Compiled using XeLaTeX engine.

A Brief Summary of Statistics Minor Courses

清华大学统计学辅修课程知识总结

Tuorui Peng

v1ncent19

Department of Physics, Tsinghua University, Beijing, China.

Department of Statistics and Data Science, Northwestern University, Evanston, IL, USA.

Compiled using XeLaTeX engine.

A Brief Summary of Statistics Minor Courses

清华大学统计学辅修课程知识总结

Tuorui Peng[1]

v1ncent19

[1]Department of Physics, Tsinghua University, Beijing, China.

[1]Department of Statistics and Data Science, Northwestern University, Evanston, IL, USA.

2024 年 9 月 20 日

[1]E-mail: v1ncent19@outlook.com | HomePage: https://v1ncent19.github.io

前言

The statistics course summary actually started from the exam cheatsheets during my first year in minor (2020 fall). At first it only contained some notion explanation / important results. Later I began to add mathematical deductions, intuition of methods, useful external links, etc. LaTeX is used to compile so that I could include cross reference, citation, index in the note so that I could look something up in case of forgetting. Also I treat it as an opportunity to practice TeXlanguage. I made it publicly available in 2022 fall and from then on I've been trying to enrich early chapters, and correct / improve contents. I hope that this summary note could benefit students like me (e.g. statistics minor students in Tsinghua Univ.). Also I am considering to produce a ZH translation, which might be more convenient for beginners / people not used to English. Also it could act as a ZH-EN contrast. Now the translation of some chapters are already in progress (but slow lol). Please contact me if you are also interested.

Since it's first written as a reminder for statistics, I may not contain some basic knowledge like calculus, linear algebra, etc in this note. And usually I just write it in a physics student perspective. Besides, for consistency within file and familiarity for myself, I might not use some commonly used notations, so please be careful if you decide to use some of my notes for reference. You can look up the denotation table in the case of confusion. Also I am not a native English speaker, so please forgive me for any grammatical errors. I will try to correct them as soon as possible.

致谢

The project is mostly based on the course content of statistics minor at Center of Statistical Science at Tsinghua. I would like to express my gratitude to the faculty for their careful preparation and for their help to me. They include (in the order I first took their courses): Wanlu Deng, Jiangdian Wang, Zaiying Zhou, Dong Li, Tianying Wang, Sheng Yu, Pengkun Yang.

In particular, I would like to express my appreciation to professor Zaiying Zhou for her guidance, encouragement and inspiration, which are largely the driving force for my further study into the field of statistics. And also great thanks to Tianying Wang and Sheng Yu for advising my undergraduate research training. Their passion for research as a scholar really impressed me and I learned a lot from them.

I would like to thank Suqing Liu for his invaluable contributions to ZH translation in this project. And I would like to thank my classmates Mai Zhang, Hao Meng, Bufan Li, Dong Huang, and Yong Qin, who offered me help by discussing problems with me, proofreading the draft and tackling capstone projects together. Also I express my thanks to Xieheng Wang, Kaixing Liu, Ximing Li, Runyu Qi, Sifan Tao, Jiarui Chen, Yikun Li, Zeyu Zhang who supported me in the path of statistics. Also I thank my friends, including by not limited to: Shiyi Peng, Yiduo Xu, Peixin Weng, Haoran Zhang, Yuxin Huang, Guancheng Chen, Peng Zhang, Shaohan Wang, Weijing Yin, etc. Wish everyone all the best in the future.

Finally, deepest gratitude to my family. I would not become who I am today without their support.

目录

记号表

Here is a list of frequently used symbols and their meanings in this summary note. I sometimes use different notation from people used to in the literature, or from the notation in lecture notes of Tsinghua University Statistics minor courses, or simply follow convention in my Physics major.

Symbol listed here are notations 'by default' in this summary note. Specially defined symbols, especially those different from convention, would be explained in the text.

General Convention

Greek / Latin	Greek alphabet is used to describe the intrinsic property while Latin alphabet is used for estimator. e.g. $\sigma_X^2 = var(X)$, $\hat{\sigma}_X^2 = \hat{var}(X) = S^2$. e.g. θ is used to denote the parameter of distribution (family).

Used in Most Cases

$(\wedge i)$	Used in a sequence, means that we dropout the i^{th} item in the sequence indexed $1, 2, \ldots$
#	number of, or number of elements in, $\cdots$
$\bar{\cdot}$, $S_\cdot^2$, $S_\cdot$	Sample mean, sample variance, sample standard deviation. e.g. $\bar{X}$, S_X^2, S_X. **But in multivariate case I directly use S_X for sample covariance matrix.**
δ_{ij}, $\delta(x)$	Kronecker delta, Dirac delta.
$\equiv, :=$	is defined as
$\perp\!\!\!\perp$	independent of
$\mathbb{E}$, var, $corr$	r.v. Expectation, variance (in multivariate case, covariance matrix), and correlation coefficient. e.g. say $\mathbb{E}[X]$, $var(X)$. Sometimes subscript is used to clarify to which r.v. we are considering expectation. e.g. $\mathbb{E}_{X\sim f_X(x)}[\log X]$. Sometimes I simply use μ_X for expectation and σ_X^2 or Σ_X for variance, ρ_X for correlation coefficient.
$\mathbb{I}_\cdot$	Indicator Function, in which subscript is the set that the indicator function takes value 1.
$\mathbb{P}$	Probability measure
$\mathrm{Re}(\cdot)$, $\mathrm{Im}(\cdot)$	real part of, imaginary part of.

SSE, SSR, SST	For SSError, SSRegression, SSTotal.
$\mathscr{B}(\cdot)$	Backshift operator.
$\mathscr{F}[\cdot], \fallingdotseq, \risingdotseq$	Fourier transform; Inversed fourier transform.
$\vec{\cdot}$	is used to stress that $\cdot$ is a multi-dim vector.
$A = \arg\max_{\alpha} f(\alpha)$	A is the value of α that maximizes $f(\alpha)$.
H_0; H_1, H_a	Null hypothesis and alternative hypothesis.
N_α, $F_{m,n,\alpha}$, etc.	(Upper α) Quantile of distributions. I use N_α for quantiles of normal distribution instead of z_α.
$O(n), \mathcal{O}(n); o(n)$	Remainder in function series. Or used for complexity of algorithm
$X^{(1)}, \ldots, X^{(t)}$	Superscript with bracket is used in iteration algorithms to denote the value of this X in the t^{th} iteration.
$X_{(1)}, \ldots, X_{(n)}$	Order statistics.
i.i.d. or $\underset{i.i.d.}{\sim}$	independent identically distributed.

Used Frequently

$\mathcal{D}$	Dataset.
$\mathcal{L}$	Loss function.
$\underset{(p+1)\times 1}{\beta}$	Regression coefficient vector, with β_0 as the first element.
$\underset{n\times(p+1)}{X}$	Design matrix in regression, with a default intercept column.
L, l, ℓ	Likelihood, log-likelihood.
$M_\cdot(s)$	Moment generating function. Usually functions with s as argument are all generating functions, say $g(s)$ for probability generating function; $\phi(s)$ for characteristic function.
$S(\theta)$, $I(\theta)$, $J(\theta)$	Score Function, Fisher Information, Observed Information.
$T; \chi^2, F$	(Usually) T for test statistic; occasionally χ^2 for when testing is χ^2 test, F for when testing is F-test.
Z	If not used as r.v. In most cases used as the normalize constant in unnormalized distribution, say $f = \frac{1}{Z}\tilde{f} = \frac{1}{\int \tilde{f}(x)\,\mathrm{d}x}\tilde{f}$

Chapter. I　概率论部分

Instructor: Wanlu Deng

This part introduces some probabilitics tools used in statistics. Theory in this part is not based on measure theory, but is quite enough of (applied) statistics lectures.

Section 1.1　Some Important Distributions

X	$p_X(k)/f_X(x)$	$\mathbb{E}$	var	MGF
Bern(p)		p	pq	$q+pe^s$
$B(n,p)$	$C_n^k p^k(1-p)^{n-k}$	np	npq	$(q+pe^s)^n$
Geo(p)	$(1-p)^{k-1}p$	$\frac{1}{p}$	$\frac{q}{p^2}$	$\frac{pe^s}{1-qe^s}$
$H(n,M,N)$	$\frac{C_M^k C_{N-M}^{n-k}}{C_N^n}$	$n\frac{M}{N}$	$\frac{nM(N-n)(N-M)}{N^2(n-1)}$	
$P(\lambda)$	$\frac{\lambda^k}{k!}e^{-\lambda}$	λ	λ	$e^{\lambda(e^s-1)}$
$U(a,b)$	$\frac{1}{b-a}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	$\frac{e^{sb}-e^{sa}}{(b-a)s}$
$N(\mu,\sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ^2	$e^{\frac{\sigma^2 s^2}{2}+\mu s}$
$\epsilon(\lambda)$	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\frac{\lambda}{\lambda-s}$
$\Gamma(\alpha,\lambda)$	$\frac{\lambda^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\lambda x}$	$\frac{\alpha}{\lambda}$	$\frac{\alpha}{\lambda^2}$	$\left(\frac{\lambda}{\lambda-s}\right)^\alpha$
$B(\alpha,\beta)$	$\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$	
χ_n^2	$\frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}x^{\frac{n}{2}-1}e^{-\frac{x}{2}}$	n	$2n$	$(1-2s)^{-n/2}$
t_ν	$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(1+\frac{x^2}{\nu})^{-\frac{\nu+1}{2}}$	0	$\frac{\nu}{\nu-2}$	
$F_{m,n}$	$\frac{\Gamma(\frac{m+n}{2})}{\Gamma(\frac{m}{2})\Gamma(\frac{n}{2})}\frac{m^{\frac{m}{2}}n^{\frac{n}{2}}x^{\frac{m}{2}-1}}{(mx+n)^{\frac{m+n}{2}}}$	$\frac{n}{n-2}$	$\frac{2n^2(m+n-2)}{m(n-2)^2(n-4)}$	

Definition of PGF, MGF, CF see section 1.5 ~ page 27.

More Properties of χ^2, t, F see section 1.8.2 ~ page 34.

Relation between distributions and more properties see http://www.math.wm.edu/~leemis/chart/UDR/UDR.html. Distribution support in R. see https://CRAN.R-project.org/view=Distributions

Use the following command for all distributions supported in R. `stats::`.

```
?Distributions
```

Section 1.2 Probability and Probability Model

What is **Probability**? A 'belief' in 'what would happen. Different people may have different answers.

1.2.1 Sample Space and σ-Field

□ **Experiment and Sample Space**

Def. sample space Ω: The set of all possible outcomes of one particular **experiment** . Conducting the experiment would result in a result/sample point ω in sample space Ω. These results should be mutually exclusive, e.g. Tossing two coins simultaneously, the sample space is the set of all possible results

$$\Omega = \{(0,0),(0,1),(1,0),(1,1)\}, \quad \omega \in \Omega \tag{1.1}$$

On the sample space, the 'belief' in results happening is measured by probability $\mathbb{P}(\omega)$, $\omega \in \Omega$

Note: Randomness comes from the random result ω that an experiment generates.

□ **Event**

We may care about a conbination of some results, say 'at least one of the coin lands tails-up'. It's like a kind of 'structure' on sample space describing how we put results together to form **Events**. The definition is a σ-field(or a σ-algebra) $\mathscr{F}$ as a collection of some subsets of Ω, with properties:

- $\Omega \in \mathscr{F}$
- if $A \in \mathscr{F}$,then $A^{\complement} \in \mathscr{F}$
- if $A_n \in \mathscr{F}$, then $\bigcup_{n=1}^{\infty} A_n \in \mathscr{F}$

And $(\Omega, \mathscr{F})$ is a measurable space, on which we can select the events that we care about.

Events (and their properties) can be described in the language of set, e.g. for events $A, B \in \mathscr{F}$

- $A = B$ means they are the same event
- $A \cup B$ means one of them happens
- $A \cap B$ or AB means both happen

And some more complex ones

- $A \cup B = B \cup A$, $A \cap B = B \cap A$
- $A \cup (B \cup C) = A \cup B \cup C$, $A \cap (B \cap C) = A \cap B \cap C$
- $A \cap (B \cup C) = (A \cap B) \cup (B \cap C)$, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- $A \cup B = A + A^{\complement} \cap B$, $A = A \cap B + A \cap B^{\complement}$

Δ $(A \cup B)^{\complement} = A^{\complement} \cap B^{\complement}$, $(A \cap B)^{\complement} = A^{\complement} \cup B^{\complement}$

- $(\bigcup_{j=1}^{\infty} A_j)^{\complement} = \bigcap_{j=1}^{\infty} A_j^{\complement}$
- $(\bigcap_{j=1}^{\infty} A_j)^{\complement} = \bigcup_{j=1}^{\infty} A_j^{\complement}$

1.2.2 Axioms of Probability

$\mathbb{P}(\cdot) : \mathscr{F} \mapsto [0,1]$ is the probability measure (or probability function) defined on $(\Omega, \mathscr{F})$ describing the possibility that some event $A \in \mathscr{F}$ happens. Definition of probability $\mathbb{P}(A)$ in useful models:

$$\mathbb{P}(A) := \begin{cases} \dfrac{\#A}{\#\Omega} & \text{Classical Model} \\ \dfrac{m(A)}{m(\Omega)} & \text{Geometric Model} \end{cases} \tag{1.2}$$

Where $m(\cdot)$ is some measure of events in continuous space, say integral in Euclidean Space $\mathbb{R}^r$

$$m_{\mathbb{R}^r}(A) = \int_A \mathrm{d}x_1 \, \mathrm{d}x_2 \ldots \mathrm{d}x_r \tag{1.3}$$

□ **Basic Axioms of Probability Mearure** $\mathbb{P}(\cdot)$

- Non-negativity

$$\mathbb{P}(A) \geq 0 \qquad \forall A \in \Omega \tag{1.4}$$

- Normalization[1]

$$\mathbb{P}(\Omega) = 1 \tag{1.6}$$

- Countable Subadditivity

$$\mathbb{P}(A_1 \cup A_2 \cup \cdots) = \mathbb{P}(A_1) + \mathbb{P}(A_2) + \cdots \quad , (A_i \perp\!\!\!\perp A_j \quad \forall i \neq j) \tag{1.7}$$

  where 'countable subadditivity' means the events can be sequentially listed. e.g. $[0,1] = \bigcup_{x \in [0,1]} \{x\}$ is not countable, intuition:

$$1 = \mathbb{P}([0,1]) = \mathbb{P}\left(\bigcup_{x \in [0,1]} \{x\} \right) \neq \sum_{x \in [0,1]} \mathbb{P}(x) = 0 \tag{1.8}$$

Then $(\Omega, \mathscr{F}, \mathbb{P})$ is probability space, where Ω for experiment outcomes and randomness, $\mathscr{F}$ for events and their algebra, $\mathbb{P}$ for probability measure.

□ **Properties of Probability:**

[1] Note: In other sections when dealing with not-yet-normalized distribution (say in Bayesian statistics), I usually use Z as the normalize constant, following the tradition in statistical physics where Z is the partition function.

$$\mathbb{P} = \frac{1}{Z}\tilde{\mathbb{P}}, \quad Z = \int \tilde{\mathbb{P}} \tag{1.5}$$

- Addition Formula

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B) \tag{1.9}$$

- Monotonicity

$$\mathbb{P}(A) \leq \mathbb{P}(B) \quad \text{for } A \subset B \tag{1.10}$$

- Finite Subadditivity (Boole Inequality)

$$\mathbb{P}(\bigcup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} \mathbb{P}(A_i) \tag{1.11}$$

- Countable Subadditivity (σ-Subadditivity)

$$\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) \leq \sum_{i=1}^{\infty} \mathbb{P}(A_i) \tag{1.12}$$

- Inclusion-Exclusion Formula (Jordan Formula)

$$\mathbb{P}(\bigcup_{i=1}^{n} A_i) = \sum_{1 \leq i \leq n} \mathbb{P}(A_i) - \sum_{1 \leq i < j \leq n} \mathbb{P}(A_i \cap A_j) \tag{1.13}$$

$$+ \sum_{1 \leq i < j < k \leq n} \mathbb{P}(A_i \cap A_j \cap A_k) - \cdots \tag{1.14}$$

$$+ (-1)^{n-1} \mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_n) \tag{1.15}$$

Or in condensed notation:

$$\mathbb{P}(\bigcup_{i=1}^{n} A_i) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq j_1 < j_2 < \ldots < j_k \leq n} \mathbb{P}\left(A_{j_1} \cap A_{j_2} \cap \ldots \cap A_{j_k}\right) \tag{1.16}$$

- Borel-Cantelli Lemma

$$\sum_{n=1}^{\infty} \mathbb{P}(A_n) < \infty \Rightarrow \mathbb{P}(\lim_{n \to \infty} \sup A_n) = 0 \tag{1.17}$$

$$\sum_{n=1}^{\infty} \mathbb{P}(A_n) = \infty \Rightarrow \mathbb{P}(\lim_{n \to \infty} \sup A_n) = 1 \quad \text{if } A_i \text{ independent} \tag{1.18}$$

□ **An Example**

We have n different balls. Draw m times with replacement. What is the number of results regardless of order the balls drawn (e.g. $\{\text{red}, \text{red}, \text{black}\}$ is the same as $\{\text{red}, \text{black}, \text{red}\}$)?

The model is the same as we are 'voting' for n different balls, with total ballot ticket m. The m tickets are divided by $n-1$ plates (making them similar to ballot boxes), e.g. here's a $n = 4, m = 6$ vote corresponding to a result $\omega \in \Omega$:

$$\bullet || \bullet\bullet\bullet \,|\, \bullet\bullet \tag{1.19}$$

which the same as inserting plates sequentially and then cancel the order of plates:

$$\#\Omega = (m+1) * (m+2) \dots (m+n-1) \Big/ (n-1)! = \frac{(n+m-1)!}{m!(n-1)!} = \binom{n+m-1}{m} \tag{1.20}$$

(The idea of spacer plate is quite useful in dealing with some troublesome discrete cases, I think.)

表 1.1: #Ω of Sampling n balls m draw

	Replacement	
	With	Without
Ordered	n^m	A_n^m
Unordered	$\binom{n+m-1}{m}$	$\binom{n}{m}$

1.2.3 Conditional Probability

Motivation: To update the knowledge of probability measure.

Def. **Conditional Probability** of B given A:

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} \tag{1.21}$$

Actually it's a change of σ-field: $\Omega \to B$

$$\mathbb{P}\left(B|A\right) = \frac{m(B)}{m(A)} \tag{1.22}$$

□ **Application of conditional probability:**

- Multiplication Formula

$$\mathbb{P}(\bigcap_{i=1}^{n} A_i) = \mathbb{P}(A_1) \prod_{i=2}^{n} \mathbb{P}(A_i|A_1 \cap A_2 \cap \dots \cap A_{i-1}) \tag{1.23}$$

- Total Probability Theorem

$$\mathbb{P}(B) = \sum_{i=1}^{n} \mathbb{P}(A_i)\mathbb{P}(B|A_i) \tag{1.24}$$

where $\{A_i\}$ is a partition of Ω: $\Omega = \bigcup_i A_i,\ A_i \cap A_j = \delta_{ij}\emptyset$

(Actually just $B \subset \bigcup_i A_i$ is enough, similar for Bayes's rule)

- Bayes's Rule

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(A_i)\mathbb{P}(B|A_i)}{\sum_{j=1}^{n} \mathbb{P}(A_j)\mathbb{P}(B|A_j)}, \quad 1 \le i \le n \tag{1.25}$$

where $\{A_i\}$ is a partition of Ω: $\Omega = \bigcup_i A_i,\ A_i \cap A_j = \delta_{ij}\emptyset$

1.2.4 Independency

Statistical Independency is defined as:

$$A \perp\!\!\!\perp B : \mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B) \tag{1.26}$$

Properties

- Complement set and indepency

$$A \perp\!\!\!\perp B \Leftrightarrow A^{\complement} \perp\!\!\!\perp B \tag{1.27}$$

- Independency of multiple events

$$A_1 \perp\!\!\!\perp A_2 \perp\!\!\!\perp \ldots \perp\!\!\!\perp A_n \Leftrightarrow \mathbb{P}\left(A_{j_1} \cap A_{j_2} \cap \ldots \cap A_{j_k}\right) = \mathbb{P}\left(A_{j_1}\right)\mathbb{P}\left(A_{j_2}\right)\ldots\mathbb{P}\left(A_{j_k}\right) \tag{1.28}$$

$$\forall 1 \leq j_1 \leq j_2 \leq \ldots \leq j_k \leq n \quad \forall k \leq n, \quad n < \infty \tag{1.29}$$

Section 1.3 Random Variable and Distribution

Motivation: defining events is troublesome, and unhelpful to extract the key feature of events. A wise approach is to map samples & events to numbers $\Omega \mapsto \mathbb{R}^r$.

1.3.1 Random Variable

Def. Random Variable: a **function**/mapping X defined on sample space Ω, from Ω to some $\mathscr{X} \in \mathbb{R}$.

$$X(\omega) : \Omega \mapsto \mathscr{X} \in \mathbb{R} \tag{1.30}$$

Note: The mapping itself is non-random, the heart of randomness is still sample ω experimented.

Naturally X induces a mapping of probability measure

$$F_X : \mathscr{X} \mapsto \Omega \mapsto \mathbb{P} \tag{1.31}$$

To describe the mapping of probability, def. Cumulative Distribution Function (CDF). (Here $X(\omega)$ is still used to remind the origin of randomness, in most case we simply use X.)

$$F_X(x) = \mathbb{P}(X(\omega) \leq x) \tag{1.32}$$

- PMF:

$$p_X(x) = F_X(x^+) - F_X(x^-) \tag{1.33}$$

PDF:

$$f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x} \tag{1.34}$$

- Right-Continuity of CDF: A physical perspective is that PMF could be written as[2]

$$p_X(x) = \sum_{\tilde{x}\in\mathcal{X}} \mathbb{P}\left(X=\tilde{x}\right)\delta(x-\tilde{x}) \tag{1.35}$$

where discrete X take values in $\mathcal{X}$. In this way for any infinitesimal interval containing x: $\mathbb{I}_x \ni x$, we have

$$F_X(x^+) - F_X(x^-) = \int_{\mathbb{I}_x} p_X(x)\,\mathrm{d}x = \int_{\mathbb{I}_x} \sum_{\tilde{x}\in\mathcal{X}} \mathbb{P}\left(X=\tilde{x}\right)\delta(x-\tilde{x})\,\mathrm{d}x = \begin{cases} F_X(x^+) - F_X(x^-), & x\in\mathcal{X} \\ 0, & \text{others} \end{cases} \tag{1.36}$$

With such notation, in this note I sometimes ignore the difference between discrete cases / continuous cases.

- Representation of events: We could use random variable to express, say event A defined as

$$A := \{\omega : X(\omega) \le x\} \tag{1.37}$$

- Indicator function:

$$\mathbb{I}_{x\in A}(x) = \begin{cases} 1 & x\in A \\ 0 & x\notin A \end{cases} \tag{1.38}$$

- Convolution
  - $W = X + Y$

$$f_W(w) = \int_{-\infty}^{\infty} f_X(x) f_Y(w-x)\mathrm{d}x \tag{1.39}$$

  - $V = X - Y$

$$f_V(v) = \int_{-\infty}^{\infty} f_X(x) f_Y(x-v)\mathrm{d}x \tag{1.40}$$

  - $Z = XY$

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{1}{|x|} f_X(x) f_Y(\frac{z}{x})\mathrm{d}x \tag{1.41}$$

  Examples:
  - Poisson[3]

$$P(\lambda_1) + P(\lambda_2) \sim P(\lambda_1+\lambda_2) \tag{1.42}$$

  - Binomial

$$B(n_1,p) + B(n_2,p) \sim B(n_1+n_2,p) \tag{1.43}$$

  - Gamma / Exponential

$$\Gamma(\alpha_1,\lambda) + \Gamma(\alpha_2,\lambda) \sim \Gamma(\alpha_1+\alpha_2,\lambda) \tag{1.44}$$

  with

$$\varepsilon(\lambda) = \Gamma(1,\lambda) \tag{1.45}$$

[2] Definition of Dirac δ function see section 12.4.3 ~ page 334.

[3] More about Poisson Distribution / Poisson Process see section 12.1.4 ~ page 315

- More relations of distributions see http://www.math.wm.edu/~leemis/chart/UDR/UDR.html
- Relation between Poisson Process and Exponential and Uniform distribution see section 12.2.5 ~ page 319.

- Order Statistics[4]

  Def $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$ as order statistics of $\vec{X}$

$$g_{X_{(i)}} = n! \prod_i f(x_i) \qquad \text{for } x_1 < x_2 \cdots < x_n \tag{1.46}$$

  PDF of $X_{(k)}$

$$g_k(x_k) = nC_{n-1}^{k-1}[F(x_k)]^{k-1}[1 - F(x_k)]^{n-k} f(x_k) \tag{1.47}$$

- p-fractile

$$\xi_p = F^{-1}(p) = \inf\{x | F(x) \geq p\} \tag{1.48}$$

1.3.2 Random Vector

A general case of random variable. Its definition is similar

$$\vec{X}(\omega) : \Omega \mapsto \mathscr{X} \in \mathbb{R}^n \tag{1.49}$$

a n-dimension Random Vector $\vec{X} = (X_1, X_2, \ldots, X_n)$ defined on $(\Omega, \mathscr{F}, \mathbb{P})$.

CDF $F(x_1, \ldots, x_n)$ defined on $\mathbb{R}^n$:

$$F(x_1, \ldots, x_n) = \mathbb{P}(X_1 \leq x_1, \ldots, X_n \leq x_n) \tag{1.50}$$

Joint PDF of random vector:

$$f(x_1, \ldots, x_n) = \frac{\partial^n F(x_1, \ldots, x_n)}{\partial x_1 \ldots \partial x_n} \tag{1.51}$$

k-dimensional Marginal Distribution: For $1 \leq k < n$ and index set $S_k = \{i_1 \ldots, i_k\}$, distribution of $\vec{X} = (X_{i_1}, X_{i_2}, \ldots, X_{i_k})$

$$F_{S_k}(X_{i_1} \leq x_{i_1}, X_{i_2} \leq x_{i_2} \ldots, X_{i_k} \leq x_{i_k}) = \mathbb{P}(X_{i_1} \leq x_{i_1}, \ldots, X_{i_k} \leq x_{i_k}; X_{i_{k+1}}, \ldots, X_{i_n} \leq \infty) \tag{1.52}$$

Marginal distribution:

$$g_{S_k}(x_{i_1}, \ldots, x_{i_k}) = \int_{\mathbb{R}^{n-k}} f(x_1, \ldots, x_n) \mathrm{d}x_{i_{k+1}} \ldots \mathrm{d}x_{j_n} = \frac{\partial^{n-k} F(x_1, \ldots, x_n)}{\partial x_{i_{k+1}} \ldots \partial x_{i_n}} \tag{1.53}$$

Δ Function of r.v.

