Model Assessment and Selection

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.

Model Assessment

- Cross Validation
- Bootstrap

Linear Model Selection

- Subset selection: Forward and Backward stagewise
- Ridge Regression
- The Lasso
- Principal Component Regression
- *Early Stopping Regularization in Gradient Descent Method

Outline

Model Assessment
- Cross Validation
- Bootstrap

Linear Model Selection
- Subset selection: Forward and Backward stagewise
- Ridge Regression
- The Lasso
- Principal Component Regression
- *Early Stopping Regularization in Gradient Descent Method

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 (prediction) error, by, for example, making adjustments of the training error.
  - The **adjusted R-squared, Mallow's C_p, AIC, BIC**, etc serve this purpose.
- 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.

Validation set approach

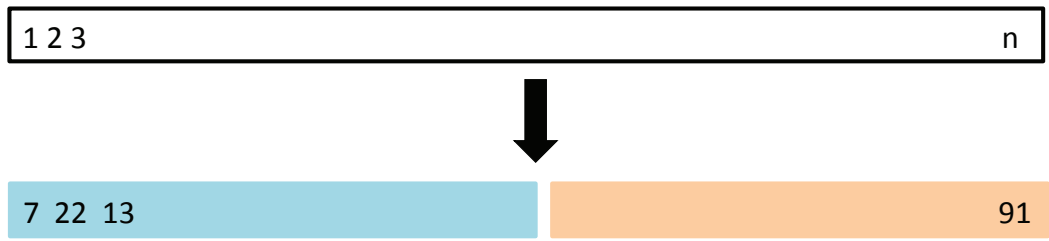

Figure: 5.1. A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.

Example: Auto Data

- A non-linear relationship between mpg and horsepower
- mpg $\sim$ horsepower $+$ horsepower^2 is better than mpg $\sim$ horsepower.
- Should we add higher terms into the model? E.g. cubic or even higher?
- One can check the p-values of regression coeffeicients to answer the question.
- In fact, a model selection problem, and we can use validation set approach.

Example: Auto Data

- randomly split the 392 observations into two sets:
- a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.
- fit various regression models on the training sample
- The validation set error rates result from evaluating their performance on the validation sample.
- Here MSE as a measure of validation set error, are shown in the left-hand panel of Figure 5.2.

Validation Errors

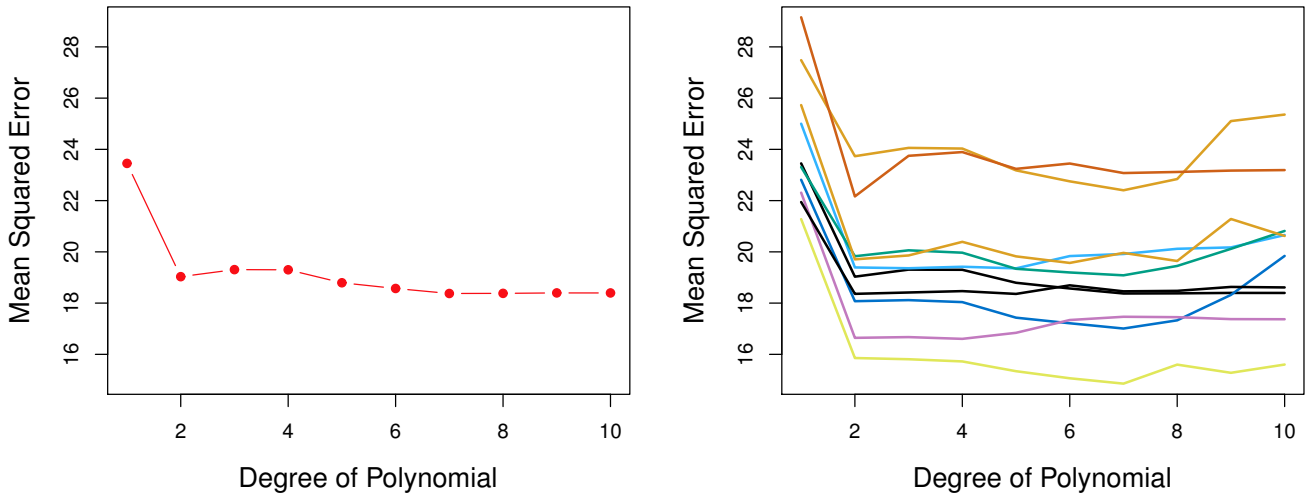

Figure: 5.2. The validation set approach was used on the Auto data set in order to estimate the test error that results from predicting mpg using polynomial functions of horsepower. Left: Validation error estimates for a single split into training and validation data sets. Right: The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.

Example: Auto Data

- The validation set MSE for the **quadratic** fit is considerably smaller than for the linear fit.
- Validation set MSE for the **cubic** fit is actually slightly larger than for the quadratic fit.
- This implies that including a cubic term in the regression does NOT lead to better prediction than simply using a quadratic term.
- Repeat the process of randomly splitting the sample set into two parts, we will get a somewhat different estimate for the test MSE.

Example: Auto Data

- A quadratic term has a dramatically smaller validation set MSE than the model with only a linear term.
- Not much benefit in including cubic or higher-order polynomial terms in the model.
- Each of the ten curves results in a different test MSE estimate for each of the ten regression models considered.
- No consensus among the curves as to which model results in the smallest validation set MSE.
- Based on the variability among these curves, all that we can conclude with any confidence is that the linear fit is not adequate for this data, and **quadratic** fit is preferred.

A summary

- The validation estimate of the test error rate can be highly variable, depending on the random split.
- Only a subset of the observations–the training set are used to fit the model.
- Statistical methods tend to perform worse when trained on fewer observations.
- The validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

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.

The leave-one-out cross-validation

- Suppose the data contain n data points.
- First, pick data point 1 as validation set, the rest as training set. fit the model on the training set, evaluate the test error, on the validation set, denoted as say MSE_1.
- Second, pick data point 2 as validation set, the rest as training set. fit the model on the training set, evaluate the test error on the validation set, denoted as say MSE_2.
- ..... (repeat the procedure for all data point.)
- Obtain an estimate of the test error by combining the MSE_i, $i =, ..., n$:

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

LOOCV

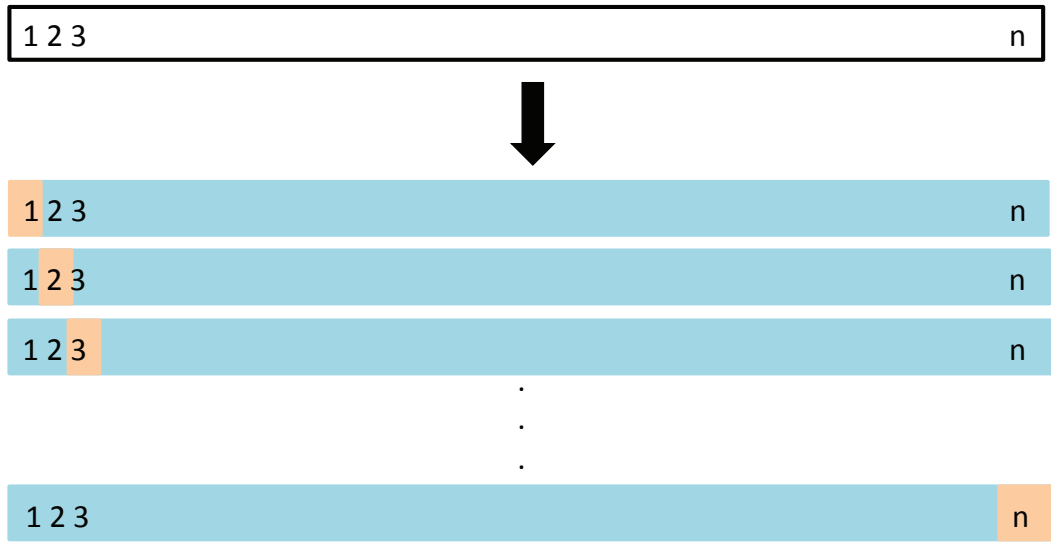

Figure: 5.3. A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.

