Statistical Learning

Wenjie Li

October 24, 2019

1 Lec 1 Introduction

Where do we get the test data:

1. Train - Test
2. Cross Validation

K-Nearest Neighbor algorithm: K controls the flexibility

The weight for each data point is either 1 or 0

$$\frac{\sum_{i=1}^{n} N(x_i, x) y_i}{\sum_{i=1}^{n} N(x_i, x)}$$

The weight by K-nearest neighbor is too rigid, either 1 or 0. We can weight the data points based on distance

$$K(x_0, x) = e^{-\frac{(x-x_0)^2}{\lambda}}$$

Classifier for a new x:

$$\frac{\sum_{i=1}^{n} K(x_i, x) y_i}{\sum_{i=1}^{n} K(x_i, x)}$$

2 Lec 2 Linear Regression

2.1 Simple Linear Regression

We assume a model

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

Given the training data $\{x_i, y_i\}, i = 1, 2, ...n$, then we have some estimates for the parameters $\hat{\beta}_0, \hat{\beta}_1$. We predict y_i using

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

The residual $e_i = y_i - \hat{y}_i$

Residual sum of squares (RSS)$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$

By minimizing the RSS, we would derive the estimated parameter $\hat{\beta}_0, \hat{\beta}_1$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta_1}\bar{x}$$

The estimates that minimize RSS are the same as the Maximum Likelihood Estimate(MLE) assuming Gaussian noise.
If $\text{Var}(\epsilon) = \sigma^2$

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$Var(\hat{\beta}_0) = \sigma^2[\frac{1}{n} + \frac{\sum_{i=1}^{n}}{(x_i - \bar{x})^2}]$$

Standard errors can be used to perform hypothesis tests on the coefficients.
H_0 : There is no relationship between X and Y.
This is equal to $H_0 : \beta_1 = 0$ and the model is $Y = \beta_0 + \epsilon$
To test the null hypothesis, we compute a t-statistic, given by

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

This will be a t-distribution with $n - 2$ degrees of freedom, assuming $\beta_1 = 0$
Using statisticall sofeware, we can compute the probability of observing any value equal to $|t|$ or larger, we call it the $p - value$.
R-squared $R^2 = \frac{TSS-RSS}{TSS} = 1 - \frac{RSS}{TSS}$
It can be shown that in the simple linear regression, $R^2 = r^2$, where r is the correlation between X and Y

2.2 Multiple Linear Regression

The new model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + epsilon$
Given estimates $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$, we can make predictions

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p$$

we estimate $\hat{\beta}_0$, $\hat{\beta}_1$, ...,$\hat{\beta}_p$ as the values that minimize the sum of squared residuals.

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

$$RSS = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + ... + \hat{\beta}_p x_{ip})^2$$

Question 1: Is at least one of the predictors $X_1, X_2, ..., X_p$ useful in predicting the response.
We can use the F-statistic

$$F = \frac{(TSS - RSS)/p}{RSS/(n - p - 1)} \ F_{p,n-p-1}$$

Question 2: Are all the predictors useful?
The most direct approach is called all subsets or best subsets. Compute the least square fit for all possible subsets, and choose the minimum RSS

Forward Selection

1. Fit p simple linear regressions and add the variable that results in the lowest RSS.
2. Add the second variable, fit $p-1$ two-variable models and add the variable that results in the lowest RSS
3. Stop when the remaining variables have a p-value above some threshold.

Backward Selection

1.Fit all variables in the model, remove the one with the largest p-value
2. Fit the new $n-1$ variable model, and the variable with the largest p-value is removed
3. Stop when the remaining variables have a p-value below some threshold.

2.3 Categorical variables in Regression

E.g. Male and Female

$$x_i = \{ \begin{array}{cc} 1 & female \\ 0 & male \end{array}$$

Resulting model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \{ \begin{array}{cc} \beta_0 + \beta_1 + \epsilon_i & female \\ \beta_0 + \epsilon_i & male \end{array}$$

x is called a dummy variable, and male is the baseline

If more than two levels of categorical variables, then we can use more than 1 dummy variable.

Ethnicity: Caucasian, African American, Asian. If we use African American as the baseline.

$$x_{i1} = \{ \begin{array}{cc} 1 & Asian \\ 0 & NotAsian \end{array} \quad x_{i2} = \{ \begin{array}{cc} 1 & Caucasian \\ 0 & NotCaucasian \end{array}$$

2.4 Interactions between variables

3 Lec 3 Classification

3.1 Logistic Regression

$$Pr(Y=1|X) = p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$

$p(X)$ will have values between 0 and 1.

