Computational Statistics Review

Guirong Fu

July 8 2019

Contents

Chapter 1

Regression

1.1 Simple Linear Regression

The simple linear regression is to modeling **linear** dependence of a *dependent/response/output* variable y against one (or more) *independent/feature/predictor*.

It is a **stochastic** linear model.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- stochastic means randomness. Here, x_i is fixed, ϵ_i is random and so is y_i
- randomness may come from
  - measurement error
  - unknown effects

1.1.1 Theory of model

Assumptions

We make some assumptions on (1)ϵ and (2)parameters $\beta_0, \beta_1, \sigma^2$.

Assumptions on ϵ

1. $E(\epsilon) = 0$. Because we try to avoid systematic bias on ϵ.
2. Independence. ϵ_i are independent from each other
3. $Var(\epsilon) = \sigma^2$. The variance should be the same for different i. This means homoscedasticity.
   $\Rightarrow$ Asuumption 2 and 3 can be written as

$$Var(\vec{\epsilon}) = \begin{bmatrix} Var(\epsilon_1) & Cov(\epsilon_1, \epsilon_2) & \cdots & Cov(\epsilon_1, \epsilon_N) \\ Cov(\epsilon_1, \epsilon_2) & Var(\epsilon_2) & \cdots & Cov(\epsilon_2, \epsilon_N) \\ \cdots & \cdots & \cdots & \cdots \\ Cov(\epsilon_1, \epsilon_N) & Cov(\epsilon_2, \epsilon_N) & \cdots & Var(\epsilon_N) \end{bmatrix} = \sigma^2 I_N$$

Assumptions on Parameters $\beta_0, \beta_1, \sigma^2$ are fixed but **unknown**.

Estimation

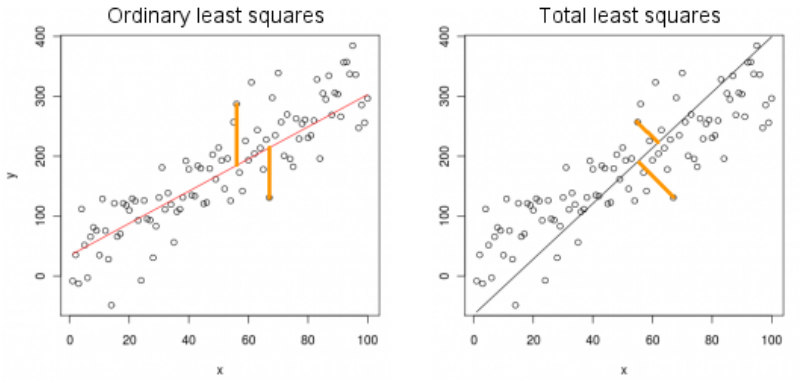
Settings: given a series of data points $\{x_i, y_i\}$, we want to estimate $\beta_0, \beta_1, \sigma^2$ by $\hat{\beta_0}, \hat{\beta_1}, \hat{\sigma^2}$. These estimations should have the following properties:

1. randomness. Given different datasets, we can get different values. The randomness come from the randomness of the datasets. In other words, these estimated parameters are determined by the datasets, and they are not the "true" values of the model.

2. closed to true values. There are many different ways to evaluate **distance**. And different best solutions correspondent to different norms, e.g.

   (a) **RSS** means residual sum square $\Sigma_{i=1}(y_i - \hat{y_i})^2$, and the corresponding method is LSR. Reflected in a figure, is the vertical distance.

   (b) Vector cross product corresponds to the perpendicular distance. The method is PCA.



RSS and LSR

RSS residual sum square is a method to calculate the residuals. It depends on $x_i, y_i, \beta_0, \beta_1$. Given $\{x_i, y_i\}$, for any combinations of β_0, β_1, we can get a RSS. For each dataset, we can get a pair of $\tilde{\beta_0}, \tilde{\beta_1}$. From the distribution of $\tilde{\beta_0}, \tilde{\beta_1}$, we indeed can get some information about σ^2.

LRS: least square regression is a method to find the optimal $\hat{\beta_0}, \hat{\beta_1}$. We would like to take such $\tilde{\beta_0}, \tilde{\beta_1}$ as $\hat{\beta_0}, \hat{\beta_1}$, which minimize the RSS.

$$\hat{\beta_0}, \hat{\beta_1} = \arg\min_{\tilde{\beta_0}, \tilde{\beta_1}} RSS(\{x_i, y_i\})$$

The minimal RSS criterion is quadratic to β_0, β_1. According to the optimization theory, we indeed can find such a pair of $\hat{\beta_0}, \hat{\beta_1}$ that hit the minimal value.

Estimation of β_0, β_1

$$L(\beta_0, \beta_1) = \Sigma_{i=1}^{N}(\beta_0 + \beta_1 x_i - y_i)^2$$

$$\frac{\partial L}{\partial \beta_0} = 2\Sigma_{i=1}^{N}(\beta_0 + \beta_1 x_i - y_i) = 0 \Rightarrow N\beta_0 = \Sigma_{i=1} y_i - \beta_1 \Sigma_{i=1} x_i$$

$$\frac{\partial L}{\partial \beta_1} = 2\Sigma_{i=1}^{N}(\beta_0 + \beta_1 x_i - y_i)x_i = 0 \Rightarrow \beta_1 \Sigma_{i=1} x_i^2 = \Sigma_{i=1} x_i y_i - \beta_0 \Sigma_{i=1} x_i$$

$$\begin{bmatrix} N & \Sigma_{i=1} x_i \\ \Sigma_{i=1} x_i & \Sigma_{i=1} x_i^2 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \Sigma_{i=1} y_i \\ \Sigma_{i=1} x_i y_i \end{bmatrix} \Rightarrow \begin{bmatrix} \hat{\beta_0} \\ \hat{\beta_1} \end{bmatrix} = \frac{1}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \begin{bmatrix} \Sigma_{i=1} x_i^2 \Sigma_{i=1} y_i - \Sigma_{i=1} x_i \Sigma_{i=1} x_i y_i \\ N\Sigma_{i=1} x_i y_i - \Sigma_{i=1} x_i \Sigma_{i=1} y_i \end{bmatrix}.$$

$$\begin{aligned}
E(\hat{\beta_0}) &= \frac{E[\Sigma_{i=1} x_i^2 \Sigma_{i=1} y_i - \Sigma_{i=1} x_i \Sigma_{i=1} x_i y_i]}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{\Sigma_{i=1} x_i^2 E[\Sigma_{i=1} y_i] - \Sigma_{i=1} x_i \Sigma_{i=1} E[x_i y_i]}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{\Sigma_{i=1} x_i^2 E[\Sigma_{i=1} x_i \beta_1 + \beta_0 + \epsilon_i] - \Sigma_{i=1} x_i \Sigma_{i=1} E[x_i(x_i \beta_1 + \beta_0 + \epsilon_i)]}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{\Sigma_{i=1} x_i^2 (\Sigma_{i=1} x_i \beta_1 + N\beta_0) - \Sigma_{i=1} x_i \Sigma_{i=1} (x_i^2 \beta_1 + x_i \beta_0)}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{\beta_1 \Sigma_{i=1} x_i \Sigma_{i=1} x_i^2 + N\beta_0 \Sigma_{i=1} x_i^2 - \beta_1 \Sigma_{i=1} x_i \Sigma_{i=1} x_i^2 \beta_1 - \beta_0 (\Sigma_{i=1} x_i)^2}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \beta_0
\end{aligned}$$

$$\begin{aligned}
Var(\hat{\beta_0}) &= \frac{1}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} Var(\Sigma_{i=1} x_i^2 \Sigma_{i=1} \epsilon_i - \Sigma_{i=1} x_i \Sigma_{i=1} x_i \epsilon_i) \\
&= \frac{\Sigma_{i=1} Var(\epsilon_i (\Sigma_{j=1} x_j^2 - x_i \Sigma_{j=1} x_j))}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} \\
&= \frac{\Sigma_{i=1} (\Sigma_{j=1} x_j^2 - x_i \Sigma_{j=1} x_j)^2}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} \sigma^2
\end{aligned}$$

$$\begin{aligned}
E[\hat{\beta_1}] &= \frac{NE[\Sigma_{i=1} x_i y_i] - \Sigma_{i=1} x_i E[\Sigma_{i=1} y_i]}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{NE[\Sigma_{i=1} x_i (x_i \beta_1 + \beta_0 + \epsilon_i)] - \Sigma_{i=1} x_i E[\Sigma_{i=1} (x_i \beta_1 + \beta_0 + \epsilon_i)]}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \frac{N\Sigma_{i=1} x_i^2 \beta_1 + \beta_0 N\Sigma_{i=1} x_i - \beta_1 (\Sigma_{i=1} x_i)^2 - \beta_0 N\Sigma_{i=1} x_i}{N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2} \\
&= \beta_1
\end{aligned}$$

$$\begin{aligned}
Var(\hat{\beta_1}) &= \frac{1}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} Var(N\Sigma_{i=1} x_i \epsilon_i - \Sigma_{i=1} x_i \Sigma_{i=1} \epsilon_i) \\
&= \frac{1}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} Var(\Sigma_{i=1} \epsilon_i (N x_i - \Sigma_{j=1} x_j)) \\
&= \frac{\Sigma_{i=1} (N x_i - \Sigma_{j=1} x_j)^2}{(N\Sigma_{i=1} x_i^2 - (\Sigma_{i=1} x_i)^2)^2} \sigma^2
\end{aligned}$$

Estimation of σ^2

$$\begin{aligned}
E(\hat{\epsilon^2}) &= E[(Y-\bar{Y})^2] \\
&= E[Y^2+\hat{Y}^2-2Y\hat{Y}] \\
&= E[Y^2]+E[\hat{Y}^2]-2EYE\hat{Y} \\
&= E[\hat{Y}^2]-E[Y^2] \\
&= E[(X\hat{\beta_1}+\hat{\beta_0})^2]-E[(X\beta_1+\beta_0+\epsilon)^2] \\
&= E[(X\hat{\beta_1})^2]+E\hat{\beta_0}^2+2E[X\hat{\beta_1}\hat{\beta_0}]-E[(X\beta_1+\beta_0)^2]-E\epsilon^2-2E[X\beta_1+\beta_0]E\epsilon \\
&= EX^2E\hat{\beta_1}^2+E\hat{\beta_0}^2+2EXE\hat{\beta_1}E\hat{\beta_0}-X^2\beta_1^2-\beta_0^2-2EXE\beta_1E\beta_0-\sigma^2-0 \\
&= EX^2Var(\hat{\beta_1})+Var(\hat{\beta_0})-\sigma^2 \\
&= \frac{\Sigma_{i=1}x_i^2}{N}Var(\hat{\beta_1})+Var(\hat{\beta_0})-\sigma^2 \\
&= (\frac{2\Sigma_{i=1}x_i^2}{N\Sigma_{i=1}x_i^2-(\Sigma_{i=1}x_i)^2}-1)\sigma^2
\end{aligned}$$

1.2 Multiple Linear Regression

1.2.1 Theory of model

We have p independent variables.

$$y_i=\beta_1x_{i1}+\beta_2x_{i2}+\cdots+\beta_px_{ip}+\epsilon_i=x_i^T\beta+\epsilon_i \tag{1.1}$$

Here, $x_i\in\mathbb{R}^{P\times1},\beta\in\mathbb{R}^{P\times1},\epsilon\in\mathbb{R}$. Written as a matrix format:

$$Y=X\beta+\epsilon \tag{1.2}$$

Here, $X\in\mathbb{R}^{N\times P},\beta\in\mathbb{R}^{P\times1},\epsilon\in\mathbb{R}^{N\times1}$, x_i is the i-th row of **design matrix** X.

Assumptions

The same to 1.1.1. Also, we assume $N>P$ to avoid the low-rank problem.