Pros and Cons of LOOCV

- **Advantages**:
  - Far less bias, since the training data size $(n-1)$ is close to the entire data size (n).
  - One single test error estimate (thanks to the averaging), without the variability in validation set approach.
- **Disadvantages**:
  - A disadvantage: could be computationally expensive since the model need to be fit n times.
  - The MSE_i may be too much correlated.

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

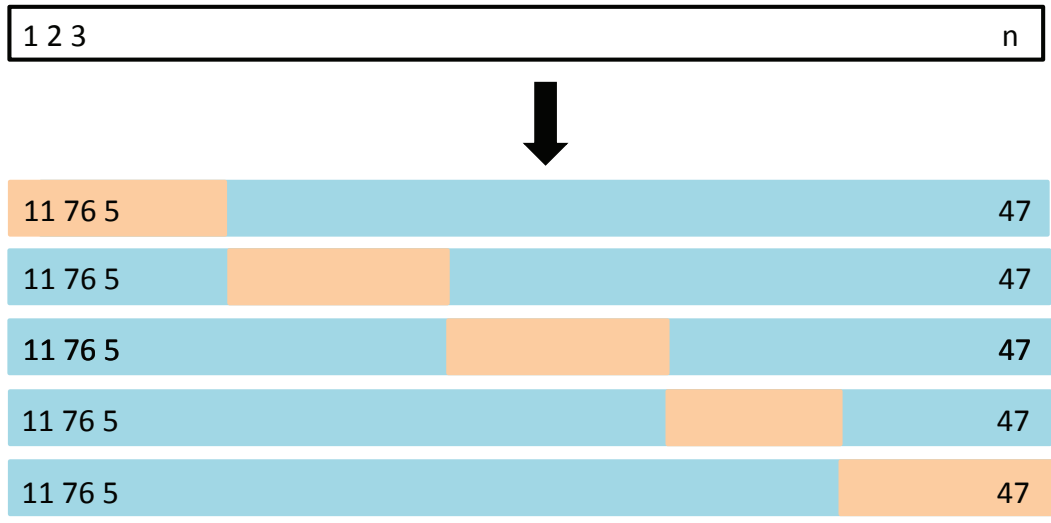

Figure: 5.5. A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE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

*Cross Validation in Time Series

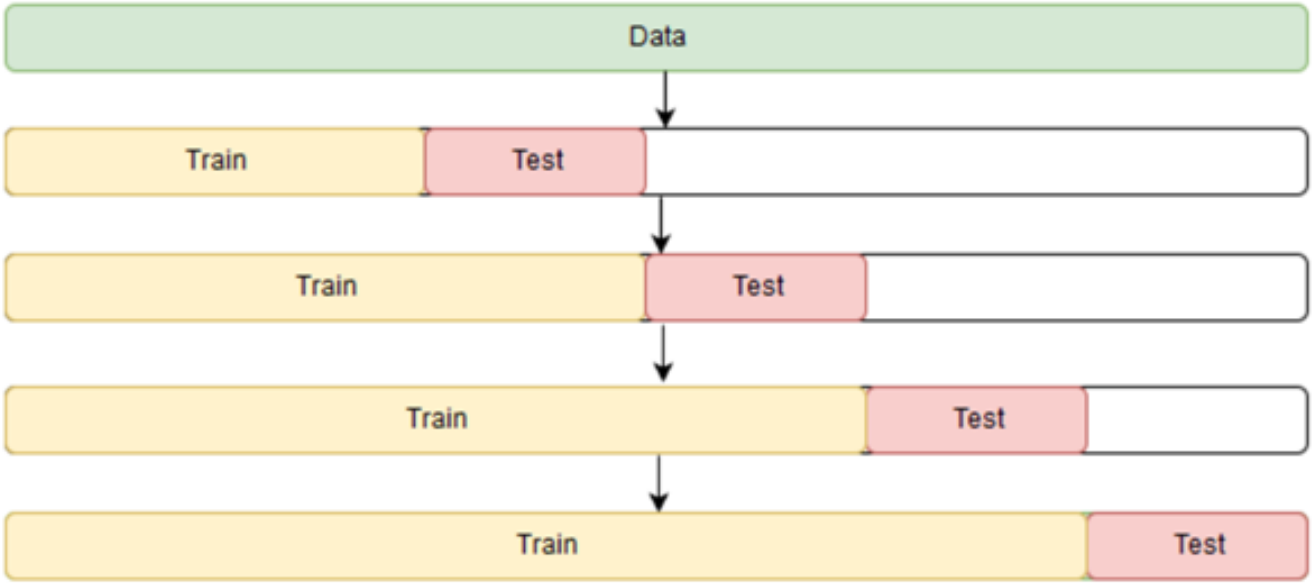

Figure: A schematic illustration for 4-fold cross validation for time series. For time-series models, cross-validation is on a rolling basis. Start with a small subset of data for training purpose, forecast for the later data points and then checking the accuracy for the forecasted data points. The same forecasted data points are then included as part of the next training dataset and subsequent data points are forecasted.

K-fold cross validation

- Common choices of K: $K = 5$ or $K = 10$.
- Advantage over LOOCV:
  - Computationally lighter, especially for complex model with large data.
  - Likely less variance.
- Advantage over validation set approach: Less variability resulting from the data-split, thanks to the averaging.

LOOCV applied to Auto data:

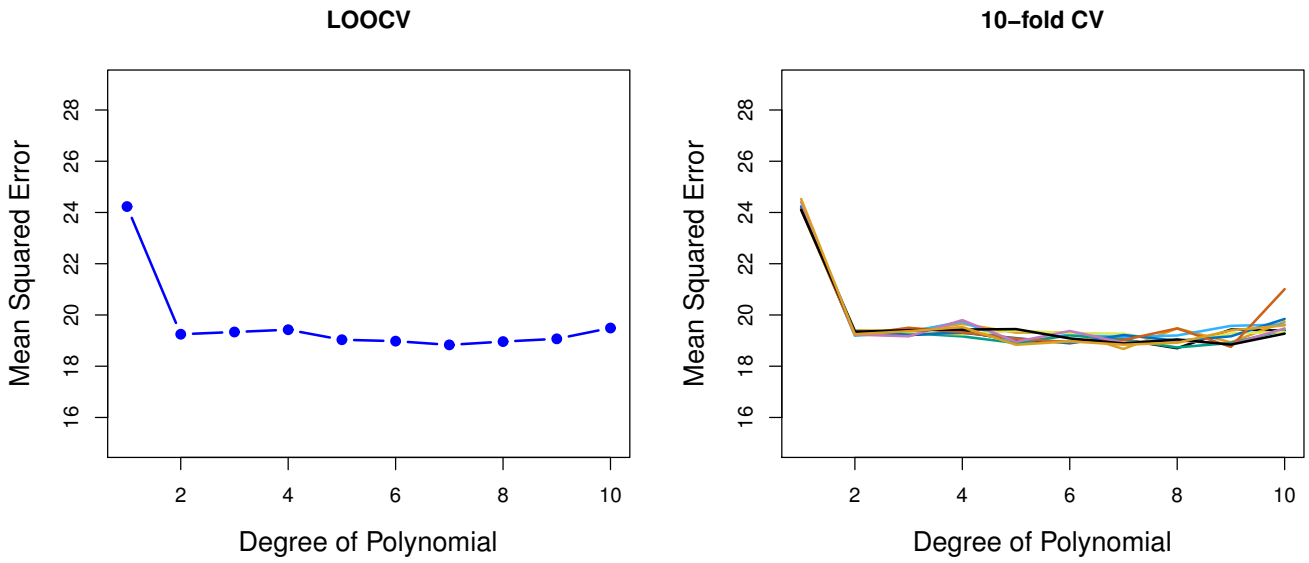

Figure: 5.4. Cross-validation was used on the Auto data set in order to estimate the test error that results from predicting mpg using polynomial functions of horsepower. Left: The LOOCV error curve. Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.