For $\vec{X} = (X_1, X_2, \cdots, X_n)$ with PDF $f(\vec{X})$ and define

$$\vec{Y} = (Y_1, Y_2, \cdots, Y_n) = \left(y_1(\vec{X}), y_2(\vec{X}), \cdots, y_n(\vec{X})\right) \tag{1.54}$$

[4] A relative object is Rank statistics, see section 2.4.6 ~ page 65.

with inverse mapping

$$\vec{X} = (X_1, X_2, \cdots, X_n) = \left(x_1(\vec{Y}), x_2(\vec{Y}), \cdots, x_n(\vec{Y})\right) \tag{1.55}$$

then

$$g(\vec{Y}) = f\left(x_1(\vec{Y}), x_2(\vec{Y}), \cdots, x_n(\vec{Y})\right) \left|\frac{\partial \vec{X}}{\partial \vec{Y}}\right| \mathbb{I}_{D_Y} \tag{1.56}$$

(Intuitively: $g(\vec{Y})\mathrm{d}\vec{Y} = \mathrm{d}\mathbb{P} = f(\vec{X})\mathrm{d}\vec{X}$)

Section 1.4 Expectation $\mathbb{E}$, Variance var and Covariance cov

Motivation: what would happen 'on average'?

Expectation and Variance of common distributions see section 1.1 ~ page 17.

1.4.1 Expection $\mathbb{E}(\cdot)$

Expectation of r.v. $g(X)$ def.:

$$\mathbb{E}[g(X)] = \begin{cases} \int_\Omega g(x) f_X(x)\mathrm{d}x = \int_\Omega g(x)\mathrm{d}F(x) \\ \sum_\Omega g(x) f_X(x) \end{cases} \tag{1.57}$$

Sometimes when there are more than 1 variables, say x, y, we would use notation $\mathbb{E}_X\left(g(X,Y)\right)$ or $\mathbb{E}_{X\sim f_X(x)}\left(g(X,Y)\right)$ to specify the variable and distribution to avoid confusion.

Note: For discrete r.v. the expectation always exists, but for continuous & unbounded r.v. the expectation might diverge, rigorously speaking:

$$\mathbb{E}\left[X\right] \exists : \int_{\mathbb{R}} |x| f(x)\,\mathrm{d}x < \infty \tag{1.58}$$

□ **Properties of Expectation $\mathbb{E}(\cdot)$:**

- Linearity of Expectation

$$\mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y) \tag{1.59}$$

- Conditional Expectation

$$\mathbb{E}(X|A) = \frac{\mathbb{E}(X\mathbb{I}_A)}{\mathbb{P}(A)} \tag{1.60}$$

Note: if take A as Y is also a r.v. then conditional expectation is actually a function of Y

$$\xi(Y) = \mathbb{E}(X|Y) = \int x f_{X|Y}(x)\mathrm{d}x \tag{1.61}$$

- Law of Total Expectation

$$\mathbb{E}_Y\left\{\mathbb{E}_X[g(X)|Y]\right\} = \mathbb{E}_X[g(X)] \tag{1.62}$$

- r.v.& Event

$$\mathbb{P}(A|X) = \mathbb{E}(\mathbb{I}_A|X) \Rightarrow \mathbb{E}[P(A|X)] = \mathbb{E}(\mathbb{I}_A) = \mathbb{P}(A) \tag{1.63}$$

- Conditional Expectation

$$\mathbb{E}\big[h(Y)g(X)|Y\big] = h(Y)\mathbb{E}[g(X)|Y] \tag{1.64}$$

1.4.2 Variance $var(\cdot)$

Variance of r.v. X:

$$var(X) = \mathbb{E}\big[(X - \mathbb{E}(X))^2\big] = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 \tag{1.65}$$

(sometimes denoted as σ_X^2.)

Another definition comes from the MMSE estimation,

$$var(X) = \min_c \mathbb{E}\left[(X - c)^2\right] \tag{1.66}$$

its solution is $c = \mathbb{E}[X]$. See section 12.4.1 ~ page 331 for more.

□ **Properties:**

- Linear combination of Variance

$$var(aX + b) = a^2 var(X) \tag{1.67}$$

- Conditional Variance

$$var(X|Y) = \mathbb{E}[X - \mathbb{E}(X|Y)]^2|Y \tag{1.68}$$

- Law of Total Variance

$$var(X) = \mathbb{E}[var(X|Y)] + var[\mathbb{E}(X|Y)] \tag{1.69}$$

Standard Deviation def. as :

$$\sigma_X = \sqrt{var(X)} \tag{1.70}$$

Then can construct **Standardization** of r.v.

$$X_{\text{sd}} = \frac{X - \mathbb{E}(X)}{\sqrt{var(X)}} \tag{1.71}$$

1.4.3 Covariance $cov(\cdot)$ and Correlation $corr(\cdot)$

Covariance of r.v. X and Y:

$$cov(X, Y) = \mathbb{E}\big[(X - \mu_X)(Y - \mu_Y)\big] = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) \tag{1.72}$$

And Correlation Coefficient

$$\rho_{X,Y} = corr(X, Y) = \frac{cov(X, Y)}{\sqrt{var(X)var(Y)}} \tag{1.73}$$

Remark: correlation $\nRightarrow$ cause and effect. Detail on causal effect topic see Chapter 11 ~ page 285.

Properties:

- Bilinear of Covariance

$$cov(X+Y,Z) = cov(X,Z) + cov(Y,Z) \tag{1.74}$$

$$cov(X,Y+Z) = cov(X,Y) + cov(X,Z) \tag{1.75}$$

- Variance and Covariance

$$var(X+Y) = var(X) + var(Y) + 2cov(X,Y) \tag{1.76}$$

- Covariance Matrix

Def $\Sigma = \mathbb{E}\big[(X-\mu)(X-\mu)^T\big] = \{\sigma_{ij}\}$ (where X should be considered as a column vector)

$$\Sigma = \begin{pmatrix} var(X_1) & cov(X_1,X_2) & \dots & cov(X_1,X_n) \\ cov(X_2,X_1) & var(X_2) & \dots & cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_n,X_1) & cov(X_n,X_2) & \dots & var(X_n) \end{pmatrix} \tag{1.77}$$

Attachment: Independence:

$$X_i \perp\!\!\!\perp X_j \Rightarrow \begin{cases} f(x_1,x_2,\cdots,x_n) = \prod f(x_i) \\ F(x_1,x_2,\cdots,x_n) = \prod F(x_i) \\ E(\prod X_i) = \prod E(X_i) \\ var(\sum X_i) = \sum var(X_i) \end{cases}, \qquad n < \infty \tag{1.78}$$

Section 1.5 PGF, MGF and C.F

Generating Function: Representation of $\mathbb{P}$ in function space. $\mathbb{P} \Leftrightarrow$ Generating Function.

1.5.1 Probability Generating Function

PGF: used for non-negative, integer X, which is the z-transformation of p_X

$$g(s) = \mathbb{E}(s^X) = \sum_{j=0}^{\infty} s^j \mathbb{P}(X=j), s \in [-1,1] \tag{1.79}$$

□ **Properties**

- $\mathbb{P}(X=k) = \dfrac{g^{(k)}(0)}{k!}$
- $\mathbb{E}(X) = g^{(1)}(1)$
- $var(X) = g^{(2)}(1) + g^{(1)}(1) - [g^{(1)}(1)]^2$
- For $X_1, X_2, \cdots, X_n$ independent with $g_i(s) = \mathbb{E}(s^{X_i})$, $Y = \sum_{i=1}^{n} X_i$, then

$$g_Y(s) = \prod_{i=1}^{n} g_i(s), s \in [-1,1] \tag{1.80}$$

- For X_i i.i.d with $\psi_i(s)=\psi(s)\equiv\mathbb{E}(s^{X_i})$, Y with $G(s)\equiv\mathbb{E}(s^Y)$, $W=X_1+X_2+\cdots+X_Y$,then

$$g_W(s)=G[\psi(s)] \tag{1.81}$$

- 2-Dimensional PGF of (X,Y)

$$g(s,t)=\mathbb{E}(s^Xt^Y)=\sum_{i=o}^{\infty}\sum_{j=0}^{\infty}\mathbb{P}_{(X,Y)}(X=i,Y=j)s^it^j,\quad s,t\in[-1,1] \tag{1.82}$$

1.5.2 Moment Generating Function

MGF: used for non-negative X, which is the Laplace transformation of f_X.

$$M_X(s)=\mathbb{E}(e^{sX})=\begin{cases}\sum_j e^{sx}\mathbb{P}(X=x_j)\\ \int_{-\infty}^{\infty}e^{sx}f_X(x)\mathrm{d}x\end{cases} \tag{1.83}$$

Properties

- MGF of $Y=aX+b$: $M_Y(s)=e^{sb}M(sa)$
- $\mathbb{E}(X^k)=M^{(k)}(0)$
- $\mathbb{P}(X=0)=\lim\limits_{s\to-\infty}M(s)$
- For $X_1,X_2,\cdots,X_n$ independent with $M_{X_i}(s)=\mathbb{E}(e^{sX_i})$, $Y=\sum\limits_{i=1}^{n}X_i$, then

$$M_Y(s)=\prod_{i=1}^{n}M_{X_i}(s) \tag{1.84}$$

1.5.3 Characteristic Function

C.F is actually the Fourier Transform (FT) of f_X.[5]

$$\phi(t)=\mathbb{E}(e^{itX})=\int_{-\infty}^{\infty}e^{itx}f_X(x)\mathrm{d}x \tag{1.85}$$

Properties

- if $E(|X|^k)<\infty$,then

$$\phi^{(k)}(t)=i^k\mathbb{E}(X^ke^{itX})\qquad\phi^{(k)}(0)=i^k\mathbb{E}(X^k) \tag{1.86}$$

- For $X_1,X_2,\cdots,X_n$ independent with $\phi_{X_i}(t)=\mathbb{E}(e^{itX_i})$, $Y=\sum\limits_{i=1}^{n}X_i$, then

$$\phi_Y(t)=\prod_{i=1}^{n}\phi_{X_i}(t) \tag{1.87}$$

- Inverse (Fourier) Transform

$$f(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-itx}\phi(t)\mathrm{d}t \tag{1.88}$$

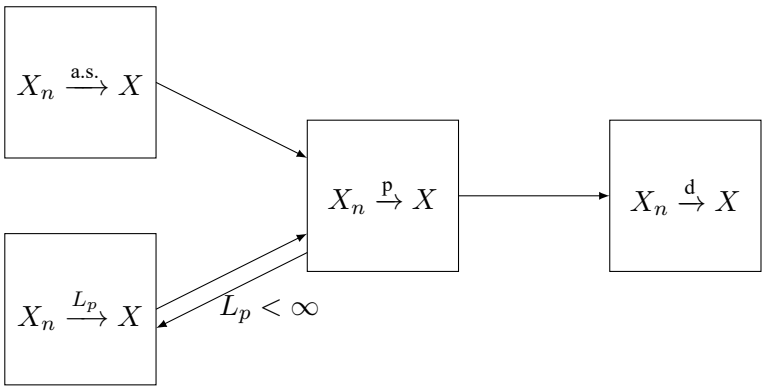
[5]More about FT see section 12.4.3 ~ page 334.

Section 1.6 Convergence and Limit Distribution

1.6.1 Convergence Mode

$$\begin{cases} \text{Convergence in Distribution} & X_n \xrightarrow{\text{d}} X : \lim\limits_{n\to\infty} F_n(x) = F(x) \\ \text{Convergence in Probability} & X_n \xrightarrow{\text{p}} X : \lim\limits_{n\to\infty} \mathbb{P}(|X_n - X) \geq \varepsilon) = 0\,, \forall \varepsilon > 0 \\ \text{Almost Sure Convergence} & X_n \xrightarrow{\text{a.s.}} X : \mathbb{P}(\lim\limits_{n\to\infty} X_n = X) = 1 \\ L_p \text{ Convergence} & X_n \xrightarrow{L_p} X : \lim\limits_{n\to\infty} \mathbb{E}(|X_n - X|^p) = 0 \end{cases} \tag{1.89}$$

Relations between convergence:



Note: L_2 convergence is also denoted m.s. (mean squared) convergence $\xrightarrow{\text{m.s.}}$.

Useful Theorem:

- Continuous Mapping Theorem: For continuous function $g(\cdot)$
  1. $X_n \xrightarrow{\text{a.s.}} X \Rightarrow g(X_n) \xrightarrow{\text{a.s.}} g(X)$
  2. $X_n \xrightarrow{\text{p}} X \Rightarrow g(X_n) \xrightarrow{\text{p}} g(X)$
  3. $X_n \xrightarrow{\text{d}} X \Rightarrow g(X_n) \xrightarrow{\text{d}} g(X)$
- Slutsky's Theorem: For $X_n \xrightarrow{\text{d}} X, Y_n \xrightarrow{\text{p}} c$
  1. $X_n + Y_n \xrightarrow{\text{d}} X + c$
  2. $X_n Y_n \xrightarrow{\text{d}} cX$
  3. $X_n / Y_n \xrightarrow{\text{d}} X/c$
- Continuity Theorem for characteristic function:

$$\lim_{n\to\infty} \phi_n(t) = \varphi(t) \Leftrightarrow X_n \xrightarrow{\text{d}} X \tag{1.90}$$

1.6.2 Law of Large Number & Central Limit Theorem

- m.s. LLN: For X_i with $cov(X_i, X_j) = 0$, if $i \neq j$, and $\mathbb{E}[X_i] = \mu < \infty$

$$\frac{1}{n}\sum X_i \xrightarrow{L_2} \mathbb{E}[X_1] \tag{1.91}$$

- WLLN: For X_i i.i.d. $\sim f_X$, with $\mathbb{E}[X_i] = \mu < \infty$

$$\frac{1}{n}\sum X_i \xrightarrow{\text{p}} \mu \tag{1.92}$$

- SLLN: For X_i i.i.d. $\sim f_X$, with $\mathbb{E}[X_i] = \mu < \infty$

$$\frac{1}{n}\sum X_i \xrightarrow{\text{a.s.}} \mu \tag{1.93}$$

- CLT: For X_i i.i.d. $\sim f_X$, with $\mathbb{E}[X_i] = \mu < \infty$, $var(X_i) = \sigma^2 < \infty$

$$\frac{\sqrt{n}\left(\bar{X} - \mu\right)}{\sigma} \xrightarrow{\text{d}} N(0,1) \tag{1.94}$$

or in equivalent form

$$\frac{1}{\sigma\sqrt{n}}\sum (X_k - \mu) \xrightarrow{\text{d}} N(0,1) \tag{1.95}$$

$$\bar{X} \xrightarrow{\text{d}} N(\mu, \frac{\sigma^2}{n}) \tag{1.96}$$

□ *Proof:* Denote the characteristic function of $X \sim f_X(x)$ as $\phi_X(t) := \mathbb{E}\left[e^{itX}\right]$, with expectation $\mu := \mathbb{E}[X]$ and variance $\sigma^2 := var(X) = \mathbb{E}\left[X^2\right] - \mu^2$.

Define $Z = \dfrac{X - \mu}{\sigma}$ The taylor series of $\phi_Z(t)$ at $t = 0$ yields:

$$\phi_Z(t) = 1 - \frac{t^2}{2} + o(t^2) \tag{1.97}$$

The characteristic function of mean $\bar{Z} := \dfrac{1}{n}\sum_{i=1}^{n} Z_i = \dfrac{1}{n}\sum_{i=1}^{n} \dfrac{X_i - \mu}{\sigma}$ w.r.t. X_i i.i.d. $\sim f_X(x)$

$$\phi_{\bar{Z}}(t) = \mathbb{E}\left[e^{it\bar{Z}}\right] = \left[\phi_Z(\frac{t}{n})\right]^n = \left[1 - \frac{t^2}{2n^2}\right]^n \tag{1.98}$$

with $n \to \infty$ limit:[6]

$$\lim_{n\to\infty} \phi_{\bar{Z}}(t) = \lim_{n\to\infty}\left[1 - \frac{1}{n}\frac{t^2}{2n}\right]^n = e^{-\frac{t^2}{2n}} \Rightarrow \bar{Z} = \frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{d}} N(0, \frac{1}{n}) \tag{1.100}$$

□

- de Moivre-Laplace Theorem is a special case of CLT at $S_n \sim B(n,p)$

$$\mathbb{P}(k \le S_n \le m) \approx \Phi(\frac{m + 0.5 - np}{\sqrt{npq}}) - \Phi(\frac{k - 0.5 - np}{\sqrt{npq}}) \tag{1.101}$$

- Stirling Eqa. derived from CLT

$$\frac{\lambda^k}{k!}e^{-\lambda} \approx \frac{1}{\sqrt{\lambda}\sqrt{2\pi}}e^{-\frac{(k-\lambda)^2}{2\lambda}} \xrightarrow[\lambda=n]{k=n} n! \approx \sqrt{2\pi n}(\frac{n}{e})^n \sim O\left((\frac{n}{e})^n\right) \tag{1.102}$$

[6]Note: if use characteristic function of X_i directly, notice that

$$n\log\left(1 + \frac{at}{n} - \frac{bt^2}{2n^2}\right) = at - \left(b + a^2\right)\frac{t^2}{2n} + \mathcal{O}(\frac{1}{n^2}) \tag{1.99}$$

using the taylor series of $\log(1+\xi)$ at $\xi = 0$.

Section 1.7 Inequalities

- Cauchy-Schwarz Inequality

$$|\mathbb{E}(XY)| \leq \sqrt{\mathbb{E}(X^2)\mathbb{E}(Y^2)} \tag{1.103}$$

- Bonferroni Inequality

$$\mathbb{P}(\bigcup_{i=1}^{n} A_i) \geq \sum_{1\leq i\leq n} \mathbb{P}(A_i) + \sum_{1\leq i<j\leq n} \mathbb{P}(A_i \cap A_j) \tag{1.104}$$

- Markov Inequality

$$\mathbb{P}(|X| \geq \epsilon) \leq \frac{\mathbb{E}(|X|^\alpha)}{\epsilon^\alpha} \tag{1.105}$$

with $\alpha = 1$, and ε selected as a multiple of $\mathbb{E}[|X|]$:

$$\mathbb{P}(|X| \geq m\mathbb{E}[|X|]) \leq \frac{1}{m} \tag{1.106}$$

- Chebyshev Inequality

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq \epsilon) \leq \frac{var(X)}{\epsilon^2} \tag{1.107}$$

Chebyshev inequality is used to proof WLLN equation 1.92 ~ page 30

- Jensen Inequality: For convex function $h(x)$:[7]

$$\mathbb{E}[h(X)] \geq h(\mathbb{E}(X)) \tag{1.109}$$

Example of using Jensen Eqa. to proof some other inequalities:

– Non-negativity of Kullback-Leibler Divergence: For two distributions $f(\cdot)$ and $g(\cdot)$, the K-L Divergence is defined as

$$\mathrm{KL}(f\|g) := -\int f(x) \log \frac{g(x)}{f(x)}\, \mathrm{d}x \tag{1.110}$$

Take $h(\xi) := \log \xi$ a concave function for $\xi \in (0, \infty)$ and $Z := \dfrac{g(X)}{f(X)}$ with $X \sim f(x)$, then

$$\mathbb{E}\left(h(Z)\right) = \int_A (\log z)\, f_Z(z)\, \mathrm{d}z = \int_A \left(\log \frac{g(x)}{f(x)}\right) f(x)\, \mathrm{d}x \tag{1.111}$$

$$\leq h(\mathbb{E}\left(Z\right)) = \log \int_A z f_Z(z)\, \mathrm{d}z = \log \int_A \frac{g(x)}{f(x)} f(x)\, \mathrm{d}x = 0 \tag{1.112}$$

$$\Rightarrow -\int_A \log f(x) \frac{g(x)}{f(x)}\, \mathrm{d}x \geq 0 \tag{1.113}$$

[7] Or equivalently for concave function $\tilde{h}(x)$:

$$\mathbb{E}[\tilde{h}(X)] \leq \tilde{h}(\mathbb{E}(X)) \tag{1.108}$$

- Cantelli Inequality

$$\mathbb{P}\left(X-\mathbb{E}\left[X\right]\geq\lambda\right)\leq\frac{var(X)}{var(X)+\lambda^2} \tag{1.114}$$

with $\lambda=\sqrt{var(X)}:=\sigma$, we have

$$\begin{cases}\mathbb{P}\left(X\geq\mathbb{E}\left[X\right]+\sigma\right)\leq\dfrac{1}{2}\\ \mathbb{P}\left(X\leq\mathbb{E}\left[X\right]-\sigma\right)\leq\dfrac{1}{2}\end{cases} \tag{1.115}$$

i.e. difference between mean and median is upperly bounded by standard deviation

$$\left|\mathbb{E}\left[X\right]-\text{med}(X)\right|\leq\sigma \tag{1.116}$$

- Hoeffding Inequality: with independent r.v. sequence $X_i\in[a_i,b_i]$, and $S_n:=\sum_{i=1}^n X_i$

$$\mathbb{P}\left(\left|S_n-\mathbb{E}\left[S_n\right]\right|\geq\varepsilon\right)\leq 2\exp\left[-\frac{2\varepsilon^2}{\sum_{i=1}^n(b_i-a_i)^2}\right] \tag{1.117}$$

Or in equivalent form $\varepsilon=nt$

$$\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n\left(X_i-\mathbb{E}\left[X_i\right]\right)\right|\geq t\right)\leq 2\exp\left[-\frac{2n^2t^2}{\sum_{i=1}^n(b_i-a_i)^2}\right] \tag{1.118}$$

For special case of $[a_i,b_i]=[a,b],\forall i,\ |[a,b]|:=L$,

$$\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n\left(X_i-\mathbb{E}\left[X_i\right]\right)\right|\geq t\right)\leq 2\exp\left[-\frac{2nt^2}{L^2}\right] \tag{1.119}$$

The proof needs the Hoeffding Lemma: for $\mathbb{E}\left[Z\right]=0$ and $Z\in[a,b]$

$$\mathbb{E}\left[e^{tZ}\right]\leq\exp\left[\frac{t^2(b-a)^2}{8}\right],\quad\forall t \tag{1.120}$$

- McDiarmid Inequality: with independent r.v. sequence X_i, and a function $f(\,\cdot\,)$ with bounded difference c_i:

$$\left|f\left(X_1,\ldots,X_n\right)-f\left(X_1,\ldots,\tilde{X}_i,\ldots,X_n\right)\right|\leq c_i,\quad\forall i \tag{1.121}$$

we have McDiarmid inequality

$$\mathbb{P}\left(\left|f(X_1,\ldots,X_n)-\mathbb{E}\left[f(X_1,\ldots,X_n)\right]\right|\geq nt\right)\leq 2\exp\left[-\frac{2n^2t^2}{\sum_{i=1}^n c_i^2}\right] \tag{1.122}$$

Section 1.8 Multivariate Normal Distribution

General Case and more discussion see section 4.2.1 ~ page 123.

Distribution of Normal r.v. $X\sim N(\mu,\sigma^2)$:

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.123}$$

For $X_1, X_2, \cdots, X_n$ independent and $X_k \sim N(\mu_k, \sigma_k^2)$, $k = 1, \cdots, n$, $T = \sum_{k=1}^{n} c_k X_k$, ($c_k$ const), then

$$T \sim N(\sum_{k=1}^{n} c_k \mu_k, \sum_{k=1}^{n} c_k^2 \sigma_k^2) \tag{1.124}$$

Deduction in some special cases:

- Given $\mu_1 = \mu_2 = \cdots = \mu_n = \mu$, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_n^2 = \sigma^2$, i.e. X_k i.i.d., then

$$T \sim N(\mu \sum_{k=1}^{n} c_k, \sigma^2 \sum_{k=1}^{n} c_k^2) \tag{1.125}$$

- Further take $c_1 = c_2 = \cdots = c_n = \frac{1}{n}$, i.e. $T = \sum_{k=1}^{n} X_k/n = \bar{X}$, then

$$T = \bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \tag{1.126}$$

1.8.1 Linear Transform

First consider $\epsilon_1, \epsilon_2, \cdots, \epsilon_m$ i.i.d. $\sim N(0,1)$, $n \times 1$ const column vector $\vec{\mu}$, $n \times m$ const matrix $\boldsymbol{B} = \{b_{ij}\}$, def.$X_i = \sum_{j=1}^{m} b_{ij}\epsilon_j$, i.e.

$$\vec{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{pmatrix} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{pmatrix} + \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} = \boldsymbol{B}\vec{\varepsilon} + \vec{\mu} \tag{1.127}$$

We have: $\vec{X} \sim N(\vec{\mu}, \Sigma)$, where Σ, as defined in equation 1.77 ~ page 27 is

$$\Sigma = \mathbb{E}[(\vec{X} - \vec{\mu})(\vec{X} - \vec{\mu})^T] = \boldsymbol{B}\boldsymbol{B}^T = \begin{pmatrix} var(X_1) & cov(X_1, X_2) & \dots & cov(X_1, X_n) \\ cov(X_2, X_1) & var(X_2) & \dots & cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_n, X_1) & cov(X_n, X_2) & \dots & var(X_n) \end{pmatrix} = \{\Sigma_{ij}\} \tag{1.128}$$

Furthur Consider $\vec{Y} = (Y_1, \cdots, Y_n)^T$, $n \times n$ const square matrix $\mathbf{A} = \{\mathbf{a_{ij}}\}$ and def. $\vec{Y} = \mathbf{A}\vec{\mathbf{X}}$ i.e.

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \tag{1.129}$$

Then $\vec{Y} \sim N(\boldsymbol{A}\vec{\mu}, \boldsymbol{A}\Sigma\boldsymbol{A}^T)$

Special case: $X_1, \cdots, X_n$ i.i.d. $\sim N(\mu, \sigma^2)$, $\vec{X} = (X_1, \cdots, X_n)^T$,

$$\mathbb{E}(Y_i) = \mu \sum_{k=1}^{n} a_{ik} \tag{1.130}$$

$$var(Y_i) = \sigma^2 \sum_{k=1}^{n} a_{ik}^2 \tag{1.131}$$

$$cov(Y_i, Y_j) = \sigma^2 \sum_{k=1}^{n} a_{ik} a_{jk} \tag{1.132}$$

Specially when $\boldsymbol{A} = \{a_{ij}\}$ orthonormal, we have $Y_1, \cdots, Y_n$ independent

$$Y_i \sim N(\mu \sum_{k=1}^{n} a_{ik}, \sigma^2) \tag{1.133}$$

□ **Definition of Jointly Gaussian/Normal**

A random vector $\vec{X}$ is called jointly Gaussian if and only if any (finite) linear combination of $\vec{X}$ is still Gaussian (Normal)

$$\sum_{k=1}^{m} \alpha_k X_{i_k} \sim N(\cdot, \cdot),\ \forall \{\alpha_k\}_{k=1}^{m},\ \forall \{i_k\}_{k=1}^{m},\ \forall m \le n \tag{1.134}$$

Counter Example: $[X, Y]$ in which $X \sim N(0,1)$, $Y = -X$ is not jointly Gaussian.