Objective

$$\hat{\beta}=\arg\min_b\Sigma_{i=1}^N(y_i-x_i^Tb)^2\Rightarrow\hat{\beta}=\arg\min_{b\in\mathbb{R}^P}(Y-Xb)^T(Y-Xb) \tag{1.3}$$

Estimation

This is a nice convex quadratic function. We can solve by setting the derivative as 0.

Estimation of β

$$2X^T(Y-Xb)=0\Rightarrow\hat{\beta}=(X^TX)^{-1}(X^TY)$$

Estimation of σ^2

$$\begin{aligned}\hat{\sigma^2} &= \frac{1}{N-P}(Y - X\hat{\beta})^T(Y - X\hat{\beta}) \\ &= \frac{1}{N-P}\Sigma_{i=1}^{N}(y_i - x_i^T\hat{\beta})^2\end{aligned}$$

Note that we should use $N - P$ instead of N, so that $E(\hat{\sigma^2}) = \sigma^2$ is unbiased.

Properties

The following are true.

- $E(\hat{\beta}) = \beta$, $\hat{\beta}$ is unbiased.
- $Var(\hat{\beta}) = \sigma^2(X^TX)^{-1}$, we can get information of σ^2 from $\hat{\beta}$ or vise verse.
- if ϵ follows a Gaussian distribution.
  - $\hat{\beta}$ follows a multi-variate Gaussian distribution $\mathcal{N}(\beta, \sigma^2(X^TX)^{-1})$,
  - $\hat{Y}$ follows a multi-variate Gaussian distribution $\mathcal{N}(X\beta, \sigma^2X(X^TX)^{-1}X^T)$
  - $\hat{e} = Y - \hat{Y}$ follows a multi-variate Gaussian distribution $\mathcal{N}(0, \sigma^2[I - X(X^TX)^{-1}X^T)]$
- $\hat{\sigma^2} = \frac{1}{N-P}\Sigma_{i=1}e_i^2, E(\hat{\sigma^2}) = \sigma^2$

Proof

$$\begin{aligned}E(\hat{\beta}) &= E((X^TX)^{-1}X^TY) = (X^TX)^{-1}X^TE(Y) \\ &= (X^TX)^{-1}X^TE(X\beta + \epsilon) \\ &= (X^TX)^{-1}X^TXE(\beta) + 0 \\ &= \beta\end{aligned}$$

Using the assumption $E(\epsilon) = 0$

$$\begin{aligned}Var(\hat{\beta}) &= Var((X^TX)^{-1}X^TY) \\ &= E[(X^TX)^{-1}X^TY)(X^TX)^{-1}X^TY)^T] \\ &= (X^TX)^{-1}X^TVar(Y)X(X^TX)^{-1} \\ &= (X^TX)^{-1}X^TVar(X\beta + \epsilon)X(X^TX)^{-1} \\ &= (X^TX)^{-1}X^TVar(\epsilon)X(X^TX)^{-1} \\ &= (X^TX)^{-1}X^T\sigma^2I_NX(X^TX)^{-1} \\ &= \sigma^2(X^TX)^{-1}\end{aligned}$$

Using the assumption: independence.

$$\begin{aligned}E(\hat{e}) &= E(X\beta + \epsilon - X\hat{\beta}) \\ &= X\beta - X\beta + E(\epsilon) \\ &= 0\end{aligned}$$

$$\begin{aligned}
Var(\hat{e}) &= Var(Y - \hat{Y}) \\
&= Var(Y - X\hat{\beta}) \\
&= Var(Y - X(X^TX)^{-1}X^TY), P := X(X^TX)^{-1}X^T \\
&= Var(Y - PY) \\
&= (I - P)Var(Y)(I - P)^T \\
&= (I - P)\sigma^2 I_N (I - P) \\
&= \sigma^2(I + P^2 - 2P) \\
&= \sigma^2(I - P)
\end{aligned}$$

Here, $P^2 = X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T = P$.

$$\begin{aligned}
E(\Sigma_{i=1}e_i^2) &= E[trace(\sigma^2(I - P))] \\
&= \sigma^2 trace(I - P) \\
&= \sigma^2(N - p)
\end{aligned}$$

1.2.2 Confidence/Prediction Interval

Confidence Interval

p-value only gives us the extreme level of our estimation against the hypothesis. Confidence interval will give us an interval and how possible the hypothesis is in this interval.

It is an interval I_j, where

$$P(\beta_j \in I_j) = (1 - \alpha) * 100\%$$

$$\Leftrightarrow P(\cdots < \frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} < \cdots) = 1 - \alpha$$

Here I_j is based on our data, and β_j is fixed, given by the hypothesis. If we had bad data samples, it may happen $\beta_j \notin I_j$, even with very high α.

$$I = \hat{\beta}_j \pm \hat{se}(\hat{\beta}_j)t_{1-\frac{\alpha}{2},n-p}$$

We can say confidence interval to the estimated parameters β_j, the independent variable y.

Prediction Interval

Prediction interval only works for the independent, to be predicted, variable y, not for the to be estimated parameters β_j. The difference with CI is that the y itself is a random variable, bringing some random error ϵ_0, but β_j is fixed, not random. In another way, in CI, we use $E(y_0)$, but in PI we use y_0.

We want to give a prediction interval for a new independent variable y_0, when we know its dependent variable vector $\vec{x_0}$.

- CI:

$$\frac{x_0^T\hat{\beta} - E(y_0)}{\sigma\sqrt{x_0^T(x^Tx)^{-1}x_0}} \sim t_{n-p}$$

- PI:

$$\frac{x_0^T\hat{\beta} - y_0}{\sigma\sqrt{x_0^T(x^Tx)^{-1}x_0 + 1}} \sim t_{n-p}$$

1.3 KNN Regression

1.3.1 Idea

For a data point x, we estimate $f(x)$ by its k nearest neighbors.

$$f(x) = \frac{1}{k}\Sigma_{i\in N(x)}y_i \tag{1.4}$$

Here, $N(x)$ denotes the set of k nearest neighbors of x.

1.3.2 Bias-Variance Analysis

k	Bias	Variance	smoothness
small	small	large	low
large	large	small	large

Table 1.1: K and its relationship with bias and variance

It is easy to understand that when k is small, to predict a new data point, we only refer to few neighbors, so the predicted value depends largely on its neighbors values. If the training data are different, the prediction will change a lot. That means the variance of models is large. But the bias is small, since we refer to neighbors very closed to it. On the contrary, when k is large, we refer to neighbors which may even be not closed to the point, that lead to a large bias. But the stability of models increase with referred neighbors increasing, not so easily influenced by the randomness.

1.3.3 Compare to linear regression

- the real function is linear: KNN-regression will lose to linear regression.
- the real is moderately non-linear: more or less
- the real is strongly non-linear: KNN may perform better.

Generally, the bigger k means a better performance(less MSE).

1.3.4 Curse of Dimension

More variables may cause big damage to KNN-regression due to the norm definition. The curse of dimension may be avoided by a suitable neighbor definition (norm definition).

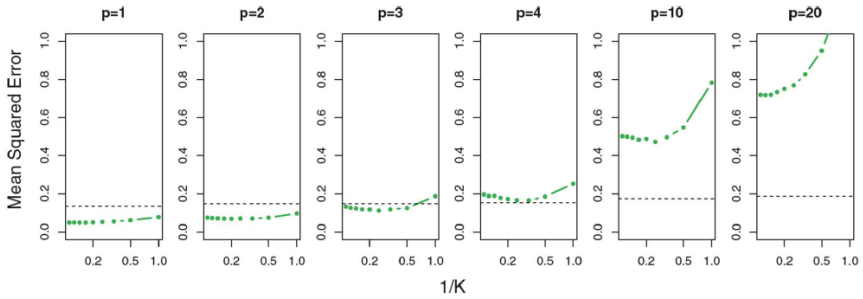


FIGURE 3.20. *Test MSE for linear regression (black dashed lines) and KNN (green curves) as the number of variables p increases. The true function is non–linear in the first variable, as in the lower panel in Figure 3.19, and does not depend on the additional variables. The performance of linear regression deteriorates slowly in the presence of these additional noise variables, whereas KNN's performance degrades much more quickly as p increases.*

1.4 Measures of Goodness-of-fit

1.4.1 residual standard error

$\hat{\sigma}$

- error interval: 2/3 possible values are covered within one $\hat{\sigma}$ errors, 95% are covered within two $\hat{\sigma}$ errors.
- this is a measurement of goodness-of-fit.
  - if $\hat{\sigma}$ is small, that means much of variables have been explained by the model, the model is nice.
  - big means not good.
- using $\frac{\hat{\sigma}}{y}$: the relative value of $\hat{\sigma}$ to evaluate the goodness-of-fit.

1.4.2 R-square

$$R^2 = \frac{TSS - RSS}{TSS} \tag{1.5}$$

It means how large the estimated y can be captured by the model.

- TSS: total sum of square error: $\Sigma_{i=1}(y_i - \bar{y})^2$
- RSS: the sum of square error of residual: $\Sigma_{i=1}(y_i - \hat{y_i})^2$

The bigger R^2 means the larger part of the variance of y can be explained by our model, the model is better.

Relation to LSR

$\max R^2 \Leftrightarrow \min RSS \Leftrightarrow$ LSR

Adjusted R-square

Only comparing different models by R-square is not fair. Because models with more variables (furtherly more parameters) will always win. Because TSS will not change but the RSS will never go down with more parameters. Here, we include adjusted R-square to **models with different amount of parameters**

$$R^2_{\text{adjusted}} = 1 - \frac{RSS/(n-p)}{TSS/(n-1)} \tag{1.6}$$

In principle, R^2_{adjusted} should be positive, but it cannot when $\frac{RSS}{n-p} > \frac{TSS}{n-1}$

1.4.3 Diagnostic Plots

- Residual VS Fitted plot. To test if $E(\epsilon) = 0$. [1]
- Normal Q-Q: to test if the errors follow a normal distribution. The x-axis is the theoretical Gaussian distribution, the y-axis is the empirical distribution got from the given data.
  - If it follows the Gaussian distribution, the line should be straight.
  - Otherwise, it tells us something about which tail is heavy, short.

1.4.4 Cook's distance and leverage

It works for each data point, to measure the effect of deleting a given observation. If the cook's distance is high, it means deleting this data point will change the model a lot, or vice verse. https://en.wikipedia.org/wiki/Cook%27s_distance#cite_note-mathworks-5

A leverage point is defined as an observation that has a value of x that is far away from the mean of x

1.5 Testing index

1.5.1 P-value

Meaning: the probability of observing a value of the test statistics that is as extreme as or more extreme than the observed ones, if we considered the H_0 is true.

Usually it works with a significance level α given in advance.

It can happen that the model's p-value is significant but each parameter's p-value is not. It may be for that variables are highly-correlated.

1.5.2 Power

Power is the probability of rejecting H_0 when H_α is true.

1.5.3 F-statistic

Compare a model with more parameters to the "mother" model to see if these parameters are important.

[1] It is also called Tukey-Anscombe plot.

1.6 Category Variables

Sometimes, there are many categories, and for each category, we can fit a different lines. It is time to import **dummy variables**.