*LOOCV in linear model

- Consider linear model:

$$y_i = \mathbf{x}_i^T \beta + \epsilon_i, \qquad i = 1, .., n$$

and the fitted values $\hat{y}_i = \mathbf{x}_i^T \hat{\beta}$, where $\hat{\beta}$ is the least squares estimate of β based on all data $(\mathbf{x}_i, y_i), i = 1, ..., n$.

- Using LOOCV, the

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i^{(i)})^2$$

where $\hat{y}_i^{(i)} = \mathbf{x}_i^T \hat{\beta}^{(i)}$ is the model predictor of y_i based on the linear model fitted by all data except $(\mathbf{x}_i, y_i)$ (delete one), i.e., $\hat{\beta}^{(i)}$ is the least squares estimate of β based on all data but $(\mathbf{x}_i, y_i)$.

Simple Formula of LOOCV in linear model

- In fact, one does NOT need to compute least squares estimate n times.
- Easy formula:

Theorem

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

where $\hat{y}_i$ is the fitted values of least squares method based on all data, h_i is the **leverage**.

Definition: Leverage

- Recall the hat matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \quad \text{as } \hat{y} = \mathbf{H}y.$$

  Let $h_{ij} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_j$ be the (i, j) elements of $\mathbf{H}$.
- The **leverage** of the i-th observation is just the i-th diagonal element of $\mathbf{H}$, denoted as h_{ii}.
- A high leverage implies that observation is quite influential. Note that the average of h_{ii} is $(p + 1)/n$.
- E.g., if h_{ii} is greater than $2(p + 1)/n$, twice of the average, is generally considered large.

Why? Fast computation of cross-validation I

- The leave-one-out cross-validation statistic is given by

$$CV = \frac{1}{N}\sum_{i=1}^{N} e_{[i]}^2,$$

where $e_{[i]} = y_i - \hat{y}_{[i]}$, and $\hat{y}_{[i]}$ is the predicted value obtained when the model is estimated with the ith case deleted.

- Suppose we have a linear regression model $\mathbf{Y} = \mathbf{X}\beta + \mathbf{e}$. The $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ and $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the hat matrix. It has this name because it is used to compute $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{HY}$. If the diagonal values of $\mathbf{H}$ are denoted by $h_1, \ldots, h_N$, then the leave-one-out cross-validation statistic can be computed using

$$CV = \frac{1}{N}\sum_{i=1}^{N}[e_i/(1-h_i)]^2,$$

where $e_i = y_i - \hat{y}_i$ is predicted value obtained when the model is estimated with all data included.

Fast computation of cross-validation II

Proof

- Let $\mathbf{X}_{[i]}$ and $\mathbf{Y}_{[i]}$ be similar to $\mathbf{X}$ and $\mathbf{Y}$ but with the ith row deleted in each case. Let $\mathbf{x}_i^T$ be the ith row of $\mathbf{X}$ and let

$$\hat{\beta}_{[i]} = (\mathbf{X}_{[i]}^T\mathbf{X}_{[i]})^{-1}\mathbf{X}_{[i]}^T\mathbf{Y}_{[i]}$$

  be the estimate of β without the ith case. Then $e_{[i]} = y_i - \mathbf{x}_i^T\hat{\beta}_{[i]}$.
- Now $\mathbf{X}_{[i]}^T\mathbf{X}_{[i]} = (\mathbf{X}^T\mathbf{X} - \mathbf{x}_i\mathbf{x}_i^T)$ and $\mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i = h_i$. So by the **Sherman-Morrison-Woodbury** formula,

$$(\mathbf{X}_{[i]}^T\mathbf{X}_{[i]})^{-1} = (\mathbf{X}^T\mathbf{X})^{-1} + \frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i\mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}}{1-h_i}.$$

Fast computation of cross-validation III

Proof

- Also note that $\mathbf{X}_{[i]}^T \mathbf{Y}_{[i]} = \mathbf{X}^T\mathbf{Y} - \mathbf{x}_i y_i$. Therefore

$$\begin{aligned}
\hat{\beta}_{[i]} &= \left[(\mathbf{X}^T\mathbf{X})^{-1} + \frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i\mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}}{1-h_i}\right](\mathbf{X}^T\mathbf{Y} - \mathbf{x}_i y_i) \\
&= \hat{\beta} - \left[\frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i}{1-h_i}\right][y_i(1-h_i) - \mathbf{x}_i^T\hat{\beta} + h_i y_i] \\
&= \hat{\beta} - (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i e_i/(1-h_i)
\end{aligned}$$

- Thus

$$\begin{aligned}
e_{[i]} &= y_i - \mathbf{x}_i^T\hat{\beta}_{[i]} \\
&= y_i - \mathbf{x}_i^T\left[\hat{\beta} - (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i e_i/(1-h_i)\right] \\
&= e_i + h_i e_i/(1-h_i) = e_i/(1-h_i).
\end{aligned}$$

□

Simplicity of LOOCV in linear model

- One fit (with all data) does it all!
- The prediction error rate (in terms of MSE) is just weighted average of the least squares fit residuals.
- High leverage point gets more weight in prediction error estimation.

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**:

Bootstrap: 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 B times. We have $\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_B^*$.
- Use the distribution of $\bar{x}_1^* - \bar{x}, ..., \bar{x}_B^* - \bar{x}$ to approximate that of $\bar{x} - \mu$.

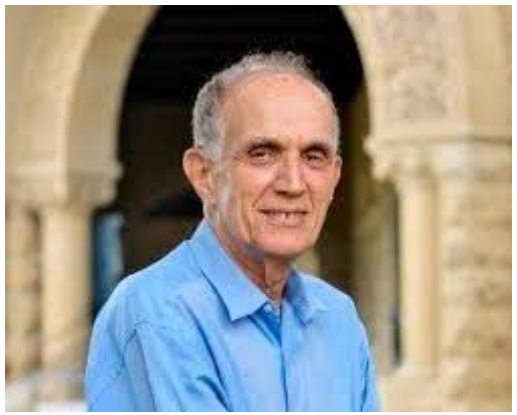

Figure: Bootstrap is firstly developed by Bradley Efron (1979)

- Essential idea: Treat the data distribution (more professionally called empirical distribution) 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.
- Evaluate your statistical procedure (which may be difficult because it involves randomness and the unknown population distribution) by evaluating your analogue procedures on the re-samples.