Take the log odds or logit transformation

$$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

Training data : $(x_i, y_i), i = 1, 2, ..., n$

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x^i) \prod_{i:y_i=0} (1-p(x^i))$$

The logistic regression with more than two classes:

$$Pr(Y=k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\beta_{2k}X_2+...+\beta_{pk}X_p}}{\sum_{l=1}^{K} \beta_{0l} + \beta_{1l}X_1 + \beta_{2l}X_2 + ... + \beta_{pl}X_p}$$

3.2 Discriminant Analysis

Both X and Y are random

$$Pr(Y = k) = \pi_k$$

$$Pr(X = x|Y = k) = f(x; \theta = \theta_k) = f_k(x)$$

π_k is the probability for each class
$f_k(x)$ is a probability density function
θ_k are class specific parameters
Step 1: Learning, estimate π_k and θ_k from training data
Step 2: Predication, calculate $P(Y = k|X = x)$
Linear Discriminant analysis: $f_k(x) = N(\mu_j, \Sigma)$
Quadratic discriminant analysis: $f_k(x) = N(\mu_k, \Sigma = \Sigma_k$
Naive Bayes Classifier: $f_k(x) = \prod_{j=1}^{p} f_k j(x_j)$

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{1}{2}(\frac{x-\mu_k}{\sigma_k})^2}$$

Here μ_k is the mean, and σ_k^2 the variance in class k. We will assume that all the $\sigma_k = \sigma$. Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = Pr(Y = k|X = x)$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{1}{2}(\frac{x-\mu_k}{\sigma_k})^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{1}{2}(\frac{x-\mu_l}{\sigma})^2}}$$

$$p_k(x) = \frac{A}{B} ln(A) = ln\pi_k + ln\frac{1}{\sqrt{2\pi}\sigma} + (-1/2)(\frac{x-\mu_k}{\sigma})^2$$

The discriminant score is $\delta_k(x) = x\frac{\dot{\mu_k}}{\sigma^2} - \frac{\mu_k^2}{2|sigma^2} + log(\pi_k)$
Note that $\sigma_k(x)$ is linear function of x
Given classes 0 and 1, we wan t to estimate $\pi_0, \pi_1, \mu_0, \mu_1, \sigma$
Training dataset: $(x_i, y_i)_{i=1,2,...,n}$

$$\begin{aligned} L(\bar{\pi}, \bar{\mu}, \sigma) &= \prod_{i=1}^{n} p(y_i)p(x_i|y_i) \\ &= \prod_{i=1}^{n} \pi_1^{y_i} \pi_0^{1-y_i} f_1^{y_i}(x_i) f_0(x_i)^{1-y_i} \end{aligned} \tag{1}$$

$$log(L) = log(\pi_1)(\sum y_i) + log(1 - \pi_1) \sum (1 - y_i) + \sum y_i log f_1(x_i) + \sum (1 - y_i) log f_0(x_i)$$

Let $part1 = log(\pi_1)(\sum y_i) + log(1 - \pi_1) \sum (1 - y_i)$, $part2 = \sum y_i log f_1(x_i) + \sum (1 - y_i) log f_0(x_i)$

$$\frac{\partial part1}{\partial \pi_1} = \frac{\sum y_i}{\pi_1} + \frac{\sum (1 - y_i)}{\pi_1 - 1} = 0$$

$$\pi_1 = \frac{\sum y_i}{n}, \pi_0 = 1 - \pi_1$$

Let $part2 = \sum y_i(-log(\sqrt{2\pi}\sigma) - \frac{(x_i-\mu_1)^2}{\sigma^2}) + \sum (1 - y_i)(-log(\sqrt{2\pi}\sigma) - \frac{(x_i-\mu_0)^2}{\sigma^2})$

$$\hat{\mu_1} = \frac{\sum_{y_i=1} x_i}{n_1}, \hat{\mu_0} = \frac{\sum_{y_i=0} x_i}{n_0}$$

$$\hat{\sigma^2} = \frac{n_0}{n_0+n_1}\frac{1}{n_0}\sum_{y_i=0}(x_i-\mu)^2 + \frac{n_1}{n_0+n_1}\frac{1}{n_1}\sum_{y_i=1}(x_i-\mu)^2$$