1.8.2 Distributions of Function of Normal Variable: χ^2, t & F

Consider $X_1, X_2, \ldots, X_n$ i.i.d. $\sim N(0,1)$; $Y, Y_1, Y_2, \ldots, Y_m$ i.i.d. $\sim N(0,1)$

- χ^2 Distribution: Def. χ^2 distribution with degree of freedom n:

$$\xi = \sum_{i=1}^{n} X_i^2 \sim \chi_n^2 \tag{1.135}$$

PDF of χ_n^2:

$$g_n(x) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{\frac{n}{2}-1} e^{-x/2} \mathbb{I}_{x>0} \tag{1.136}$$

Properties

- $\mathbb{E}$ and var of $\xi \sim \chi_n^2$

$$\mathbb{E}(\xi) = n \qquad var(\xi) = 2n \tag{1.137}$$

- For independent $\xi_i \sim \chi_{n_i}^2$, $i = 1, 2, \ldots, k$:

$$\xi_0 = \sum_{i=1}^{k} \xi_i \sim \chi_{n_1+\ldots+n_k}^2 \tag{1.138}$$

- Denoted as $\Gamma(\alpha, \lambda)$:

$$\xi = \sum_{i=1}^{n} X_i^2 \sim \Gamma(\frac{n}{2}, \frac{1}{2}) = \chi_n^2 \tag{1.139}$$

- t Distribution: Def. t distribution with degree of freedom n:

$$T = \frac{Y}{\sqrt{\frac{\sum_{i=1}^{n} X_i^2}{n}}} = \frac{Y}{\sqrt{\xi/n}} \sim t_n \tag{1.140}$$

(Usually take ν instead of n as degree of freedom for t distribution)

PDF of t_ν:

$$t_\nu(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{1.141}$$

Denote: Upper α-fractile of t_ν, satisfies $\mathbb{P}(T \geq c) = \alpha$:

$$t_{\nu,\alpha} = \arg_c \mathbb{P}(T \geq c) = \alpha, \quad T \sim t_\nu \tag{1.142}$$

(Similar for N, χ_n^2 and $F_{m,n}$ etc.)

- F Distribution: Def. F distribution with degree of freedom m and n:

$$F = \frac{\sum_{i=1}^{m} Y_i^2/m}{\sum_{i=1}^{n} X_i^2/n} \sim F_{m,n} \tag{1.143}$$

PDF of $F_{m,n}$:

$$f_{m,n}(x) = \frac{\Gamma(\frac{m+n}{2}) m^{\frac{m}{2}} n^{\frac{n}{2}}}{\Gamma(\frac{m}{2})\Gamma(\frac{n}{2})} x^{\frac{m}{2}-1} (mx+n)^{-\frac{m+n}{2}} \mathbb{I}_{x>0} \tag{1.144}$$

Properties

  - If $Z \sim F_{m,n}$, then $\frac{1}{Z} \sim F_{n,m}$.
  - If $T \sim t_n$, then $T^2 \sim F_{1,n}$
  - $F_{m,n,1-\alpha} = \frac{1}{F_{n,m,\alpha}}$

□ **Some useful Lemma (uesd in statistic inference, see section 2.3.3 ~ page 54):**

- For $X_1, X_2, \ldots, X_n$ independent with $X_i \sim N(\mu_i, \sigma_i^2)$, then

$$\sum_{i=1}^{n} \left(\frac{X_i - \mu_i}{\sigma_i}\right)^2 \sim \chi_n^2 \tag{1.145}$$

- For $X_1, X_2, \ldots, X_n$ i.i.d.$\sim N(\mu, \sigma^2)$, then

$$T = \frac{\sqrt{n}(\bar{X} - \mu)}{S} \sim t_{n-1} \tag{1.146}$$

For $X_1, X_2, \ldots, X_m$ i.i.d.$\sim N(\mu_1, \sigma^2)$, $Y_1, Y_2, \ldots, Y_n$ i.i.d.$\sim N(\mu_2, \sigma^2)$,
denote sample pooled variance $S_\omega^2 = \frac{(m-1)S_1^2 + (n-1)S_2^2}{m+n-2}$, then

$$T = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{S_\omega} \cdot \sqrt{\frac{mn}{m+n}} \sim t_{m+n-2} \tag{1.147}$$

- For $X_1, X_2, \ldots, X_m$ i.i.d.$\sim N(\mu, \sigma^2)$, $Y_1, Y_2, \ldots, Y_n$ i.i.d.$\sim N(\mu_2, \sigma^2)$, then

$$T = \frac{S_1^2}{S_2^2}\frac{\sigma_2^2}{\sigma_1^2} \sim F_{m-1,n-1} \tag{1.148}$$

- For $X_1, X_2, \ldots, X_n$ i.i.d. $\sim \epsilon(\lambda)$, then

$$2\lambda n\bar{X} = 2\lambda \sum_{i=1}^{n} X_i \sim \chi_{2n}^2 \tag{1.149}$$

Remark: for $X_i \sim \epsilon(\lambda) = \Gamma(1, \lambda) \Rightarrow 2\lambda \sum_{i=1}^{n} X_i \sim \Gamma(n, 1/2) = \chi_{2n}^2$.

Chapter. II　统计推断部分

Instructor: Jiangdian Wang

Statistical Inference: Given sample $X = (x_1, x_2, \ldots, x_n)$, we want to estimate some features of the population. This part focus on parametric statistical inference, thus our task is to estimate/testing parameters.

□ **Example of statistical inference**

- Sample item x_i, estimate its mean and variance
- Sample item $x_i = (\vec{x_i}, y_i)$, use multivariate linear model $Y \sim \vec{X}'\beta + \beta_0$, estimate slope & intercept β and variance σ^2

□ **Two main tasks of Statistical Inference**

- Parameter Estimation
  - Point Estimation: section 2.2 ~ page 43
  - Interval Estimation: section 2.3 ~ page 52
- Hypothesis Testing: section 2.4 ~ page 57

▷ R. **Code**

Example data x, y, df used in this section:

```r
set.seed(42)
x <- rnorm(n = 50, mean = 2, sd = 2)
y <- 0.5*x + rnorm(n = 50, mean = 2.1, sd = 2.1)
df <- data.frame(x=x,y=y)
```

Section 2.1　Statistical Model and Statistics

Random sample comes from population X. In parametric model case, we have population distribution family:

$$\mathscr{F} = \{f(x;\vec{\theta})|\vec{\theta} \in \Theta\} \tag{2.1}$$

where parameter $\vec{\theta}$ reflect some quantities of population (e.g. mean, variance, etc.), each $\vec{\theta}$ corresponds to a distribution of population X.

Sample space: Def. as $\mathscr{X} = \{\{x_1, x_2, \ldots, x_n\}, \forall x_i\}$, then $\{X_i\} \in \mathscr{X}$ is random sample from population $X \sim f(x; \vec{\theta})$.

2.1.1 Statistics

Statistic(s): function of random sample $\vec{T}(X_1, X_2, \ldots, X_n)$, **but not a function of parameter**.[1]

□ **Some useful statistics, e.g.**

- Sample mean (Consider X_i i.i.d.)

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.2}$$

- Sample variance

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \tag{2.3}$$

- Sample moments
  - Origin moment

$$a_{n,k} = \frac{1}{n}\sum_{i=1}^{k} X_i^k \qquad k = 1, 2, 3, \ldots \tag{2.4}$$

  - Center moment

$$m_{n,k} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^k \qquad k = 2, 3, 4, \ldots \tag{2.5}$$

- Pearson's Correlation Coefficient r

$$r_{X,Y} = \hat{cov}(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2.6}$$

  Multivariate version see equation 4.24 ~ page 117

- Order statistics

$$(X_{(1)}, X_{(2)}, \ldots, X_{(n)}), \text{ for } X_{(1)} \le X_{(2)} \le \ldots \le X_{(n)} \tag{2.7}$$

- Sample p-fractile

$$m_p = X_{(m)}, \quad m = \lfloor (n+1)p \rfloor \tag{2.8}$$

- Sample coefficient of variation

$$\hat{\nu} = \frac{S}{\bar{X}} \tag{2.9}$$

- Skewness and Kurtosis

$$\hat{g}_1 = \frac{m_{n,3}}{m_{n,2}^{3/2}} \qquad \hat{g}_2 = \frac{m_{n,4}}{m_{n,2}^2} - 3 \tag{2.10}$$

[1] Maybe to be more precise, the sample are drawn from the distribution $f(x; \vec{\theta})$, so naturally the data $\{X_i\}$ is related to parameters. Here a better description would be 'expression of statistics does not contain parameters explicitly'. And thus we could calculate the value to statistics as long as we have the sample data. Detail see Sampling Distribution.

▷ R. **Code**

R. code for some statistics

```r
# mean function
mean(x)
mean(df)
# variance / covariance
var(x)
var(x,y)
var(df)
cov(x,y)
cov(df)
# correlation
cor(x,y)
cor(df)
cov2cor(cov(df))
# moments
library('moments')
moments::moment(df, order = ORDER_OF_M, central = FALSE, na.rm =
    FALSE)
```

□ **Properties**

SamplingDistribution T is a function of random sample $\vec{X} = \{X_i\}$. Since the sample is drawn 'at random', then$T(\vec{X})$ certainly also has its own distribution (say $g_T(t)$) called **Sampling Distribution**.[2]

For X_i i.i.d. from $X \sim f(x)$ with population mean μ and variance σ^2

- Calculation of sample variance S^2

$$(n-1)S^2 = \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 \tag{2.11}$$

- $\mathbb{E}$ and var of $\bar{X}$ and S^2

$$\mathbb{E}(\bar{X}) = \mu \qquad var(\bar{X}) = \frac{\sigma^2}{n} \qquad \mathbb{E}(S^2) = \sigma^2 \tag{2.12}$$

Further if X_i i.i.d. from $X \sim N(\mu, \sigma^2)$ where μ and σ^2 unknown.

- Independence of $\bar{X}$ and S^2 [3]

$$\bar{X} \perp\!\!\!\perp S^2 \tag{2.13}$$

[2]Now recap the statement that 'statistic is not a function of parameter θ': Recall that random variable (the sample is a set of r.v.) X is a mapping, so $T: \Omega \mapsto \mathscr{X} \mapsto \mathbb{R}$ also. Parameter θ does not involve in the mapping process, instead it influence the sample probability $\mathbb{P}_\theta(\vec{X})$, and thus the distribution of statistics $g_T(t(\vec{X});\theta)$. Sometimes I use notation like $T(\vec{X};\theta)$ to remind me of this, but actually statistic should not contain θ (at least in its expression).

[3]A brief proof is here https://v1ncent19.github.io//texts/indepencyXS/

– Distribution of $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \tag{2.14}$$

– Distribution of $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1} \tag{2.15}$$

Comment: the independence here can explain the $n-1$ degree of freedom of χ^2_{n-1}

2.1.2 Exponential Family

Motivation: parametric statistical inference needs a (priorly assumed) distribution family, e.g. Normal $N(\mu, \sigma^2)$, Poisson $P(\lambda)$, Gamma $\Gamma(\alpha, \lambda)$, etc. Exponential Family is a framework to represent them in the same form Exponential Family can extract some key features of the distribution, and has some nice properties.

Def. $\mathscr{F}_\Theta = \{f(x;\vec{\theta}|\vec{\theta} \in \Theta)\}$ is **Exponential Family** if $f(x;\vec{\theta})$ has the form as

$$f(x;\vec{\theta}) = C(\vec{\theta})h(x)\exp\left[\sum_{i=1}^{k} Q_i(\vec{\theta})T_i(x)\right] \quad \vec{\theta} \in \Theta \tag{2.16}$$

Or equivalently express $c(\vec{\theta}) = \ln C(\vec{\theta})$:

$$f(x;\vec{\theta}) = h(x)\exp\left[\sum_{i=1}^{k} Q_i(\vec{\theta})T_i(x) + c(\vec{\theta})\right] \quad \vec{\theta} \in \Theta \tag{2.17}$$

Canonical Form: Take $Q_i(\vec{\theta}) = \varphi_i$, then $\vec{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_k) = (Q_1(\vec{\theta}), Q_2(\vec{\theta}), \ldots, Q_k(\vec{\theta}))$ is a transform from $\Theta \mapsto \Theta^*$, s.t. $\mathscr{F}$ has canonical form, i.e.

$$f(x;\vec{\varphi}) = C^*(\vec{\varphi})h(x)\exp\left[\sum_{i=1}^{k} \varphi_i T_i(x)\right] \quad \vec{\varphi} \in \Theta^* \tag{2.18}$$

Θ^* is canonical parameter space.

□ **Examples of Exponential Family**

- Normal Distribution $X \sim N(\mu, \sigma^2)$

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{x^2-2x\mu+\mu^2}{2\sigma^2}\right] \tag{2.19}$$

$$C(\theta) = \frac{1}{\sqrt{2\pi}\sigma} \tag{2.20}$$

$$Q_1(\theta) = -\frac{1}{2\sigma^2} \tag{2.21}$$

$$T_1(x) = x^2 \tag{2.22}$$

$$Q_2(\theta) = \frac{\mu}{\sigma^2} \tag{2.23}$$

$$T_2(x) = x \tag{2.24}$$

$$Q_3(\theta) = -\frac{\mu^2}{2\sigma^2} \tag{2.25}$$

$$T_3(x) = 1 \tag{2.26}$$

- Gamma Distribution $X \sim \Gamma(\alpha, \lambda)$

$$f(x;\alpha,\lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} \tag{2.27}$$

$$C(\theta) = \frac{\lambda^\alpha}{\Gamma(\alpha)} \tag{2.28}$$

$$Q_1(\theta) = -\lambda \tag{2.29}$$

$$T_1(x) = x \tag{2.30}$$

$$Q_2(\theta) = \alpha - 1 \tag{2.31}$$

$$T_2(x) = \log x \tag{2.32}$$

- Binomial Distribution $X \sim B(n,p)$

$$p(k;n,p) = \binom{n}{k} p^k (1-p)^{n-k} = \binom{n}{k} (1-p)^n \exp\left[k \log \frac{p}{1-p}\right] \tag{2.33}$$

$$C(\theta) = (1-p)^n \tag{2.34}$$

$$h(x) \underset{x=k}{=} \binom{n}{k} \tag{2.35}$$

$$Q_1(\theta) = \log \frac{p}{1-p} \tag{2.36}$$

$$T_1(x) \underset{x=k}{=} k \tag{2.37}$$

2.1.3 Sufficient and Complete Statistics

(Note: For simplification, the following parts denote $\vec{\theta}, \vec{T}, \ldots$ as $\theta, T, \ldots$ etc.) Now say we are trying to estimate θ by a statistic $T(\vec{X})$. We hope that $T(\vec{X})$ contains 'necessary/enough useful information' when estimating θ. [4]

► A **Sufficient Statistic** $T(\vec{X})$ for θ contains '*enough*' information of sample when infering θ, knowing more would not help us get a better estimation. i.e. the (conditional) distribution of sample given $T(\vec{X})$ is the same as that given the parameter.

$$f(\vec{X}; T(\vec{X})) = f(\vec{X}; T(\vec{X}), \theta) \tag{2.38}$$

Or, $T(\vec{X})$ condensedly stores all information about θ contained in sample $\vec{X}$.

Properties

- **Factorization Theorem** $T(\vec{X})$ is sufficient **if and only if** $f_{\vec{X}}(\vec{x};\theta) = f(\vec{x};\theta)$ can be written as

$$f(\vec{x};\theta) = g\left(t(\vec{x});\theta\right) h(\vec{x}) \tag{2.39}$$

- If $T(\vec{X})$ sufficient, then $T'(\vec{X}) = g[T(\vec{X})]$ also.(requires g single-valued and invertible)

[4] Intuition: 'Information' might be described by 'distribution of $T(\vec{X})$ for different θ'. i.e. the distribution family $\{g_T : \theta \in \Theta\}$ measures the performance of estimator.

- If $T(\vec{X})$ sufficient, then $[T, T_1]$ also.
- Sufficient statistic is **not** unique.
- Usually dimension of $\vec{T_\theta}$ and $\vec{\theta}$ equals.

► A **Complete Statistic** $T(\vec{X})$ for θ satisfies

$$\text{any } \phi(\,\cdot\,) \text{ with: } \mathbb{E}\left[\phi(T(\vec{X}))\right] = 0,\ \forall\theta,\ \text{must have } \mathbb{P}\left(\phi(\vec{X}) = 0\right) = 1 \forall\theta \tag{2.40}$$

Explanation: $T(\vec{X})$ as a function of sample, has its sampling distribution, say $T \sim g_T(t)$. 'Complete' is the description to the distribution family of $T(\vec{X})$: $\left\{g_T\left(t(x;\vec{\theta})\right) : \theta \in \Theta\right\}$. The above equation is rewritten as

$$\int \varphi(t) g_T(t)\,\mathrm{d}t = 0\,\forall\theta \underset{\text{compl stat}}{\Rightarrow} \varphi(\,\cdot\,) = 0 \text{ a.s. } \forall\,\theta \tag{2.41}$$

Another perspective: Recall that $\int \imath(u)\jmath(u)\,\mathrm{d}u$ is a kind of inner product $\langle \imath, \jmath \rangle$, the above statement is saying that: functional space of g_T, denoted $\underline{\text{span}\{g_T(t); \forall\theta\} \text{ is a complete function space}}$.

Another statement for complete statistic is that

$$\varphi(T) \neq 0\ \forall\theta \Rightarrow \mathbb{E}[\varphi(T(\vec{X}))] \neq 0 \tag{2.42}$$

Intuition: Not complete means $\exists\phi(\,\cdot\,)$, θ s.t. $\mathbb{E}\left[\phi(T(\vec{X}))\right] = 0$, and also $\exists$ another function $\tilde{\phi}(\,\cdot\,) = \phi(\,\cdot\,) + \text{const}$ so that $\mathbb{E}\left[\tilde{\phi}(T(\vec{X}))\right]$ can be any const $\rightarrow$ some information is **unnecessary**. So maybe complete means containing' *no extra*' information, to a certain degree.

Properties

- If $T(\vec{X})$ complete, then $T'(\vec{X}) = g[T(\vec{X})]$ also.(requires g measurable)
- A complete statistic does not always exists.

► A **Complete Sufficient Statistics**: with both sufficient and complete properties satisfied.

► A **Minimal Sufficient Statistics** $T(\vec{X})$ for θ contains 'just enough necessary' information about θ. Definition:

$$\forall \text{ sufficient statistic } \tilde{T}(\vec{X}),\ \exists\, q_{\tilde{T}}(\,\cdot\,),\ \text{s.t. } T(\vec{X}) = q_{\tilde{T}}(\tilde{T}(\vec{X})) \tag{2.43}$$

Intuition: $T(\vec{X})$ is a function of $\tilde{T}(\vec{X})$ suggests that T contains no more information than $\tilde{T}$. And if sufficient statistic T can be function of all sufficient statistics, then $T(\vec{X})$ contains 'enough and minimal information' about θ.

Properties

- Sufficient & Complete $\Rightarrow$ Minimal Sufficient ($\not\Leftarrow$)

  Sufficient as 'enough' + complete as 'no extra' = minimal sufficient as 'just enough'.
- A minimal sufficient statistic does **not** always exists.

– If minimal sufficient statistic exists, then any complete statistic is also minimal sufficient ⇒ complete & sufficient.

▶ An **Ancillary Statistic** $S(\vec{X})$ is a statistic whose distribution does not depend on θ

Basu Theorem: $\vec{X} = (X_1, X_2, \ldots, X_n)$ is sample from $\mathscr{F} = \{f(x;\theta), \theta \in \Theta\}$. $T(\vec{X})$ is a complete and minimal sufficient statistic, $S(\vec{X})$ is ancillary statistic, then $S(\vec{X}) \perp\!\!\!\perp T(\vec{X})$. Intuitively $S(\vec{X})$ contains no information about θ and minimal sufficient $T(\vec{X})$ contains all and necessary information about θ.

▶ Exponential family: For $\vec{X} = (X_1, X_2, \ldots, X_n)$ from exponential family with canonical form, i.e.

$$f(\vec{x};\theta) = C(\theta)h(\vec{x}) \exp\left[\sum_{i=1}^{k} \theta_i T_i(\vec{x})\right], \quad \theta \in \Theta \tag{2.44}$$

Then if $\Theta \in \mathbb{R}^k$ interior point exists, then $T(\vec{X}) = (T_1(\vec{X}), T_2(\vec{X}), \ldots, T_k(\vec{X}))$ is sufficient & complete statistic.

Section 2.2 Point Estimation

For parametric distribution family $\mathscr{F} = \{f(x,\theta), \theta \in \Theta\}$, random sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ from $\mathscr{F}$. $g(\theta)$ is a function defined on Θ.

Mission: use sample $\{X_i\}$ to estimate $g(\theta)$, called **Parameter Estimation**.

$$\text{Parameter Estimation} \begin{cases} \text{Point Estimation} & \surd \\ \text{Interval Estimation} \end{cases} \tag{2.45}$$

Point estimation: when estimating θ or $g(\theta)$, denote the estimator (defined on sample space $\mathscr{X}$) as

$$\hat{\theta}(\vec{X}) \xrightarrow{\text{estimates}} \theta \quad \text{or} \quad \hat{g}(\vec{X}) \xrightarrow{\text{estimates}} g(\theta) \tag{2.46}$$

Estimator is a statistic, with sampling distribution. In the following part we only give the expression for $\hat{\theta}(\vec{X}) \xrightarrow{\text{estimates}} \theta$ ($\hat{g}$ version is similar).

2.2.1 Optimal Criterion

Some nice properties of estimators (that we expect). They might not be satisfied simultaneously, e.g. we usually have to face trade-off between bias & precision.

- Unbiasedness: say $\hat{\theta}(\vec{X})$ or $\hat{g}(\vec{X})$ is unbiased estimator of θ or $g(\theta)$, if

$$\mathbb{E}(\hat{\theta}) = \theta, \quad \mathbb{E}(\hat{g}) = g(\theta) \tag{2.47}$$

Otherwise, say $\hat{\theta}$ or $\hat{g}$ is biased. Define **Bias**:

$$Bias(\hat{\theta}) := \mathbb{E}(\hat{\theta}) - \theta \tag{2.48}$$

in this way, an unbiased estimator is one with $Bias(\hat{\theta}) = 0$

Asymptotic unbiasedness with n as sample size:

$$\lim_{n\to\infty} \mathbb{E}(\hat{\theta}_n(\vec{X})) = \theta \tag{2.49}$$

- Efficiency: say $\hat{\theta}_1(\vec{X})$ is more efficient than $\hat{\theta}_2(\vec{X})$, if

$$var(\hat{\theta}_1) \leq var(\hat{\theta}_2) \quad \forall\theta \in \Theta \tag{2.50}$$

Can we find a estimator with minimum variance / the most efficient? See section 2.2.4 ~ page 47.

- Minimum Mean Squared Error (MMSE): Most efficient in the sense with bias-variance trade-off. More about Minimum MSE estimation see section 12.4.1 ~ page 331.

$$\text{MSE} = \mathbb{E}[(\hat{\theta} - \theta)^2] = \mathbb{E}\left[(\hat{\theta} - \mathbb{E}\left[\hat{\theta}\right] + \mathbb{E}\left[\hat{\theta}\right] - \theta)^2\right] = var(\hat{\theta}) + [Bias(\hat{\theta})]^2 \tag{2.51}$$

For unbiased estimator, i.e. $Bias(\hat{\theta}) = 0$, we have

$$\text{MSE} = \mathbb{E}[(\hat{\theta} - \theta)^2] = var(\hat{\theta}) \tag{2.52}$$

More about MMSE see section 12.4.1 ~ page 331.

- (Weak) Consistency

$$\lim_{n\to\infty} \mathbb{P}(|\hat{\theta}_n(\vec{X}) - \theta| \geq \varepsilon) = 0 \quad \forall\varepsilon > 0 \tag{2.53}$$

- Asymptotic Normality

$$\hat{\theta}_n - \theta \xrightarrow{d} N(0, \sigma^2_{\hat{\theta}}) \tag{2.54}$$

2.2.2 Method of Moments

Review: Population moments & Sample moments

$$\alpha_k = \mathbb{E}(X^k) \qquad \mu_k = \mathbb{E}[(X - \mathbb{E}(X))^k] \tag{2.55}$$

$$a_{n,k} = \hat{\alpha}_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k \qquad m_{n,k} = \hat{\mu}_k = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^k \tag{2.56}$$

Property: $a_{n,k}$ is the unbiased estimator of α_k.(while $m_{n,k}$ unually biased for μ_k)

For sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ from $\mathscr{F} = \{f(x; \theta, \theta \in \Theta)\}$, unknown parameter (or its function) $g(\theta)$ can be written as

$$g(\theta) = G(\alpha_1, \alpha_2, \ldots, \alpha_k; \mu_2, \mu_3, \ldots, \mu_l) \tag{2.57}$$

Then its **Moment Estimate** $\hat{g}(\vec{X})$ is

$$\hat{g}(\vec{X}) = G(a_{n,1}, a_{n,2}, \ldots, a_{n,k}; m_{n,2}, m_{n,3}, \ldots, m_{n,l}) \tag{2.58}$$

Example: coefficient of variation ν & skewness β_1

$$\hat{\nu} = \frac{S}{\bar{X}} \qquad \hat{\beta}_1 = \frac{m_{n,3}}{m_{n,2^{3/2}}} = \sqrt{n}\frac{\sum_{i=1}^{n}(X_i - \bar{X})^3}{\left[\sum_{i=1}^{n}(X_i - \bar{X})^2\right]^{\frac{3}{2}}} \tag{2.59}$$

□ **Note:**

- G may not have explicit expression.
- Moment estimate may not be unique.
- If $G = \sum_{i=1}^{k} c_i\alpha_i$ (linear combination of α, without μ), then $\hat{g}(\vec{X}) = \sum_{i=1}^{k} c_i a_{n,i}$ unbiased.

  Usually $\hat{g}(\vec{X})$ is asymptotically unbiased.
- For small sample, not so accurate.
- May not contain all the information about θ, i.e. may not be sufficient statistic.
- Do not require a statistic model, as long as you can express $G(\ldots)$.

2.2.3 Maximum Likelihood Estimation

For sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ with distribution $f(\vec{x};\theta)$ from $\mathscr{F} = \{f(x;\theta), \theta \in \Theta\}$, def. **Likelihood Function** $L(\theta;\vec{x})$, defined on Θ (as a function of θ)

$$L(\theta;\vec{x}) = f(\vec{x};\theta) \qquad \theta \in \Theta,\ \vec{x} \in \mathscr{X} \tag{2.60}$$

for X_i i.i.d. $\sim f(x;\theta)$ case:

$$L(\theta;\vec{x}) = \prod_{i=1}^{n} f(x_i;\theta) \tag{2.61}$$

Also def. log-likelihood function $\ell(\theta;\vec{x}) = \ln L(\theta;\vec{x})$.