1.6.1 Dummy Variables

On Interception

Suppose there is one category variable and it affects the interception, the original functions should be:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon, i \in \text{category A}$$
$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon + \gamma, i \notin \text{category A}$$

We can merge these two by a **dummy variable** as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon + d_i \gamma$$

If the category has more than 2 levels, say it is m, we can set

- $m - 1$ dummy variables. The first type: all dummy variables are 0. The i-th type: the (i-1)-th dummy variables are 1, others are 0 (i¿1).
- $\lceil \log_2(m) \rceil$. Using the 2-system coding method. e.g. 0000, 0001, 0010, 0011, $\cdots$

On Direction

When a dummy variable affects the direction, it is also called **interaction**[2]. Because the corresponding dummy variable will multiply with the other variables. To achieve this, we include $\gamma d_i x_{ik}$

For instance,

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon, i \in \text{category A}$$
$$y_i = (\beta_0 + \gamma) + (\beta_1 + \alpha_1) x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon, i \notin \text{category A}$$

Can be written as

$$y_i = \beta_0 + \gamma d_i + (\beta_1 + \alpha_1 d_i) x_{i1} + (\beta_2 + \alpha_2 d_i) x_{i2} + \cdots + \epsilon$$

1.6.2 More about interaction

There can also be interaction between two quantitative variables

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1} x_{i2} + \cdots + \epsilon_i$$

There can also be interaction between two category variables

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 d_{i1} d_{i2} + \cdots + \epsilon_i$$

1.7 Ideas

If we only do an observational study, we cannot generate casual conclusions from that.

[2] Interaction VS correlation: we cannot say there is interaction between x_1 and x_2 without saying what y is. Correlation means if i know x_1, I also have some information on x_2

Chapter 2

Model Selection

2.1 Bias-Variance Trade-off

- What is Bias? The mathematical formula?
- What is Variance? The mathematical formula?
- How to balance these two?
- Their performance on different linearity?

2.1.1 Motivation

We want to choose the model which performs best on the **test dataset**, but not on the train dataset. So the expected mean square error on the test data should be:

$$E[(y - f\hat{(}x))^2] \tag{2.1}$$

In this formula,

- y is a random variable, which is a random data point in the possible space. $y = f(x) + \epsilon$. Here, $f(x)$ is a fixed value, but ϵ is random.
- $f\hat{(}\cdot)$ is also random, which is relied on the randomness of the training dataset.

Decompose this formula, we get

$$\begin{aligned}
E[(y - f\hat{(}x))^2] &= E[(f(x) + \epsilon - f\hat{(}x))^2] \\
&= E[(f(x) - f\hat{(}x))^2 + \epsilon^2 + 2\epsilon(f(x) - f\hat{(}x))] \\
&= E[(f(x) - E[f\hat{(}x)] + E[f\hat{(}x)] - f\hat{(}x))^2] + E[\epsilon^2] + 2 \cdot 0 \\
&= E[(f(x) - Ef\hat{(}x))^2] + E[(E[f\hat{(}x)] - f\hat{(}x))^2] + 2E[f(x) - Ef\hat{(}x)]E[E[f\hat{(}x)] - f\hat{(}x)] + \sigma^2 \\
&= (f(x) - Ef\hat{(}x))^2 + E[(E[f\hat{(}x)] - f\hat{(}x))^2] + 2(f(x) - Ef\hat{(}x)) \cdot 0 + \sigma^2 \\
&= (f(x) - Ef\hat{(}x))^2 + E[(E[f\hat{(}x)] - f\hat{(}x))^2] + \sigma^2
\end{aligned}$$

We can view it from the following three parts:

- Bias: it reflects something essence of our model choice, e.g. whether it is good to choose a linear model. $(f(x) - Ef(\hat{x}))^2$. $Ef(\hat{x})$ reflects the expectation of the prediction of x over all possible training datasets under the specific model formulation.
- Variance: it reflects how the model trained on one specific training dataset performs differently to the expectation of models throughout all the training datasets.
- Data's randomness: σ^2

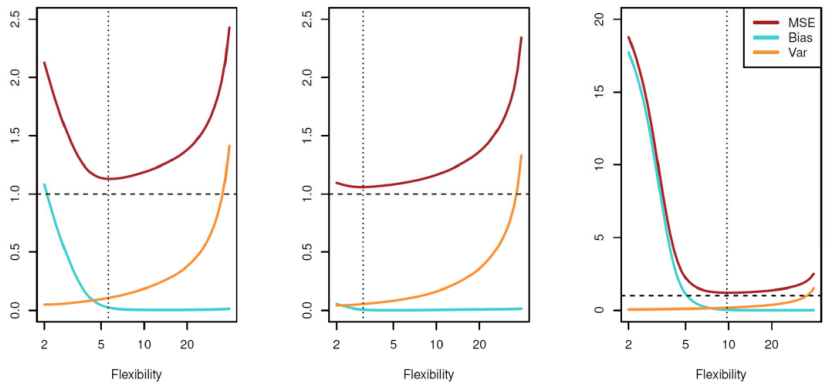


Figure 2.1: Left: moderately non-linear, Middle: strongly linear. Right: strongly non-linear.

From the Fig2.1.1, we can see with the flexibility increasing, the bias of the expectation of models, which may be trained on different training set, decreases while the variance increases. We want to find the appropriate model specification, which both make each trained model perform similarly, (in other words, has low variance) and all predict closely to the real value (reflected by the low bias).

2.2 Cross Validation

2.2.1 Motivation

The objective is to find the optimal parameters that give the smallest expected **test** squared error.

$$E[(Y_{new} - \hat{f}_\alpha(X_{new}))^2] = E_{trainingdata}[E_{new}[(Y_{new} - \hat{f}(X_{new}))^2 | trainingdata]]$$

Three randomness are involved in this goal

- X_{new} the new tested data point is random
- Y_{new}, there is a random error in Y_{new}

- training data: different training dataset will give different $\hat{f}$

To solve the first two randomness, we can build a large test dataset and estimate on that.

To solve the third randomness, we need many training dataset. Cross validation is one way to generate multiple training datasets from one given set.

2.2.2 Approach

Split the given set into training set and validation set. We can use different splitting strategies to get more set pairs.

Remark:

- when the training data is fewer, estimation will not be as good.
- this solution cannot solve the original dataset's bias. That is, if the given training dataset is not good, we get more not good set pairs.
- The variances among different pairs may be large. We can solve this by averaging.

Question on the final model of CV: which is the final model if using CV?

The final model is trained on the whole dataset, instead of the optimal one on different set pairs. We use CV to evaluate how such a setting of model parameters could be. For instance, in KNN-model, we need to decide to set k as 5, or k as 10, or k as 15. Then train models based on the k. CV is to tell us which k to select, instead of the final model. If CV chooses k as 10, we train a model on the whole dataset and get the final model.

2.2.3 Submodels

2.2.4 Validation Once

Split the dataset into training and validation dataset only once. Train on the training dataset and validate on the validation dataset.

k-fold CV

1. split the dataset as k folds, each fold with the same size.
2. iterate for k times. For the i-th time, select the i-th fold as the validation set and the left as training dataest
3. average the results.

LOOCV

Similar to k-fold, setting k as the size of the whole dataset.

In another words, we train $\hat{f}^{(i)}$ on the dataset $\{x_1, \cdots, x_{i-1}, x_{i+1}, \cdots x_n\}$ and get $SE_i = (y_i - \hat{f}^{(i)}(x_i))^2$ and the final expected squared error is $\frac{1}{n}\Sigma_i SE_i$

Efficient computation of LOOCV Denote $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$, $\mathbf{H}_{(-i)} = \mathbf{X}_{(-i)}[\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)}]^{-1}\mathbf{X}_{(-i)}^\top$, we have $\hat{y}_i = (\mathbf{HY})_i$ and $\hat{y}_i^{(-i)} = (\mathbf{H}_{(-i)}\mathbf{Y})_i$.

Meanwhile, we define $\hat{\beta} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, $\hat{\beta}_{(-i)} = (\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)})^{-1}\mathbf{X}_{(-i)}^\top\mathbf{Y}_{(-i)}$. This gives us (the proof is ignored)

$$\hat{y}_i = (\mathbf{HY})_i = \mathbf{x}_i^\top\hat{\beta} \tag{2.2}$$

$$\hat{y}_i^{(-i)} = (\mathbf{H}_{(-i)}\mathbf{Y})_i = \mathbf{x}_i^\top\hat{\beta}_{(-i)} \tag{2.3}$$

The parameters and their dimensions are as below:

- $\mathbf{X} \in \mathbb{R}^{N\times P}, \mathbf{X}_{(-i)} \in \mathbb{R}^{(N-1)\times P}$,
- $\mathbf{x}_i \in \mathbb{R}^P, \mathbf{x}_i^\top$ is the i-th row of $\mathbf{X}$,
- $\mathbf{Y} \in \mathbb{R}^N, \mathbf{Y}_{(-i)} \in \mathbb{R}^{N-1}$
- $\mathbf{H} \in \mathbb{R}^{N\times N}, \mathbf{H}_{(-i)} \in \mathbb{R}^{(N-1)\times(N-1)}$
- $\hat{\beta}, \hat{\beta}_{(-i)} \in \mathbb{R}^P$

The objective is to prove that

$$\hat{y}_i^{(-i)} = \mathbf{x}_i^\top\hat{\beta}_{(-i)} = (\mathbf{H}_{(-i)}\mathbf{Y})_i = \frac{(\mathbf{HY})_i - \mathbf{H}_{ii}y_i}{1-\mathbf{H}_{ii}} \tag{2.4}$$

The idea is to rewrite $\mathbf{H}_{(-i)}$ or $(\hat{\beta}_{(-i)})$ by other parameters.

$$\begin{aligned}\hat{y}_i^{(-i)} &= \mathbf{x}_i^\top\hat{\beta}_{(-i)} \\ &= \mathbf{x}_i^\top(\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)})^{-1}\mathbf{X}_{(-i)}^\top\mathbf{Y}_{(-i)}\end{aligned}$$

We gonna prove that

$$\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i = \mathbf{H}_{ii} \tag{2.5}$$

$$\mathbf{X}_{(-i)}^\top\mathbf{Y}_{(-i)} = \mathbf{X}^\top\mathbf{Y} - y_i\mathbf{x}_i \tag{2.6}$$

$$(\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)})^{-1} = [\mathbf{X}^\top\mathbf{X}]^{-1} + \frac{[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}}{1-\mathbf{H}_{ii}} \tag{2.7}$$

From those, we can get

$$\begin{aligned}\hat{y}_i^{(-i)} &= \mathbf{x}_i^\top(\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)})^{-1}\mathbf{X}_{(-i)}^\top\mathbf{Y}_{(-i)} \\ &= \mathbf{x}_i^\top\{[\mathbf{X}^\top\mathbf{X}]^{-1} + \frac{[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}}{1-\mathbf{H}_{ii}}\}[\mathbf{X}^\top\mathbf{Y} - y_i\mathbf{x}_i] \\ &= \mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{X}^\top\mathbf{Y} + \frac{\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}}{1-\mathbf{H}_{ii}}\mathbf{X}^\top\mathbf{Y} \\ &\quad - y_i\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i - \frac{y_i}{1-\mathbf{H}_{ii}}\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i \\ &= (\mathbf{HY})_i + \frac{\mathbf{H}_{ii}}{1-\mathbf{H}_{ii}}(\mathbf{HY})_i - y_i\mathbf{H}_{ii} - \frac{y_i}{1-\mathbf{H}_{ii}}\mathbf{H}_{ii}^2 \\ &= \frac{(\mathbf{HY})_i - \mathbf{H}_{ii}y_i}{1-\mathbf{H}_{ii}}\end{aligned}$$

The proof of Equation 2.5:

$$\begin{aligned}
\mathbf{H}_{ii} &= [\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top]_{ii} \\
&= \sum_{j=1}^{N}[\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}]_{ij}\mathbf{X}^\top_{ji} \\
&= \sum_{j=1}^{N}[\sum_{k=1}^{N}\mathbf{X}_{ik}(\mathbf{X}^\top\mathbf{X})^{-1}_{kj}]\mathbf{X}_{ij} \\
&= \sum_{j=1}^{N}[\sum_{k=1}^{N}(\mathbf{x}_i)_k(\mathbf{X}^\top\mathbf{X})^{-1}_{kj}](\mathbf{x}_i)_j \\
&= \mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i
\end{aligned}$$

The proof of Equation 2.6:

$$\begin{aligned}
[\mathbf{X}^\top_{(-i)}\mathbf{Y}_{(-i)}]_k &= \sum_{j}^{N-1}(\mathbf{X}^\top_{(-i)})_{kj}(\mathbf{Y}_{(-i)})_j \\
&= \sum_{j=1}^{N}(\mathbf{X}^\top)_{kj}(\mathbf{Y})_j - (\mathbf{X}^\top)_{ki}y_i \\
&= [\mathbf{X}^\top\mathbf{Y}]_k - y_i(\mathbf{x}_i)_k
\end{aligned}$$

The proof of Equation 2.7:

First, the below equation is true, which proof is similar to that of 2.6

$$(\mathbf{X}^\top_{(-i)}\mathbf{X}_{(-i)})^{-1} = [\mathbf{X}^\top\mathbf{X}]^{-1} - \mathbf{x}_i\mathbf{x}_i^\top \tag{2.8}$$

According to Sherman–Morrison formula Wiki

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

we can rewrite the Equation 2.8 as

$$(\mathbf{X}^\top_{(-i)}\mathbf{X}_{(-i)})^{-1} = [\mathbf{X}^\top\mathbf{X}]^{-1} + \frac{[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i\mathbf{x_i}^\top[\mathbf{X}^\top\mathbf{X}]^{-1}}{1-\mathbf{x}_i^\top[\mathbf{X}^\top\mathbf{X}]^{-1}\mathbf{x}_i = 1-\mathbf{H}_{ii}}$$

Double CV

The outer loop is for model assessment and build the final model.