Example

- X and Y are two random variables (e.g. stocks). Then minimizer of $\text{var}(\alpha X + (1-\alpha)Y))$ (e.g. minimal risk) 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 evaluate $\hat{\alpha} - \alpha$, (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 $\hat{\alpha}_1^*, ..., \hat{\alpha}_B^*$ for a large B.
- Use the distribution of $\hat{\alpha}_1^* - \hat{\alpha}, ..., \hat{\alpha}_B^* - \hat{\alpha}$ to approximate the distribution of $\hat{\alpha} - \alpha$.
- For example, we can use

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

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

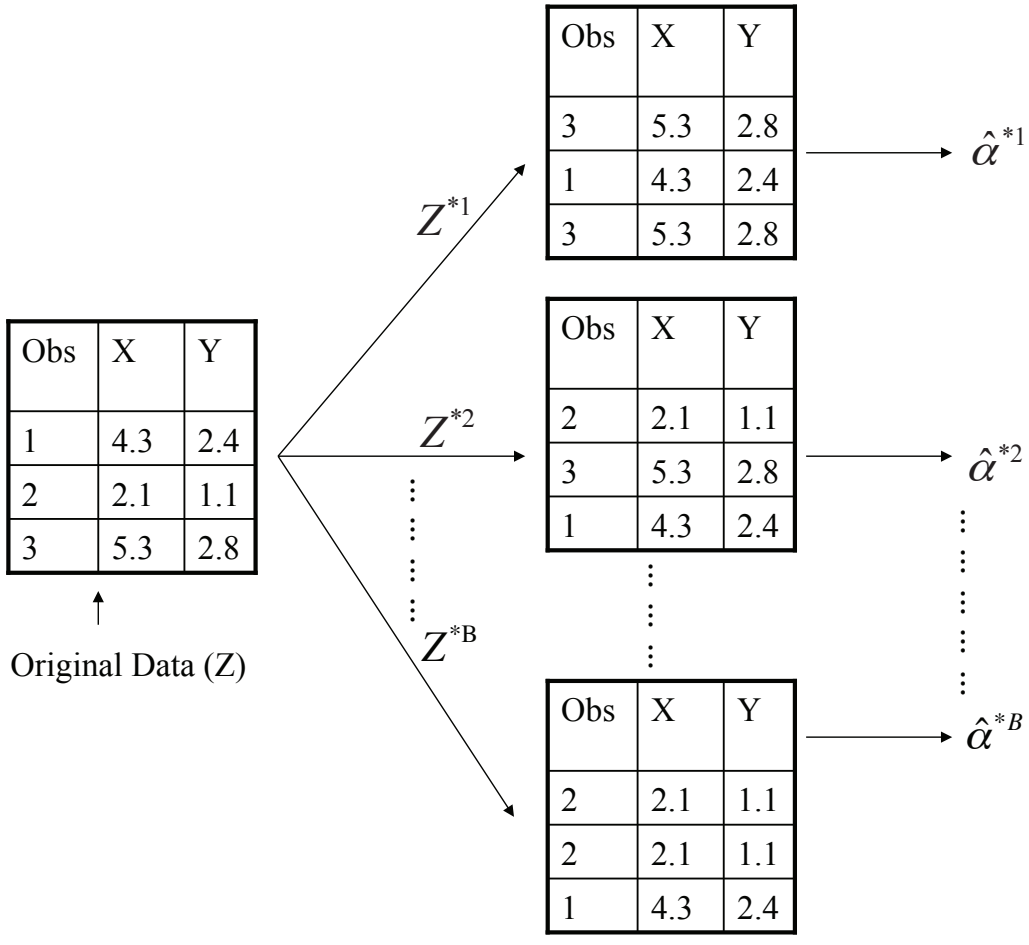
Obs X Y
1 4.3 2.4
2 2.1 1.1
3 5.3 2.8
Original Data (Z)
Z^{*1}
Z^{*2}
Z^{*B}
Obs X Y
3 5.3 2.8
1 4.3 2.4
3 5.3 2.8
Obs X Y
2 2.1 1.1
3 5.3 2.8
1 4.3 2.4
Obs X Y
2 2.1 1.1
2 2.1 1.1
1 4.3 2.4
$\hat{\alpha}^{*1}$
$\hat{\alpha}^{*2}$
$\hat{\alpha}^{*B}$

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.

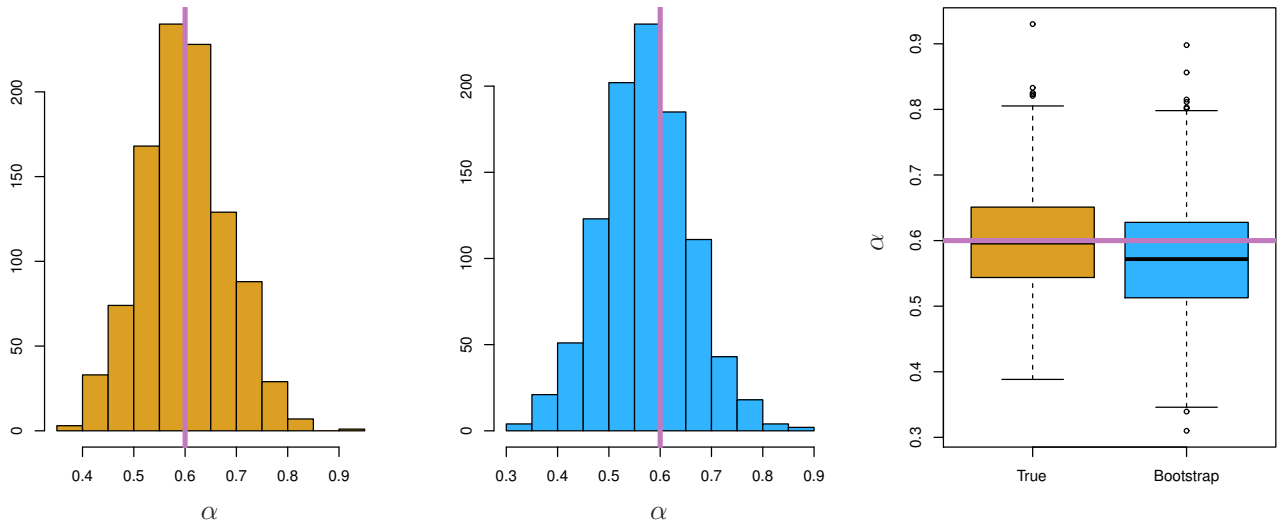

Figure: 5.10. Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population. Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set. Right: The estimates of α displayed in the left and center panels are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

Outline

Model Assessment
- Cross Validation
- Bootstrap

Linear Model Selection
- Subset selection: Forward and Backward stagewise
- Ridge Regression
- The Lasso
- Principal Component Regression
- *Early Stopping Regularization in Gradient Descent Method

Interpretability vs. Prediction

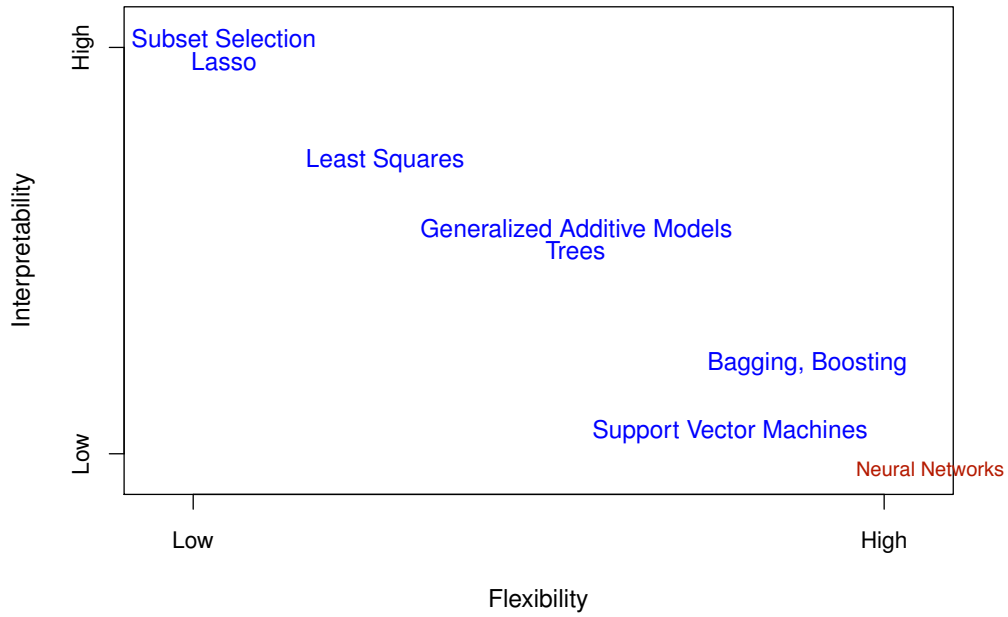

Figure: 2.7. As models become flexible, interpretability drops. **Occam Razor principle**: Everything has to be kept as simple as possible, but not simpler (Albert Einstein).