A **Maximum Likelihood Estimator** $\hat{\theta}(\vec{X})$ for θ maximizes (or finds the upper bound) likelihood, or equivalently log-likelihood:

$$L(\hat{\theta};\vec{x}) = \sup_{\theta\in\Theta} L(\theta;\vec{x}) \Leftrightarrow \ell(\hat{\theta};\vec{x}) = \sup_{\theta\in\Theta} \ell(\theta;\vec{x}), \quad \vec{x} \in \mathscr{X} \tag{2.62}$$

□ **Identification of MLE**

- Differentiation: Fermat Lemma

$$\left.\frac{\partial L}{\partial\theta_i}\right|_{\theta=\hat{\theta}} = 0 \qquad \left.\frac{\partial^2 L}{\partial\theta_i\partial\theta_j}\right|_{\theta=\hat{\theta}} \text{negative definite} \qquad \forall i,j = 1,2,\ldots,k \tag{2.63}$$

- Graphing method.

- Numerically compute maximum.

□ **Properties**

- **Not Always** unbiased, an example is variance estimator, where

$$\hat{\sigma}^2_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{2.64}$$

- Invariance of MLE: If $\hat{\theta}$ is MLE of θ, then $h(\hat{\theta})$ is MLE of $h(\theta)$, where $h(\cdot)$ is an invertible function.

- MLE and Sufficiency: $T = T(X_1, X_2, \ldots, X_n)$ is a sufficient statistic of θ, if MLE of θ exists, say $\hat{\theta}$, then $\hat{\theta}$ is a function of T, i.e.

$$\hat{\theta} = \hat{\theta}(\vec{X}) = \hat{\theta}^*(T(\vec{X})) \tag{2.65}$$

- Asymptotic Normality:

$$\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{\text{d}} N(0, \sigma_\theta^2), \quad \sigma_\theta^2 = \frac{1}{\mathbb{E}_{\vec{X}}\left[[\frac{\partial}{\partial\theta}\ln f(\vec{X};\theta)]^2\right]} \tag{2.66}$$

  i.e.

$$\hat{\theta}_n \xrightarrow{\text{d}} N(\theta, \frac{\sigma_\theta^2}{n}) \tag{2.67}$$

  We will later see (in the next section) that σ_θ^2 is the inversed Fisher Information.

$$\hat{\theta}_n \xrightarrow{d} N(\theta, \frac{I(\theta)^{-1}}{n}) \tag{2.68}$$

□ **Comparison: MoM and MLE**

- MoM do not require statistic model; MLE need to know PDF.

- MoM is more robust than MLE.

□ **MLE in Exponential Family**

For sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ from canonical exponential family $\mathscr{F} = \{f(x;\theta), \theta \in \Theta\}$

$$f(x;\theta) = C(\theta)h(x)\exp\left[\sum_{i=1}^{k}\theta_i T_i(x)\right] \quad \theta = (\theta_1, \ldots, \theta_k) \in \Theta \tag{2.69}$$

Likelihood function $L(\theta, \vec{x}) = \prod_{j=i}^{n} f(x_j;\theta)$ and log-likelihood function $l(\theta, \vec{x})$

$$L(\theta, \vec{x}) = C^n(\theta)\prod_{j=1}^{n}h(x_j)\exp\left[\sum_{i=1}^{k}\theta_i\sum_{j=1}^{n}T_i(x_j)\right] \tag{2.70}$$

$$\ell(\theta, \vec{x}) = n\ln C(\theta) + \sum_{j=1}^{n}\ln h(x_j) + \sum_{i=1}^{k}\theta_i\sum_{j=1}^{n}T_i(x_j) \tag{2.71}$$

Solution of MLE: (Require $\hat{\theta} \in \Theta$)

$$\frac{n}{C(\theta)}\frac{\partial C(\theta)}{\partial\theta_i}\bigg|_{\theta=\hat{\theta}} = -\sum_{j=1}^{n}T_i(x_j), \quad i = 1, 2, \ldots, k \tag{2.72}$$

2.2.4 Uniformly Minimum Variance Unbiased Estimator

Recall MSE: If $\hat{g}(\vec{X})$ is an estimator of $g(\theta)$,then MSE

$$\text{MSE}(\hat{g}(\vec{X})) = \mathbb{E}[(\hat{g}(\vec{X}) - g(\theta))^2] = var(\hat{g}) + [Bias(\hat{g})]^2 \tag{2.73}$$

Note: Unbiased estimator (i.e. $Bias(\hat{g}) = 0$) is not unique; and not always exists. But now anyway for UMVUE we only consider the case that unbiased estimators of $g(\theta)$ exists, say $\hat{g}(\vec{X})$, then

$$\text{MSE}(\hat{g}(\vec{X})) = var(\hat{g}(\vec{X})) \tag{2.74}$$

If $\forall$ unbiased estimate $\tilde{g}(\vec{X})$, $\hat{g}$ satisfies

$$var[\hat{g}(\vec{X})] \leq var[\tilde{g}(\vec{X})] \tag{2.75}$$

Then $\hat{g}(\vec{X})$ is **Uniformly Minimum Variance Unbiased Estimator(UMVUE)** of $g(\theta)$

□ **How to determine UMVUE? (Which is not an easy task)**

1. **Zero Unbiased Estimate Method**

   Let $\hat{g}(\vec{X})$ be an unbiased estimator $\mathbb{E}\left[\hat{g}(\vec{X})\right] = 0$ with $var(\hat{g}) < \infty$. If $\forall$ other unbiased estimator $\mathbb{E}(\hat{l}(\vec{X})) = 0$, $\hat{g}$ holds that

   $$cov(\hat{g}, \hat{l}) = \mathbb{E}(\hat{g} \cdot \hat{l}) = 0, \quad \forall \theta \in \Theta \tag{2.76}$$

   Then $\hat{g}$ is a UMVUE of $g(\theta)$ (sufficient & necessary condition).

2. **Sufficient and Complete Statistic Method**. This method relies on two theorems:

   - **Rao-Blackwell Theorem**: For $T(\vec{X})$ sufficient statistic, $\hat{g}(\vec{X})$ unbiased estimate of $g(\theta)$, then

     $$h(T) = \mathbb{E}(\hat{g}(\vec{X})|T) \tag{2.77}$$

     is an unbiased estimate of $g(\theta)$ and $var(h(T)) \leq var(\hat{g})$.

     Remark:

     – A method to improve estimator.

     – A UMVUE has to be a function of sufficient statistic.

   - **Lehmann-Scheffé Theorem**: For $T(\vec{X})$ sufficient & complete, $\hat{g}(T()\vec{X})$ an unbiased estimate of $g(T(\theta))$, then $\hat{g}(T(\vec{X}))$ is the unique UMVUE.

     Using this theorem, we can actually construct UMVUE by conditional expectation: with any unbiased estimator $\tilde{g}(\vec{x})$ and sufficient & complete statistic $T(\vec{X})$, we have

     $$\mathbb{E}\left[\tilde{g}(\vec{x})|T(\vec{x})\right] \text{ the unique UMVUE for } g(\theta)$$

3. **Cramer-Rao Inequality**

   Core idea: determine a lower bound of $var(\hat{g})$.

   Consider $\theta = \theta$ (One dimension parameter); For $\{X_i\}$ i.i.d. $f(x, \theta)$: def.

- **Score function**: Reflects the steepness/slope of likelihood function.

$$S(\vec{x};\theta)=\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}=\frac{\partial \ell(\theta;\vec{x})}{\partial\theta}=\sum_{i=1}^{n}\frac{\partial \ln f(x_i;\theta)}{\partial\theta} \tag{2.78}$$

Property:[5]

$$\mathbb{E}[S(\vec{X};\theta)]=0 \tag{2.82}$$

- **Fisher Information**: Variance of $S(\vec{x};\theta)$, reflects the accuracy to conduct estimation, i.e. reflects information of statistic model that sample brings.[6]

$$I(\theta)=\mathbb{E}\left[\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\right)^2\right]=-\mathbb{E}\left[\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta^2}\right] \tag{2.89}$$

Consider $\mathscr{F}$ satisfies some regularity conditions(in most cases, regularity conditions do hold), then the lower bound of $var(\hat{g})$ satisfies **Cramer-Rao Inequality**:

$$var(\hat{g}(\vec{X}))\geq\frac{[g'(\theta)]^2}{nI(\theta)} \tag{2.90}$$

Special case: $g(\theta)=\theta$ then

$$var(\hat{\theta})\geq\frac{1}{nI(\theta)} \tag{2.91}$$

note:

- C-R Inequality determine a lower bound, not the infimum(i.e. UMVUE$\not\Rightarrow var(\hat{g}(\vec{X}))=\frac{[g'(\theta)]^2}{nI(\theta)}$).
- Take '=': Only some cases in Exponential family.

[5]Proof of $\mathbb{E}(S(\vec{x};\theta))=0$:

$$\mathbb{E}(S|\theta)=\int f(\vec{x};\theta)\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\,\mathrm{d}\vec{x} \tag{2.79}$$

$$=\int f(\vec{x};\theta)\frac{1}{f(\vec{x};\theta)}\frac{\partial f(\vec{x};\theta)}{\partial\theta}\,\mathrm{d}\vec{x} \tag{2.80}$$

$$=\frac{\partial}{\partial\theta}\int f(\vec{x};\theta)\,\mathrm{d}\vec{x}=\frac{\partial}{\partial\theta}1=0 \tag{2.81}$$

[6]Proof of $I(\theta)=\mathbb{E}\left[\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\right)^2\right]=-\mathbb{E}\left[\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta^2}\right]$:

$$0=\frac{\partial}{\partial\theta^T}\mathbb{E}(S|\theta) \tag{2.83}$$

$$=\int\frac{\partial}{\partial\theta^T}\left\{\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}f(\vec{x};\theta)\right\}\mathrm{d}\vec{x} \tag{2.84}$$

$$=\int\left\{\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta\partial\theta^T}f(\vec{x};\theta)+\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\frac{\partial f(\vec{x};\theta)}{\partial\theta^T}\right\}\mathrm{d}\vec{x} \tag{2.85}$$

$$=\int\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta\partial\theta^T}f(\vec{x};\theta)\,\mathrm{d}\vec{x}+\int\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta^T}f(\vec{x};\theta)\,\mathrm{d}\vec{x} \tag{2.86}$$

$$=\mathbb{E}\left(\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta\partial\theta^T}\right)+\mathbb{E}\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta^T}\right) \tag{2.87}$$

$$\Rightarrow\mathbb{E}\left(\frac{\partial^2 \ln f(\vec{x};\theta)}{\partial\theta\partial\theta^T}\right)=-\mathbb{E}\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta^T}\right) \tag{2.88}$$

- **Efficiency** $e_{\hat{g}}$: How good the estimator is.

$$e_{\hat{g}(\vec{X})}(\theta) = \frac{[g'(\theta)]^2/(nI(\theta))}{var(\hat{g}(\vec{X}))} \tag{2.92}$$

4. **Multi-Dimensional Cramer-Rao Inequality**

   ReDef. Fisher Information:

$$\boldsymbol{I}(\theta) = \{I_{ij}(\theta)\} = \{\mathbb{E}\left[\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial \theta_i}\right)\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial \theta_j}\right)\right]\} \tag{2.93}$$

   Then covariance matrix $\Sigma(\theta)$ satisfies **Cramer-Rao Inequality**

$$\Sigma(\theta) \succeq (n\boldsymbol{I}(\theta))^{-1} \tag{2.94}$$

   Note: '$\succeq$' means '$\geq$' holds for all diagonal elements, i.e.

$$var(\hat{\theta}_i) \geq \frac{I^*_{ii}(\theta)}{n}, \quad \forall\, i = 1, 2, \ldots, k \tag{2.95}$$

2.2.5 OLS, MoM, and MLE in Linear Regression

Note: More detailed knowledge see Chapter 3 ~ page 71 Linear Regression Analysis.

□ **Linear Regression Model (1-dimension case):**

$$y_i = \beta_0 + \beta_1 x_0 + \epsilon_i \tag{2.96}$$

where β_0, β_1 are regression coefficient, and ϵ_i are unknown random **error**.

Basic Assumptions (Guass-Markov Assumption):

$$\text{Zero-Mean: } \mathbb{E}(\epsilon_i|x_i) = 0 \tag{2.97}$$

$$\text{Homogeneity of Variance: } var(\epsilon_i) = \sigma^2 \tag{2.98}$$

$$\text{Independent: } \epsilon_i \text{ are i.d.} \tag{2.99}$$

further for MLE we need normality assumption

$$\epsilon_i \sim N(0, \sigma^2)$$

Mission: use data $\{(x_i, y_i)\}$ to estimate β_0, β_1(i.e. regression line), and error ϵ_i.

1. OLS (Ordinary Least Squares): Take β_0, β_1 so that MSE min, i.e. SSE min

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{2.100}$$

   (Express in Matrix Notation equation 2.118 ~ page 51, so that it can be generalized to multidimensional case) SSE can be expressed as the **Eucliean Distance** between $\{y_i\}$ and $\{\hat{\beta}_0 + \hat{\beta}_1 x_i\}$, i.e.

$$\hat{\beta} = \arg\min_{\beta} D(y, X\hat{\beta}) \tag{2.101}$$

i.e. $\hat{\beta}$ is the Projection of y onto hyperplane X, then

$$(X\hat{\beta})^T(y - X\hat{\beta}) = 0 \Rightarrow \hat{\beta} = (X^TX)^{-1}X^Ty \tag{2.102}$$

Solution for 1-D case:

$$\hat{\beta} = \begin{bmatrix}\hat{\beta}_0 \\ \hat{\beta}_1\end{bmatrix} = \begin{bmatrix}\bar{y} - \hat{\beta}_1\bar{x} \\ \dfrac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\end{bmatrix} \tag{2.103}$$

So get regression line:$y = \hat{\beta}_0 + \hat{\beta}_1 x$

Def. Residuals

$$e_i = \hat{\epsilon}_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \tag{2.104}$$

Residuals can be used to estimate ϵ_i: $E[(\epsilon_i)^2] = \sigma^2$

$$\hat{\sigma^2} = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \tag{2.105}$$

2. MoM: Consider r.v. $\epsilon \sim f(\varepsilon; x, y, \beta_0, \beta_1)$, sample $\{\epsilon_i | \epsilon_i = y_i - \beta_0 - \beta_1 x_i\}$, then obviously

$$\bar{\epsilon} = \bar{y} - \beta_0 - \beta_1\bar{x} \tag{2.106}$$

Take moment estimate of ϵ, we have

$$\mathbb{E}(\epsilon_i) = 0 \qquad \mathbb{E}(\epsilon_i x_i) = 0 \text{ (note that } \mathbb{E}(\epsilon|x) = 0) \tag{2.107}$$

$$\text{i.e.} \begin{cases} \dfrac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0 \\ \dfrac{1}{n}\sum_{i=1}^{n} x_i(y_i - \beta_0 - \beta_1 x_i) = 0 \end{cases} \tag{2.108}$$

Solution:

$$\begin{cases} \hat{\beta}_0 &= \bar{y} - \beta_1\bar{x} \\ \hat{\beta}_1 &= \dfrac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \end{cases} \tag{2.109}$$

(the same as OLS estimation)

Moment estimate of σ^2

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \tag{2.110}$$

3. MLE: Assume $\epsilon_i \sim N(0, \sigma^2)$, then $y_i|x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$. Get likelihood function:

$$L(\beta_0, \beta_1, \sigma^2; x_1, \ldots, x_n, y_1, \ldots, y_n) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)}{2\sigma^2}\right] \tag{2.111}$$

Log-likelihood:

$$\ell(\beta_0, \beta_1, \sigma^2; x_1, \ldots, x_n, y_1, \ldots, y_n) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \tag{2.112}$$

MLE, use Fermat Lemma:

$$\begin{cases} \dfrac{\partial \ell}{\partial \beta_0} = 0 & \Rightarrow -\dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0 \\ \dfrac{\partial \ell}{\partial \beta_1} = 0 & \Rightarrow -\dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{n} x_i(y_i - \beta_0 - \beta_1 x_i) = 0 \\ \dfrac{\partial \ell}{\partial \sigma^2} = 0 & \Rightarrow -\dfrac{n}{2}\dfrac{1}{\sigma^2} + \dfrac{1}{2(\sigma^2)^2}\displaystyle\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0 \end{cases} \tag{2.113}$$

Solution:

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x} \tag{2.114}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.115}$$

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \tag{2.116}$$

□ **Linear Regression Model (Multi-dimension case):**

Detailed derivation see section 3.3 ~ page 81

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i \tag{2.117}$$

Denote: $\vec{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$, $\vec{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ip})$, then for each i: $y_i = x_i^T \beta + \epsilon_i$

Further denote: Matrix form:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix} = X\vec{\beta} + \vec{\epsilon} \tag{2.118}$$

Basic Assumptions: Gauss-Markov Assumptions and Normal Assumption (for MLE).

Residuals:

$$e_i = \hat{\epsilon}_i = y_i - \hat{y}_i = y_i - x_i^T \hat{\beta} \tag{2.119}$$

Def. Error Sum of Squares (SSE)

$$\text{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}(y_i - x_i^T \hat{\beta})^2 \tag{2.120}$$

Estimator exists and unique:($\hat{\sigma}^2$ is after bias correction)

$$\begin{aligned} \hat{\beta} &= (X^T X)^{-1} X^T y \\ \hat{\sigma}_n^2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T \hat{\beta})^2 \\ \hat{\sigma}^2 &= \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - x_i^T \hat{\beta})^2 \end{aligned} \tag{2.121}$$

▷ R. **Code**

Example of linear regression model $Y = \beta_0 + x\beta + \varepsilon$

```r
lmfit <- lm(y~x, df)
summary(lmfit)
```

2.2.6 Kernel Density Estimation

Given random sample $\{X_i\}$. Def. Empirical CDF.

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{(-\infty, X_i]}(x) \tag{2.122}$$

Problem: Overfitting when getting $\hat{f}$. Solution: Using **Kernel Estimate**, replace $I_{(-\infty,x]}(\cdot)$ with Kernel function $K(\cdot)$, then

$$\hat{f}_n(x) = \frac{F_n(x+h_n) - F_n(x-h_n)}{2h_n} = \frac{1}{nh_n}\sum_{i=1}^{n} K(\frac{x - X_i}{h_n}) \tag{2.123}$$

where h_n is **bandwidth**. Take proper kernel function K to get estimate of f.

Kernel density estimation can be considered as a convolution $\otimes$ of sample $\{X_i\}$ and kernel function $K(\cdot)$.

$$\hat{f}_K = \frac{1}{n}\sum_{i=1}^{n} \delta(x - X_i) \otimes K(x) \tag{2.124}$$

□ **Useful Kernel Functions**

$$K(x) := \begin{cases} \mathbb{I}_{[-\frac{1}{2},\frac{1}{2}]}, & \text{Square Kernel} \\ (1-|x|)\mathbb{I}_{[-1,1]}, & \text{Triangle Kernel} \\ \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, & \text{Gaussian Kernel} \\ \frac{1}{\pi(1+x^2)}, & \text{Cauchy Kernel} \\ \frac{1}{2\pi}\text{sinc}^2\frac{x}{2} = \frac{1}{2\pi}\left(\frac{\sin x/2}{x/2}\right)^2, & \text{sinc Kernel} \end{cases} \tag{2.125}$$

▷ R. **Code**

Plot kernel density in R.

```r
plot(density(x, kernel = KERNEL_TO_USE))
```

Section 2.3 Interval Estimation

$$\text{Parameter Estimation} \begin{cases} \text{Point Estimation} \\ \text{Interval Estimation} \quad \surd \end{cases} \tag{2.126}$$

Interval Estimation: to estimate $g(\theta)$, give **two** estimators $\hat{g}_1(\vec{X})$, $\hat{g}_2(\vec{X})$ defined on $\mathscr{X}$ as the two ends of interval (i.e. give an interval $[\hat{g}_1(\vec{X}),\ \hat{g}_2(\vec{X})]$), then random interval $[\hat{g}_1(\vec{X}),\ \hat{g}_2(\vec{X})]$ is an **Interval Estimation** of $g(\theta)$.

> Δ **NOTE:** Here $g(\theta)$ is the parameter, which is fixed, while confidence interval, as a function of data, is random. So all the probabilities discussed below are **Probability that the interval covers the true value**, rather that the true value falls in the interval. There is a huge difference.[a]
>
> [a] A good example: Consider a bi-classification task into $\uparrow$ or $\downarrow\}$. A confidence interval algorithm can randomly produce $\{\uparrow,\downarrow\}$ 19 times, and $\emptyset$ 1 time. This is still a 95% cofidence interval algorithm (covers true label 19 in 20), but true label falls in $\{\uparrow,\downarrow\}$ with pr 1, and in $\emptyset$ with pr 0.

2.3.1 Confidence Interval

How to judge an interval estimation?

- Reliability

$$\mathbb{P}_{\hat{g}_1,\hat{g}_2}([\hat{g}_1,\hat{g}_2] \ni g(\theta)) \tag{2.127}$$

- Precision

$$\mathbb{E}(\hat{g}_2 - \hat{g}_1) \tag{2.128}$$

Trade off: (in most cases) Given a level of reliability, find an interval with the highest precision with reliability above the level.

□ **For a given** $0 < \alpha < 1$**, if**

$$\mathbb{P}(\hat{g}_1 \le g(\theta) \le \hat{g}_2) \ge 1 - \alpha \tag{2.129}$$

then $[\hat{g}_1, \hat{g}_2]$ is a **Confidence Interval** for $g(\theta)$, with **Confidence Level** $1 - \alpha$.

Confidence Coefficient:

$$\inf_{\forall\theta\in\Theta} \mathbb{P}(\theta \in \mathrm{CI}) \tag{2.130}$$

Other cases:

- **Confidence Limit**: (One-way) Upper/Lower Confidence Limit

$$\mathbb{P}(g \le \hat{g}_U) \ge 1 - \alpha \tag{2.131}$$

$$\mathbb{P}(\hat{g}_L \le g) \ge 1 - \alpha \tag{2.132}$$

- **Confidence Region**: For high dimensional parameters $\vec{g} = (g_1, g_2, \ldots, g_k)$

$$\mathbb{P}(\vec{g} \in S(\vec{X})) \ge 1 - \alpha \quad \forall\theta \in \Theta \tag{2.133}$$

Mission: Determine $\hat{g}_1, \hat{g}_2$.

2.3.2 Pivot Variable Method

Idea: Based on point estimation, construct a new variable and thus find the interval estimation.

Def. **Pivot Variable** T, satisfies:

- Expression of T contains θ (thus T is not a statistic).
- Distribution of T independent of θ.[7]

In different cases, construct different pivot variable, usually base on sufficient statistics and transform.

Knowing a proper pivot variable $T = T(\hat{\varphi}, g(\theta)) \sim f$, ($f$ is some distribution independent of θ), $\hat{\varphi}$ is a sufficient statistic), then we can take T satisfies:

$$\mathbb{P}(f_{1-\frac{\alpha}{2}} \leq T \leq f_{\frac{\alpha}{2}}) = 1 - \alpha \tag{2.134}$$

Construct the inverse mapping of $T = T(\hat{\varphi}, g(\theta)) \rightleftarrows g(\theta) = T^{-1}(T, \hat{\varphi})$, we get

$$\mathbb{P}[T^{-1}(f_{1-\frac{\alpha}{2}}, \hat{\varphi}) \leq \hat{g} \leq T^{-1}(f_{\frac{\alpha}{2}}, \hat{\varphi})] = 1 - \alpha \tag{2.135}$$

Thus get a confidence interval for θ with confidence coefficient $1 - \alpha$.

2.3.3 Confidence Interval for Common Distributions

Some important properties of χ^2, t and F see section 1.8.2 ~ page 34.

1. Single normal population: $\vec{X} = \{X_1, X_2, \ldots, X_n\} \in \mathscr{X}$ i.i.d from Normal Distribution population $N(\mu, \sigma^2)$. Denote sample mean and sample variance:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad S_\mu = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2, \text{(for the case } \mu \text{ known)} \tag{2.136}$$

Estimating μ & σ^2: construction of pivot variable under different circumstances:

2. Double normal population: $\vec{X} = \{X_1, X_2, \ldots, X_m\}$ i.i.d. from $N(\mu_1, \sigma_1^2)$; $\vec{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ i.i.d. from $N(\mu_2, \sigma_2^2)$

Denote sample mean, sample variance and pooled sample variance:

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{n} X_i \quad S_X^2 = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \bar{X})^2 \qquad S_{\mu_1}^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \mu_1)^2, (\mu_1 \text{ known}) \tag{2.137}$$

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \qquad S_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2 \qquad S_{\mu_2}^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \mu_2)^2, (\mu_2 \text{ known}) \tag{2.138}$$

$$S_\omega^2 = \frac{(m-1)S_X^2 + (n-1)S_Y^2}{m+n-2} \tag{2.139}$$

[7] Comment: $T(X, \theta)$ is both function of sample X an parameter in statistics model. Note that X also depends on θ, but is fixed once we complete a sample.

Estimation	Pivot Variable	Confidence Interval
σ^2 known, estimate μ	$Z=\dfrac{\sqrt{n}(\bar{X}-\mu)}{\sigma}\sim N(0,1)$	$\left[\bar{X}-\dfrac{\sigma}{\sqrt{n}}N_{\frac{\alpha}{2}},\bar{X}+\dfrac{\sigma}{\sqrt{n}}N_{\frac{\alpha}{2}}\right]$
σ^2 unknown, estimate μ	$T=\dfrac{\sqrt{n}(\bar{X}-\mu)}{S}\sim t_{n-1}$	$\left[\bar{X}-\dfrac{S}{\sqrt{n}}t_{n-1,\frac{\alpha}{2}},\bar{X}+\dfrac{S}{\sqrt{n}}t_{n-1,\frac{\alpha}{2}}\right]$
μ known, estimate σ^2	$T=\dfrac{nS_\mu^2}{\sigma^2}\sim\chi_n^2$	$\left[\dfrac{nS_\mu^2}{\chi^2_{n,\frac{\alpha}{2}}},\dfrac{nS_\mu^2}{\chi^2_{n,1-\frac{\alpha}{2}}}\right]$
μ unknown, estimate σ^2	$T=\dfrac{(n-1)S^2}{\sigma^2}\sim\chi^2_{n-1}$	$\left[\dfrac{(n-1)S^2}{\chi^2_{n-1,\frac{\alpha}{2}}},\dfrac{(n-1)S^2}{\chi^2_{n-1,1-\frac{\alpha}{2}}}\right]$

(a) Estimating $\mu_1-\mu_2$:

When $\sigma_1^2\neq\sigma_2^2$ and unknown, estimate $\mu_1-\mu_2$: Behrens-Fisher Problem, remains unsolved[8], but we can deal with simplified cases.