The inner loop is for model selection, selecting the parameters.

The relationship between this two loop is part of the data set of the outer loop is used as the whole set for the inner loop, and the left part is used as the validation data for the outer loop.

2.2.5 Theoretical Comparison

Randomness Table 2.2.5 shows the comparison. There is no randomness in LOOCV. each time you do this you will get the same result.

Models	Random?	Interpretation
LOOCV	No	Repeat the experiment you will get the same result
K-fold	Yes	Depends on the randomness on splitting the set
Valid	Yes	A big randomness

Computation Cost

- LOOCV: high
- k-fold: lower
- validation: much lower.

Estimated test MSE We use $\hat{\theta}$ to denote the estimated test MSE.

- LOOCV: $\hat{\theta}_{LOOCV} = \frac{1}{n}\Sigma_i(y_i - \hat{f}^{(i)}(x_i))^2$
- k-fold: $\hat{\theta}_{k-fold} = \frac{1}{K}\Sigma_{j=1}^{K}\frac{1}{|I_j|}\Sigma_{i\in I_j}(y_i - \hat{f}^{(j)}(x_i))^2$
- valid: $\hat{\theta}_{Valid} = \frac{1}{|Valid|}\Sigma_{i\in Valid}(y_i - \hat{f}(x_i))^2$

Bias To compare the bias, we want to see the expectation of $\hat{\theta}$, i.e. $E[\hat{\theta}]$ estimated by different methods, not the $\hat{\theta}$ itself. And compare the $E[\hat{\theta}]$ with the true θ.

- LOOCV: Since there is no randomness in LOOCV, $E[\hat{\theta}_{LOOCV}] = \hat{\theta}_{LOOCV}$. We can also know the variance $Var[\hat{\theta}_{LOOCV}]$is 0. The $\hat{\theta}_{LOOCV}$ should be closed to θ if our training set is good enough. Because in LOOCV, we use nearly all the data points to train.
- k-fold: similarly, since each time we use $\frac{k-1}{k}$ data points to train the model, the bias should be a little higher than LOOCV.
- Valid: the training data is even less than k-fold, which means it has the highest bias.

The following relationship is based on the assumption that we can get the real θ from the given dataset.

$$E[\hat{\theta}_{val}] > E[\hat{\theta}_{K-fold}] > E[\hat{\theta}_{LOOCV}] \approx \theta$$

Variance There is no clear theory for comparing the variance among CV methods. What teacher taught is as follows:

It is difficult to formally compare the variance of these estimators. In practice for K-fold CV, $MSE_1, ..., MSE_K$ are highly correlated due to the overlap in the training datasets, but still to get some idea of the variance some might use $\frac{\hat{Var}(MSE_i)}{K}$ as an estimator for $Var(\hat{\theta}_{K-fold})$.

Remark Be careful in this case because the model selection step is now a new step that has to be included when you would like to assess your overall performance. So, if you use cross-validation for model selection, when you assess the model with cross-validation you should do the model selection step within each cross-validation step. See the 1-NN example in the Rcode4.r. There, in the first approach, we do pre selection before sending data to the model but and tested on that. So the model we evaluate has the different input from the original one. In the second approach, we do pre selection within each cross validation step. Due to this, the model we evaluate still has the same input as the original one.

Is it reasonable to do millions of times of 10-CV to estimate the confidence interval of accuracy?

No, it isn't. In 10-CV, the randomness lies on the choice of 10-folds, not on the data itself. Repeating millions of times will lead to many repeated groups, which is a waste, invalid. Imagine the LOOCV, we will get the same number each time. So, after millions of times, we may get a data point, instead of a distribution.

Chapter 3

Bootstrap

3.1 Idea

Numerous simulations are very useful to understand the estimation of an estimator. **The understanding of the data distribution** is crucial. But the problem is that in practice, we lack the knowledge of the data distribution, i.e. the distribution which generates the data. Bootstrap is to solve this problem.

The problem setting is,

- data: $Z_1, Z_2, \cdots, Z_n$ are i.i.d from P. P is a fixed but unknown distribution
- Example: $Z_i = (x_i, y_i), x_i \in \mathbb{R}^b, y \in \mathbb{R}$.We want to estimate $E[y|x]$. (A regression or classification problem in essence).
- Goal: find the probability distribution of $\hat{\Theta} = q(z_1, \cdots, z_n)$. Θ is parameter which can be got from $(z_1, \cdots, z_n)$,e.g. the average of z_i or the medium of z_i

Bootstrap approach:

1. We build an empirical distribution $\hat{P_n}$ from the given data points that puts mass $\frac{1}{n}$ at each data point.
2. From this empirical $\hat{P_n}$, we sample multiple times and get many datasets. Each time:
   (a) In turn i, we generate n new data points: $(z_1^{(i)}, z_2^{(i)}, \cdots, z_n^{(i)})$
   (b) Calculate $\hat{\Theta_n}^{(i)} = f(z_1^{(i)}, \cdots, z_n^{(i)})$
3. $(\hat{\Theta_n}*^{(1)}, \cdots, \hat{\Theta_n}*^{(B)})$ gives us the distribution $p^*$of $\hat{\Theta}_n$, which is the estimation on the whole original dataset.

There are several kinds of bootstrap:

- Non-parametric: using empirical $\hat{p_n}$ to replace p.
- Parametric: We assume the distribution is from some distribution family with unknown parameters η. We want to estimate θ.

  - we estimateη and θ, (sometimes, η and θ could be the same) from the given data.
  - we resample new dataset to estimate $\hat{eta}_i$
  - We use $\hat{\eta}_i$ to generate new dataset to estimate $\hat{\theta}_i^*$
  - We estimate $\hat{\theta}$ from $\{\hat{\theta}_i\}$

  It works well if the parametric assumption is right. Verse vice.

- Smoothed Bootstrap: different from parametric bootstrap, we don't assume the distribution family of the data, but assume the distribution is smooth. Under this condition, to generate the estimated distribution.

3.2 Consistency

3.2.1 Definition

We define bootstrap consistency for $\hat{\theta_n}$ if

- we can find an increasing sequence $\{a_n\}$, e.g. $a_n = \frac{1}{\sqrt{n}}$. Usually, we can the a_n^{-1} the convergence rate of $\hat{\theta}_n$.
- The rescaled distribution of $\hat{\theta_n} - \theta$ from p is close to the rescaled distribution of $\hat{\theta_n *} - \hat{\theta_n}$ from p^*. p is the true distribution of parameter θ, p^* is the bootstrap simulated distribution of $\hat{\theta_n}$.

Mathematically,

$$\underbrace{P(a_n(\hat{\theta}_n - \theta) \le x)}_{\substack{\text{True rescaled distribution} \\ \text{of parameter of interest (deterministic)}}} - \underbrace{P^*(a_n(\hat{\theta}_n^* - \hat{\theta}_n) \le x)}_{\substack{\text{Bootstrap rescaled} \\ \text{distribution (random)}}} \xrightarrow{P} 0 \quad \text{as } n \to \infty$$

Here, x can be any number to give the limit the rescaled interval.

An example of the consistency: take $a_n = \sqrt{n}$, which works as a scaling factor here. We have

$$\sqrt{n}(\hat{\theta_n} - \theta) \approx \sqrt{n}(\hat{\theta_n^*} - \hat{\theta_n}) \to 0$$

This means the distribution of $\sqrt{n}(\hat{\theta_n} - \theta)$ is almost the same as $\sqrt{n}(\hat{\theta_n^*} - \hat{\theta_n})$

3.2.2 Usages

Suppose the bootstrap consistency holds, we can get:

$$\frac{Var^*[\hat{\theta}_n^*]}{Var[\hat{\theta}_n]} \xrightarrow{P} 1$$
$$\frac{E^*[\hat{\theta}_n^*] - \hat{\theta}_n}{E[\hat{\theta}_n] - \theta} \xrightarrow{P} 1$$

Therefore, we can approximate $Bias(\hat{\theta}_n) = E[\hat{\theta}_n] - \theta$ by $E^*[\hat{\theta}_n^*] - \hat{\theta}_n$ where

$$E^*[\hat{\theta}_n^*] = \int \hat{\theta}_n^* dP^* \approx \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}_n^{*b} = \bar{\hat{\theta}}_n^*$$

Similarly we can estimate $Var[\hat{\theta}_n]$ by

$$Var[\hat{\theta}_n] \approx Var^*[\hat{\theta}_n^*] = E^*[(\hat{\theta}_n^* - E^*[\hat{\theta}_n^*])^2] = \int (\hat{\theta}_n^* - E^*[\hat{\theta}_n^*])^2 dP^* \approx \frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_n^{*b} - \bar{\hat{\theta}}_n^*)^2$$

The first $\approx$ needs bootstrap consistency$\Leftrightarrow$ n large, the second $\approx$ needs "variance similar to average" $\Leftrightarrow$ B large.

In another way, if we set

- B large enough, $Var^*[\hat{\theta}_n^*] \approx \frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_n^{*b} - \bar{\hat{\theta}}_n^*)^2$ will hold.
- n large enough, $Var[\hat{\theta}_n] \approx Var^*[\hat{\theta}_n^*]$ will hold.

3.2.3 Conclusion

Two points to be noticed:
1. depend on the given dataset.
2. bootstrap distribution is centered around $\hat{\theta_n}$

3.2.4 When does Bootstrap Consistency hold

Basically, when $\sqrt{n}(\hat{\theta_n} - \theta)$ is **asymptotically normal**.

3.3 Applications

3.3.1 Construct Confidence Interval

The problem setting: we use $\hat{\theta_n}$ as an estimator to estimate θ, where $\hat{\theta_n}$ is get from n i.i.d random variables from the real distribution.
The goal is to build a confidence interval I for θ with convergence $1-\alpha$, s.t. $P(\theta \in I) \geq 1-\alpha$

For any random variable w, we define $q_w(\alpha)$ as the α-quantile of w.