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:
  - Subset selection.
  - Shrinkage/regularization: constraining some regression parameters to 0, e.g. Ridge and Lasso.
  - 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-variable model, add one more variable to get the best two-variable model.
- With the best two-variable 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 as the Training Error

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

- R-squared

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

where the Total Sum of Squares $\text{TSS} := \sum_{i=1}^{n} (y_i - \bar{y})^2$.

Example: Credit data

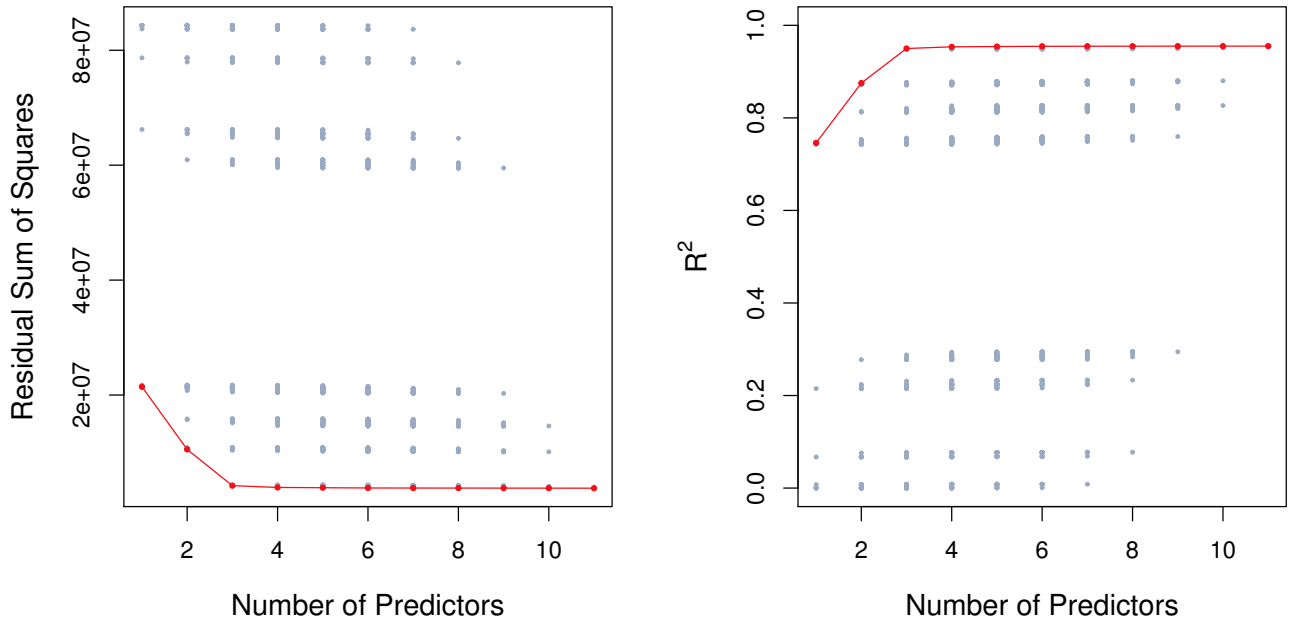

Figure: 6.1. For each possible model containing a subset of the ten predictors in the Credit data set, the RSS and R^2 are displayed. The red frontier tracks the best model for a given number of predictors, according to RSS and R^2. Though the data set contains only ten predictors, the x-axis ranges 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

$$\text{adjusted } R^2 \quad = \quad 1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)}$$

- 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.
- Maximizing the adjusted R-squared, is equivalent to

$$\text{minimize } \text{RSS}/(n-p-1).$$

b) Mallow's C_p

- Recall that our linear model (2.1) has p covariates, and $\hat{\sigma}^2 = RSS/(n-p-1)$ is the unbiased estimator of σ^2.
- Suppose we use only d predictors and RSS(d) is the residual sum of squares for the linear model with d predictors.
- The statistic of Mallow's C_p is defined as

$$C_p = \frac{\text{RSS}(d) + 2d\hat{\sigma}^2}{n}$$

- The smaller Mallows' C_p is, the better the model is.
- The following AIC is more often used, despite that Mallows' C_p and AIC usually give the same best model.

c) AIC

- AIC stands for Akaike information criterion, defined as

$$\text{AIC} = \frac{\text{RSS}(d) + 2d\hat{\sigma}^2}{n\hat{\sigma}^2},$$

  for a linear model with $d \leq p$ predictors, where $\hat{\sigma}^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2/(n-p-1)$ is the unbiased estimator of σ^2 using the full model.
- AIC aims at maximizing the expected predictive likelihood (or minimizing the expected predictive error). The model with the smallest AIC is preferred.

d) BIC

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

$$\text{BIC} = \frac{\text{RSS}(d) + d\hat{\sigma}^2 \log(n)}{n\hat{\sigma}^2},$$

  for a linear model with d inputs.
- The model with the smallest BIC is preferred. 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 samples, $\log n > 2$.

Penalized log-likelihood

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

$$\textbf{minimize} - (\log \text{ likelihood}) + d \log(n)/n$$

where, the first term is called deviance (some refer it to $-2 \log$ likelihood). 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

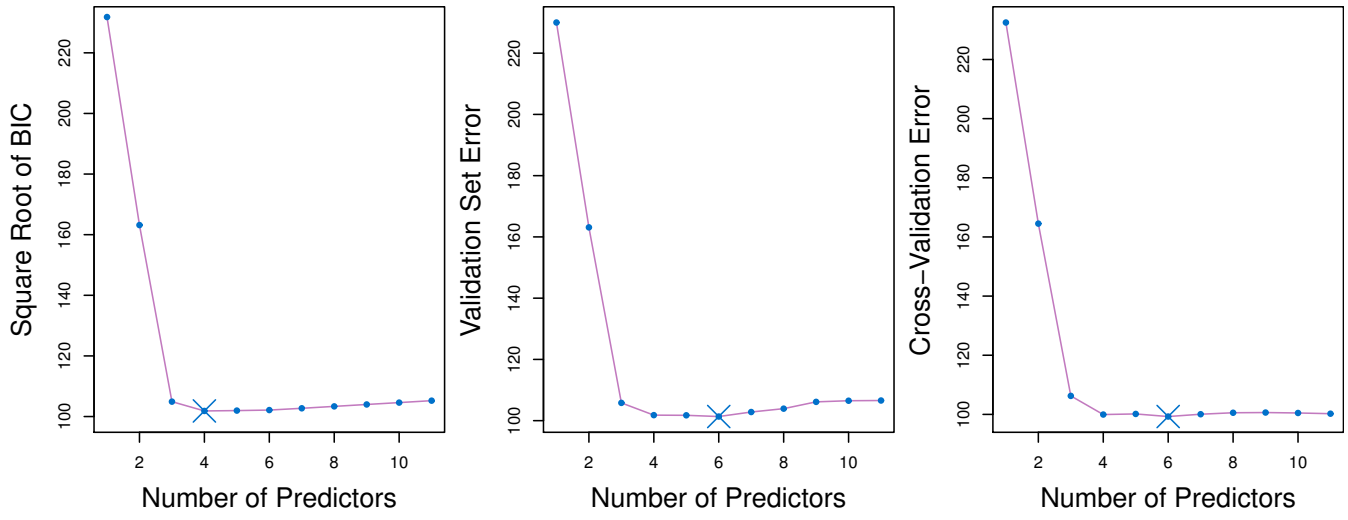

Figure: 6.3. For the Credit data set, three quantities are displayed for the best model containing d predictors, for d ranging from 1 to 11. The overall best model, based on each of these quantities, is shown as a blue cross. Left: Square root of BIC. Center: Validation set errors (75% training data). 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 measures goodness of fit, the smaller the better.
- The second term $\lambda\sum_{j=1}^{p}\beta_j^2$ is called shrinkage 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, $\hat{\beta}_\infty^R = 0$.
- 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
- Closed form estimator