Estimation	Pivot Variable	Confidence Interval
σ_1^2 & σ_2^2 known, estimate $\mu_1-\mu_2$	$Z=\dfrac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{\sqrt{\dfrac{\sigma_1^2}{m}+\dfrac{\sigma_2^2}{n}}}\sim N(0,1)$	$\left[\bar{X}-\bar{Y}-N_{\frac{\alpha}{2}}\sqrt{\dfrac{\sigma_1^2}{m}+\dfrac{\sigma_2^2}{n}},\ \bar{X}-\bar{Y}+N_{\frac{\alpha}{2}}\sqrt{\dfrac{\sigma_1^2}{m}+\dfrac{\sigma_2^2}{n}}\right]$
$\sigma_1^2=\sigma_2^2$ unknown, estimate $\mu_1-\mu_2$	$T=\dfrac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{S_\omega\sqrt{\dfrac{1}{m}+\dfrac{1}{n}}}\sim t_{m+n-2}$	$\left[\bar{X}-\bar{Y}-S_\omega t_{m+n-2,\frac{\alpha}{2}}\sqrt{\dfrac{1}{m}+\dfrac{1}{n}},\ \bar{X}-\bar{Y}+S_\omega t_{m+n-2,\frac{\alpha}{2}}\sqrt{\dfrac{1}{m}+\dfrac{1}{n}}\right]$
Welch's t-Interval (when m, n large enough)	$T=\dfrac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{\sqrt{\dfrac{S_X^2}{m}+\dfrac{S_Y^2}{n}}}\xrightarrow{\text{d}}N(0,1)$	$\left[\bar{X}-\bar{Y}-N_{\frac{\alpha}{2}}\sqrt{\dfrac{S_1^2}{m}+\dfrac{S_2^2}{n}},\ \bar{X}-\bar{Y}+N_{\frac{\alpha}{2}}\sqrt{\dfrac{S_1^2}{m}+\dfrac{S_2^2}{n}}\right]$

(b) Estimating $\dfrac{\sigma_1^2}{\sigma_2^2}$:

[8] An approximation is Welch-Satterthwaite equation. Detail see section 14.1.2 ~ page 361.

Estimation	Pivot Variable	Confidence Interval
μ_1, μ_2 known, estimate $\frac{\sigma_1^2}{\sigma_2^2}$	$F = \frac{S_{\mu_2}^2}{S_{\mu_1}^2}\frac{\sigma_1^2}{\sigma_2^2} \sim F_{n,m}$	$\left[\frac{S_{\mu_1}^2}{S_{\mu_2}^2}\frac{1}{F_{m,n,\frac{\alpha}{2}}}, \frac{S_{\mu_1}^2}{S_{\mu_2}^2}\frac{1}{F_{m,n,1-\frac{\alpha}{2}}}\right]$ or $\left[\frac{S_{\mu_1}^2}{S_{\mu_2}^2}F_{m,n,\frac{\alpha}{2}}, \frac{S_{\mu_1}^2}{S_{\mu_2}^2}F_{n,m,\frac{\alpha}{2}}\right]$
μ_1, μ_2 unknown, estimate $\frac{\sigma_1^2}{\sigma_2^2}$	$F = \frac{S_Y^2}{S_X^2}\frac{\sigma_1^2}{\sigma_2^2} \sim F_{n-1,m-1}$	$\left[\frac{S_X^2}{S_Y^2}\frac{1}{F_{m-1,n-1,\frac{\alpha}{2}}}, \frac{S_X^2}{S_Y^2}\frac{1}{F_{m-1,n-1,1-\frac{\alpha}{2}}}\right]$ or $\left[\frac{S_X^2}{S_Y^2}F_{m-1,n-1,\frac{\alpha}{2}}, \frac{S_X^2}{S_Y^2}F_{n-1,m-1,\frac{\alpha}{2}}\right]$

3. Non-normal population:

Estimation	Pivot Variable	Confidence Interval
Uniform Distribution: $\vec{X}$ i.i.d. from $U(0,\theta)$	$T = \frac{X_{(n)}}{\theta} \sim U(0,1)$	$\left[X_{(n)}, \frac{X_{(n)}}{\sqrt[n]{\alpha}}\right]$
Exponential Distribution: $\vec{X}$ i.i.d. from $\epsilon(\lambda)$	$T = 2n\lambda\bar{X} \sim \chi_{2n}^2$	$\left[\frac{\chi_{2n,1-\frac{\alpha}{2}}^2}{2n\bar{X}}, \frac{\chi_{2n,\frac{\alpha}{2}}^2}{2n\bar{X}}\right]$
Bernoulli Distribution: $\vec{X}$ i.i.d. from $B(1,\theta)$	$T = \frac{\sqrt{n}(\bar{X}-\theta)}{\sqrt{\bar{X}(1-\bar{X})}} \xrightarrow{\text{d}} N(0,1)$	$\left[\bar{X} - N_{\frac{\alpha}{2}}\sqrt{\frac{\bar{X}(1-\bar{X})}{n}}, \bar{X} + N_{\frac{\alpha}{2}}\sqrt{\frac{\bar{X}(1-\bar{X})}{n}}\right]$
Poisson Distribution: $\vec{X}$ i.i.d. from $P(\lambda)$	$T = \frac{\sqrt{n}(\bar{X}-\lambda)}{\sqrt{\bar{X}}} \xrightarrow{\text{d}} N(0,1)$	$\left[\bar{X} - N_{\frac{\alpha}{2}}\sqrt{\frac{\bar{X}}{n}}, \bar{X} + N_{\frac{\alpha}{2}}\sqrt{\frac{\bar{X}}{n}}\right]$

4. General Case: Use asymptotic normality of MLE to construct CLT for large sample. MLE of θ satisfies:

$$\sqrt{n}(\hat{\theta}^* - \theta) \xrightarrow{\text{d}} N(0, \frac{1}{I(\theta)}) \tag{2.140}$$

where $\hat{\theta}^*$ is MLE of θ. Replace $\frac{1}{I(\theta)}$ by $\sigma^2(\hat{\theta}^*)$, then

$$T = \frac{\sqrt{n}(\hat{\theta}^* - \theta)}{\sigma(\hat{\theta}^*)} \xrightarrow{\text{d}} N(0,1) \tag{2.141}$$

If $I(\theta)$ is unknown, we can estimate it by sample:

$$\hat{I}(\theta) = \hat{\mathbb{E}}\left[\left(\frac{\partial \ln f(\vec{x};\theta)}{\partial\theta}\right)^2\right] = \sum_{i=1}^{n}\left(\frac{\partial \ln f(x_i;\theta)}{\partial\hat{\theta}^*}\right)^2 \tag{2.142}$$

confidence interval:

$$\left[\hat{\theta}^* - \frac{N_{\frac{\alpha}{2}}}{\sqrt{n}}\sigma(\hat{\theta}^*), \hat{\theta}^* + \frac{N_{\frac{\alpha}{2}}}{\sqrt{n}}\sigma(\hat{\theta}^*)\right] \tag{2.143}$$

2.3.4 Fisher Fiducial Argument*

(Not complete yet) Idea: When sample is known, we can get '**Fiducial Probability**' of θ, thus can find an interval estimation based on fiducial distribution.(Similar to the idea of MLE)

Remark: Fiducial probability (denoted as $\tilde{\mathbb{P}}(\theta)$) is 'probability of parameter', in the case that sample is known. **Fiducial probability is different from Probability**.

Thus get

$$\tilde{\mathbb{P}}(\hat{g}_1 \leq g(\theta) \leq \hat{g}_2) = 1 - \alpha \tag{2.144}$$

Section 2.4 Hypothesis Testing

Hypothesis is a statement about the characteristic of population, e.g. distribution form, parameters, independency, etc.

Mission: Use sample to test the hypothesis, i.e. judge whether population has some characteristic.

2.4.1 Basic Concepts

Parametric hypothesis testing.

For random sample $\vec{X} = (X_1, X_2, \ldots, X_n) \in \mathscr{X}$ i.i.d. from $\mathscr{F} = \{f(x;\theta); \theta \in \Theta\}$

- Null Hypothesis H_0 & Alternative Hypothesis H_1 (Sometimes denoted H_a): Wonder whether a statement is true. Def. **Null Hypothesis**: $H_0 : \theta \in \Theta_0 \subset \Theta$, **a statement that we try to reject based on sample**; $H_1 : \theta \in \Theta_1 = \Theta / \Theta_0$ is **Alternative Hypothesis**.

  □ **Note:** **Cannot** exchange H_0 and H_1, because when the evidence is ambiguity, we have to accept H_0, regardless of what H_0 is. So it is **very important** to pick the proper H_0[9].

  Thus Hypothesis Testing:

  $$H_0 : \theta \in \Theta_0 \longleftrightarrow H_1 : \theta \in \Theta_1 \tag{2.145}$$

- Rejection Region R & Acceptance Region R^C: Judge whether to reject H_0 from sample, Def. **Rejection Region**:

  $$R \subset \mathscr{X} \text{: reject } H_0 \text{ if } \vec{X} \in R \tag{2.146}$$

  Acceptance Region: accept H_0 if $\vec{X} \in R^C$

[9] So when being uncertain about which to put on H_0, think about which one we are more intended to assume when evidence is ambiguous.

Examples:

- Clinical test, in which we should put 'being ill' in H_0, and 'all right' in H_1.
- Court trial, in which we should put 'innocent' in H_0, and 'guity' in H_1.

- Test Function: It's hard and unparctical to really dividing regions in $\mathscr{X}$. Instead the regions are usually described by some test function, basically it's like some indicator function.

  – Continuous Case:

  $$\varphi(\vec{X}) = \begin{cases} 1, & \vec{X} \in R \\ 0, & \vec{X} \in R^{\complement} \end{cases} \tag{2.147}$$

  i.e. $R = \{\vec{X} : \varphi(\vec{X}) = 1\}$. Where R to be determined.

  – Discrete Case: Randomized Test Function

  $$\varphi(\vec{X}) = \begin{cases} 1, & \vec{X} \in R - \partial R \\ r, & \vec{X} \in \partial R \\ 0, & \vec{X} \in R^{\complement} \end{cases} \tag{2.148}$$

  Where R and r to be determined. ∂R means the boundary of R

Δ Type I Error & Type II Error: Sample is random, possible to make a wrong judge.

  – Type I Error (弃真): H_0 is true but sample falls in R, thus H_0 is rejected.

  $$\mathbb{P}(\text{type I error}) = \mathbb{P}(\vec{X} \in R | H_0) = \alpha(\theta) \tag{2.149}$$

  – Type II Error (取伪): H_0 is wrong but sample falls in R^C, thus H_0 is accepted.

  $$\mathbb{P}(\text{type II error}) = \mathbb{P}(\vec{X} \notin R | H_1) = \beta(\theta) \tag{2.150}$$

		Judgement	
		Accept H_0	Reject H_0
Truth	H_0	$\surd$	Type I Error
	H_1	Type II Error	$\surd$

表 2.1: 'Confusion Matrix'

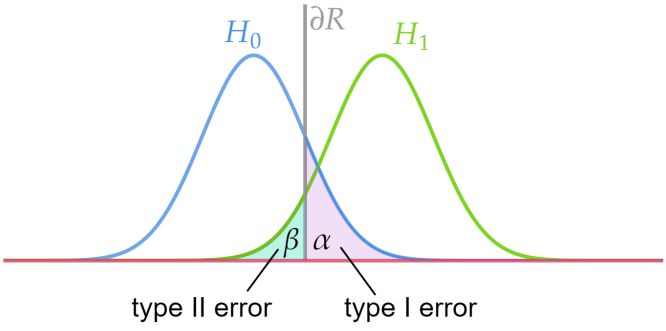


图 2.1: Illustration of type I&II error

It's impossible to make probability of Type I & II Error small simultaneously, how to pick a proper test $\varphi(\vec{x})$?

□ **Neyman-Pearson Principle: First control $\alpha \leq \alpha_0$, then take** min β**.**

How to determine α_0? Depend on specific problem.[10]

Δ p-value: probability to get larger bias (or simply 'more extreme data') than observed $\vec{x}_0$ if H_0 as ground truth, and H_1 as alternative.

e.g. For reject region defined with statistics $R = \{\vec{X}|T(\vec{X}) \geq C\}$, p-value:

$$p_{H_0,H_1}(\vec{x}) = \mathbb{P}[T(\vec{X}) \geq t(\vec{x}_0)|H_0, H_1] \tag{2.151}$$

Remark: We believe that sample should reflect the property of model parameter, and p-value is that under H_0, the probability to get a **worse** result than $\vec{x}$. If the probability is small, then our assumption H_0 might be invalid.

Rule: Reject H_0 if $p(\vec{x}_0) \leq \alpha_0$.

Note :

- p-value is **different from** α or type I error. p-value is generated before we make decision while α arises after we decide how to make decisions. (But they do target the same result.)
- p-value is calculated **after** $H_0 \leftrightsquigarrow H_1$ pair is given. Avoid abusing the concept of p-value.

- Power Function: After $H_0 : \theta \in \Theta_0$ is given, and we have determined the rejection region R, the probability that sample falls in R, i.e. reject H_0 by sampling, as a function of ground truth θ.

$$\pi(\theta) = \mathbb{P}\left(\vec{X}(\theta) \in R|H_0\right) = \begin{cases} \mathbb{P}(\text{type I error}), & \theta \in \Theta_0 \\ 1 - \mathbb{P}(\text{type II error}), & \theta \in \Theta_1 \end{cases} = \begin{cases} \alpha(\theta), & \theta \in \Theta_0 \\ 1 - \beta(\theta), & \theta \in \Theta_1 \end{cases} \tag{2.152}$$

Express as test function:

$$\pi(\theta) = \mathbb{E}[\varphi(\vec{X})|\theta] \tag{2.153}$$

Power function is a measure of the goodness of test: $\pi(\theta)$ should be small under H_0, and be large under H_1 (and grows very fast at the boundary of H_0 and H_1).

□ **General Steps of Hypothesis Testing:**

1. Propose H_0 & H_1.
2. Select a proper α (to determine c).
3. Determine R (usually in the form of a statistic, e.g. $R = \{\vec{X} : T(\vec{X}) \geq c\}$).
4. Sampling, get sample (as well as $t(\vec{x})$), then
   - compare with R and determine whether to reject/accept H_0, or
   - calculate p-value and determine whether to reject/acceptH_0

[10] In most cases, take $\alpha_0 = 0.05$.

2.4.2 Hypothesis Testing of Common Distributions

For some common distribution populations, determine rejection region R under certain H_0 with confidence coefficient α.

Definition of necessary statistics see section section 2.3.3 ~ page 54.

1. Single normal population:

Condition	H_0	H_1	Testing Statistic T	Rejection Region R
σ^2 known, test μ	$\mu = \mu_0$ $\mu \le \mu_0$ $\mu \ge \mu_0$	$\mu \ne \mu_0$ $\mu > \mu_0$ $\mu < \mu_0$	$T = \dfrac{\sqrt{n}(\bar{X} - \mu_0)}{\sigma} \sim N(0,1)$	$\lvert T\rvert > N_{\frac{\alpha}{2}}$ $T > N_\alpha$ $T < -N_\alpha$
σ^2 unknown, test μ	$\mu = \mu_0$ $\mu \le \mu_0$ $\mu \ge \mu_0$	$\mu \ne \mu_0$ $\mu > \mu_0$ $\mu < \mu_0$	$T = \dfrac{\sqrt{n}(\bar{X} - \mu_0)}{S} \sim t_{n-1}$	$\lvert T\rvert > t_{n-1,\frac{\alpha}{2}}$ $T > t_{n-1,\alpha}$ $T < -t_{n-1,\alpha}$
μ known, test σ^2	$\sigma^2 = \sigma_0^2$ $\sigma^2 \le \sigma_0^2$ $\sigma^2 \ge \sigma_0^2$	$\sigma^2 \ne \sigma_0^2$ $\sigma^2 > \sigma_0^2$ $\sigma^2 < \sigma_0^2$	$T = \dfrac{nS_\mu^2}{\sigma_0^2} \sim \chi_n^2$	$T < \chi^2_{n,1-\frac{\alpha}{2}} \cup T > \chi^2_{n,\frac{\alpha}{2}}$ $T > \chi^2_{n,\alpha}$ $T < \chi^2_{n,1-\alpha}$
μ unknown, test σ^2	$\sigma^2 = \sigma_0^2$ $\sigma^2 \le \sigma_0^2$ $\sigma^2 \ge \sigma_0^2$	$\sigma^2 \ne \sigma_0^2$ $\sigma^2 > \sigma_0^2$ $\sigma^2 < \sigma_0^2$	$T = \dfrac{(n-1)S^2}{\sigma_0^2} \sim \chi_{n-1}^2$	$T < \chi^2_{n-1,1-\frac{\alpha}{2}} \cup T > \chi^2_{n-1,\frac{\alpha}{2}}$ $T > \chi^2_{n-1,\alpha}$ $T < \chi^2_{n-1,1-\alpha}$

2. Double normal population:

   σ_1^2, σ_2^2 unknown case see Welch Test in section 14.1.2 ~ page 361.

3. None normal population:

4. More than two normal population: Analysis of Variance.

2.4.3 Likelihood Ratio Test

Idea: To test $H_0 : \theta \in \Theta_0 \longleftrightarrow H_1 : \theta \in \Theta_1$ known $\vec{x}$, examine the likelihood function $L(\theta; \vec{x})$ and **compare** $L_{\theta\in\Theta_0}$ and $L_{\theta\in\Theta}$ to see the likelihood that H_0 is true.

Def. **Likelihood Ratio** (LR):

$$\Lambda(\vec{x}) = \frac{\sup\limits_{\theta\in\Theta_0} L(\theta;\vec{x})}{\sup\limits_{\theta\in\Theta} L(\theta;\vec{x})} \tag{2.154}$$

Reject H_0 if $\Lambda(\vec{x}) < \Lambda_0$. Or equivalently: Reject H_0 if $-2\ln\Lambda(\vec{x}) > C(= -2\ln\Lambda_0)$.

Condition	H_0	H_1	Testing Statistic T	Rejection Region R
σ_1^2, σ_2^2 known, test $\mu_1 - \mu_2$	$\mu_1 - \mu_2 = \mu_0$ $\mu_1 - \mu_2 \le \mu_0$ $\mu_1 - \mu_2 \ge \mu_0$	$\mu_1 - \mu_2 \ne \mu_0$ $\mu_1 - \mu_2 > \mu_0$ $\mu_1 - \mu_2 < \mu_0$	$T = \dfrac{\bar{X} - \bar{Y} - \mu_0}{\sqrt{\dfrac{\sigma_1^2}{m} + \dfrac{\sigma_2^2}{n}}} \sim N(0,1)$	$\lvert T\rvert > N_{\frac{\alpha}{2}}$ $T > N_\alpha$ $T < -N_\alpha$
σ_1^2, σ_2^2 unknown but same, test $\mu_1 - \mu_2$	$\mu_1 - \mu_2 = \mu_0$ $\mu_1 - \mu_2 \le \mu_0$ $\mu_1 - \mu_2 \ge \mu_0$	$\mu_1 - \mu_2 \ne \mu_0$ $\mu_1 - \mu_2 > \mu_0$ $\mu_1 - \mu_2 < \mu_0$	$T = \dfrac{\bar{X} - \bar{Y} - \mu_0}{S_\omega}\sqrt{\dfrac{mn}{m+n}}$ $\sim t_{m+n-2}$	$\lvert T\rvert > t_{m+n-2,\frac{\alpha}{2}}$ $T > t_{m+n-2,\alpha}$ $T < -t_{m+n-2,\alpha}$
μ_1, μ_2 known, test $\dfrac{\sigma_1^2}{\sigma_2^2}$	$\sigma_1^2 = \sigma_2^2$ $\sigma_1^2 \ge \sigma_2^2$ $\sigma_1^2 \le \sigma_2^2$	$\sigma_1^2 \ne \sigma_2^2$ $\sigma_1^2 < \sigma_2^2$ $\sigma_1^2 > \sigma_2^2$	$T = \dfrac{S_{\mu_2}^2}{S_{\mu_1}^2} \sim F_{n,m}$	$T < F_{n,m,1-\frac{\alpha}{2}}$ $\cup T > F_{n,m,\frac{\alpha}{2}}$ $T > F_{n,m,\alpha}$ $T < F_{n,m,1-\alpha}$
μ_1, μ_2 unknown, test $\dfrac{\sigma_1^2}{\sigma_2^2}$	$\sigma_1^2 = \sigma_2^2$ $\sigma_1^2 \ge \sigma_2^2$ $\sigma_1^2 \le \sigma_2^2$	$\sigma_1^2 \ne \sigma_2^2$ $\sigma_1^2 < \sigma_2^2$ $\sigma_1^2 > \sigma_2^2$	$T = \dfrac{S_2^2}{S_2^2} \sim F_{n-1,m-1}$	$T < F_{n-1,m-1,1-\frac{\alpha}{2}}$ $\cup T > F_{n-1,m-1,\frac{\alpha}{2}}$ $T > F_{n-1,m-1,\alpha}$ $T < F_{n-1,m-1,1-\alpha}$

Condition	H_0	H_1	Testing Statistic T	Rejection Region R
$\vec{X}$ from $B(1,p)$, test p	$p = p_0$	$p \ne p_0$	$T = \dfrac{\sqrt{n}(\bar{X} - p_0)}{\sqrt{p_0(1-p_0)}} \xrightarrow{\text{d}} N(0,1)$	$\lvert T\rvert > N_{\frac{\alpha}{2}}$
$\vec{X}$ from $P(\lambda)$, test λ	$\lambda = \lambda_0$	$\lambda \ne \lambda_0$	$T = \dfrac{\sqrt{n}(\bar{X} - \lambda_0)}{\sqrt{\lambda_0}} \xrightarrow{\text{d}} N(0,1)$	$\lvert T\rvert > N_{\frac{\alpha}{2}}$

where Λ_0 (or equivalently $C = -2\ln\Lambda_0$) satisfies:

$$\mathbb{E}_{\Theta_0}[\varphi(\vec{X})] \leq \alpha, \quad \forall\theta \in \Theta_0 \tag{2.155}$$

LR and sufficient statistic: $\Lambda(\vec{x})$ can be expressed as $\Lambda(\vec{x}) = \Lambda^*(T(\vec{x}))$, where $T(\vec{X})$ is sufficient statistic. We usually denote $\lambda = \log\Lambda$

□ **LRT for one-sample t-test: For $X_1, X_2, \ldots, X_n$ i.i.d. $\sim N(\mu, \sigma^2)$, test**

$$H_0 : \mu = \mu_0 \longleftrightarrow H_1 : \mu \neq \mu_0 \quad \textbf{when } \sigma^2 \textbf{ unknown} \tag{2.156}$$

Can prove:

$$\lambda^{2/n} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \mu_0)^2} \tag{2.157}$$

Denote $T = \dfrac{\sqrt{n}(\bar{x} - \mu_0)}{S}$, then LRT could be expressed in equivalent form

$$\lambda = \left(1 + \frac{T^2}{n-1}\right)^{-n/2} \tag{2.158}$$

The Multivariate case see section 4.2.4 ~ page 126, where T^2 itself is the Hotelling's T^2 statistic.

□ **Limiting Distribution of LRT: Wilks' Theorem**

If $\dim\Theta = k > \dim\text{span}\{\Theta_0\} = s$[11], then under $H_0 : \theta \in \Theta_0$:

$$-2\lambda = -2\ln\Lambda(\vec{x}) \xrightarrow{\text{d}} \chi^2_{k-s} \tag{2.159}$$

2.4.4 Uniformly Most Powerful Test

Idea: Neyman-Pearson Principle: control α, find min β. i.e. control α, find max $\pi(\theta)$

Def. **Uniformly Most Powerful Test** (UMP) φ_{UMP} with level of significance α satisfies

$$\pi_{\text{UMP}}(\theta) \geq \pi(\theta), \forall\theta \in \Theta_1 \tag{2.160}$$

Neyman-Pearson Lemma: For $\vec{X} = (X_1, X_2, \ldots, X_n)$ i.i.d. from $f(\vec{x};\theta)$.

Test hypothesis $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta = \theta_1$. Def. test function φ as:

$$\varphi(\vec{x}) = \begin{cases} 1, & \dfrac{f(\vec{x};\theta_1)}{f(\vec{x};\theta_0)} > C \\ r, & \dfrac{f(\vec{x};\theta_1)}{f(\vec{x};\theta_0)} = C \\ 0, & \dfrac{f(\vec{x};\theta_1)}{f(\vec{x};\theta_0)} < C \end{cases} \tag{2.161}$$

Then there exists C and r such that

[11] Here 'dimension' refers to 'degree of freedom'.

- $\mathbb{E}[\varphi(\vec{x})|\theta_0] = \mathbb{P}(\frac{f(\vec{x};\theta_1)}{f(\vec{x};\theta_0)} > C) + r\mathbb{P}(\frac{f(\vec{x};\theta_1)}{f(\vec{x};\theta_0)} = C) = \alpha$
- This φ is UMP of level of significance α

Actually kind of 1-dimensional case of LRT.

Note: UMT exist for **simple** H_0, H_1, otherwise may not exist.

UMP and sufficient statistics: Test function $\varphi(\vec{X})$ given by equation 2.161 ~ page 62 is function of sufficient statistics $T(\vec{X})$, i.e. $\varphi(\vec{X}) = \varphi^*(T(\vec{X}))$.

UMP and Exponential Family: For sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ from exponential family:

$$f(\vec{x};\theta) = C(\theta)h(\vec{x})\exp\{Q(\theta)T(\vec{x})\} \tag{2.162}$$

Test single hypothesis $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta = \theta_1$, (where $\theta_0 < \theta_1$). If

- θ_0 is inner point of Θ
- $Q(\theta)$ monotone increase with θ

Then UMP exists, in the form of:

$$\varphi(\vec{x}) = \begin{cases} 1, & T(\vec{x}) > C \\ r, & T(\vec{x}) = C \\ 0, & T(\vec{x}) < C \end{cases} \tag{2.163}$$

where C and r satisfies $\mathbb{E}[\varphi(\vec{x})|\theta_0] = \alpha$.

Note: or take $Q(\theta)$ mono decreased, then in equation 2.163 ~ page 63, take opposite inequality operators.

□ **General Steps of UMP:**

1. Find a point $\theta_0 \in \Theta_0$ and a point $\theta_1 \in \Theta_1$. (Note: **one** point)
2. Construct test function in the form of equation 2.161 ~ page 62, use $\mathbb{E}[\varphi(\vec{x})|\theta_0] = \alpha$ to determine C and r.
3. Get R and $\varphi(\vec{x})$.
4. If φ does **not** depend on θ_1, then H_1 can be generalized to $H_1 : \theta \in \Theta_1$.
5. If φ satisfies $\mathbb{E}_{\theta\in\Theta_0}(\varphi) \le \alpha$, then H_0 an be generalized to $H_0 : \theta \in \Theta_0$.