Using (1)$\hat{\theta_n} - \theta$ and (2) a two-sided centered interval[1] to rewrite this goal is:

$$1-\alpha = P(q_{\hat{\theta_n}-\theta}(\frac{\alpha}{2}) \leq \hat{\theta_n} - \theta \leq q_{\hat{\theta_n}-\theta}(1-\frac{\alpha}{2}))$$

$$\Leftrightarrow \mathcal{I} = [\hat{\theta_n} - q_{\hat{\theta_n}-\theta}(1-\frac{\alpha}{2}), \hat{\theta_n} - q_{\hat{\theta_n}-\theta}(\frac{\alpha}{2})]$$

However, we don't know $q_{\hat{\theta_n}-\theta}$. Inspired by bootstrap, the distribution of $\hat{\theta_n} - \theta$ is similar to $\hat{\theta_n^*} - \hat{\theta_n}$, and we can have several ways to approximate this interval.

Reversed quantile CI (pivotal CI)

Idea: using $q_{\hat{\theta_n^*}-\hat{\theta_n}}$ to replace $q_{\hat{\theta_n}-\theta}$

$$[\hat{\theta_n} - q_{\hat{\theta_n^*}-\hat{\theta_n}}(1-\frac{\alpha}{2}), \hat{\theta_n} - q_{\hat{\theta_n^*}-\hat{\theta_n}}(\frac{\alpha}{2})]$$

[1] There are two reasons to use this interval: 1. it is the most narrow interval, 2. it is centered

Normal Bootstrap CI

Idea: assuming $\hat{\theta_n}$ is asymptotically Gaussian $\mathcal{N}(\mu, Var(\hat{\theta_n}))$ (that's why it is called normal Bootstrap). Since we don't know $Var(\hat{\theta_n})$, by virtue of bootstrap method, we use $Var(\hat{\theta_n^*})$ to replace.

$$[\hat{\theta_n} - q_z(1-\frac{\alpha}{2})\sqrt{Var(\hat{\theta_n^*})}, \hat{\theta_n} + q_z(1-\frac{\alpha}{2})\sqrt{Var(\hat{\theta_n^*})}]$$

Here, $z \sim \mathcal{N}(0,1)$

Quantile Bootstrap CI (Naive CI)

Idea: directly use the quantile of $\hat{\theta_n^*}$.

$$[q_{\hat{\theta_n^*}}(\frac{\alpha}{2}), q_{\hat{\theta_n^*}}(1-\frac{\alpha}{2})]$$

This is very simple, but not theoretically justified. There are too many conditions to satisfy: 1. the bootstrap consistency, 2. the distribution of $\hat{\theta_n^*}$ is symmetric.

Boostrap T

Idea: inspired by t-distribution (here, it is not necessary to be t-distribution, name as T because the format like t-distribution). The assumption of reversed quantile CI, the approximality of $\hat{\theta_n} - \theta$ and $\hat{\theta_n^*} - \hat{\theta_n}$ may not be true. For example, the shape $Var(\hat{(\theta_n^*)})$ may be related to $\hat{\theta_n}$. In this case, we build another two distributions:

$$t = \frac{\hat{\theta}_n - \theta}{\hat{sd}(\hat{\theta}_n)} \quad \text{and} \quad t^* = \frac{\hat{\theta}_n^* - \hat{\theta}_n}{\hat{sd}(\hat{\theta}_n^*)}$$

These two can be the approximately equal.

$$[\hat{\theta_n} - q_{t^*}(1-\frac{\alpha}{2})sd(\hat{\theta_n}), \hat{\theta_n} + q_{t^*}(\frac{\alpha}{2})sd(\hat{\theta_n})]$$

We also need to know the value of $sd(\hat{\theta_n})$. This is estimated by double bootstrap. Note that if we would simply set $\hat{sd}(\hat{\theta}_n^*) = \hat{sd}(\hat{\theta}_n)$ we would get back to the reversed quantile Bootstrap CI.

for *b in* $1, ..., B$ **do**
  Generate Bootstrap Sample $z_1^*, ..., z_B^*$ from $z_1, ..., z_n$. ;
  Compute Boostrapped estimator $\hat{\theta}_n^{*b}$;
  for *c in* $1, ..., C$ **do**
    Generate Bootstrap Sample $z_1^{**bc}, ..., z_n^{**bc}$ from $z_1^*, ..., z_B^*$;
    Compute Bootstrapped estimator $\hat{\theta}_n^{**bc}$;
  end
  Compute $\hat{sd}(\hat{\theta}_n^{*b}) = \sqrt{\frac{1}{C-1}\sum_{c=1}^{C}(\hat{\theta}_n^{**bc} - \bar{\hat{\theta}}_n^{**bc})^2}$;
  Compute $t^{*b} = \frac{\hat{\theta}_n^{*b} - \hat{\theta}_n}{\hat{sd}(\hat{\theta}_n^{*b})}$.
end
Compute $\hat{sd}(\hat{\theta}_n) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_n^{*b} - \bar{\hat{\theta}}_n^{*b})^2}$;
Compute the desired quantile of $t^{*1}, ..., t^{*B}$;

This method is good at location parameters, e.g. mean, median, trimmed mean and it is second order accurate, converging much faster. (First Order: $P(I \ni \theta) = 1 - \alpha + O(1/\sqrt{n})$, Second order: $P(I \ni \theta) = 1 - \alpha + O(1/n)$).

3.3.2 Testing

We want to test:

$$H_0 : \theta = a, H_1 : \theta \neq a$$

We can solve this by building CI.

H_0 is accepted $\Rightarrow a \in (1-\alpha)$ CI for θ.

H_1 is accepted $\Rightarrow a \notin (1-\alpha)$ CI for θ.

3.3.3 For Regression

Bootstrap is used to **generate new dataset** for regression.

The regression model is

$$y_i = \mu(x_i) + \epsilon_i$$

We have different options to generate new datasets:

- resample $\{(x_i, y_i)\}$ directly. It works if the data points are discrete.
- resample ϵ. Keep X fixed. Get Y' by $\hat{\mu}(X)$, generate ϵ_i' and get the newly generated pair $\{(x_i, \hat{\mu(x_i)} + \epsilon_i')\}$. It requires we have already estimated the $\hat{\mu(\cdot)}$
- resample Y'. Keep X fixed. For each x_i, generate new y_i' according to the distribution $P(Y|X = x_i)$. It requires we have multiple observations of y given x_i, by which can we get the distribution of y given x_i.
- Nothing fixed.
  1. Generate new X, according to the distribution of X.
  2. Then, generate new Y. This can be done by options above.

Chapter 4

Test

Statistical test is to test if a hypothesis is true. According to the construction of the dataset, it can be divided into paired test and unpaired test.

Web resource

Paried means that both samples consist of the same test subjects. A paired t-test is equivalent to a one-sample t-test.

Unpaired means that both samples consist of distinct test subjects. An unpaired t-test is equivalent to a two-sample t-test.

For example, if you wanted to conduct an experiment to see how drinking an energy drink increases heart rate, you could do it two ways.

The "paired" way would be to measure the heart rate of 10 people before they drink the energy drink and then measure the heart rate of the same 10 people after drinking the energy drink. These two samples consist of the same test subjects, so you would perform a paired t-test on the means of both samples.

The "unpaired" way would be to measure the heart rate of 10 people before drinking an energy drink and then measure the heart rate of some other group of people who have drank energy drinks. These two samples consist of different test subjects, so you would perform an unpaired t-test on the means of both samples.

Test can be divided into parametric and non-parametric. Here, **parametric** refers to the parameters used for describe a distribution. They are parameters in the distribution math formulas.

4.1 Parametric Test

- z-test: normal distribution with known variance.
- t-test: normal distribution with unknown variance.

4.2 Non-parametric Test

4.2.1 Motivation

Web resource. In short, non-parametric test doesn't put any assumption on the distribution of data.

4.2.2 Wilcoxon rank sum test (Mann-Witney U test)

Why to use wilcoxon rank sum test

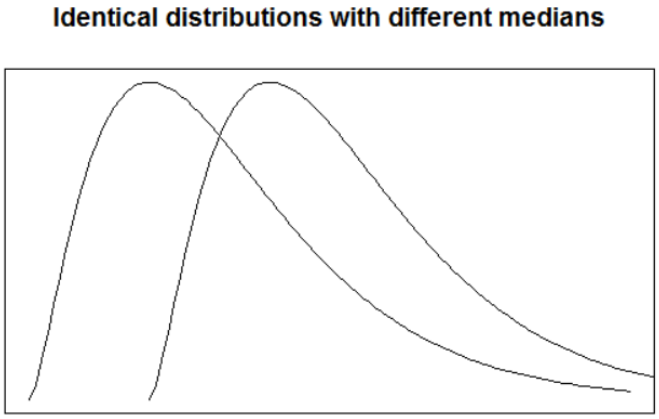
Why don't we use t-test or z-test to test if the two samples from a same distribution? Refer to web resource

Recall that these two parametric tests need the following assumptions:

1. The two samples are independent of one another
2. The two populations have equal variance or spread
3. The two populations are normally distributed

But in wilcoxon rank sum test, we only assumes the first two, not the third. (Yes, we still assume the second condition, the equal variance.) This suits when our samples are small and our data skew or non-normal.

Whereas the null hypothesis of the two-sample t test is equal means, the null hypothesis of the Wilcoxon test is usually taken as equal medians. Another way to think of the null is that the two populations have the same distribution with the same median. If we reject the null, that means we have evidence that one distribution is shifted to the left or right of the other.



Model and Procedure

Set-up

- Observations $Y_1^{(1)}, \cdots, Y_{n_1}^{(1)} \sim F_1$ and $Y_1^{(2)}, \cdots, Y_{n_2}^{(2)} \sim F_2$, all independent
- $H_0 : F_1 = F_2$, H_A:F_1 is a shifted version of F_2.

Procedure

1. Determine the ranks of all data points $(1, \cdots, n_1 + n_2)$
2. Compute U: the sum of ranks in each group

3. The distribution of U under H_0 is given in tables. For larger sample size, it can be approximated by a Normal distribution.

Whenever doing test, we need the observed value and a lot of compared values. How do we get compared values in this test? We randomly reassign the groups to compute the statistics.

4.2.3 Randomization/Permutation test

This is also reflected in the Wilcoxon rank sum test's generation of compared values.

The general framework is if we want to test, we must have multiple datasets to compare the statistics in our given sample dataset. Actually, we don't have those datasets except for the sample one. How should we generate by ourself?

- generate y from some distribution. (This is the idea of parametric test)
- just reshuffle, change the corresponding between x and y. (This is the origin of non-parametric test)

Permutation is one way to do reshuffling. It is very applicable when the sample dataset is small.

Procedure

A little different for paired datasets: the possible rearrangement's number decrease (only done within a group, can be realized by organized in two columns.)

For example, in the former example, our data can be organized as (x_i, y_i). Here x_i denotes the group, 1 or 2, and y_i denotes the value.

In the paired dataset, say we have k pairs, our dataset is like

$$\{((x_1, y_1), (x_2, y_2)), ((x_3, y_3), (x_4, y_4)), \cdots, ((x_{n-1}, y_{n-1}), (x_n, y_n))\}$$

with $n = 2k$. Also, we know if $x_{2k+1} = 1$, then $x_{2k+2} = 2$. So we can use two columns, one column corresponds to the group 1 and the other correspondents to group 2 to reorganize our data. Each row represents one sample. This is a $k * 2$ table, with $e_{ij} = y_{2i+1}$

Permutation p-value

the more extreme ratio among all (large enough if the size is big) the permutation datasets.

Variants

multiple regression Suppose we have n data, and each data has m variables. We can permute y to test **global null**.

But we cannot do no straightforward test for **individual coefficients**. Do not permute the X-columns individually, as this would destroy the correlation structure among the X variables!

4.3 Multiple testing

Multiple means we test multiple **hypotheses** at the same time.