The class proportion: $\pi_k = \frac{n_k}{n}$, The class center: $\hat{\mu_k} = \frac{1}{n_k}\sum_{i:y_i=k} x_i$
LDA assumes that sigma is the same across classes

$$\begin{aligned}\hat{\sigma^2} &= \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu_k})^2 \\ &= \sum_{k=1}^{K}\frac{n_k-1}{n-K}\hat{\sigma_k}^2\end{aligned} \tag{2}$$

where $\hat{\sigma_k} = \frac{1}{n_k-1}\sum_{i:y_i=k}(x_i-\hat{\mu_k})^2$
Comparison:

- Discriminant functions for logistic regression : $\delta_k(x) = x\beta_{1k} + \beta_{0k}$
- Discriminant function for LDA: $\delta_k(x) = x\frac{\mu_k}{\sigma} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on $P(Y|X)$, known as discriminative learning
- LDA uses the full likelihood based on $P(X,Y)$, known as generative learning

3.3 Quadratic Discriminant Analysis

when p = 1

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k}e^{\frac{1}{2}(\frac{x-\mu_k}{\sigma_k})^2}$$

Plugging this into Bayes formula

$$p_k(x) = \frac{\pi_k\frac{1}{\sqrt{2\pi}\sigma_k}e^{\frac{1}{2}(\frac{x-\mu_k}{\sigma_k})^2}}{\sum_{l=1}^{K}\pi_l\frac{1}{\sqrt{2\pi}\sigma_l}e^{\frac{1}{2}(\frac{x-\mu_l}{\sigma_l})^2}}$$

3.4 LDA and QDA for $p > 2$

LDA:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}e^{\frac{1}{2}((x-\mu_k)^T\Sigma^{-1}(x-\mu_k)}$$

Discriminant function: $\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k$, it is a linear function
QDA:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Discriminant function: $\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) - \frac{1}{2}log|\Sigma_k| + \log\pi_k$, it is not linear

Parameters in LDA: K classes and X has dimension p:

$$\pi_k, k = 1, ..., K : K$$
$$\mu_k, k = 1, ..., K : K * p$$
$$\Sigma, p + (p^2 - p)/2 = p(p+1)/2$$

Suppose that Σ is a diagonal matrix,

$$\Sigma = diag(D_1, ..., D_P)$$

$$\begin{aligned} f(x) &= \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \\ &= \frac{1}{(2\pi)^{p/2}(\prod_{j=1}^p D_j)^{1/2}} e^{-\frac{1}{2}\sum_{j=1}^p (\frac{(x_j-\mu_j)^2}{D_j}} \\ &= \prod_{j=1}^p f_j(x_j) \end{aligned} \tag{3}$$

where $f_j(x_j) = \frac{1}{\sqrt{2\pi D_j}} e^{-\frac{(x_j-\mu_j)^2}{2D_j}}$, across the dimensions $x_1, ...x_p$ are independent with each other

3.5 Naive Bayes Classifier

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

$f_k(x)$ is the density function for X in class k

π_k is the marginal or prior probability for each class k

Naive Bayes classifier: $f_k(x) = \prod_{j=1}^p f_{kj}(x_j)$

with the covariance matrix is diagonal, LDA is a naive Bayes classifier.

Naive Bayes Classifier is useful when p is large, which means that multivariate methods like QDA and LDA break down because of density estimation problem.

4 Lec 4 Model Selection

ProblemChoose a model that minimizes the testing error

4.1 Variance-bias decomposition

$Testingerror = Bias^2 + variance + irreducibleerror$

Bias-variance trade-off:

A. Data generating machinery

(1) Given x, x is not random

(2) $y = f(x) + \epsilon$

$\mathbb{E}(\epsilon) = 0$, $\text{Var}(\epsilon) = \sigma_\epsilon^2$

B. Training data $(x_i, y_i)_{i=1,2,...,n}$

$y_i = f(x_i) + \epsilon_i$

Goal: $g(*) \to f(*)$

C. Testing data

$x_0, y_0 = f(x_0) + \epsilon_0, \hat{g}$: estimation from training data

$\hat{g}(x_0) = \hat{y_0}$

$$\begin{aligned}\mathbb{E}[(y_0 - \hat{y_0})^2] &= \mathbb{E}[(f(x_0) - \hat{y_0})^2 + \sigma_0^2 + 2\sigma_0(f(x_0) - \hat{y_0}] \\ &= \mathbb{E}[(f(x_0) - \hat{y_0})^2] + \mathbb{E}(\sigma_0^2) \\ &= \mathbb{E}[(f(x_0) - \hat{y_0})^2] + \sigma_{\epsilon_0}^2\end{aligned} \tag{4}$$