□ **Upgrade to one-sided test:**

Using Neyman-Pearson we have only two-points test $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta = \theta_1$. With certain condition we could upgrade two-points two one-sided test $H_0 : \theta \le \theta_0 \longleftrightarrow H_1 : \theta > \theta_0$ by **Karlin-Rubin Theorem**.

1. Monotone Likelihood Ratio Condition : If $\forall\theta > \tilde{\theta}$ we have

$$\frac{L(\theta)}{L(\tilde{\theta})} \text{ is monotone in sufficient stat} T(\vec{X})$$

then we say that the MLR condition is satisfied.

2. With MLR condition, we could upgrade two-points test to one-sided test $H_0 : \theta < \theta_0 \longleftrightarrow H_1 : \theta > \theta_0$. We would have a UMP test of form

$$\text{Rejection Region } R = \{\vec{X} : T(\vec{X}) > C\}$$

Note: Upgrade to two-sided test $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta \neq \theta_0$ is not always possible. e.g. $H_0 : \mu = \mu_0 \longleftrightarrow H_1 : \mu \neq \mu_0$ for $X \overset{i.i.d.}{\sim} N(\mu, \sigma_0^2$ known) does not have UMP test.

2.4.5 Duality of Hypothesis Testing and Interval Estimation

- Theorem: $\forall \theta_0 \in \Theta$ there exists hypothesis testing $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta \neq \theta_0$ of level α with rejection region R_{θ_0}. Then

$$C(\vec{X}) = \{\theta : \vec{X} \in R_\theta^C\} \tag{2.164}$$

  is a $1 - \alpha$ confidence region for θ

- Theorem: $C(\vec{X})$ is a $1 - \alpha$ confidence region for θ. Then $\forall \theta_0 \in C(\vec{X})$, the rejection region of hypothesis testing $H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta \neq \theta_0$ of level α satisfies

$$R_{\theta_0}^{\complement} = \{\vec{X} : \theta_0 \in C(\vec{X})\} \tag{2.165}$$

□ **Idea:**

$$H_0 : \theta = \theta_0 \longleftrightarrow H_1 : \theta \neq \theta_0$$

$$\updownarrow$$

$$\mathbb{P}(R^{\complement}(\vec{X})|H_0) = \mathbb{P}(R^{\complement}(\vec{X})|\theta_0) = 1 - \alpha$$

$$\updownarrow$$

$$\text{Confidence Interval: } \theta_0 \in R^{\complement}(\vec{X})$$

Similar for Confidence Limit and One-Sided Testing.

▷ R. **Code**

The test function for one-way / two-way test. The function gives both interval estimation and hypothesis testing results.

```
# one-way
t.test(x, alternative = c("two.sided", "less", "greater"), mu = 0,
    conf.level = 0.95, ...)
# two-way
t.test(x, y, alternative = c("two.sided", "less", "greater"), mu =
    0, paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)
t.test(df, ...)
```

where `paired = TRUE` for pairwise comparison requires $|x| = |y|$.

2.4.6 Introduction to Non-Parametric Hypothesis Testing

Motivation: Usually distribution form unknown, cannot use parametric hypothesis testing.

Useful Method:

- Sign Test: Used for paired comparison $\vec{X} = (X_1, X_2, \ldots, X_n)$, $\vec{Y} = (Y_1, Y_2, \ldots, Y_n)$.

  Take $Z_i = Y_i - X_i$ i.i.d., denote $\mathbb{E}(Z) = \mu$. Test $H_0 : \mu = 0 \longleftrightarrow H_1 : \mu \neq 0$. Denote $n_+ = \#(\text{positive } Z_i)$ and $n_- = \#(\text{negative } Z_i)$, $n_0 = n_+ + n_-$. Then $n_+ \sim B(n_0, \theta)$, thus the test is $H_0 : \theta = \dfrac{1}{2} \longleftrightarrow H_1 : \theta \neq \dfrac{1}{2}$

  Then use Binomial Testing or large sample CLT Normal Testing on H_0.

  Remark:

  - Also can test $H_0 : \theta \leq \dfrac{1}{2} \longleftrightarrow H_1 : \theta > \dfrac{1}{2}$
  - Drawback: ignores magnitudes.

- Wilcoxon Signed Rank Sum Test: Improvement of Sign Test. Based on order statistics.

  Order Statistics of Z_i: $Z_{(1)} < Z_{(2)} < \ldots < Z_{(n)}$, where each $Z_{(j)}$ corresponds to some Z_i as $Z_i = Z_{(R_i)}$, then R_i is the rank of Z_i.[12]

$$R_i = \sum_{i=1}^{n} \tag{2.166}$$

  Def. $\vec{R} = (R_1, R_2, \ldots, R_n)$ is **Rank Statistics** of $(Z_1, Z_2, \ldots, Z_n)$

$$R_i = \sum_{j=1}^{n} \mathbb{I}_{Z_j < Z_i} + \frac{1}{2}\left(1 + \sum_{j=1}^{n} \mathbb{I}_{Z_j = Z_i}\right) \tag{2.167}$$

  Def. **Sum of Wilcoxon Signed Rank**:

$$W^+ = \sum_{i=1}^{n_0} R_i \mathbb{I}_{Z_{R_i} > 0} \tag{2.168}$$

  Distribution of W^+ is complex. $\mathbb{E}$ and var of W^+ under H_0:

$$\mathbb{E}(W^+) = \frac{n_0(n_0+1)}{4} \qquad var(W^+) = \frac{n_0(n_0+1)(2n_0+1)}{24} \tag{2.169}$$

  Usually consider large sample CLT, construct normal approximation:

$$T = \frac{W^+ - \mathbb{E}(W^+)}{\sqrt{var(W^+)}} \xrightarrow{\text{d}} N(0,1) \tag{2.170}$$

  Rejection Region: $R = \{|T| > N_{\frac{\alpha}{2}}\}$

[12] If some $X_i, X_j, \ldots$ equal, then take same rank $R = \text{mean}\{R_i, R_j, \ldots\}$.

- Wilcoxon Two-Sample Rank Sum Test: Used for two independent sample comparison.

  Assume $\vec{X} = (X_1, \ldots, X_m)$ i.i.d. $\sim f(x)$; $\vec{Y} = (Y_1, \ldots, Y_n)$ i.i.d. $\sim f(x-\theta)$, test $H_0 : \theta = 0 \longleftrightarrow H_1 : \theta \neq 0$.

  Rank X_i and Y_i as:

  $$Z_1 \leq Z_2 \leq \ldots \leq Z_{m+n} \tag{2.171}$$

  in which denote rank of Y_i as R_i, and def. **Wilcoxon two-sample rank sum**:

  $$W = \sum_{i=1}^{n} R_i \tag{2.172}$$

  $\mathbb{E}$ and var of W under H_0:

  $$\mathbb{E}(W) = \frac{n(m+n+1)}{2} \qquad var(W) = \frac{mn(n+m+1)}{12} \tag{2.173}$$

  Use large sample approximation, construct CLT:

  $$T = \frac{W - \mathbb{E}(W)}{\sqrt{var(W)}} \xrightarrow{\text{d}} N(0,1) \tag{2.174}$$

  ▷ R. **Code**

  ```r
  wilcox.test(x, y, alternative = c("two.sided", "less", "greater"), mu = 0, paired = FALSE)
  ```

- Goodness-of-Fit Test: For $\vec{X} = (X_1, X_2, \ldots, X_n)$ i.i.d. from some certain population X. Test $H_0 : X \sim F(x)$.

  where F is theoretical distribution, can be either parametric or non-parametric.

  Idea: Define some *quantity* $D = D(X_1, \ldots, X_n; F)$ to measure the difference between F and sample. And def. *Goodness-of-fit* when observed value of D (say d_0) is given:

  $$p(d_0) = \mathbb{P}(D \geq d_0 | H_0) \tag{2.175}$$

  Goodness-of-Fit Test: Reject H_0 if $p(d_0) < \alpha$.

  Pearson χ^2 Test: Usually used for discrete case.

  Test $H_0 : \mathbb{P}(X_i = a_i) = p_i$, $i = 1, 2, \ldots, r$. Denote $\#(X_j = a_i) = \nu_i$, take D as:

  $$K_n = K_n(X_1, \ldots, X_n; F) = \sum_{i=1}^{r} \frac{(\nu_i - np_i)^2}{np_i} \tag{2.176}$$

  Pearson Theorem: For K_n defined as equation 2.176 ~ page 66, then under H_0:

  $$K_n \xrightarrow{\text{d}} \chi^2_{r-1-s} \tag{2.177}$$

  Here s is number of unknown parameter, $r - 1 - s$ is the degree of freedom.

  Note:

– a_i must **not** depend on sample.

– For continuous case, construct division:

$$\mathbb{R} \to (-\infty, a_1, a_2, \ldots, a_{r-1}, \infty = a_r) \tag{2.178}$$

and test $H_0 : \mathbb{P}(X \in I_j) = p_j$

Criterion: Pick proper interval so that np_i and ν_i both ≥ 5.

- Contingency Table Independence & Homogeneity Test: Detailed knowledge and more complex appplication cases see section 7.2.4 ~ page 222 and section 8.2.1 ~ page 240

– Independence Test:

Test a two-parameter sample and to see whether these two parameters(features) are independent. Denote $Z = (X, Y)$ are some 'level' of sample, n_{ij} is number of sample with level (i, j)

Contingency Table:

X \ Y	1	…	j	…	s	$\sum$
1	n_{11}	…	n_{1j}	…	n_{1s}	$n_{1\cdot}$
⋮	⋮	⋱	⋮	⋱	⋮	⋮
i	n_{i1}	…	n_{ij}	…	n_{is}	$n_{i\cdot}$
⋮	⋮	⋱	⋮	⋱	⋮	⋮
r	n_{r1}	…	n_{rj}	…	n_{rs}	$n_{r\cdot}$
$\sum$	$n_{\cdot 1}$	…	$n_{\cdot j}$	…	$n_{\cdot s}$	n

Test $H_0 : X$ & Y are independent. i.e. $H_0 : P(X = i, Y = j) = P(X = i)P(Y = j) = p_{i\cdot}p_{\cdot j}$.

Construct χ^2 test statistic:

$$K_n = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{[n_{ij} - n(\frac{n_{i\cdot}}{n})(\frac{n_{\cdot j}}{n})]^2}{n(\frac{n_{i\cdot}}{n})(\frac{n_{\cdot j}}{n})} = n\left(\sum_{i=1}^{r}\sum_{j=1}^{s}\frac{n_{ij}^2}{n_{i\cdot}n_{\cdot j}} - 1\right) \tag{2.179}$$

Then under H_0, $K_n \xrightarrow{\text{d}} \chi^2_{rs-1-(r+s-2)} = \chi^2_{(r-1)(s-1)}$

Reject H_0 if $p(k_0) = P(K_n \geq k_0) < \alpha$

– Homogeneity Test:

Test R groups of sample with category rank, to see whether these groups has similar rank distribution.

Group \ Category	Category 1	…	Category j	…	Category C	$\sum$
Group 1	n_{11}	…	n_{1j}	…	n_{1C}	$n_{1\cdot}$
⋮	⋮	⋱	⋮	⋱	⋮	⋮
Group i	n_{i1}	…	n_{ij}	…	n_{iC}	$n_{i\cdot}$
⋮	⋮	⋱	⋮	⋱	⋮	⋮
Group R	n_{R1}	…	n_{Rj}	…	n_{RC}	$n_{R\cdot}$
$\sum$	$n_{\cdot 1}$	…	$n_{\cdot j}$	…	$n_{\cdot C}$	n

Denote $P(\text{Category } j|\text{Group } i) = p_{ij}$. Test $H_0 : p_{ij} = p_j, \forall 1 \le i \le R$.

Construct χ^2 test statistic:

$$D = \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{[n_{ij} - n(\frac{n_{i\cdot}}{n})(\frac{n_{\cdot j}}{n})]^2}{n(\frac{n_{i\cdot}}{n})(\frac{n_{\cdot j}}{n})} = n\left(\sum_{i=1}^{R}\sum_{j=1}^{C}\frac{n_{ij}^2}{n_{i\cdot}n_{\cdot j}} - 1\right) \tag{2.180}$$

Then under H_0, $D \xrightarrow{\text{d}} \chi^2_{R(C-1)-(C-1)} = \chi^2_{(R-1)(C-1)}$

▷ R. **Code**

Contingency table test example:

```r
table_df <- matrix(c(10,20,15,25), 2, 2)
chisq.test(table_df)
fisher.test(table_df)
```

- Test of Normality: normality is a good & useful assumption.

  For $\vec{Y} = (Y_1, Y_2, \dots, Y_n)$,

  Test H_0 : exists μ & σ^2 such that Y_i i.i.d. $\sim N(\mu, \sigma^2)$.

  – Kolmogorov-Smirnov Test: Assume $\vec{X}$ form population CDF $F(x)$, test $H_0 : F(x) = F_0(x)$(where can take $F_0 = \Phi$ or some other known CDF).

  use $F_n(x)$ (as defined in equation 2.122 ~ page 52) as approx. to $F(x)$, test

  $$D_n = \sum_{-\infty<x<+\infty} |F_n(x) - F_0(x)| \tag{2.181}$$

  Reject H_0 if $D_n > c$

  or use goodness-of-fit: denote observed value of D_n as d_n. Reject H_0 if

  $$p(d_n) = \mathbb{P}(D_n > d_n|H_0) < \alpha \tag{2.182}$$

  – Shapiro-Wilk Test:

Test H_0 : exists μ & σ^2 such that X_i i.i.d. $\sim N(\mu, \sigma^2)$.

Denote $Y_{(i)} = \dfrac{X_{(i)} - \mu}{\sigma}$, $m_i = \mathbb{E}(Y_{(i)})$

Under H_0, $(X_{(i)}, m_i)$ falls close to straight line. Test Statistic: Correlation

$$R^2 = \frac{\left(\sum_{i=1}^n (X_{(i)} - \bar{X})(m_i - \bar{m})\right)^2}{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (m_i - \bar{m})^2} = corr(X_{(i)}, m_i) \tag{2.183}$$

Reject H_0 if $R^2 < c$

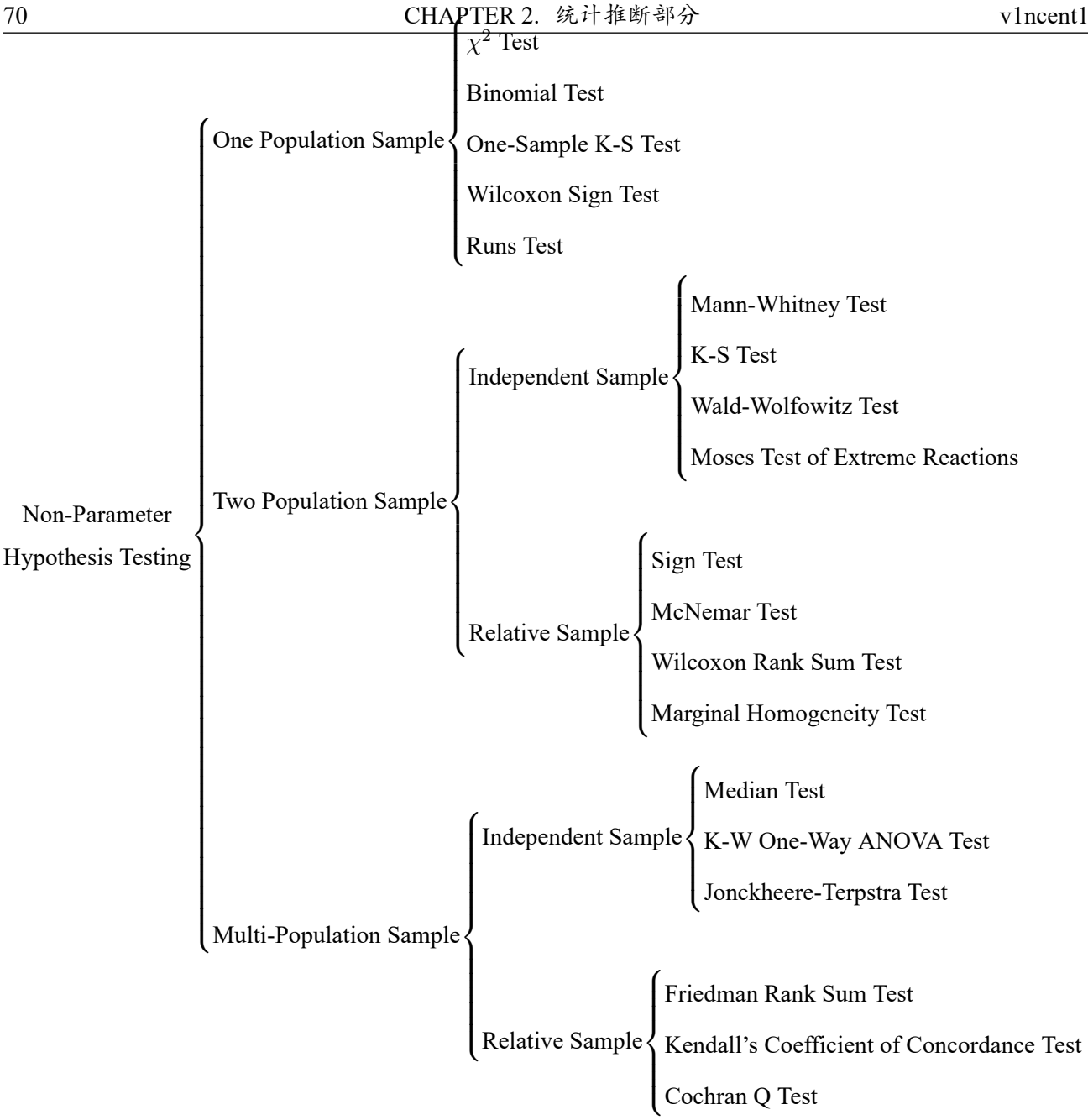
Shapiro-Wilk correction:

$$W = \frac{\left(\sum_{i=1}^{[n/2]} a_i (X_{(n+1-i)} - X_{(i)})\right)^2}{\sum_{i=1}^n (X_{(i)} - \bar{X})^2} \tag{2.184}$$

▷ R. **Code**

```r
shapiro.test(x)
```

□ **Summary: Useful Non-Parameter Hypothesis Testing.**


Non-Parameter Hypothesis Testing
One Population Sample
χ^2 Test
Binomial Test
One-Sample K-S Test
Wilcoxon Sign Test
Runs Test
Two Population Sample
Independent Sample
Mann-Whitney Test
K-S Test
Wald-Wolfowitz Test
Moses Test of Extreme Reactions
Relative Sample
Sign Test
McNemar Test
Wilcoxon Rank Sum Test
Marginal Homogeneity Test
Multi-Population Sample
Independent Sample
Median Test
K-W One-Way ANOVA Test
Jonckheere-Terpstra Test
Relative Sample
Friedman Rank Sum Test
Kendall's Coefficient of Concordance Test
Cochran Q Test

Chapter. III　线性回归分析部分

Instructor: Zaiying Zhou

☐ **Steps in Regression Analysis**

1. Statement of the problem;
2. Selection of potentially relevant **variables**;
3. Data collection;
4. Exploratory Data Analysis (**EDA**)
5. **Model** specification;
6. Choice of fitting method;
7. Model fitting;
8. Model validation and criticism;
9. Using the chosen model(s) for the solution of the posed problem;
10. **Explain** the result.

R. Code for EDA

```
libaray('GGally')
head(df)
ggpairs(df)
str(df)
summary(df)
```

☐ **Used Packages in R.**

```
library('ggplot2')
libaray('GGally')
library('car')
library('moments')
library('lmtest')
library('nortest')
```

```r
library('MASS')
library('tseries')

source('package.r')
```

Section 3.1 Regression Model

In regression model, we will observe pairs of variables, called 'cases'(样本点). A sample is $(X_1; Y_1), \ldots, (X_n; Y_n)$, where X_i can be multivariate $X_i = \vec{X_i} = (X_{i1}, X_{i2}, \ldots, X_{ip})$.

If X is continuous `numeric` variable, use Regression Model(s), else if X is discrete `factor` variable, use Factor Model(s).

▷ `R`. **Code**

Example data import:

```r
df <- read.table('dataset/testdata.txt', header=FALSE, sep=',', col.
   names = c('y','x1','x2'))
```

3.1.1 Linear Regression Model

Regression Model focuses on how Y changes with continuous variables $X \in \mathbb{R}$. As a basic situation, we use **Linear Regression**, i.e. $Y \sim X$ in linear relation.

□ **Sample Geometry Notation (Full Version)**

For most general case, in sample matrix notation:

$$Y = X\beta + \varepsilon \leftrightharpoons Y_j = X\beta_j + \varepsilon_j, \forall j = 1, 2, \ldots, q \tag{3.1}$$

in Einstein Summation Convention:

$$Y_{ij} = X_{ij'}\beta_{j'j} + \varepsilon_{ij} \tag{3.2}$$

Why we need ε as 'random error term'?

- It represents the intrinsic random property of the model.
- Based on ε, we can take r.v. into our statistic model.

where

$$\underset{n\times q}{Y}=\begin{bmatrix}y_{11}&y_{12}&\cdots&y_{1q}\\y_{21}&y_{22}&\cdots&y_{2q}\\\vdots&\vdots&\ddots&\vdots\\y_{n1}&y_{n2}&\cdots&y_{nq}\end{bmatrix}=\begin{bmatrix}y_1,y_2,\ldots,y_q\end{bmatrix}\qquad y_j=\begin{bmatrix}y_{1j}\\y_{2j}\\\vdots\\y_{nj}\end{bmatrix}\tag{3.3a}$$

$$\underset{n\times(p+1)}{X}=\begin{bmatrix}1&x_{11}&x_{12}&\ldots&x_{1p}\\1&x_{21}&x_{22}&\ldots&x_{2p}\\\vdots&\vdots&\ddots&\vdots&\\1&x_{n1}&x_{n2}&\ldots&x_{np}\end{bmatrix}=\begin{bmatrix}x_1'\\x_2'\\\vdots\\x_n'\end{bmatrix}\qquad x_i=\begin{bmatrix}1\\x_{i1}\\\vdots\\x_{ip}\end{bmatrix}\tag{3.3b}$$

$$\underset{(p+1)\times q}{\beta}=\begin{bmatrix}\beta_{01}&\beta_{02}&\cdots&\beta_{0q}\\\beta_{11}&\beta_{12}&\cdots&\beta_{1q}\\\beta_{21}&\beta_{22}&\cdots&\beta_{2q}\\\vdots&\vdots&\ddots&\vdots\\\beta_{p1}&\beta_{p2}&\cdots&\beta_{pq}\end{bmatrix}=\begin{bmatrix}\beta_1,\beta_2,\ldots,\beta_q\end{bmatrix}\qquad \beta_j=\begin{bmatrix}\beta_{j0}\\\beta_{j1}\\\vdots\\\beta_{jp}\end{bmatrix}\tag{3.3c}$$

$$\underset{n\times q}{\varepsilon}=\begin{bmatrix}\varepsilon_{11}&\varepsilon_{12}&\cdots&\varepsilon_{1q}\\\varepsilon_{21}&\varepsilon_{22}&\cdots&\varepsilon_{2q}\\\vdots&\vdots&\ddots&\vdots\\\varepsilon_{n1}&\varepsilon_{n2}&\cdots&\varepsilon_{nq}\end{bmatrix}=\begin{bmatrix}\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_q\end{bmatrix}\qquad \varepsilon_j=\begin{bmatrix}\varepsilon_{1j}\\\varepsilon_{2j}\\\vdots\\\varepsilon_{nj}\end{bmatrix}\tag{3.3d}$$

with Guass-Markov Assumption:

$$\begin{aligned}\text{Zero-Mean: }&\mathbb{E}(\epsilon_i|X_i)=0\\\text{Homogeneity of Variance: }&var(\epsilon_i)=\sigma^2\\\text{Independent: }&\epsilon_i\text{ i.i.d. }\sim\varepsilon\end{aligned}\tag{3.4}$$

and Normality Error Assumption:

$$\text{Normality: }\varepsilon_i\text{ i.i.d. }\sim N(0,\sigma^2)\tag{3.5}$$

Under matrix notation, model and assumptions equation 3.4 ~ page 73 (equation 3.5 ~ page 73) can be expressed in condensed notation:

$$Y_j=X\beta_j+\varepsilon_j\sim N_n(X\beta_j,\sigma_j^2I_n),\quad j=1,2,\ldots,q\tag{3.6}$$

> Δ **Note:** In this section we only focus on $q=1$, i.e.
>
> $$\underset{n\times1}{Y}=\underset{n\times(p+1)}{X}\underset{(p+1)\times1}{\beta}+\underset{n\times1}{\varepsilon}\tag{3.7}$$

Regression Problem in Bayesian Statistics Statement see section 13.4.9 ~ page 357.

3.1.2 Factor Analysis Model

Regression Model focuses on continuous variables $X \in \mathbb{R}$ while factor model focus on discrete variable. More specifically, the 'value' of X is just a label, not necessarily a 'numeric value'.

Here only introduce one-way factor analysis,(single factor analysis) i.e. Y with only one factor with r levels: fac $= 1, 2, \ldots, r$. Re-denote $Y_{ij} =$ the observation outcome of the j^{th} item labelled the i^{th} level.

Model:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij},\ i = 1, \ldots, r,\ j = 1, \ldots, n_i \tag{3.8}$$

$$w.r.t. \sum_{i=1}^{r} \tau_i = 0 \tag{3.9}$$

where μ is the average effect of all r factor levels, τ_i is the level effect of the i^{th} factor level, and ε i.i.d. $\sim N(0, \sigma^2)$ is noise error.

In matrix notation:

$$Y = \begin{bmatrix} y_{11} & \ldots & y_{1n_1} & y_{21} & \ldots & y_{2n_2} & \ldots \end{bmatrix}^T \tag{3.10a}$$

$$X = \begin{bmatrix} 1 & 1 & 0 & \ldots \\ 1 & 1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \ddots \\ 1 & 0 & 1 & \ldots \\ 1 & 0 & 1 & \ldots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -1 & -1 & \ldots \\ 1 & -1 & -1 & \ldots \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{1}_{n_1} & 0 & \ldots & 0 \\ \mathbf{1}_{n_2} & 0 & \mathbf{1}_{n_2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{1}_{n_{r-1}} & 0 & 0 & \ldots & \mathbf{1}_{n_{r-1}} \\ \mathbf{1}_{n_r} & -\mathbf{1}_{n_r} & -\mathbf{1}_{n_r} & \ldots & -\mathbf{1}_{n_r} \end{bmatrix} \tag{3.10b}$$

$$\tau = \begin{bmatrix} \mu & \tau_1 & \tau_2 & \ldots & \tau_{r-1} \end{bmatrix}^T \tag{3.10c}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_{11} & \ldots & \varepsilon_{1n_1} & \varepsilon_{21} & \ldots & \varepsilon_{2n_2} & \ldots \end{bmatrix}^T \tag{3.10d}$$

For more factor model e.g. two-way factor analysis with k denoting item and i, j denoting factor:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \tag{3.11}$$

cannot be simply expressed in matrix notation $\longrightarrow$ use index notation.