Testing several hypotheses at significance level α simultaneously can lead to a scenario when at least one of the tests find significant evidence against the null hypotheses when if all the null hypotheses are true since allow a false positive probability of α for each test. Over the years several approaches have been developed to overcome this issue and being able to conduct simultaneous tests at an overall significance level α.

The table shows the notations we gonna use:

	H_0 is true	H_a is true	Total
H_0 is not rejected	U	T	$m-R$
H_0 is rejected	V	S	R
Total	m_0	$m-m_0$	m

- m: we totally have m hypothesis to test. This is a fixed and known (given) number.
- m_0, $m-m_0$: are the fixed but unknown numbers, showing exactly how many hypothesis are true, how many are false.
- R,$m-R$: are what we get from our experiments. Random but known.
- random based on test: U,V, T, S, R.
- fixed: m,$m-m_0$.
- known: m, R
- unknown: m_0,V, S, U, T.

4.3.1 Example

A medical experiment setting is

- n people in one gene experiments (e.g. 50 in the control group and 50 in the cancer group),
- m tested genes (e.g. 6033).
- x_{ij}: the (i,j) of the matrix $X \in \mathbb{R}^{n\times m}$.
- y_i: the predict variable, in indicator of cancer.

We want to test if the gene expression relates to cancer. Totally, we have m (e.g. 6033) hypothesis for each gene.

There are two points of view:

- Global test: Is $\bigcap_{i=1}^{m} H_{0,i}$ true? Is there any genetic basis for this cancer?
- Individual test: $\forall i, H_{0,i}$ is true?

4.3.2 Error Measurements

FWER

Family-wise error rate.

- the probability of making at least one Type I Error.
- "family-wise" error rate comes from family of tests, which is the technical definition for a series of tests on data.
- $P(V \geq 1)$
- This is a probability, not a random variable. This means we cannot get the value from only one experiments.
- Each of these hypothesis is controlled by α

FDR

False discovery rate.

- Discovery: means discovering the "wrong" hypothesis. It refers to all the hypothesis we think they are true, based on statistical testing.
- False: means what we think is wrong is not wrong actually.
- $E[Q]$,$Q = \frac{V}{R}$.
- This is an expectation. Not a random variable.

Relationships

- FWER $\geq$ FDR. It means controlling FWER is stricter than FDR.
- $\alpha \leq$ FWER $\leq m\alpha$

Proof of the first relationship

Global null Global null means all the hypothesis are true. This means $S = 0 \Rightarrow R = V$.

FWER $= P(V \geq 1)$, FDR $= E[\frac{V}{R}] = E[\frac{V}{V}]$

We set

$$\frac{V}{V} = \begin{cases} 0, & \text{if } V = 0 \\ 1, & \text{otherwise} \end{cases}$$

Therefore,

$$\text{FDR} = E[\frac{V}{V}] = P(V = 0) \cdot 0 + P(V > 0) \cdot 1 = P(V \geq 1) = \text{FWER}$$

Not global null When the global null is not true, we know $S \geq 0$ Similarly, we have

$$\frac{V}{R} = \frac{V}{V+S} = \begin{cases} 0, & \text{if } V = 0 \\ \leq 1, & \text{otherwise} \end{cases}$$

Therefore,

$$\begin{aligned} \text{FDR} &= E[\frac{V}{V+S}] = P(V=0)\cdot 0 + P(V>0)\cdot E[\frac{V}{V+S}|V>0] \\ &\leq P(V>0)E[\frac{V}{V}|V>0] = P(V>0) = \text{FWER} \end{aligned}$$

Proof of the second relationship

Right side

$$P(V \geq 1) = 1 - P(V = 0)$$

$P(V = 0)$ means we don't reject all the true hypothesis under the level α. Suppose we have m_0 true hypothesis,

- all the (true) hypothesis are independent: $P(V = 0) = (1-\alpha)^{m_0} \Rightarrow$ FWER $= P(V > 1) = 1 - (1-\alpha)^{m_0} \leq 1 - (1-\alpha)^m$ $(1-\alpha)^m = 1 - m\alpha + \frac{m(m-1)}{2}\alpha^2 + \cdots$. Therefore, FWER$\leq m\alpha - \frac{m(m-1)}{2}\alpha^2 + O(\alpha^2)$.
- some of the (true) hypothesis are dependent: under this conditions, we reduce m_0, but it doesn't change the more powerful control m. The inequality is still true.

Another more intuitive solution:

$$\begin{aligned} \text{FWER} &= P(\text{at least one false rejection among } T_1, T_2, \cdots, T_m) \\ &= P(\bigcup_{i=1}^{m}\{\text{false rejection of Test}T_i\}) \\ &\leq \Sigma_{i=1}^{m} P(\text{false rejection of Test}T_i) \\ &= am \end{aligned}$$

Left side Denote the case "reject a random true hypotheses" as A.Since the probability of A is controlled by α, and whenever A happens, $V \geq 1$ must happen. That means $P(V \geq 1) \geq P(A) = \alpha$.

4.3.3 Control Methods

We want to control the false rejection for false rejection under the multiple testing settings. There are two concepts: **strong control** and **weak control**.

Weak control Control under the global null conditions.

Strong control Control under all the configurations of hypothesis.
There are two ways to control,

- control FWER. Examples: Bonferroni control, Westfall-Young permutation procedure.
- control FDR. Examples: Benjamini-Hochberg

Intuitive (Bonferroni) control

Inspired by the right side of the second relationship, we can choose $\alpha = \frac{\alpha^*}{m}$ for each hypotheses testing.

However, this control is too strict. Considering two situations:

- m is very large. From the above proof FWER$\leq m\alpha - \frac{m(m-1)}{2}\alpha^2 + O(\alpha^2)$, the second item will be large and reduce the actual FWER obviously. It means we push too much control on FWER.

- hypothesis are dependent. Suppose we do 100 hypothesis by repeating one hypotheses 100 times. We only have need $\alpha = \alpha^*$ actually. But using this way, we will get $\alpha = \frac{\alpha^*}{100}$.

Westfall Young permutation procedure

The effect of Bonferroni control is closely related to the relations among hypothesis, e.g. whether the hypothesis are independent. But we don't have such knowledge.

Now, we try to consider this problem by viewing each p-value of those hypothesis. suppose we do each test at level δ, how to choose δ to make FWER$\leq \alpha$?

Under the global null configuration, we have

$$\begin{aligned}\text{FWER} &= P(V \geq 1) \\ &= P(\text{at least one p-value} \leq \delta) \\ &= P(\min\{p_1, p_2, \cdots, p_m\} \leq \delta) \\ &\leq \alpha\end{aligned}$$

The last line is our objective. If we take $\min\{p_1, p_2 \cdots, p_m\}$ as a new variable D, our goal becomes: find δ s.t. $P(D \leq \delta) \leq \alpha$. Obviously, δ is the α-quantile of D. So, what we need to do is do find the α-quantile of D. We can achieve an empirical δ by repeating many experiments.

Determine δ The whole procedure is given as:

for *number of permutations* **do**
  Permute the y-column of the data matrix and do a two sample test (e.g. Wilcoxon or a "nested" permutation test) for each x_j-column. Let $p_j, j = 1, ..., m$ be the corresponding p-value and store $p_{min} = \min(p_1, ..., p_m)$;
end
Set δ = empirical α-quantile of the permutation distribution of $\{p_{min}\}$;
Reject any null hypothesis where the two-sample test on the original data has p-value $\leq \delta$;

Algorithm 1: Westfall Young Permutation test

Bonferroni-like Weak control

There is another two-step idea to control.

1. Reject global null if $\min_{i=1,\cdots,m} p_i \leq \frac{\alpha}{m}$. This criterion is like Bonferroni control, using $\frac{\alpha}{m}$, also like the idea of Westfall Young permutation control, using $\min_{i=1,\cdots,m} p_i$.

2. If the global null is rejected, reject hypothesis $H_{0,i}$ with $p_i \leq \alpha$.

It is still a **weak** control, not a strong control. An example will explain the weakness.

Suppose $X_i \sim \mathcal{N}(\mu_i, 1), i = 1, 2, \cdots, m$ The reality is μ_1 is very large while $\mu_2 = \mu_3 = \cdots = \mu_m = 0$.

Our hypotheses are $\mu_1 = \mu_2 = \cdots = \mu_m = 0$. (We know it is not true.)

Then we test our data and get $p_1, p_2, \cdots, p_m$, the p-value for each hypothesis. Obviously p_1 could be really small to be less than $\frac{\alpha}{m}$. Therefore, the global null will be rejected. Then, we check each p-value p_i. Ideally, we will make around $\alpha(m-1)$ false rejection. That means we don't control FWER.

Remark From Bonferroni(-like) idea, we can see if m is large, FWER could be too strict. We turn to another control, FDR, to see if it has a "large-m" problem.

Benjamini-Hochberg

The problem is that we do not know V (it is a random variable) so is $Q = \frac{V}{R}$ but we can use $E\left[\frac{V}{R}\right] = FDR$. Note that FDR control is not a statement about individual experiment. If we repeat a procedure many times on average we control FDP. The following Benjamini-Hochberg procedure is meant to achieve this.

Web example

The procedure of Benjamini-Hochberg control is:

1. we order original p-values by ascending order and get a new series of p-values: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$.

2. We choose a FDR q.

3. We compare each $p_{(i)}$ with $\frac{i}{m}q$ and find the smallest i_0, with $p_i \geq \frac{i}{m}q$ $\forall i > i_0$. In the example 4.3.3, i_0 is 5. Note that it doesn't require that $\forall i < i_0, p_i < \frac{i}{m}q$. In the example, $p_2 < \frac{2}{m}q$ but $p_3 > \frac{3}{m}q$

4. We reject all the former i_0 hypotheses.

Variable	P Value	Rank	(I/m)Q
Depression	0.001	1	0.01
Family History	0.008	2	0.02
Obesity	0.039	3	0.03
Other health	0.041	4	0.04
Children	**0.042**	**5**	**0.05**
Divorce	0.060	6	0.06
Death of Spouse	0.074	7	0.07
Limited income	0.205	8	0.08

Figure 4.1: The example from web

Theorem For independent test statistics (or p-values), the Benjamini-Hochberg procedure controls the FDR at level q. More precisely, FDR $= \frac{m_0}{m} q \leq q$.

Proof Let $m_0 \geq 1$ (If $m_0 = 0$, FDR control is undefined as $V = 0$). Define $V_i = \mathbb{1}_{\{H_i \text{ is rejected}\}}$ for $i = 1, ..., m$. Then

$$FDP = \begin{cases} \frac{V}{R} & \text{If } R > 0 \\ 0 & \text{If } R = 0, \text{also } V = 0 \end{cases} = \frac{V}{\max(R,1)} = \sum_{i \in \mathcal{H}_0} \frac{V_i}{\max(R,1)}$$

$\mathcal{H}_0$ is the set of true hypotheses.

If we can show that $E\left[\frac{V_i}{\max(R,1)}\right] = \frac{q}{m}$, which means the probability of a true hypothesis is rejected, then

$$\text{FDR} = E[\text{FDP}] = E\left[\sum_{i \in \mathcal{H}_0} \frac{V_i}{\max(R,1)}\right] = \sum_{i \in \mathcal{H}_0} E\left[\frac{V_i}{\max(R,1)}\right] = \sum_{i \in \mathcal{H}_0} \frac{q}{m} = \frac{m_0}{m} q$$

Let $i \in \mathcal{H}_0$, we write

$$\frac{V_i}{\max(R,1)} = \sum_{k=1}^{m} \frac{V_i \mathbb{1}_{\{R=k\}}}{k}$$

In a given experiment, R is a fixed but random number. We rewrite R as an indicator to get a fraction without the maximum term. Note that when there are k rejections, $H_{(i)}$ is rejected if $p_{(i)} \leq p_{(k)} = \frac{k}{m} q$. So

$$V_i \mathbb{1}_{\{R=k\}} = \mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} \mathbb{1}_{\{R=k\}}$$

Let $R(p_i \to 0)$ be the number of rejections when p_i is set to 0. Then

$$V_i \mathbb{1}_{\{R=k\}} = \mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} \mathbb{1}_{\{R=k\}} = \mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} \mathbb{1}_{\{R(p_i \to 0)=k\}}$$

The usage of this transform is that we split R from p_i. In the left form, R is a random variable depends on p_i. But in the form $\mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} \mathbb{1}_{\{R(p_i \to 0)=k\}}$, $R(p_i \to 0)$ is independent of p_i. Therefore, we can calculate the expectation by extracting R later.