$\sigma_{\epsilon_0}^2$ is called the irreducible error, for the other part

$$\begin{aligned}\mathbb{E}[(f(x_0) - \hat{y_0})^2] &= \mathbb{E}[(f(x_0) - \mathbb{E}\hat{y_0})^2] + \mathbb{E}[(\hat{(}y_0) - \mathbb{E}[\hat{y_0}])^2] + 2\mathbb{E}[(f(x_0) - \mathbb{E}(\hat{y_0}))(\mathbb{E}[\hat{(}y_0)] - \hat{y_0})] \\ &= \mathbb{E}[(f(x_0) - \mathbb{E}\hat{y_0})^2] + \mathbb{E}[(\hat{(}y_0) - \mathbb{E}[\hat{y_0}])^2]\end{aligned} \tag{5}$$

because $\mathbb{E}[(f(x_0) - \mathbb{E}(\hat{y_0}))(\mathbb{E}[\hat{(}y_0)] - \hat{y_0})] = (f(x_0) - \mathbb{E}(\hat{y_0}))\mathbb{E}[\mathbb{E}[\hat{(}y_0)] - \hat{y_0})] = 0$.

$\mathbb{E}[(f(x_0) - \mathbb{E}\hat{y_0})^2] + \mathbb{E}[(\hat{(}y_0) - \mathbb{E}[\hat{y_0}])^2]$ is called $bias^2$

Example:Variance-bias decomposition for KNN

KNN prediction: $\hat{y_0} = \frac{1}{K}\sum_{i \in N_0} y_i$

$$\begin{aligned}Bias^2 &= \mathbb{E}[(f(x_0) - \mathbb{E}(\frac{1}{K}\sum y_i)]^2 \\ &= \mathbb{E}[(f(x_0) - \frac{1}{K}\sum \mathbb{E}(y_i)]^2 \\ &= \mathbb{E}[(f(x_0) - \frac{1}{K}\sum \mathbb{E}(f(x_i) + \epsilon)]^2 \\ &= \mathbb{E}[(f(x_0) - \frac{1}{K}\sum \mathbb{E}(f(x_i))]^2\end{aligned} \tag{6}$$

$$\begin{aligned}Variance &= \mathbb{E}[(\hat{y_0} - \mathbb{E}(\hat{y_0}))^2] \\ &= \mathbb{E}[(\frac{1}{K}\sum (y_i - \mathbb{E}(y_i))^2] \\ &= \frac{1}{K^2}\mathbb{E}[[\sum (y_i - \mathbb{E}(y_i))]^2] \\ &= \frac{1}{K^2}\mathbb{E}[\sum (y_i - \mathbb{E}(y_i))^2 + cross - product] \\ &= \frac{1}{K^2}\mathbb{E}(\epsilon_i^2) \qquad\qquad = \frac{\sigma_\epsilon^2}{K}\end{aligned} \tag{7}$$

$$cross - product = \mathbb{E}[(y_i - \mathbb{E}(y_i))(y_j - \mathbb{E}(y_j))] \quad = \mathbb{E}[\epsilon_i \epsilon_j] = 0 \tag{8}$$

4.2 Model Selection for Linear Regression

We consider alternatives to least squares.

We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting.

Mallow's C_p

$$C_p = \frac{1}{n}(RSS + 2d\sigma^2)$$

where d is the total number of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement.

AIC and BIC Criterion

The AIC criterion is defined for a large class of models fit by maximum likelihood:

$$AIC = -2\log L + 2 * d$$

where L is the maximized value of the likelihood function for the estimates model

$$BIC = -2\log L + \log n * d$$

Adjusted R^2

For a least squares model with d variables, the adjusted R^2 statistics is calculated as

$$A - R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

where TSS is the total sum of squares.

Unlike C_p, AIC, and BIC, a large value of adjusted R^2 indicates a model with a small test error.