Assumption: Normal, Equal variance, independent

- One-way: $Y_{j|i}$ i.i.d. $\sim N(\mu + \tau_i, \sigma^2), \forall i$
- Two-way: $Y_{k|ij}$ i.i.d. $\sim N(\mu + \alpha_i + \beta_j, \sigma^2), \forall i, j$

Section 3.2 Monovariate Linear Regression Model

First focus on the simpliest monovariate case $\vec{X}_i = X_i$. Monovariate Linear Model[1] with Gauss-Markov assumption & Normal Error assumption:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i,\ \varepsilon_i \text{ i.i.d. } \sim N(0, \sigma^2) \tag{3.12}$$

What does Linear Regression do? Try to estimate

- β_0 (intercept) ;
- β_1 (slope) ;
- σ^2 (variance of error).

(Thus Linear Regression is also a Statistics Inference process: deduce properties of model from data)

3.2.1 The Ordinary Least Square Estimation

Aim: use (x_i, y_i) to estimate $\beta_0, \beta_1, \sigma^2$. The idea is to define a 'loss function' to reflect the 'distance' from sample point to estimation point.

Estimate Principle: [2]

- Ordinary Least Squares:

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y - \beta_0 - \beta_1 x_i)^2 \tag{3.13}$$

- MLE or MoM Estimation.

And get $\hat{\beta}_1, \hat{\beta}_0$ as well as $\hat{\sigma^2}$(see equation 3.18 ~ page 76:[3]

$$\begin{aligned} \hat{\beta}_1 =& \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ \hat{\beta}_0 =& \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\sigma^2} =& \frac{1}{n-p-1} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{3.15}$$

Def. **Residuals**: distance from sample point to estimate point, to reflect how the sample points fit the model.

$$e_i = y_i - \hat{y}_i = \text{observed value of } \varepsilon_i \tag{3.16}$$

[1] Here in linear regression, we consider X_i only as real number, **without** randomness. So here Y_i can be considered as an r.v. with X_i as parameter, i.e. $Y_i|_{X_i = x_i}$

[2] Detailed Definition and Derivation see section 2.2.5 ~ page 49 or section 3.3 ~ page 81.

[3] A memory trick: use $\frac{Y}{\sqrt{s_Y}} = r_{XY} \frac{X}{\sqrt{s_X}}$ to get formular of $Y \sim X$:

$$\hat{\beta}_1 = r_{XY} \frac{\sqrt{s_Y}}{\sqrt{s_X}} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \tag{3.14}$$

Note: under least square estimation, we have[4]

$$\sum e_i = 0 \qquad \sum x_i e_i = 0 \tag{3.17}$$

Then use e_i to estimate σ^2 (because it is ε_0 that are i.i.d., not Y_i), where $(n-p-1)$ is Degree of Freedom (df or dof)[5]

$$\begin{aligned} \hat{\sigma_n^2} &= \frac{1}{n}\sum e_i^2 \quad \text{(use MLE or MoM)} \\ \hat{\sigma^2} &= \frac{1}{n-p-1}\sum e_i^2 = \frac{1}{n-2}\sum e_i^2 \quad \text{(use OLS, unbiased)} \end{aligned} \tag{3.18}$$

Degree of Freedom of a Quadric Form:

- Intuitively: the number of independent variable;
- Rigorously: for quadric SS $= x'Ax$:

$$dof_{SS} = \text{rank}(A) \tag{3.19}$$

Which comes from Cochran's Theorem. A proof can be found here: https://v1ncent19.github.io//texts/Cochran/

▷ R. **Code**

```r
lmfit <- lm(formula,df)
summary(lmfit,cor=TRUE)
ggcoef(lmfit)
```

`lmfit` includes parameters `lmfit$coefficient` and `lmfit$residuals`

Example `lm()` output:

```
    Call:
    lm(formula = y ~ x, data = df)

    Residuals:
         Min       1Q   Median       3Q      Max
    -16.1368  -6.1968  -0.5969   6.7607  23.4731

    Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
    (Intercept) 156.3466     5.5123   28.36   <2e-16 ***
    x            -1.1900     0.0902  -13.19   <2e-16 ***
```

[4] Intuitively, they each means '$E(\varepsilon) = 0$' and '$X \parallel \varepsilon$'.

[5] Generally, MLE and OLSE are different.

Comment from R.A.Fisher: $\sum e_i^2$ should be divided by 'number of e_i^2 that contribute to variance'. Here $(n-p-1)$ corresponds to 'degree of freedom' $= (n-2)$, $p=1$ corresponds to 'one' variable (see section 2.2.5 ~ page 49, equation 2.121 ~ page 51), and correponds to the two equations of e_i, equation 3.17 ~ page 76

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.173 on 58 degrees of freedom
Multiple R-squared:  0.7501,    Adjusted R-squared:  0.7458
F-statistic: 174.1 on 1 and 58 DF,  p-value: < 2.2e-16
```

3.2.2 Statistical Inference to β_0, β_1, σ^2

□ **Sampling Distribution of $\hat{\beta}_1, \hat{\beta}_0$**

Consider $\hat{\beta}_1, \hat{\beta}_0$ as statistics of sample, then we can examine the sampling distribution of $\hat{\beta}_1, \hat{\beta}_0$. Their randomness comes from

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{3.20}$$

(The following part treats $\hat{\beta}_1, \hat{\beta}_0$ as r.v., and note that X_i are **not** r.v.. And for convenience and conciseness, denote $S_{XX} = \sum_{i=1}^{n}(X_i - \bar{X})^2$)

$$\hat{\beta}_1 = \beta_1 + \sum_{i=1}^{n} \frac{X_i - \bar{X}}{S_{XX}} \varepsilon_i \tag{3.21}$$

$$\hat{\beta}_0 = \beta_0 + \sum_{i=1}^{n} \left(\frac{1}{n} - \frac{(X_i - \bar{X})\bar{X}}{S_{XX}} \right) \varepsilon_i \tag{3.22}$$

Denote corresponding variance as $\sigma^2_{\hat{\beta}_1}$ and $\sigma^2_{\hat{\beta}_0}$, using equation 1.125 ~ page 33 to get:

$$\sigma^2_{\hat{\beta}_1} = \frac{\sigma^2}{S_{XX}} \qquad \sigma^2_{\hat{\beta}_0} = \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{S_{XX}}\right) \tag{3.23}$$

And under normal error assumption, distribution of $\hat{\beta}_1, \hat{\beta}_0$ are

$$\hat{\beta}_1 \sim N(\beta_1, \sigma^2_{\hat{\beta}_1}) = N(\beta_1, \frac{\sigma^2}{S_{XX}}) \tag{3.24}$$

$$\hat{\beta}_0 \sim N(\beta_0, \sigma^2_{\hat{\beta}_0}) = N(\beta_0, \sigma^2(\frac{1}{n} + \frac{\bar{X}^2}{S_{XX}})) \tag{3.25}$$

Based on sampling distribution of $\hat{\beta}_1, \hat{\beta}_0$, we can conduct statistical inference, including CI and HT.[6]

Note: In linear regression model, we usually focus more on β_1. And note that when 0 is **not** within the fitting range,β_0 is not so important.[7]

[6] Detail see section 2.4 ~ page 57, estimating/testing $\hat{\beta}_1, \hat{\beta}_0$ usually corresponds to 'estimate μ, with σ^2 unknown'.

[7] Two reason:

- The etimation error of Y from $\hat{\beta}_1$ increases with $X_h - \bar{X}$;
- $\beta_1 == 0$ is important: decides whether linear model can be used.

□ **Sampling Distribution of e_i** Consider e_i as r.v. satisfies

$$e_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i \tag{3.26}$$

and get the expression of e_i

$$\hat{e}_i = \varepsilon_i - \sum_{k=1}^{n} \left(\frac{1}{n} + \frac{(X_i - \bar{X})^2}{S_{XX}} \right) \varepsilon_k \tag{3.27}$$

$$e_i = 0 \qquad \sigma_{e_i}^2 = \sigma^2 \left(1 - \frac{1}{n} - \frac{(X_i - \bar{X})^2}{S_{XX}} \right) \tag{3.28}$$

Under normal assumption:

$$e_i \sim N(0, \sigma^2 \left(1 - \frac{1}{n} - \frac{(X_i - \bar{X})^2}{S_{XX}} \right)) \tag{3.29}$$

Further we can get $var(\hat{\sigma}^2) = \mathbb{E}(\frac{1}{n-2} \sum_{i=1}^{n} e_i^2)$, where $e_i^2 \sim \sigma^2 \left(1 - \frac{1}{n} - \frac{(X_i - \bar{X})^2}{S_{XX}} \right) \chi^2$

$$var(\hat{\sigma}^2) = \frac{1}{n-2} \sigma^2 \sum_{i=1}^{n} (1 - \frac{1}{n} - \frac{(X_i - \bar{X})^2}{S_{XX}}) \tag{3.30}$$

More definition of refined residuals see section 3.4.3 ~ page 88 in page 3.4.3.

□ **Why we choose OLS to get regression coefficients?**

Gauss-Markov Theorem: the OLS estimator has the lowest sampling variance within the class of linear unbiased estimators, i.e. OLS is the **Best Linear Unbiased Estimator(BLUE)**.[8]

3.2.3 Prediction to Y_h

For a new X_h at which we wish to predict the corresponding Y_h (based on other known points $\{(X_i, Y_i)\}_{i=1}^n$), denote the estimator of mean as $\hat{\mu}_h$:

$$\hat{\mu}_h = \hat{\beta}_1 X_h + \hat{\beta}_0 = \beta_1 X_h + \beta_0 + \sum_{i=1}^{n} \left(\frac{1}{n} + \frac{(X_i - \bar{X})(X_h - \bar{X})}{S_{XX}} \right) \varepsilon_i \tag{3.31}$$

Thus we can get[9]

$$\mathbb{E}(\hat{\mu}_h) = \beta_1 X_h + \beta_0 \qquad \sigma_{\hat{\mu}_h}^2 = \left(\frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{XX}} \right) \sigma^2 \tag{3.32}$$

Under Normal assumption:

$$\hat{\mu}_h \sim N(\beta_1 X_h + \beta_0, \left(\frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{XX}} \right) \sigma^2) \tag{3.33}$$

Base on distribution we can give CI and HT.

□ **Interval: We can either consider ...**

[8] This Theorem does **not** require normal error assumption.

[9] So $\sigma^2(\hat{\mu}_h)$ increases with $X_h - \bar{X}$. Intuitively it make sense, because $(\bar{X}, \bar{Y})$ must falls on regression line.

- $\hat{Y}_h^{\text{conf}} = \hat{\mu}_h$: a function of (estimated) parameter to get **confidence interval**: We can just use the above distribution of $\hat{\mu}_h$

$$\hat{Y}_h^{\text{conf}} = \hat{\mu}_h \sim N(\beta_1 X_h + \beta_0, \left(\frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{XX}}\right)\sigma^2) \tag{3.34}$$

▷ R. **Code**

```
predict(lmfit,newdata = 40),
    interval="confidence",level=0.95)
```

- $\hat{Y}_h^{\text{pred}} = \hat{\mu}_h + \varepsilon$: a function of (estimated) parameter and an extra random term to get **prediction interval**.

In this case we have an extra randomness. Def. Prediction Error: Y_h itself is an Y of the linear model, i.e. $Y_h = \hat{\beta}_0 + \hat{\beta}_1 X_h + \varepsilon_h$, we can define **Prediction Error**:

$$d_h = Y_h - \hat{\mu}_h \tag{3.35}$$

with

$$\mathbb{E}(d_h) = 0 \qquad \sigma_{d_h}^2 = var(Y_h - \hat{\mu}_h) = \left[1 + \frac{1}{n} + \frac{(X_h - \bar{X})}{S_{XX}}\right]\sigma^2 > \sigma_{\hat{\mu}_h}^2 \tag{3.36}$$

thus

$$\hat{Y}_h^{\text{conf}} = \hat{\mu}_h + \varepsilon_h \sim N(\beta_1 X_h + \beta_0, \left(1 + \frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{XX}}\right)\sigma^2) \tag{3.37}$$

▷ R. **Code**

```
predict(lmfit,newdata = ..., interval="confidence",level=0.95)
predict(lmfit,newdata = ..., interval="prediction",level=0.95)
```

Comment: in prediction error, we considered more random component, thus the CI is also larger.

□ **Simultaneous Confidence Band (SCB)**

Confidence Band is **not** the CI at each point, but really a **band** for the **entire** regression line.[10]

Aim: Find lower and upper function $L(x)$ and $U(x)$ such that

$$\mathbb{P}[L(x) < (\beta_0 + \beta_1 x) < U(x), \forall x \in I_x] = 1 - \alpha \tag{3.38}$$

and get **Confidence Band**:

$$\{(x, y) | L(x) < y < U(x) | \forall x \in I_x\} \tag{3.39}$$

Where $(L(x), U(x))$ can be derived as

$$(L(x), U(x)) = \hat{\mu}_x \pm s_{\hat{\mu}_x} W_{2,n-2,1-\alpha} \quad \forall x \in I_x \tag{3.40}$$

[10] Why they are different? We require the confidence band have a **simultaneous** converage probability. For the same band $(L(x), U(x))$, P(the whole line) $< P$(each point), so Confidence Band is wider than $\bigcup$CIs to hold the same $1 - \alpha$.

Also, we will see that for linear model, the boundary of SCB forms hyperbola, which make sense considering its asymptotic line.

Where W correponds to W distribution: $W_{m,n} = \sqrt{2F_{m,n}}$

Small sample case: Bonferroni correction.

▷ R. **Code**

```
library(ggplot2)
ggplot(df,aes(x,y))+geom_point()+geom_smooth(method='lm',formula=y~x
    )
```

3.2.4 Analysis of Variance: Monovariate

ANalysis **O**f **VA**riance (ANOVA): One-sample t test ⇝ Two sample t test ⇝ More-sample: ANOVA

□ **Key Idea Of ANOVA: Test whether the mean of some groups are the same, i.e.** $\mu_1 = \mu_2 = \ldots = \mu_r$

In linear regression model, modified as testing $\beta_1 == 0$. Conduction: Take Partition of Total Sum of Square To Examine **Variation**. Because Y_i are not i.i.d. (different mean value $X\beta$), so has different parts of variation from Regression Model/Error Term.

Measure of Variation: Sum of Square (SS) & Mean Sum of Square (MS).

MS: Divide each SS by corresponding dof. Definition of dof see equation 3.19 ~ page 76.

$$\text{MS} = \frac{\text{SS}}{dof} \tag{3.41}$$

- SST: Total Sum of Squares

$$\text{SST} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 \qquad dof_{\text{SST}} = n - 1 \tag{3.42}$$

- SSRegression: Variation due to Regression Model (which is explained by regression line);[11]

$$\text{SSR} = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 \qquad dof_{\text{SSR}} = 1 \tag{3.43}$$

- SSError: Variation attribtes to ε (which is reflected by residuals).

$$\text{SSE} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i) \qquad dof_{\text{SSE}} = n - 2 \tag{3.44}$$

> Δ **IMPORTANT:** In some books
>
> - SSRegression → SSExplained or SSModel;
> - SSError → SSResidual.
>
> And Cause **Confusion**! In this summary we take the former.

Idea: take partition of SST. i.e.

$$Y_i - \bar{Y} = (Y_i - \hat{Y}) + (\hat{Y} - \bar{Y}) = e_i \tag{3.45}$$

[11] $\text{SSR} = \hat{\beta}_1^2 \sum_{i=1}^{n}(X_i - \bar{X})^2$, so $dof_R = 1$

And we can prove that

$$\text{SST} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \text{SSR} + \text{SSE} \tag{3.46}$$

That is: we **partition** SST into two parts, so that we can examine them seperately.

□ **ANOVA Table**

Source	dof	SS	MS	F-Statistic
SSRegression	1	$\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$	SSR/dof_R	MSR/MSE
SSError	$n-2$	$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	SSE/dof_E	
SSTotal	$n-1$	$\sum_{i=1}^{n}(Y_i - \bar{Y})^2$	SST/dof_T	

▷ R. **Code**

```r
anova(lmfit)
```

Properties:

$$\mathbb{E}(\text{MSE}) = \sigma^2 \qquad \mathbb{E}(\text{MSR}) = \sigma^2 + \beta_1^2 S_{XX} \tag{3.47}$$

Section 3.3 Multivariate Linear Regression Model

As a more general case of $\vec{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip})$, Multivariate Linear Model is expressed as in equation 3.7 ~ page 73:

$$Y = X\beta + \varepsilon,\ \varepsilon \sim N_p(0, \sigma^2 I) \tag{3.48}$$

3.3.1 The Ordinary Least Estimation

To conduct OLS

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^{p+1}}{\arg\min}(Y - X\beta)^T(Y - X\beta) \tag{3.49}$$

Here we introduce two approaches:

- Analytical: Take matrix differciation (See section 4.1.2 ~ page 118 equation 4.41 ~ page 120)

$$0 = \frac{\partial(Y - X\beta)^T(Y - X\beta)}{\partial\beta} = \frac{\partial}{\partial\beta}(Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta) \tag{3.50}$$

$$= -X^TY - X^TY + (X^TX + XX^T)\beta = -2X^T(Y - X\beta) \tag{3.51}$$

Thus we get OLS:

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{3.52}$$

- Geometric/Algebraical: Use hyper-projection.

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^{p+1}}{\arg\min}\, d(Y, X\beta) \tag{3.53}$$

i.e. $\hat{\beta}$ is the (hyper-)projection of Y onto X (within Euclidean Space), naturally we have

$$(X\beta)^T(Y - X\beta) = 0 \Rightarrow \hat{\beta} = (X'X)^{-1}X'Y \tag{3.54}$$

□ **Matrix Notation of OLS Estimator:**

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{3.55}$$

3.3.2 Statistical Inference to β, σ^2

Properties & Extrapolation

- Sampling Distribution of $\hat{\beta}$: (Here consider normal case $Y \sim N(X\beta, \sigma^2 I_n)$, and use equation 4.66 ~ page 123)

$$\hat{\beta} = (X'X)^{-1}X'Y \sim N_p(\beta, \sigma^2(X'X)^{-1}) \tag{3.56}$$

Comment: $cov(\beta_i, \beta_j)$ are generally not 0, $\Rightarrow \beta_i, \beta_j$ dependent.

- Predicted Response & Hat Matrix H:

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y \equiv HY = P_X Y \tag{3.57}$$

where **Hat Matrix**/Influence matrix/Projection matrix $H = P_X = X(X'X)^{-1}X'$, with properties

  - Symmertric: $H^T = H$;
  - Idempotence: $H^2 = H$
  - Rank: $\text{rk}(H) = tr(H) = \text{rk}(X)$
  - H and self-influene factor h_{ii}: Note the linearity of $\hat{Y}$ on Y

$$\hat{Y} = HY \Rightarrow H = \frac{\partial \hat{Y}}{\partial Y} \tag{3.58}$$

The diagonal elements of H is

$$h_{ii} = \frac{\partial \hat{y}_i}{\partial y_i} = X_i(X'X)^{-1}X_i' \tag{3.59}$$

Comment on h_{ii}: $var(e_i) = \sigma^2(1 - h_{ii})$, for $h_{ii} \to 1$, i.e. the regression line always pass y_i, thus it's 'influential'.

- Residual:

$$e = Y - \hat{Y} = (I - H)Y \sim N_n\left(0, \sigma^2(I - H)\right) \tag{3.60}$$

where $I - H$ is the complementary projection of X

Covariance Matrix of Residual:

$$cov(e) = \sigma^2(I-H) = \sigma^2 \begin{bmatrix} 1-h_{11} & -h_{12} & \dots & -h_{1n} \\ -h_{21} & 1-h_{22} & \dots & -h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{n1} & -h_{n2} & \dots & 1-h_{nn} \end{bmatrix} \tag{3.61}$$

- Estimator and Distribution of σ^2:

  First use equation 4.67 ~ page 123 to get [12]

$$\mathbb{E}(\text{SSE}) = \mathbb{E}(e'e) = \mathbb{E}(Y'(I-H)Y) = (X\beta)'(I-H)X\beta + tr((I-H)\sigma^2 I_n) = \sigma^2(n-p-1) \tag{3.63}$$

  dof of Residual e (use definition equation 3.19 ~ page 76):

$$dof_e = dof_{(I-H)Y} = \text{rank}(I-H) = n-p-1 \tag{3.64}$$

  Thus the unbiased estimator of σ^2 is

$$\hat{\sigma}^2 = \text{MSE} = \frac{e'e}{n-p-1} = \frac{Y'(I-H)Y}{n-p-1} \tag{3.65}$$

  Distribution (under normal assumption):

$$\frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1} \tag{3.66}$$

- Gauss-Markov Theorem: OLS Estimator of β is the BLUE Estimator.

More hypethesis testing to β see section 4.2.4 ~ page 126.

3.3.3 Prediction to Y_h

For a new $\vec{X}_h$ at which we wish to **predict** the corresponding Y_h (based on other known point (X_i, Y_i)), denote the estimator of mean as $\hat{\mu}_h$:

$$\hat{\mu}_h = X_h'\hat{\beta} = X_h'(X'X)^{-1}X'Y \tag{3.67}$$

thus we get

$$\mathbb{E}(\hat{\mu}_h) = X_h'\beta \qquad \sigma^2_{\hat{\mu}_h} = \sigma^2(1 + X_h'(X'X)^{-1}X_h) \tag{3.68}$$

under normal assumption:

$$\hat{\mu}_h \sim N(X'\beta, \sigma^2(1 + X_h'(X'X)^{-1}X_h)) \tag{3.69}$$

[12] Also we need the property of idmpotnet matrix

$$\lambda_i = 0 \text{ or } 1 \Rightarrow tr(H) = \text{rank}(H) = \sum_{i=1}^{n} \lambda_i = \#(\lambda = 1) \tag{3.62}$$

3.3.4 Analysis of Variance: Multivariate

Sampling Notation see equation 3.3 ~ page 73, still consider $(p+1)$ -dim $(\mathbf{1}_n, X_i)$ v.s. 1-dim Y, and $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$

- SST:

$$\text{SST} = (Y - \bar{Y}\mathbf{1}_n)'(Y - \bar{Y}\mathbf{1}_n) \qquad dof_{\text{SST}} = n - 1 \tag{3.70}$$

- SSR:

$$\text{SSR} = (\hat{Y} - \bar{Y}\mathbf{1}_n)'(\hat{Y} - \bar{Y}\mathbf{1}_n) \qquad dof_{\text{SSR}} = p \tag{3.71}$$

Denoted in hat matrix H and $\mathcal{J}$ in equation 4.17 ~ page 116

$$\text{SSR} = Y'(H - \frac{1}{n}\mathcal{J})Y \tag{3.72}$$

- SSE:

$$\text{SSE} = (Y - \hat{Y})'(Y - \hat{Y}) \qquad dof_{\text{SSE}} = n - p - 1 \tag{3.73}$$

Denoted in residual e and hat matrix H:

$$\text{SSE} = e'e = Y'(I - H)Y \tag{3.74}$$

More knowledge about multivariate ANOVA see section 3.4.5 ~ page 95.

□ **ANOVA Table**

Source	dof	SS	MS	F-Statistic
SSRegression	p	$\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$	SSR/dof_R	MSR/MSE
SSError	$n-p-1$	$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	SSE/dof_E	
SSTotal	$n-1$	$\sum_{i=1}^{n}(Y_i - \bar{Y})^2$	SST/dof_T	

▷ R. **Code**

```r
anova(lmfit)
```

Section 3.4 Diagnostics

To apply OLS, we need the basic Gauss–Markov Assumption equation 3.4 ~ page 73; or we further need better properties of the model, e.g. take Normal Assumption.

Assumptions:

$$\begin{aligned} \text{Zero-Mean: } & \mathbb{E}(\epsilon_i | X_i) = 0 \\ \text{Homogeneity of Variance: } & var(\epsilon_i) = \sigma^2 \\ \text{Independent: } & \epsilon_i \text{ i.i.d. } \sim \varepsilon \\ \text{Normal: } & \varepsilon \sim N(0, \sigma^2) \end{aligned} \tag{3.75}$$

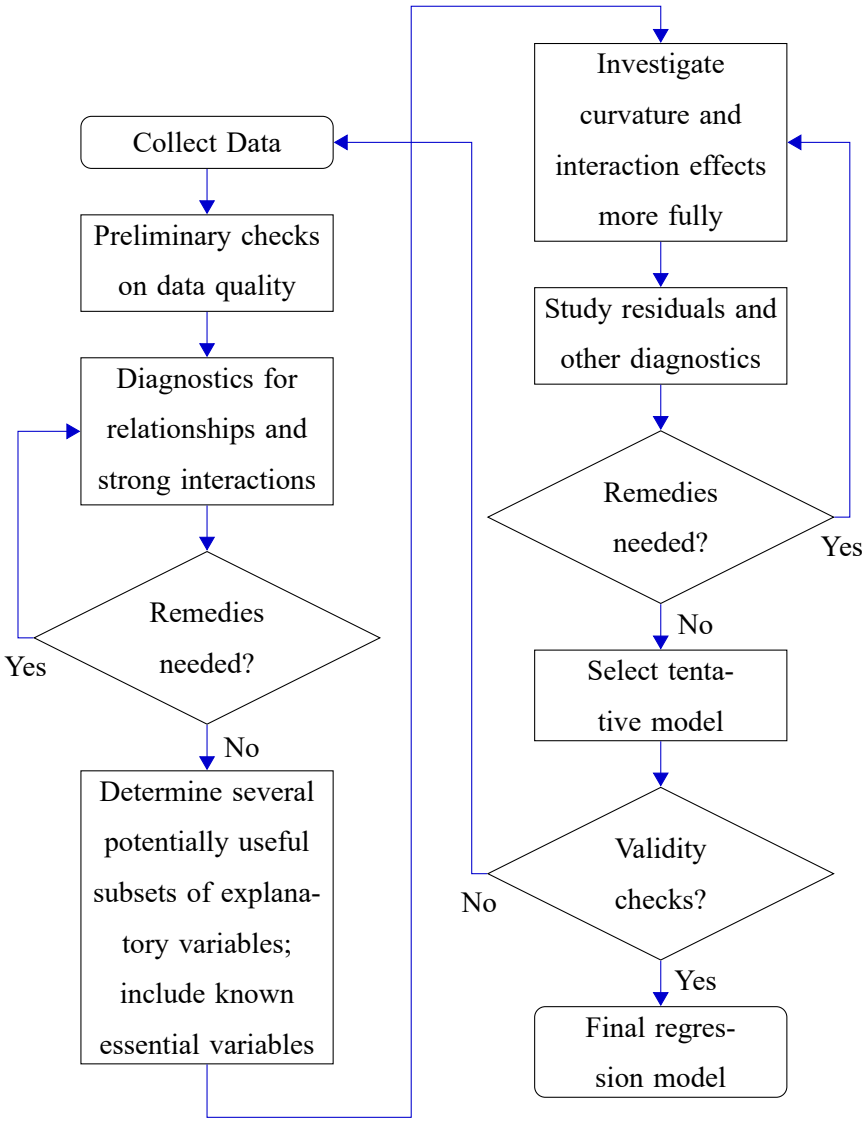


图 3.1: Diagnostics and Remedies for Regression Model

Or sum up as

$$Y \sim N_n(X\beta, \sigma^2 I_n) \tag{3.76}$$

Thus we need to conduct Diagnostics and Remedies to

- examine whether these assumptions are satisfies;
- perform correction to regression method.

Preliminary Diagnostics:

▷ R. **Code**

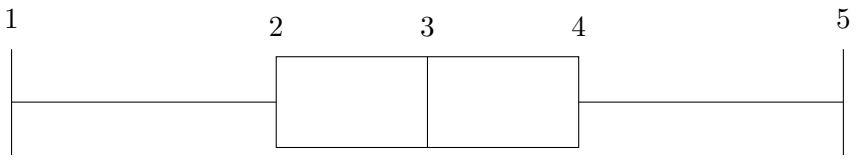
```r
lmfit <- lm(y~x,lmfit)
par(mfrow = c(2, 2))
plot(lmfit)
par(mfrow = c(1, 1))
```

3.4.1 Useful Diagnostics Plots

- BoxPlot: to examine the similarity of shape of distribution.

  Notation:

  1. min point above (25% quartile $-$ 1.5 IQR);
  2. 25% quantile;
  3. median;
  4. 75% quantile;
  5. max point below (75% quartile $+$ 1.5 IQR).



- Histogram Plots: Frequency distribution (can deal with many-peak)
- Quartile-Quartile Plots: Examine the similarity between distribution.

  For two CDF $q = F(x)$ and $q = G(x)$(where q for quartile), with $x = F^{-1}(q)$, $x = G^{-1}(q)$. And Plot $F^{-1}(q)$-$G^{-1}(q)$.

  Usually test normality, take $G = \Phi$

- Partial Regression Plot: Test non-linearity/heterogeneous-variance.

  For each X_i variable:

  – Use other $X_{(\wedge i)}$ to predict Y, get residual $e_Y|X_{(\wedge i)}$;

– Use other $X_{(\wedge i)}$ to predict X_i, get residual $e_{X_i}|X_{(\wedge i)}$

Plot $(e_Y|X_{(\wedge i)})$-$(e_{X_i}|X_{(\wedge i)})$ as Added Variable Plot (AV Plot). Used for testing non-linearity/heterogeneous-variance.

▷ R. **Code**