We show this equation is true, because

- If $V_i = 0$, then $\mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} = 0$ Leftside equals to rightside. Both are 0.
- If $V_i = 1$, then $\mathbb{1}_{\{p_i \leq \frac{qk}{m}\}} = 1 \Leftrightarrow p_i \leq \frac{k}{m} q$
  Since H_i is already rejected, at the "new experiments" where $p_i = 0$ it will still be rejected with other p-values keeping unchanged. That is it does not change number of rejections.

Let $F_i = \{p_1, ..., p_{i-1}, p_{i+1}, ..., p_m\}$. Since

$$E\left[\frac{V_i}{\max(R,1)}\right] = E_{F_i}\left[E\left[\frac{V_i}{\max(R,1)} | F_i\right]\right]$$

it suffices to show that $E\left[\frac{V_i}{\max(R,1)}|F_i\right] = \frac{q}{m}$.

$$\begin{aligned}
E\left[\frac{V_i}{\max(R,1)}|F_i\right] &= E\left[\sum_{k=1}^{m}\frac{V_i\mathbb{1}_{\{R=k\}}}{k}|F_i\right] \\
&= \sum_{k=1}^{m} E\left[\frac{V_i\mathbb{1}_{\{R=k\}}}{k}|F_i\right] \\
&= \sum_{k=1}^{m} E\left[\frac{\mathbb{1}_{\{p_i\leq\frac{qk}{m}\}}\mathbb{1}_{\{R(p_i\to 0)=k\}}}{k}|F_i\right] \\
&= \sum_{k=1}^{m}\frac{\mathbb{1}_{\{R(p_i\to 0)=k\}}E[\mathbb{1}_{\{p_i\leq\frac{qk}{m}\}}|F_i]}{k} \\
&= \sum_{k=1}^{m}\frac{\mathbb{1}_{\{R(p_i\to 0)=k\}}\frac{qk}{m}}{k} \\
&= \frac{q}{m}\underbrace{\sum_{k=1}^{m}\mathbb{1}_{\{R(p_i\to 0)=k\}}}_{=1} \\
&= \frac{q}{m}
\end{aligned}$$

$E[\mathbb{1}_{\{p_i\leq\frac{qk}{m}\}}|F_i] = \frac{qk}{m}$ because hypotheses are independent so conditioning on F_i does not matter.

Chapter 5

Model Selection

The objective is that given a dataset, select the best model for it. Two objects here: dataset and model. Several basic ideas for the goal:

- Feature selection: the object is the dataset. Trying to find the best subset of the data for modeling. Usually applied when the feature number is big.
- "Degree" selection: the object is the degree of regression model, or something like degree, e.g. the neighbors number in K-NN. The object is hyperparameter in a family of model. The model type(family) and the dataset is given, we try to find the best hyperparameter.
- "Model" prototype selection: the object is the model family. For instance, given the dataset, we try to find which model type to use, tree, regression, k-nn, boost and so on.

5.1 Feature selection

Why we want to use the subset of features instead of all of them? It could because the number of features is too big or the complexity of the model is too high.

For feature, we can melt many features as one, or select few from numerous features. Here, we focus on the latter method.

Settings Each data point $1 \times p$ (p columns).

- given k, I want to find the best subset with k features to model.
- k is not given, I want to find the best k and the corresponding features to model.

5.1.1 Judge criterion

C_p, AIC, BIC, adjusted R^2 and CV can evaluate the trade-off between bias and complexity.

- why we don't only use RSS (residual sum of squares)? Because RSS will decrease with more features, which may lead to overfitting.

All of the formulas below are under the assumption that the error of the model follows i.i.d normal distribution.

Mallow's C_p

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

- d: the number of variables.
- $\hat{\sigma}^2$: estimate of $\text{Var}(\epsilon)$. Usually taken from the full model.

choose the model with the lowest C_p.

AIC

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

The general version is defined

$$2K - 2\ln L$$

- K: the number of parameters
- L: the likelihood.

BIC

$$\frac{RSS}{n\hat{\sigma}^2} + \frac{d}{n}\log n$$

Adjusted R^2

$$1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

Mallow's Cp, AIC, BIC: minimize. the smaller the better.
Adjusted R^2: maximize. the bigger, the better.

5.1.2 Subset Construction

For p feature, there are $\binom{p}{k}$ choices of features. We can do one of the following methods:

- compare all of the combinations. Totally $\binom{p}{k}$ choices. When p is very big, this choice is not practical. If k is not given, we need to do $\binom{\Sigma_{k=0}^{p}(p}{k)=2^p}$ times experiments.
- backward step-wise: we first include all of the features. In each step, we drop the one with least contribution to our model. Repeat until we have k features. If k is not given, there are totally $1 + \Sigma_{k=0}^{p-1}(p-k) = \frac{p(p+1)}{2} + 1$
- forward step-wise: we start from 0 feature. In each step, we include one feature making most improvement of the model, combined with features we have selected. Repeat until we have k features. If k is not given, there are totally $\Sigma_{k=1}^{p}(p-k+1) = \frac{p(1+p)}{2}$.

We can select the best k according to the specific judge criterion.

5.1.3 Norm Criterion (Shrinkage)

If we restrict the subset feature size must be less than s, we can define the following optimization problem:

$$\hat{\beta}_s = \arg_{\beta:\|\beta\|_0 \leq s} \min RSS(\beta)$$

here, the space is non-convex, the objective is convex. Written in another form:

$$\arg_{\beta:\beta\in\mathbb{R}^p} \min RSS(\beta) + \lambda_s \|\beta\|_0$$

here, the objective is non-convex, the space $\mathbb{R}^p$ is convex.

Since $\|\cdot\|_0 \leq s$ is not convex, we can relax by changing the norm measurements and have more variant. The typical two are ridge and LASSO.

- Ridge: 2-norm
  $$\hat{\beta}_s = \arg_{\beta:\|\beta\|_2 \leq s} \min RSS(\beta)$$
  Written in another form:
  $$\arg_{\beta:\beta\in\mathbb{R}^p} \min RSS(\beta) + \lambda_s \|\beta\|_2$$
- LASSO(Least Absolute Shrinkage and Selection Operator): 1-norm
  $$\hat{\beta}_s = \arg_{\beta:\|\beta\|_1 \leq s} \min RSS(\beta)$$
  Written in another form:
  $$\arg_{\beta:\beta\in\mathbb{R}^p} \min RSS(\beta) + \lambda_s \|\beta\|_1$$
- P-norm:
  $$\|\beta\|_q^q = \Sigma_{i=1}^p \beta_i^q$$
  when $q < 1$, this is not a norm.

Remark

- λ_s is a fixed parameter, decided by s.
- LASSO/Ridge are computationally much easier, we don't need to search on all possible parameters to find the best one, but can have a direct solution.
  For instance,
  $$\hat{\beta}^{Ridge} = (X^T X + \lambda I)^{-1} X^T Y$$
  when $\lambda > 0$, $X^T X + \lambda I$ is invertible, well-conditioned.
- Why in ridge regression, the coefficients tend to be the same? Because one of our objective is $\lambda\|\beta\|_2$ to minimize $\|\beta\|_2$. If $\beta_i + \beta_j = c$ is fixed, when $\beta_i = \beta_j$, $\beta_i^2 + \beta_j^2$ is the smallest.
- We still need to select the appropriate λ_s or (s in the norm restriction). We can use the above judge criterion, e.g. AIC.
- If variables are highly-correlate, LASSO tends to choose one of them (can be very unstable) while ridge tends to divide the weights among them.

Variants

Elastic net We can combine LASSO and ridge. For instance Elastic net

$$\hat{\beta}_s = \arg_{\beta:(1-\alpha)\|\beta\|_1+\alpha\|\beta\|_2\leq s} \min RSS(\beta)$$

Written in another form:

$$\arg_{\beta:\beta\in\mathbb{R}^p} \min RSS(\beta) + \lambda_s\|\beta\|_1 + (1-\lambda_s)\|\beta\|_2$$

Adaptive LASSO We can put weights to variables to control which variables must be included/decluded.

$$\hat{\beta}_s = \arg_\beta \min RSS(\beta) + \lambda\sum_{j=1}^{n} w_j|\beta_j|$$

when w_j is very big, it pushes a lot of penalty on β_j and very likely to declude it.

Group LASSO The aim is to dealing with some variables must exist or disappear together. It acts like LASSO, but on the group level, not on individual variable level.

Settings: $X \in \mathbb{R}^{n\times p}$. The p variables are divided into K groups (the trivial case is $K = p$, that is each variable is viewed as one group.) Hence, we can write:

$$X\beta = x_1\beta_1 + x_2\beta_2 + \cdots + x_K\beta_K$$

The objective of group LASSO is:

$$\hat{\beta}_\lambda = \arg_{\beta\in\mathbb{R}^p} RSS(\beta) + \lambda\sum_{l=1}^{K} w_l\|\beta_l\|_1$$

Note here the penalty item has K instead of p subitems.

It can be applied to categorical variables by putting all dummy variables in one group. (If a categorical variable has n levels, it requires $n-1$ dummy variables to represent.)

Procedure for LASSO and Ridge

Given $X \in \mathbb{R}^{N\times P}$, first we need to center and standarize the data.

- Why? If we only use 0-norm, we don't need to do that, cause β_i's value will not change $\|\beta\|_0$. But for 1-norm and 2-norm, it is required. Different scale of β_i indeed changes $\|\beta\|_2$ and $\|\beta\|_1$
- How? Use $\hat{x}_{ik}$ instead of x_{ik}.
  - $\hat{x}_{ik} = \frac{x_{ik}-\bar{x_k}}{\sqrt{Var(x_k)}}$
  - $\bar{x_k} = \frac{\Sigma_{i=1}^N x_{ik}}{N}$
  - $Var(x_k) = \frac{\Sigma_{i=1}^N (x_{ik}-x_k)^2}{N}$

5.2 Regression variants

The idea of LASSO and ridge is to reduce the model complexity to reduce **variance**. Now, we try to reduce**bias** by increasing flexibility.

$$y_i = m(x_i) + \epsilon_i$$

Here, the $m(\cdot)$ is not linear anymore.

5.2.1 Polynomial regression

Model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k + \epsilon_i$$

- $y_i \in \mathbb{R}^n, x_i \in \mathbb{R}^n, \epsilon_i \in \mathbb{R}^n$
- $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$

Remark

Advantage: easy to fit.
Disadvantage: unstable near boundaries. Cannot fit functions like sin

Orthogonality

In Rcode, `poly(var,power)` is to construct orthogonal basis.An example about orthogonal or not orthogonal.

The benefits of orthogonality are:

- see whether a certain order in the polynomial significantly improves the regression over the lower orders.
- adding or removing variables does not change the fitted values.

ANOVA always works!