$$\hat{\beta}_{\lambda}^{R} = (\mathbf{X}^{T}\mathbf{X} + \lambda I)^{-1}\mathbf{X}^{T}\mathbf{y}$$

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

Example: Ridge Regularization Path in Credit data

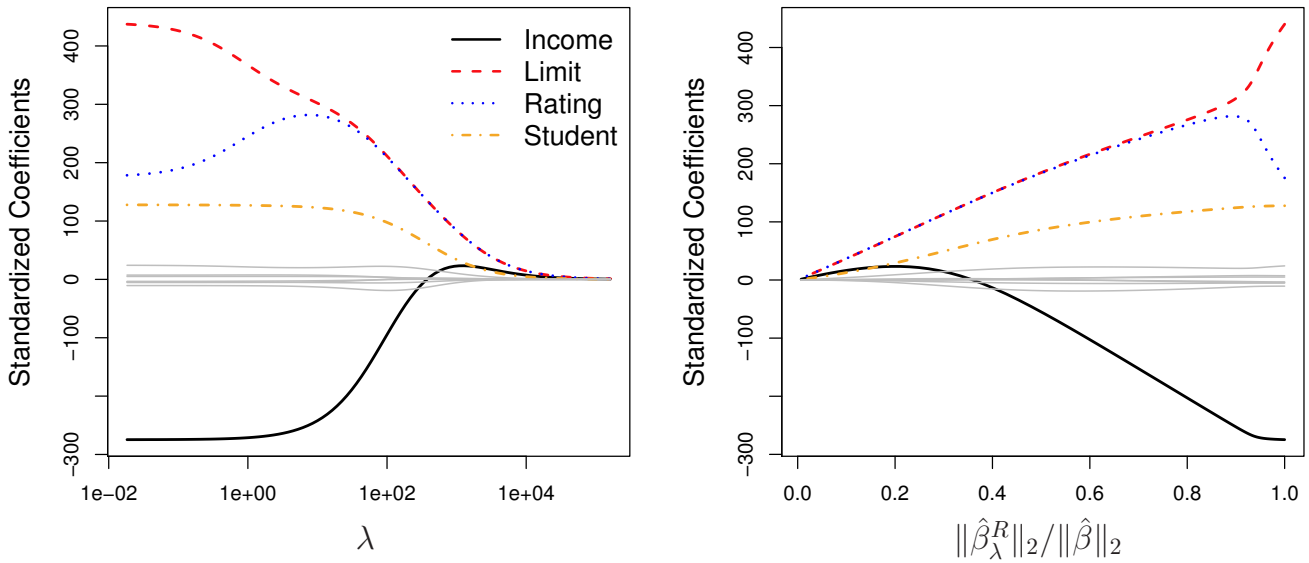

Figure: 6.4. The standardized ridge regression coefficients are displayed for the Credit data set, as a function of λ and $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. Here $\|\mathbf{a}\|_2 = \sqrt{\sum_{j=1}^{p} a_j^2}$.

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.

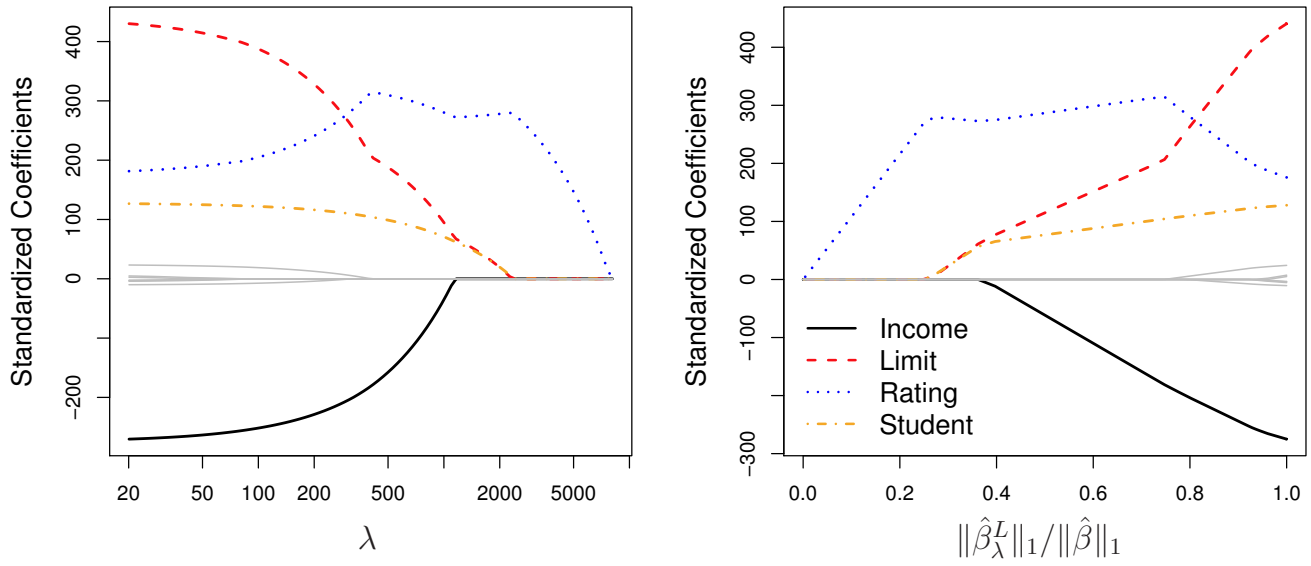

Figure: 6.6. The standardized lasso coefficients on the Credit data set are shown as a function of λ and $\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|_1$.

