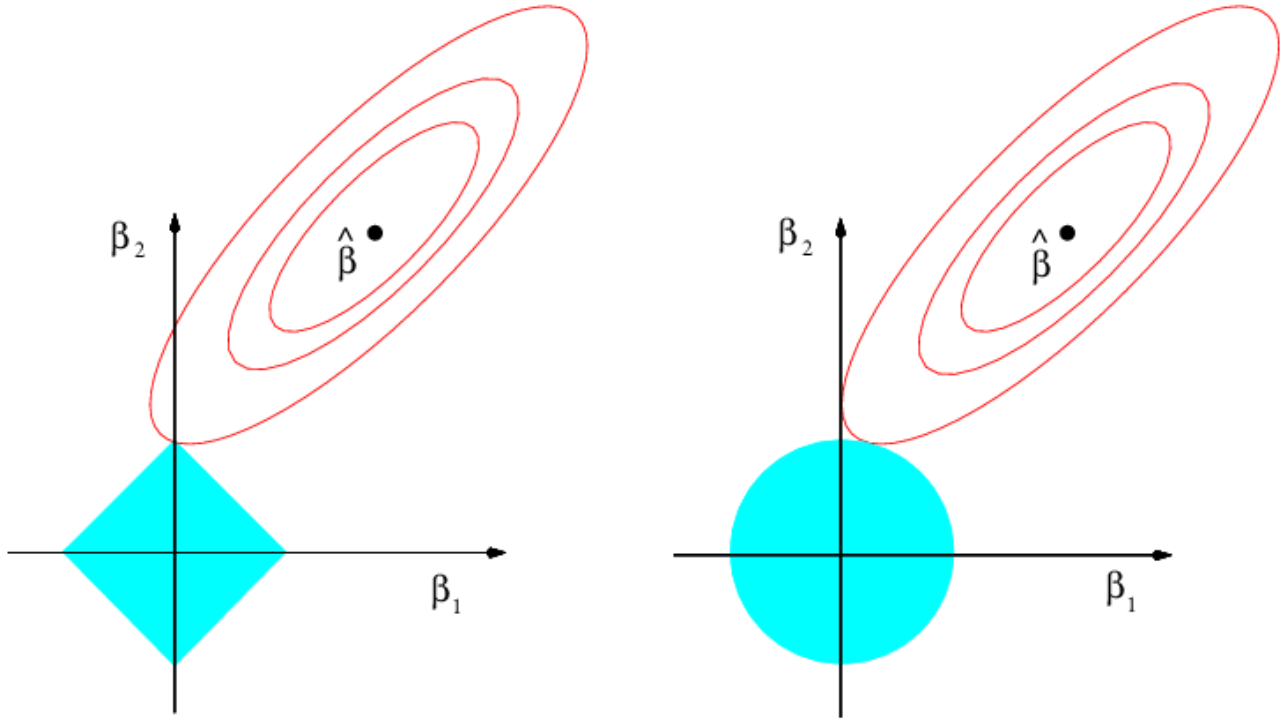

Figure: 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Simple cases

- Ridge has closed form solution. Lasso generally does not have closed form solution.
- Consider the simple model $y_i = \beta_i + \epsilon_i$, $i = 1, ..., n$ and $n = p$. Then,
  The least squares $\hat{\beta}_j = y_j$; the ridge $\hat{\beta}_j^R = y_j/(1+\lambda)$
  The Lasso $\hat{\beta}_j^L = \text{sign}(y_j)(|y_j| - \lambda/2)_+$.
- Slightly more generally, suppose input columns of the $\mathbf{X}$ are standardized to be mean 0 and variance 1 and are orthogonal.

$$\hat{\beta}_j^R = \hat{\beta}_j^{\text{LSE}}/(1+\lambda)$$

$$\hat{\beta}_j^L = \text{sign}(\hat{\beta}_j^{\text{LSE}})(|\hat{\beta}_j^{\text{LSE}}| - \lambda/2)_+$$

for $j = 1, ..., p$.