Theoretical proof

Settings:

$$Y = X\beta + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$Var(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$$

Under the condition of orthogonality, we have

$$X^T X = \begin{bmatrix} \sum_{i=1}^n x_{i1}^2 & 0 & \cdots & 0 \\ 0 & \sum_{i=1}^n x_{i2}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \sum_{i=1}^n x_{ip}^2 \end{bmatrix}$$

$$\Rightarrow (X^T X)^{-1} = \begin{bmatrix} 1/\sum_{i=1}^n x_{i1}^2 & 0 & \cdots & 0 \\ 0 & 1/\sum_{i=1}^n x_{i2}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1/\sum_{i=1}^n x_{ip}^2 \end{bmatrix}$$

$$\Rightarrow \hat{\beta}_j = \frac{\sum_{i=1}^n x_{ij} y_i}{\sum_{i=1}^n x_{ij}^2}$$

$$se(\hat{\beta}_j) = \sqrt{\sigma^2 (X^T X)_j^{-1}} = \sqrt{\hat{\sigma}^2 / \sum_{i=1}^n x_{ij}^2}$$

We can see $\hat{\beta}_j$ is only influence by the j-th column of X. If we leave out some other columns, the value of $\hat{\beta}_j$ will not be changed. However, for the standard error, since we don't know σ^2, we use $\hat{sigma}^2$ which is calculated from our data X. If we leave out some variables, they will be "added into" errors, so $\hat{\sigma}^2$ is very likely to change.

In summary, adding/removing variables(columns of X)$\hat{\beta}_j$ will not change, but $se(\hat{\beta}_j)$ and p-value are very likely to change.

5.2.2 Step functions

The most basic setting: there are some jumps in data, like steps.

Method:

1. create cut points $c_1, c_2, \cdots, c_k$ in the range of x
2. create $k+1$ new variables between/outside these points. e.g. $C_0(x) = \mathbb{1}_{x \le c_0}, C_1(x) = \mathbb{1}_{c_1 \le x \le c_2}, \cdots, C_k(x) = \mathbb{1}_{x > C_k}$

$$y_i = \sum_{i=0}^k \beta_i C_i(x) + \epsilon_i$$

Combining polynomial regression and step functions, we will get regression splines.

For instance, we fit totally two different polynomial regressions when x is in different fields.

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \cdots + \beta_{k1} x_i^k + \epsilon_i, & \text{if } x \le c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \cdots + \beta_{k2} x_i^k + \epsilon_i, & \text{otherwise} \end{cases}$$

5.2.3 Regression splines

Degree-d spline

Guarantee the $d-1$ degree continuity at the split points.

Continuity Piecewise degree d polynomial with continuity in derivatives up to $d-1$.

Examples:

- degree 0 spline: step functions, piecewise constant
- degree 1 spline: piecewise linear functions, left-/right-side of the split points have the same value. That is continuous at the split line.

Freedom How many free parameters do we have?

Example: cubic splines.

If the splines are totally free, we need 4 parameters (for items x^0, x^1, x^2, x^3 specifically) for each interval. But the continuity requirements restrict 3 parameters (of x^0, x^1, x^2) of the intervals after the first interval (actually, these k intervals can be any k intervals, not necessary to be the left k.).

$$4 * (k + 1) - 3 * k = k + 4$$

It can also be understood by:

- each interval has 4 parameters. There are $k + 1$ intervals.
- each knot restricts 3 parameters. There are k knots.

$$4 * (k + 1) - 3 * k = k + 4$$

Generalized to degree d-splines:

- each interval has d+1 parameters. There are $k + 1$ intervals.
- each knot restricts d parameters. There are k knots.

$$(d + 1) * (k + 1) - d * k = d + k + 1$$

Choices of splines

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \sum_{j=1}^{k} \gamma_j h(x, \zeta_j) + \epsilon_i$$

$$h(x, \zeta) = (x - \zeta)_+^3 = \begin{cases} (x - \zeta)^3, & x > \zeta \\ 0, & \text{otherwise} \end{cases}$$

choices of knots number and locations Number: we can use CV to find the optimal k.

Locations: in practice it is common to place knots ina uniform fashion.

Natural cubic spline

Natural cubic splines are cubic splines with additional boundary restriction that the splines are linear outside the outer knots.

The freedom: at the first and last knots, we add two more restriction on each, $\beta_2 = \beta_3 = 0$. Totally, the degree of freedom is

$$k + 4 - 2 * 2 = k$$

Natural spline is much stable than degree-d spline at the boundary.

5.2.4 Smoothing splines

Control the magnitude of the second derivative to control the smoothness of the spline. Use *lambda* to trade-off between SSE and smoothness

Consider a class of functions$\} = \{g : [a,b] \to \mathbb{R} : g''\text{exists\&} \int_a^b g''(x)^2 dx < \infty\}$

$$\hat{g} = \arg_g \min \underbrace{\sum_{i=1}^{n}(y_i - g(x_i))^2}_{\text{SSE}} + \underbrace{\lambda \int_a^b g''(x)^2 dx}_{\substack{\text{Regularization term} \\ \text{measure roughness of g}}}$$

the role of λ Control the smoothness of the spline (from the shape), the freedom of parameters.

- $\lambda = 0$: $\hat{g}$ just minimizes the sum of square error. Degree of freedom is n, the number of data points.
- $\lambda \to \infty$: $g''(x) = 0$, the LSE linear estimation. Degree of freedom is 2 (linear).

We can define "degree of freedom" from λ.
We rewrite $\hat{g}_\lambda$ by Y[1]

$$\hat{g}_\lambda = S_\lambda \cdot Y$$

with

- $S_\lambda = B(B^T B + \lambda\Omega)^{-1} B^T$
- $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$
- B: design matrix for basis of natural splines with n knots.

The "degree of freedom" is defined by $\text{tr}(S_\lambda)$

Remark It can be shown that $\hat{g}_\lambda$ is

- piecewise cubic polynomial,
- with knots at **data points**,
- 0-th, 1-st, 2-nd derivatives continuous at the knots
- linear outside the outer knots.
- it is not exactly same as the solution of natural cubic spline with knots at data points, but a shrunk version. λ controls the shrunk level.

5.2.5 Local regression

Assume we fit a linear local regression on 1-dimension independent variables.

[1] Recall the linear regression, where $\hat{Y} = X \cdot \hat{\beta} = X(X^T X)^{-1} X \cdot Y$

algorithm

1. select a data point x_i
2. give weights to its k neighbors. (e.g. 1 for all, the reverse of distance by different norm.)
3. $\hat{f}(x_i) = \hat{\beta}_{i,0} + \hat{\beta}_{i,1} x_i$, where

$$\hat{\beta}_{i,0}, \hat{\beta}_{i,1} = \arg_{\beta_{i,0}, \beta_{i,1}} \min \sum_{j=1}^{n} w_{ij} (y_j - \beta_{i,0} - \beta_{i,1} x_i)^2$$

Remark This method usually works poorly for high-dimension (larger than 3 dimensions) for few closed neighbors could be found in high dimensions.

This method is unsupervised and non-parametric. Whatever new points are given, we can find its prediction by finding its neighbors and fitting them as well as possible.

5.2.6 GAM (generalized additive model)

Using $f(x)$ instead of x to show non-linear patterns.

Using linear format to keep additivity.

$$y = \begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_k \end{bmatrix} \cdot \begin{bmatrix} 1 \\ f_1(x) \\ \vdots \\ f_k(x) \end{bmatrix} + \epsilon$$

Remark Additive: we can still examine the effect of each X on Y individually while holding all of the other variables fixed.

Also, because of this, it cannot automatically include interaction factors, which could be manually included.

5.3 Other models

5.3.1 Trees

Trees are non-linear and not smooth models. They could be used for both classification and regression.

The general model is

$$y_i = \sum_{i=1}^{M} \beta_i \mathbb{1}_{x \in R_i} + \epsilon_i$$

- R_i denotes a region of x.

Evaluation

The most difficult part of building a tree model is to find the right partition regions $\{R_i\}$.

Possible solutions:

- rectangular regions.
- greedy method: recursively binary splitting.

Gini index To evaluate whether a split will improve the result.

$$I(D) - [\frac{n_L}{n} I(D_L) + \frac{n_R}{n} I(D_R)]$$

If the equation > 0, we split.
Otherwise, no.

$$I(D) = p(1-p)$$

5.3.2 Bagging

Bootstrap aggregating.

Model

Create B bootstrap datasets from the original dataset. Each is noted as $\{(x_i^{*b}, y_i^{*b})\}$

- regression:

$$\hat{g}^{bag}(x) = \frac{1}{B} \cdot \sum_{i=1}^{B} \hat{g}^{*b}(x)$$

- classification:

$$\hat{g}^{bag}(x) = \arg_k \max \hat{p}_k^{bag}(x)$$

$\hat{p}_k^{bag}(x) := \frac{1}{B} \sum_{i=1}^{B} \hat{p}_k^{*b}(x)$, the average probability of x belonging to class k over all B functions. $\hat{p}_k^{*b}(x)$ the b-th function's result of the probability of x belonging to class k.

5.3.3 Random Forest

Using bagging method to train a series of tree models.

Chapter 6

Others

6.1 R language

6.1.1 Basic

matrix VS dataframe

refer to this webpage

Matrix In essence it is a reorganization of an array **by column**. Each element must have the same type (homogeneous). We can treat a $N \times M$ matrix as an array v, in which the $M_{ij} = v_{(j-1)\cdot N+i}$

Dataframe In essence it is a list of arrays. Each **column** corresponds to an array. These arrays must have the same length but their type can be different (heterogeneous) (e.g. the first array (column) is character while the second array (column) is numerous.)

Rcode Example:

```
library(ISLR)
Hitters<-na.omit(Hitters)
# create matrix and data.frame object:
Hitters_df<-as.data.frame(Hitters)
Hitters_mat<-as.matrix(Hitters)

typeof(Hitters_mat) # "character"
# the matrix is treated as an array, denoted by its type.
typeof(Hitters_df) # "list"
# the dataframe is treated as a list.

# length
length(Hitters_mat) # 5260 = 263*20
length(Hitters_df)  # 20 = ncol

# dimension
dim(Hitters_mat) # 263 20
dim(Hitters_df) # 263 20

Hitters_mat[2] # "479": (2,1)
Hitters_df[2] # the second column
Hitters_mat[264] # "81": (1,2) 264=263*(2-1)+1
# Hitters_df[264] # Error
```

```
names(Hitters_mat) # NULL
names(Hitters_df) # colnames(Hitters_df)
```

set.seed()

it allows us to regenerate results, which, however, is related to "times". That means the first time's result may be different from the second time. But the order is counted from each time we set the random seed.

```
set.seed(5)
sample(1:10,10) # 1-st time
# [1]  3  7  8  2  1  4 10  5  9  6
sample(1:10,10) # 2-nd time
# [1]  3  5 10  4  2  6  9  8  7  1
sample(1:10,10) # 3-rd time
# [1]  9  7  2  8  1  3 10  5  6  4
set.seed(5) # reset the seed, we count from the begining.
sample(1:10,10) # 1-st time
# [1]  3  7  8  2  1  4 10  5  9  6
sample(1:10,10) # 2-nd time
# [1]  3  5 10  4  2  6  9  8  7  1
sample(1:10,10) # 3-rd time
# [1]  9  7  2  8  1  3 10  5  6  4
```

dot in naming a function

Refer to this webpage. Generally, the dot (".") character in naming a function means assign a specific class to a function, which can be called by that class use name before the "." character.

```
# define a function as
myfunction.myclass <- function(x,...) {...}
# Then the dot has special meaning. For all objects with class myclass
    calling
myfunction(a) # a is an instance of the class: myclass
# will actualy call function myfunction.myclass myfunction.myclass(a)
```

class in R

Refer to this webpage. "In fact, everything in R is an object." "Class is a blueprint for the object". We can use class() function to define a class for an object.

```
myclass<-function(x) {
s = list(a=1,b=2,r=3)
class(s)="myclass1"
return(s)
}
class_demo<-myclass(4)
class(class_demo) # "myclass1"
```