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

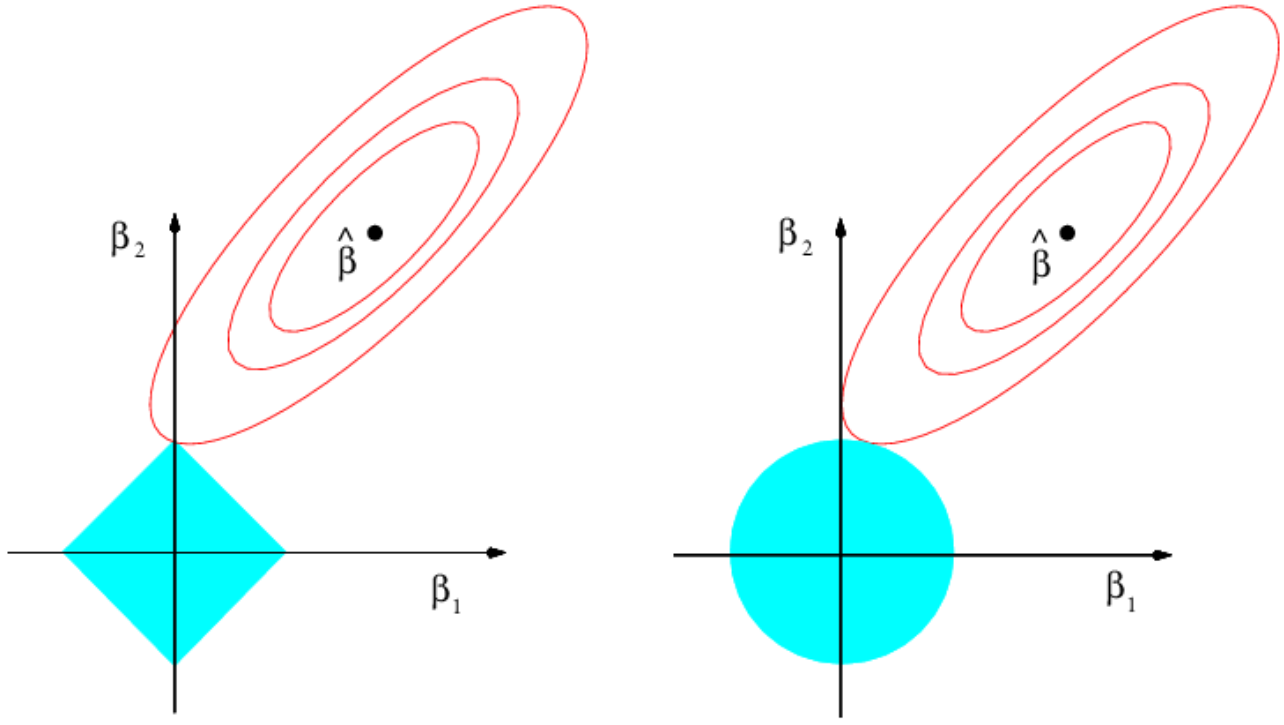

Figure: 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Simple cases

- 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

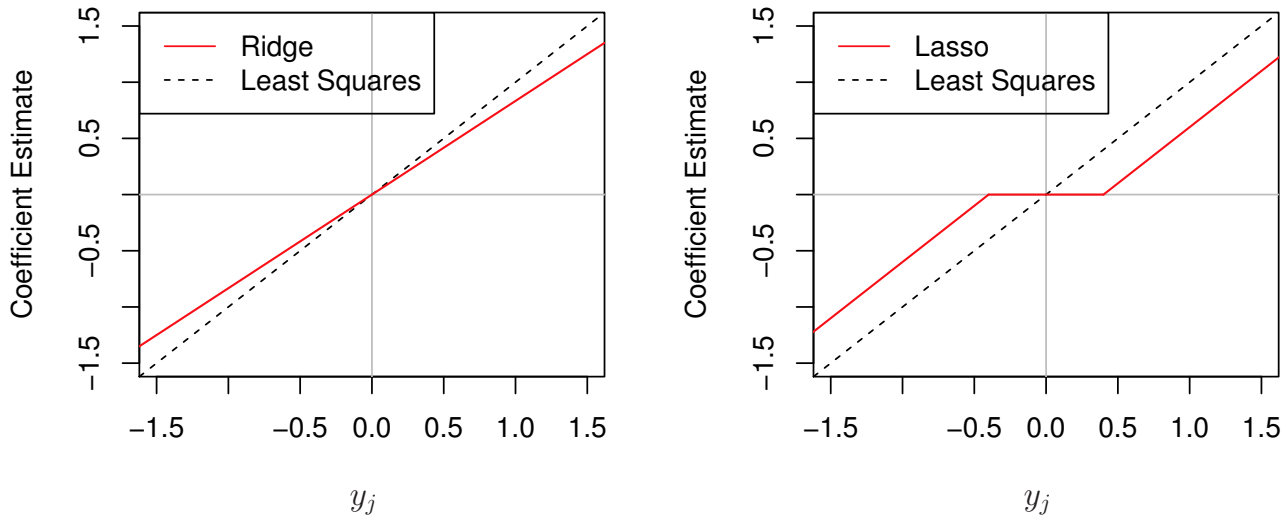

Figure: 6.10. The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and X a diagonal matrix with 1 on the diagonal. Left: The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates. Right: The lasso coefficient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

Example for curse of dimensionality

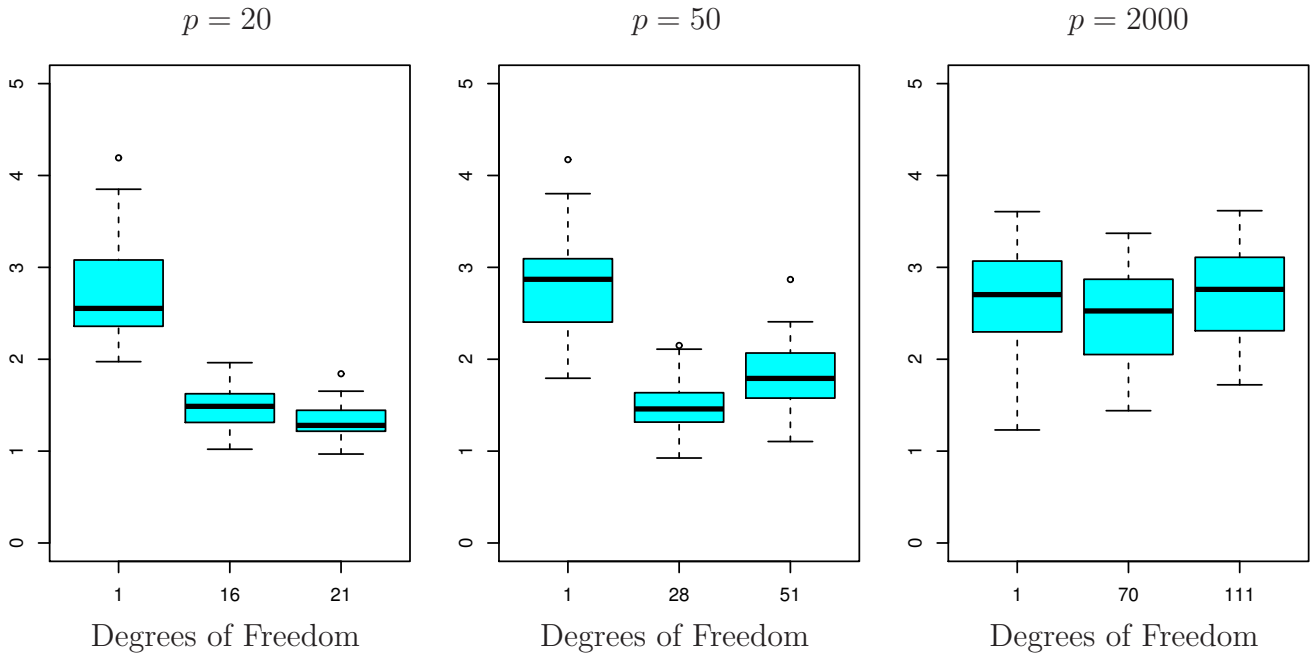

Figure: 6.24. see next page

Figure 6.24. The lasso was performed with $n = 100$ observations and three values of p, the number of features. Of the p features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter λ in (6.7). For ease of interpretation, rather than reporting , the degrees of freedom are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2,000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.

Caution when $p > n$.

- Extreme multicollinearity.
- Model selection consistency for Lasso needs weak correlation among features (**incoherence (irrepresentable)** conditions).
- Refrain from over-statement. (What we find may be one of many possible models, false discoveries)
- Avoid using sum of squares, p-values, R^2, or other traditional measures of model on training as evidence of good fit.
- Place more emphasis on test error or cross validation error.

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

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

  with $V^T V = I_p$ ($U = [v_1, \ldots, v_p]$), $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$.