Example

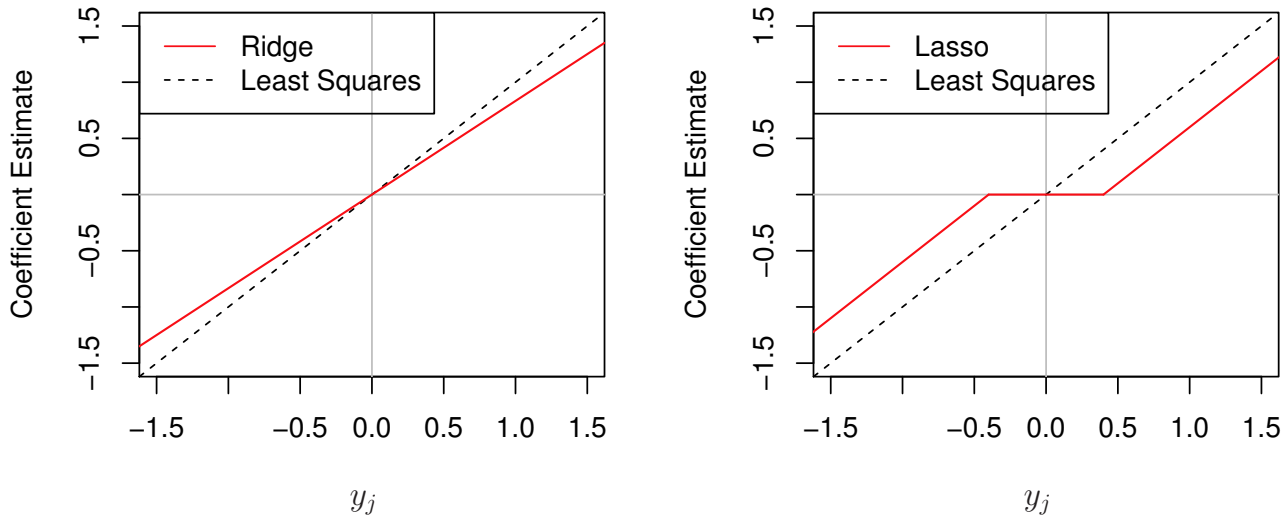

Figure: 6.10. The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and X a diagonal matrix with 1 on the diagonal. Left: The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates. Right: The lasso coefficient estimates are soft-thresholded towards zero.

Dimension reduction methods (using derived inputs)

- When p is large, we may consider to regress on, not the original inputs x, but some small number of derived features $\phi_1, ..., \phi_k$ with $k < p$.

$$y_i = \theta_0 + \sum_{j=1}^{k} \theta_j \phi_j(x_i) + \epsilon_i, \quad i = 1, ..., n.$$

  - ϕ_j can be linear: linear combinations of $X_1, \ldots, X_p$
  - ϕ_j can be nonlinear: basis, kernels, neural networks, trees, etc.

Principal Component Analysis (PCA)

- Suppose there are n observations of p variables presented as $\mathbf{X} = (x_1, \ldots x_n)^T \in \mathbf{R}^{n \times p}$, where $x_i^T \in \mathbf{R}^p$.
- Define the sample covariance matrix

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^T (x_i - \hat{\mu})$$

  where the sample mean $\hat{\mu} = \frac{1}{n} \sum_i x_i$.
- $\hat{\Sigma}$ has an eigenvalue decomposition

$$\hat{\Sigma} = U \Lambda U^T,$$

  with $U^T U = I_p$ ($U = [u_1, \ldots, u_p]$), $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$.

Principal Component Regression

For $X = (X_1, \ldots, X_p)$,

- Define ϕ_j be the projection on the j-th eigenvector of centralized data:

$$Z_j = \phi_j(X) = u_j^T (X - \hat{\mu})$$

- Principal Component Regression (PCR) model:

$$y_i = \theta_0 + \sum_{j=1}^{k} \theta_j Z_j + \epsilon_i, \quad i = 1, ..., n.$$

A summary table of PCs

		eigenvalue (variance)	eigenvector (combination coefficient)	percent of variation explained	P.C.s as projections of $X - \mu$
1st P.C.	Z_1	λ_1	u_1	$\lambda_1 / \sum_{j=1}^{p} \lambda_j$	$Z_1 = u_1'(X - \mu)$
2nd P.C.	Z_2	λ_2	u_2	$\lambda_2 / \sum_{j=1}^{p} \lambda_j$	$Z_2 = u_2'(X - \mu)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p-th P.C.	Z_p	λ_p	u_p	$\lambda_p / \sum_{j=1}^{p} \lambda_j$	$Z_p = u_p'(X - \mu)$

- where top k principal components explained the following percentage of total variations

$$\sum_{j=1}^{k} \lambda_j / \mathbf{trace}(\Sigma)$$