4.3 Shrinkage (Regularization)

Ridge Regression and Lasso Regression

Shrinkage increases the bias part but reduces the variance

Ridge Regression

Recall the least squares fitting procedure estimates $\beta_0, \beta_1, ...\beta_p$ using the value that minimize

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

In contrast, the ridge regression coefficient estimates are the values that minimize

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

where $\lambda \geq 0$ is a tuning parameter, to be determined separately.

E.g.

$$f(\beta_0, \beta_1) = \sum(y_i - \beta_0 - \beta_1 x_i)^2 + \lambda\beta_1^2$$

$$\begin{aligned}
\frac{\partial f}{\partial \beta_0} &= -2\sum(y_i - \beta_0 - \beta_1 x_i) \\
&= -2n\bar{y} + 2n\beta_0 + 2n\beta_1\bar{x} \\
\frac{\partial f}{\partial \beta_1} &= -2x_i\sum(y_i - \beta_0 - \beta_1 x_i) + 2\lambda\beta \\
&= 2\{-\sum x_i y_i + n\bar{x}\beta_0 + \beta_1 x_i^2 + \lambda\beta_1)\} \\
\beta_1 &= \frac{\sum x_i y_i - n\bar{x}y}{(\sum x_i - \bar{x})^2 + \lambda}
\end{aligned} \tag{9}$$

For ridge regression, it is important to standardize the predictors

Lasso Regression

The lasso is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients minimize the quantity:

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

It can be shown that Lasso and Ridge are equivalent to solving the following problems.

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

$$min_\beta \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2, \sum_{j=1}(\beta_j)^2 \leq s$$

5 Lec 5 Resampling methods

5.1 K-fold Cross Validation

For each fold k, the Mean Squared Error: $MSE_k = \sum_{i\in C_k}(y_i - \hat{y}_i)^2/n_k$
Cross-Validation error: $CV_K = \sum_{k=1}^{K}\frac{n_k}{n}MSE_k = \frac{1}{n}\sum_i(y_i - \hat{y}_i)^2$
In practice, we choose $K = 5$ or 10
Cross-Validation for classification problem: $CV_k = \sum_{k=1}^{K}\frac{n_k}{n}Err_k$, where $Err_k = \sum_{i\in C_k} I(y_i \neq \hat{y}_i)/n_k$
The wrong and right way to do cross-validation
Wrong:

- First select a subset of "good" predictors from the data
- Divide the samples into K folds at random
- For each fold $k = 1, 2, ..., K$, use the good predictors to build a learner using all of the samples except in fold k
- Use the learner to predict for samples in fold k

Right:

- Divide the samples into K folds at random
- For each fold $k = 1, 2, ..., K$, select a subset of "good" predictors from the data, use the good predictors to build a learner using all of the samples except those in fold k
- Use the learner to predict for samples in fold k

The wrong one already uses the whole data before cv, and will underestimate test error.

5.2 The Bootstrap

The bootstrap is commonly used to quantify the uncertainty associated with a given estimator, especially for complex models where the standard error cannot be easily derived.

Suppose that we want to choose α such that we can minimize the variance: $\text{Var}(\alpha X + (1-\alpha)Y)$.

One can show that the value that minimizes the variance is given by:

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

Ideally, if we know $f(X,Y)$, let $Z = (X_i, Y_i)_{i=1,2,...100}$, we use $f(X,Y)$ to generate B datasets, $Z_1, Z_2, ...Z_B \to \hat{\alpha_1}, ...\hat{\alpha_B}$

But we only have the dataset and Bootstrap uses the empirical distribution $f^*(X,Y)$ to generate resampled data and use that to estimate SE of α

$$f^*(X,Y) \to z_1^*, z_2^*, ..., z_B^* \to \alpha_1^*, \alpha_2^*, ..., \alpha_B^*$$

and use $\sqrt{\frac{1}{B-1}\sum_{n=1}^{B}(\hat{\alpha_n}^* - bar(\hat{\alpha_n^*}))^2}$ as the SE for $\hat{\alpha}$

Example.

Obs	X	Y		Obs	X	Y		Obs	X	Y
$i=1$	4.3	2.4		$i=3$	5.3	2.8		$i=2$	2.1	1.1
$i=2$	2.1	1.1	$\to z_1^*$	$i=2$	2.1	1.1	z_2^*	$i=3$	5.3	2.8
$i=3$	5.3	2.8		$i=3$	5.3	2.8		$i=1$	4.3	2.4

Other uses of the bootstrap. It also provides approximate confidence intervals for a population parameter. It is called a Bootstrap Percentile.