Principal Component Regression

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

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

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

- 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	v_1	$\lambda_1 / \sum_{j=1}^{p} \lambda_j$	$Z_1 = v_1'(X - \mu)$
2nd P.C.	Z_2	λ_2	v_2	$\lambda_2 / \sum_{j=1}^{p} \lambda_j$	$Z_2 = v_2'(X - \mu)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p-th P.C.	Z_p	λ_p	v_p	$\lambda_p / \sum_{j=1}^{p} \lambda_j$	$Z_p = v_p'(X - \mu)$

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

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

Ridge Regression as Shrinkage on Principal Components

- Assume the (centralized) design matrix admits the singular value decomposition

$$\mathbf{X} = \Phi S \Psi^T, \quad S = \mathrm{diag}(\sigma_i) \text{ with } \sigma_i \geq 0$$

where $\Phi^T\Phi = \Psi^T\Psi = I$, then covariance matrix has eigenvalue decomposition: $\hat{\Sigma} := \mathbf{X}^T\mathbf{X} = \Psi\Lambda\Psi^T$, where $\Lambda = \mathrm{diag}(\lambda_i = \sigma_i^2)$ $(\lambda_1 \geq \lambda_2 \geq \cdots \geq 0)$.

- Ridge regression prediction

$$\begin{aligned} \hat{\mathbf{y}} = \mathbf{X}\hat{\beta}_\lambda^R &= \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y} \\ &= \sum_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \langle \phi_j, \mathbf{y} \rangle \phi_j \end{aligned}$$

  - $\sigma_j^2 \ll \lambda$, $\sigma_j^2/(\sigma_j^2 + \lambda) \approx 0$,
  - $\sigma_j^2 \gg \lambda$, $\sigma_j^2/(\sigma_j^2 + \lambda) \approx 1$.

High dimensionality $p > n$

- $g_\lambda(x) = (x+\lambda)^{-1}$ is the regularization function, s.t. $\hat{\beta}^R_\lambda = g_\lambda(\mathbf{X}^T\mathbf{X})\mathbf{X}^T\mathbf{y}$
- $\mathbf{X}^T\mathbf{X}$ has rank r no more than $n < p$, thus not invertible
- Ridge regression as $\lambda \to 0^+$ gives pseudo-inverse

$$\begin{aligned}
\hat{\beta}^R_\varepsilon &= \mathbf{X}^\dagger\mathbf{y} \\
&= (\mathbf{X}^T\mathbf{X} + \varepsilon I)^{-1}\mathbf{X}^T\mathbf{y} \\
&= \sum_{1\le j\le r:\sigma_j^2>0} \frac{\sigma_j^2}{\sigma_j^2+\varepsilon}\langle\phi_j, \mathbf{y}\rangle\psi_j
\end{aligned}$$

Gradient Descent Method

- Gradient Descent Algorithm:

$$\widehat{\beta}_{k+1} = \widehat{\beta}_k + \gamma_k \mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\beta}_k), \tag{1}$$

  - Initialization: $\widehat{\beta}_0 = 0$ or small random ones
  - Stepsize: $\gamma_k > 0$
  - $\widehat{\beta}_1 = (1 - \gamma_1 \mathbf{X}^T\mathbf{X})\widehat{\beta}_0 + \gamma_0 \mathbf{X}^T\mathbf{y}$,
  - $\widehat{\beta}_2 = \prod_{k=0}^{1}(1 - \gamma_k \mathbf{X}^T\mathbf{X})\widehat{\beta}_0 + \widehat{\beta}_1\gamma_1 \mathbf{X}^T\mathbf{y} + \gamma_0(1 - \gamma_1 \mathbf{X}^T\mathbf{X})\mathbf{X}^T\mathbf{y}$,
  - and so on ...

Implicit Regularization in Gradient Descent

- Gradient Descent regularization path:

$$\widehat{\beta}_t = \pi_0^{t-1}(\mathbf{X}^T\mathbf{X})\widehat{\beta}_0 + g_t(\mathbf{X}^T\mathbf{X})\mathbf{X}^T\mathbf{y} \tag{2}$$

where

  - For $x \in \mathbb{R}$, define a polynomial of degree $t-k+1$,

$$\pi_k^t(x) = \begin{cases} \prod_{i=k}^{t}(1-\gamma_i x), & k \le t; \\ 1, & k > t. \end{cases} \tag{3}$$

  - *Regularization polynomial* at iteration t:

$$g_t(x) = \sum_{k=0}^{t-1} \gamma_k \pi_{k+1}^{t-1}(x); \tag{4}$$

where $1 - xg_t(x) = \pi_0^{t-1}(x)$ by telescope sum.

Implicit Regularization in Gradient Descent

- Consider constant stepsize $\gamma_k = \alpha < 1/\|\mathbf{X}^T\mathbf{X}\|_2$ and $\widehat{\beta}_0 = 0$
- *Regularization polynomial*:

$$g_t(x) = \alpha \sum_{k=0}^{t-1} (1 - \alpha x)^{t-k-1} = \frac{1 - (1 - \alpha x)^t}{x};$$

- Regularization path:

$$\widehat{\beta}_t = g_t(\mathbf{X}^T\mathbf{X})\mathbf{X}^T\mathbf{y} = \sum_j \frac{1 - (1 - \alpha\sigma_j^2)^t}{\sigma_j} \langle \phi_j, \mathbf{y} \rangle \psi_j \qquad (5)$$

  - Large σ_j (low freq): $g_t(\sigma_j)$ drops slowly;
  - Small σ_i (high freq): $g_t(\sigma_j)$ drops fast.
  - A similar role as $g_\lambda(x) = 1/(x + \lambda)$ in ridge regression, yet better in nonparametrics (Yao-Rosasco-Caponnetto (2007))

Early Stopping Regularization in Gradient Descent

- For $\mathbf{y} = \mathbf{X}\beta + \epsilon$, the estimation error

$$\begin{aligned}
\beta - \widehat{\beta}_t &= \beta - g_t(\mathbf{X}^T\mathbf{X})\mathbf{X}^T(\mathbf{X}\beta + \epsilon) \\
&= (1 - \alpha\mathbf{X}^T\mathbf{X})^t\beta + g_t(\mathbf{X}^T\mathbf{X})\mathbf{X}^T\epsilon, \\
&= \sum_j \Big\{(1 - \alpha\sigma_j^2)^t\langle\psi_j, \beta\rangle\psi_j + \dots \\
&\qquad + \frac{1 - (1 - \alpha\sigma_j^2)^t}{\sigma_j}\langle\phi_j, \epsilon\rangle\psi_j\Big\}.
\end{aligned}$$

  - The first term is *bias* that decreases exponentially with $t \to \infty$
  - The second term is *variance* that increases with t
  - Early stopping: take the optimal stopping time t^* towards a bias-variance trade-off

In comparison, Ridge Regression

- For $\widehat{\beta}_\lambda^R = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y}$ and $\mathbf{y} = \mathbf{X}\beta + \epsilon$,

$$\begin{aligned}\widehat{\beta}_\lambda^R - \beta &= \lambda(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\beta + (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\epsilon, \\ &= \sum_j \left\{ \frac{\lambda}{\sigma_j^2 + \lambda}\langle \psi_j, \beta\rangle \psi_j + \frac{\sigma_j}{\sigma_j^2 + \lambda}\langle \phi_j, \epsilon\rangle \psi_j \right\}.\end{aligned}$$

  - The first term is *bias* that decreases with $\lambda \to 0$
  - The second term is *variance* that increases as $\lambda \to 0$
  - Optimal regularization $\lambda^* \sim 1/t^*$ for a bias-variance trade-off