6 Lec 6 Nonlinear methods

6.1 Polynomial regression

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_2 x_i^3 + ... + \beta_d x_i^d + \epsilon$$

Not really interested in the coefficients, more interested in the fitted function values at any value x_0

Since $f(\hat{(}x_0)$ is a linear function of the $\hat{\beta}_l$, we can get a simple expression for pointwise-variance at any value x_0, $\text{Var}(f(\hat{x_0}))$ at any value

Logistic regression follows naturally.

$$P(y_i > Y|x_i) = \frac{exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_2 x_i^3 + ... + \beta_d x_i^d +}{1 + exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_2 x_i^3 + ... + \beta_d x_i^d +)}$$

However, polynomials have notorious tail behaviour - very bad for extrapolation.

7 Lec 7 Tree-based Methods

7.1 Decision Tree

The basic idea for tree-based methods is to stratify or segment the predictor space into a number of simple regions

Details of the tree building process:

- We first select the predictor X_j and the cut point s such that splitting the predictor space into the regions. $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$

  Selecting the best X_j and s

  $X_1 : s_1 = s_{min}, ...s_d = s_{max}$

$$RSS(X_1, s_i) = \sum_{i:x_i \in R_1} (y_i - \bar{y}_{R1})^2 + \sum_{i:x_i \in R_2} (y_i - \bar{y}_{R2})^2 \tag{10}$$

  Pick smallest $RSS(X_1, s^*)$ from $RSS(X_1, s_1), RSS(X_1, s_2), ...RSS(X_1, s_d)$

  Do the same for $RSS(X_2, s_i)$

- On each of the regions obtained in step 1, we repeat the process in step 1.
- The process continues until a stopping criterion is reached.

Remark: The tree-building process is a top-down, greedy search to minimize the objective function

$$argmin(\sum_{j=1}^{J} \sum_{x_i \in R_j} (y_i - y_{\bar{R}j})^2$$

It is greedy in the sense that:
(1) Each split is best given the past splits
(2) The process does not search through all possible rectangles

Predictions: We predict the response for a given test observation using the **mean** of the training observations in the region to which that test observation belongs.

7.2 Pruning a Tree

For a tree T, we define $|T|$, the number of terminal nodes to be its complexity,

$$T = argmin \sum_{j=1}^{|T|} \sum_{i:x_i \in R_j} (y_i - y_{\bar{R}j})^2 + \alpha|T|, \alpha > 0$$

(1) $\alpha = 0, T = T_0$ (2) $\alpha = \infty, |T| = 1$

We select an optimal value $\hat{\alpha}$ using cross-validation

If we have k folds in the cross validation, and the tests sets as

$$\begin{array}{ccc} \alpha_1 & ... & \alpha_d \\ T'_{\alpha_1} & ... & T'_{\alpha_d} \\ \frac{1}{n_1}\sum_{i \in C_1}(y_i - \hat{y}_i(T^1_{\alpha_1}))^2 & ... & \frac{1}{n_1}\sum_{i \in C_1}(y_i - \hat{y}_i(T^1_{\alpha_1}))^2 \\ ... & ... & ... \\ \frac{1}{n_k}\sum_{i \in C_k}(y_i - \hat{y}_i(T^k_{\alpha_d}))^2 & ... & \frac{1}{n_k}\sum_{i \in C_k}(y_i - \hat{y}_i(T^k_{\alpha_d}))^2 \end{array}$$

weighted sum $\frac{n_k}{n}$ for each column $MSE(\alpha_1), MSE(\alpha_2), ...MSE(\alpha_d)$, and pick $\hat{\alpha}$ with the smallest MSE

7.3 Classification Tree

For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.

In the classification setting, RSS cannot be used as a criterion for making the binary splits. A natural alternative to RSS is the classification error rate:

$$E = 1 - max_k(\hat{p}_{mk})$$

Here $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the kth class.
In each rectangle R_m, Regression loss: $\sum_{i:x_i \in R_m}(y_i - y_{\bar{Rm}})^2$. Classification loss: $N_m(1 - max(\hat{p_{mk}})$
Gini Index: The Gini index is defined by: $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
Cross Entropy: An alternative to the Gini index is cross entropy,given by $D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

8 Lec 8 Support Vector Machines

9 Lec 9 Unsupervised Learning