```r
boxplot(df$x)

hist(df$x)

hist(df$x,freq=FALSE)
lines(density(df$x))

stem(df$x)

qqnorm(df$x)
qqline(df$x,col='red')

library(car)
avPlots(lmfit)
```

3.4.2 Diagnostics to X Distribution

Considering the dependence of Y_i on X_i, to get a more reliable $\hat{\beta}_1$, we cannot just focus on the (marginal) distribution of Y_i, we would also need a better 'distribution' of X_i

- Plots: BoxPlot/QQPlot
- 4 statistics(parameters);[13]
  - Mean: Location;

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.77}$$

  - Standard Deviation: Variability;

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \tag{3.78}$$

  - Skewness: Lack of Symmertry;

$$\hat{g}_1 = \frac{m_{n,3}}{m_{n,2}^{3/2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})\right)^{3/2}} \tag{3.79}$$

[13] See section 2.1.1 ~ page 38

Adjusted Skewness (MMSE):

$$\frac{\sqrt{n(n-1)}}{n-2}\hat{g}_1 \tag{3.80}$$

* $\hat{g}_1 > 0$: Right skewness, longer right tail;
* $\hat{g}_1 < 0$: Left skewness, longer left tail.

Fisher-Pearson coefficient of skewness: $\frac{3(\text{mean}-\text{median})}{\sigma}$.

– Kurtosis: Heavy/Light Tailed.

$$\hat{g}_2 = \frac{m_{n,4}}{m_{n,2}^2} - 3 = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^2} - 3 \tag{3.81}$$

$\hat{g}_2 = 0 \Rightarrow$ similar to normal.

* $\hat{g}_2 > 0$: Leptokurtic, heavy tail, slender;
* $\hat{g}_2 < 0$: Platykurtic, light tail, broad.

Note: In expression of $\hat{g}_1$ and $\hat{g}_2$, we already divide the variance. So Skewness and Kurtosis only reflect the difference from normal, but **not** related to variance.

Best tool to determine Kurtosis: QQ-Plot.

▷ R. **Code**

```r
summary(df$x)
```

Other moments use package `moments`

- Bias: Inspect the design methodology
  - Selection Bias: Not completely random sampling;
  - Information Bias: Difference between 'designed' and 'get', e.g. no response;
  - Confounding: Exist another important variable, while the model actually focuses on a less important variable, or even reverse the causality.

3.4.3 Diagnostics to Residual

□ **Residual Reflects the properties of ε**

- **Linearity** : use Residual Plot/AV Plot to Reflect the linearity and variance assumption.

▷ R. **Code**

```r
lmfit <- lm(y~x,df)
scatter(df$x,lmfit$residuals)
abline(h=0)

```

```
library(car)
avPlots(lmfit)
```

- The Assumption of **Equal Variances** / Homoscedasticity (齐方差性):
  - AV Plot , e.g. test the R^2 of $(e_Y|X_{(\wedge i)})$-$(e_{X_i}|X_{(\wedge i)})$ relation.
  - Bartlett's test:

    Idea: divide the sample into groups g, and get each MSE

    $$\text{MSE}_g = \frac{1}{n_g}\sum_{i=1}^{n_g}(Y_{gi} - \hat{Y}_g)^2 \tag{3.82}$$

    and take statistic

    $$S = -\frac{(N-g)\ln\left[\sum_{g=1}^{G}\frac{n_g}{N-n_g}\text{MSE}_g\right] - \sum_{g=1}^{G}(n_g)\ln\frac{n_g-1}{N-n_g}\text{MSE}_g}{1+\frac{1}{3(G-1)}\left(\sum_{g=1}^{G}\frac{1}{n_g} - \frac{1}{N-G}\right)} \sim \chi^2_{G-1} \tag{3.83}$$

    to conduct test.

    Note: **sensitive** to normal assumption, not robust. Used when normal assumption is satisfied.
  - Levene's test: Divide the sample into G groups. Denote **mean** of residual within each group as $\tilde{e}_g$, and in each group compute

    $$d_{ig} = |e_{ig} - \tilde{e}_g| \Rightarrow \bar{d}_g = \frac{1}{n_g}\sum_{j=1}^{n_g} d_{ig} \tag{3.84}$$

    Then conduct ANOVA to d_{ig}.

    If $G = 2$: 2-sample t-test,

    $$T = \frac{\bar{d}_1 - \bar{d}_2}{s\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}} \xrightarrow{\text{d}} t_{n-2} \qquad s^2 = \frac{\sum(d_{i1}-\bar{d}_1)^2 + \sum(d_{i2}-\bar{d}_2)^2}{n-2} \tag{3.85}$$

  - Brown-Forsythe's Test (Modified Levene's test): For skewed sample, take the **mean** as **median**, more robust.
  - ⋆ Breusch-Pagan Test:

    Assume variance of ε_i dependent on X_i as m^{th} polynomial:

    $$\sigma_i^2 = \alpha_0 + \sum_{k=1}^{m}\alpha_k X_i^k \tag{3.86}$$

    and test

    $$H_0 : \alpha_k = 0 \,\forall k = 1, 2, \ldots, m \longleftrightarrow H_1 \tag{3.87}$$

    Method: First conduct OLS to get regression line $\hat{l}_1$ and residuals e_i and SSE, and conduct regression of e_i^2 over X_i to get another regression line $\hat{l}_2$ and corresponding SSR*.

    Then statistic

    $$S = \frac{\text{SSR}^*/2}{(\text{SSE}/n)^2} \xrightarrow{\text{d}} \chi^2_m \tag{3.88}$$

▷ R. **Code**

Example for $G = 2$:

```
group <-factor(rep(c(1,2),length.out=length(df$x),
    each=(ceiling(length(df$x)/2))))

bartlett.test(lmfit$residuals~group,group)

library(car)
leveneTest(lmfit$residuals~group,group,center=mean)
leveneTest(lmfit$residuals~group,group,center=median)

library(lmtest)
bptest(lmfit)
```

- The Assumption of **Normality** :

  In most case we use S-W Test($n < 2000$) and K-S Test($n > 2000$):

  – QQ-plot of ordered residuals.

  ⋆ Shapiro-Wilk Test (Most Powerful)[14]: To test $H_0 : \exists\sigma^2,\ s.t.\varepsilon \sim N_n(0, \sigma^2 I_n)$, denote

  $$m_i = \mathbb{E}(\frac{\varepsilon_{(i)}}{\sigma}) \tag{3.89}$$

  then under H_0, $\varepsilon_{(i)} \sim m_i \to$ linear, thus test correlation

  $$R^2 = \frac{\left(\sum_{i=1}^n (e_{(i)} - \bar{e})(m_i - \bar{m})\right)^2}{\sum_{i=1}^n (e_i - \bar{e})^2 \sum_{i=1}^n (m_i - \bar{m})^2} = corr(e_{(i)}, m_i) \tag{3.90}$$

  – Kolmogorov-Smirnov Test:

  $$D_n = \sum_e |F_n(e) - \Phi(e)| \tag{3.91}$$

  – Cramér-von Mises Test:

  $$T = n\int_{-\infty}^{\infty} (F_n(e) - \Phi(e))^2 \,\mathrm{d}\Phi(e) \tag{3.92}$$

  – Anderson-Darling Test:

  $$A^2 - n\int_{-\infty}^{\infty} (F_n(e) - \Phi(e))^2 \frac{1}{\Phi(e)(1-\Phi(e))} \,\mathrm{d}\Phi(e) \tag{3.93}$$

  – Jarque-Bera Test , using skewness $\hat{g}_1$ and kurtosis $\hat{g}_2$ of $\vec{e}$

  $$\mathrm{JB} = \frac{n}{6}(\hat{g}_1^2 + \frac{1}{4}\hat{g}_2^2) \xrightarrow{\mathrm{d}} \chi_2^2 \tag{3.94}$$

▷ R. **Code**

[14] Detail of S-W Test and K-S Test see Test of Normality in section 2.4.6 ~ page 65

```r
qqnorm(lmfit$residuals)
qqline(lmfit$residuals)

qqp <- qqnorm(lmfit$residuals)
cor(qqp$x,qqp$y)

shapiro.test(lmfit$residuals)

ks.test(jitter(lmfit$residuals), pnorm,mean(lmfit$residuals), sd
   (lmfit$residuals))

library(nortest)
cvm.test(lmfit$residuals)

ad.test(lmfit$residuals)

library(tseries)
jarque.bera.test(lmfit$residuals)
```

- The Assumption of **Independence** :
  - Durbin-Watson Test:

$$d = \frac{\sum_{j=2}^{n}(e_j - e_{j-1})^2}{\sum_{j=1}^{n} e_j^2} \tag{3.95}$$

    $d \in (1.5, 2.5)$ is fine.
  - Ljung-Box Test:

$$Q = n(n+2)\sum_{k=1}^{n}\frac{\hat{\rho}_k^2}{n-k} \tag{3.96}$$

▷ R. **Code**

```r
dwtest(lmfit)
```

3.4.4 Diagnostics to Influentials

An intuitive explanation to extreme values:

- Outliers: Extreme case for Y;
- High Leverage: Extreme case for X;

- Influentials: Cases that influence the regression line.

□ **Influentials = Outliers ∩ High Leverage**

In section 3.3 ~ page 81, we got the $\hat{\beta}$ as $\hat{\beta} = (X'X)^{-1}X'Y = HY$ and got $\hat{Y}$ as

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y = \hat{H}Y \tag{3.97}$$

where hat matrix $H \equiv X(X'X)^{-1}X' = \dfrac{\partial \hat{Y}}{\partial Y}$

Also we got statistical inference to β, σ^2, e

$$\hat{\beta} = (X'X)^{-1}X'Y \sim N(\beta, \sigma^2(X'X)^{-1}) \tag{3.98}$$

$$e = Y - \hat{Y} = (I - H)Y \sim N(0, \sigma^2(I - H)) \tag{3.99}$$

$$\hat{\sigma}^2 = \text{MSE} = \frac{e'e}{n-p-1} = \frac{Y'(I-H)Y}{n-p-1} \tag{3.100}$$

$$\frac{(n-p-1)\hat{\sigma^2}}{\sigma^2} \sim \chi^2_{n-p-1} \tag{3.101}$$

The diagonal elements of $\hat{H}$ are **self-sensitivity** h_{ii}

$$h_{ii} = X_i'(X'X)^{-1}X_i \tag{3.102}$$

□ **Some refined residuals to help conduct Diagnostics:**

- Standardized Residual:

$$e_{\text{sd}i} = \frac{e_i}{\sigma_{e_i}} = \frac{e_i}{\sigma\sqrt{1-h_{ii}}} \tag{3.103}$$

- (Internally) Studentized Residual: replace σ with $s = \hat{\sigma}$

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}} = \frac{e_i}{\sqrt{\text{MSE}}\sqrt{1-h_{ii}}} \sim t_{n-p-1} \tag{3.104}$$

- Deleted Residual:[15]

$$d_i = Y_i - \hat{Y}_{i(\wedge i)} = \frac{e_i}{1-h_{ii}} \tag{3.115}$$

[15] □ *Proof*:

Lemma: $(A+B)^{-1} = A^{-1} - \dfrac{1}{1+tr(BA^{-1})}A^{-1}BA^{-1}$, where $\text{rk}(B) = 1$.

$$\hat{\beta}_{(\wedge i)} = (X'_{(\wedge i)}X_{(\wedge i)})^{-1}X'_{(\wedge i)}Y_{(\wedge i)} \tag{3.105}$$

Using the above lemma: (here for aesthetic purpose, treat X_i as row vector)

$$(X'_{(\wedge i)}X_{(\wedge i)})^{-1} = (X'X - X_i'X_i)^{-1} \tag{3.106}$$

$$= (X'X)^{-1} + \frac{1}{1-tr[X_i'X_i(X'X)^{-1}]}(X'X)^{-1}X_i'X_i(X'X)^{-1} \tag{3.107}$$

$$= (X'X)^{-1} + \frac{1}{1-h_{ii}}(X'X)^{-1}X_i'X_i(X'X)^{-1} \tag{3.108}$$

$$X_{(\wedge i)}Y_{(\wedge i)} = X'Y - X_i'Y_i \tag{3.109}$$

then calaulate $\hat{\beta}_{(\wedge i)}$:

where $\hat{Y}_{i(\wedge i)}$ is predicted Y value at X_i obtained from the regression of dataset with the i case (X_i, Y_i) removed:

$$\hat{\beta}_{(\wedge i)} = (X'_{(\wedge i)}X_{(\wedge i)})^{-1}X'_{(\wedge i)}Y_{(\wedge i)} \qquad \hat{Y}_{i(\wedge i)} = X'_i\hat{\beta}_{(\wedge i)} \tag{3.116}$$

- (Externally) Studentized Residual: To avoid self-influence, take deleted residual in equation 3.104 ~ page 92

$$t_i = \frac{d_i}{s^2(d_i)} = \frac{e_i}{\hat{\sigma}_{(\wedge i)}\sqrt{1-h_{ii}}} = \frac{e_i}{\sqrt{\text{MSE}_{(\wedge i)}}\sqrt{1-h_{ii}}} \sim t_{n-p-2} \tag{3.117}$$

Relation between MSE and $\text{MSE}_{(\wedge i)}$:

$$(n-p-1)\text{MSE} = (n-p-2)\text{MSE}_{(\wedge i)} + \frac{e_i^2}{1-h_{ii}} \tag{3.118}$$

which also gives the relation between t_i and r_i:

$$t_i = r_i\left(\frac{n-p-2}{n-p-1-r_i^2}\right)^{1/2} \Leftrightarrow r_i = t_i\left(\frac{n-p-1}{n-p-2+t_i^2}\right)^{1/2} \tag{3.119}$$

- Diagnostics to **Outlier**: use external studentized residual for t-test with Bonferroni adjustment. Declare the i^{th} case an outlier if:

$$|t_i| > t_{\alpha/2n,n-p-2} \tag{3.120}$$

- Diagnostics to **Leverage**: use hat matrix H/self-sensitivity h_{ii}.

$$\sum_{i=1}^{n} h_{ii} = tr(H) = p+1 \Rightarrow \bar{h} = \frac{p+1}{n} \tag{3.121}$$

$$\begin{aligned}
\hat{\beta}_{(\wedge i)} =&(X'_{(\wedge i)}X_{(\wedge i)})^{-1}X'_{(\wedge i)}Y_{(\wedge i)} \\
=&\left[(X'X)^{-1} + \frac{(X'X)^{-1}X'_iX_i(X'X)^{-1}}{1-h_{ii}}\right](X'Y - X'_iY_i) \\
=&\hat{\beta} + \frac{(X'X)^{-1}X'_iX_i(X'X)^{-1}X'Y}{1-h_{ii}} - (X'X)^{-1}X'_iY_i - \frac{(X'X)^{-1}X'_iX_i(X'X)^{-1}X'_iY_i}{1-h_{ii}} \\
=&\hat{\beta} + \frac{(X'X)^{-1}X'_i\hat{Y}_i}{1-h_{ii}} - \frac{(X'X)^{-1}X'_iY_i(1-h_{ii})}{1-h_{ii}} - \frac{(X'X)^{-1}X'_iY_i}{1-h_{ii}}h_{ii} \\
=&\hat{\beta} + \frac{(X'X)^{-1}X'_i}{1-h_{ii}}(\hat{Y}_i - Y_i) \\
\Rightarrow&\hat{\beta} - \hat{\beta}_{(\wedge i)} = (X'X)^{-1}X'_i\frac{e_i}{1-h_{ii}}
\end{aligned} \tag{3.110}$$

Then

$$\begin{aligned}
Y_i - \hat{Y}_{i(\wedge i)} =&Y_i - \hat{Y}_i + \hat{Y}_i - \hat{Y}_{i(\wedge i)} && (3.111)\\
=&e_i + X_i(\hat{\beta} - \hat{\beta}_{(\wedge i)}) && (3.112)\\
=&e_i + X_i(X'X)^{-1}X'_i\frac{e_i}{1-h_{ii}} && (3.113)\\
=&\frac{e_i}{1-h_{ii}} && (3.114)
\end{aligned}$$

□

Declare the i^{th} case a leverage if:

$$h_{ii} > \kappa \bar{h} = \kappa \frac{p+1}{n} \tag{3.122}$$

where usually take $\kappa = 2$ or 3.

- Diagnostics to **Influential**: Studentized DIFference caused to FITted values (DIFFITS)

  DIFFIT:

$$\text{DIFFIT}_i = \hat{Y}_i - \hat{Y}_{i(\wedge i)} = e_i \frac{h_{ii}}{1-h_{ii}} \tag{3.123}$$

  DIFFITS:

$$\text{DIFFITS}_i = \frac{\text{DIFFIT}_i}{s(\hat{Y}_i)} = t_i \sqrt{\frac{h_{ii}}{1-h_{ii}}} \tag{3.124}$$

  Declare the i^{th} case an influential if:

$$\begin{cases} \text{DIFFITS}_i > 1 & \text{small/medium data} \\ \text{DIFFITS}_i > 2\sqrt{\dfrac{p+1}{n}} & \text{large data} \end{cases} \tag{3.125}$$

- Diagnostics to **Influential**: Cook's Distance, by quantifying the 'influence' to $\hat{\beta}$.

  Using equation 3.56 ~ page 82(equation 3.66 ~ page 83) we could construct the following Cook's Distance[16]

$$D_i = \frac{\left\| X(\hat{\beta} - \hat{\beta}_{(\wedge 1)}) \right\|^2}{(p+1)\hat{\sigma}^2} = \frac{e_i^2}{(p+1)\hat{\sigma}^2} \frac{h_{ii}}{(1-h_{ii})^2} \qquad \frac{1-h_{ii}}{h_{ii}} D_i \sim F_{p+1,n-p-1} \tag{3.126}$$

  Comment:

$$D_i = \frac{e_i^2}{(p+1)\hat{\sigma}^2} \left[\frac{h_{ii}}{(1-h_{ii})^2} \right] = \frac{1}{p+1} \frac{h_{ii}}{1-h_{ii}} \times r_i^2 \tag{3.127}$$

  where $\dfrac{1}{p+1} \dfrac{h_{ii}}{1-h_{ii}}$ correponds to hige leverage, and r_i^2 correponds to outliers, multiply to get influentials.

  Declare the i^{th} case an influential if

$$D_i > \frac{4}{n} \tag{3.128}$$

  Or conduct F-test using the distribution of D_i, with $\alpha \sim 20\%$.

- Diagnostics to **Influential**: Studentized DiFference in BETA estimates (DFBETAS). Use equation 3.56 ~ page 82, define

$$var(\hat{\beta}_k) = \sigma^2 (X'X)^{-1}_{kk} := \sigma^2 c_{kk} \tag{3.129}$$

  And studentize difference in $\hat{\beta}$ with i^{th} case removed: $\hat{\beta}_k - \hat{\beta}_{k(\wedge i)}$

$$\text{DFBETAS}_{k(\wedge i)} = \frac{\hat{\beta}_k - \hat{\beta}_{k(\wedge i)}}{\sqrt{\text{MSE}_{(\wedge i)} c_{kk}}}, \; k = 1, 2, \ldots, p \tag{3.130}$$

[16] Proof uses equation 3.110 ~ page 93.

Declare the i^{th} case an influential if

$$\begin{cases} \text{DFBETAS}_i > 1 & \text{small/medium data} \\ \text{DFBETAS}_i > \dfrac{2}{\sqrt{n}} & \text{large data} \end{cases} \tag{3.131}$$

▷ R. **Code**

```r
rstudent(lmfit)
library(car)
outlierTest(lmfit)

hatvalues(lmfit)

cooks.distance(lmfit)
plot(lmfit,which=4)

dfbetas(lmfit)
```

Leverage and Mahalanobis Distance:

Mahalanobis Distance between X and Y as defined in equation 4.29 ~ page 118

$$d_M(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1}(\vec{x}-\vec{\mu})} \tag{3.132}$$

And we can proof d_M of a case item $X_{i\cdot} = (1, X_{i1}, X_{i2}, \dots, X_{ip})$ is[17]

$$d_M^2(X_{i\cdot}) = (n-1)(h_{ii} - \frac{1}{n}) \tag{3.134}$$

here $S = \underset{(p+1)\times(p+1)}{S}$. Note that L.H.S. ≥ 0, thus it's also an evidence that $h_{ii} \geq \dfrac{1}{n}$

3.4.5 Extra Sum Of Square

Def. Extra SS: the part of SSE explained by a new X_2 when adding to model $Y \sim X_1$:

$$\text{SSR}(X_2|X_1) = \text{SSE}(X_1) - \text{SSE}(X_1, X_2) = \text{SSR}(X_1, X_2) - \text{SSR}(X_1) \tag{3.135}$$

where $\text{SS}(\cdot)$ represents the SS when the model contains variable $\cdot$.[18]

(The following part use model (Y, X_1, X_2) as example.)

[17] Proof hint: use lemma

$$(A+B)^{-1} = A^{-1} - \frac{A^{-1}BA^{-1}}{1+tr(B^{-1}A)}, \quad \text{rank}(B) = 1 \tag{3.133}$$

and note that $X_{:,1} = \mathbf{1}_n$

[18] $\text{SSE}(1) = \text{SST}$, where 1 correponds to intercept.

We could use extra SS to examine the proper regression model: examine the `F value` and `Pr(>F)` in the output.

▷ R. **Code**

```r
lm(y~x1+x2+x1:x2) %>% anova
```

Note: Three types of SS

Term	Type I SS[19]	Type II SS	Type III SS
X_1	$\text{SSR}(X_1)$	$\text{SSR}(X_1\|X_2)$	$\text{SSR}(X_1\|X_2, X_1X_2)$
X_2	$\text{SSR}(X_2\|X_1)$	$\text{SSR}(X_2\|X_1)$	$\text{SSR}(X_2\|X_1, X_1X_2)$
X_1X_2	$\text{SSR}(X_1X_2\|X_1, X_2)$	Assume no interaction term	$\text{SSR}(X_1X_2\|X_1, X_2)$
Lang.Func	`R.anova`	`python.`	`SPSS,SAS,R.lm`

To get Type II and III anova, use `Anova(lmfit,type='III')` in `'car'` package.

Hierarchical Principle: the interaction term X_1X_2 should always come in **after** marginal term X_1 and X_2.

▷ R. **Code**

```r
libaray('car')
Anova(lmfit,type='II')
Anova(lmfit,type='III')
```

3.4.6 Hypotheses Testing to Slope

Main focus: whether the linear relation exist:

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0 \longleftrightarrow H_1 : \exists \beta_i \neq 0,\ i = 1, 2, \ldots, p \tag{3.136}$$

As for general case $H_0 : \underset{q\times(p+1)}{C}\beta - \underset{(p+1)\times 1}{t} = 0$, use General Linear Test F.

- ANOVA F-Test:

  We can examine

  $$F = \frac{\text{MSR}}{\text{MSE}} \sim F_{p,n-p-1} \tag{3.137}$$

- General Linear Test (GLT)

  First we introduce the examine models:

  – Full model: Include all variable/parameters to be examined, with p variables.

  $$Y = X\beta + \varepsilon \tag{3.138}$$

  And define SSE_F with $dof_\text{F} = n - p - 1$ under Full Model.

– Reduced model: Apply the Null Hypothesis to Full Model, with $\tilde{p}$ variables

$$Y_i = \tilde{X}\tilde{\beta} + \varepsilon \tag{3.139}$$

And define SSE_R with $dof_\text{R} = n - \tilde{p} - 1$ under Reduced Model.

Then conduct test to the difference between Full model and Reduced model through SSE_F and SSE_R.

– One dimensional case: $H_0 : \beta_1 = 0$

Examine

$$F = \frac{(\text{SSE}_\text{R} - \text{SSE}_\text{F})/(dof_\text{R} - dof_\text{F})}{\text{SSE}_\text{F}/dof_F} \sim F_{1,n-2} \tag{3.140}$$

▷ R. **Code**

```r
fullmodel <- lmfit
nullmodel <- lm(y ~ 1,df)
anova(nullmodel,fullmodel)
```

– General case: Test $H_0 : \underset{q\times(p+1)}{C}\beta - \underset{(p+1)\times 1}{t} = 0$, construct F statistics as

$$F = \frac{(C\hat{\beta} - t)'\left[C(X'X)^{-1}C'\right]^{-1}(C\hat{\beta} - t)}{q\hat{\sigma}^2} \sim F_{q,n-q-1} \tag{3.141}$$

- r and Different R^2:

– Pearson's r:

$$r_{Y,\hat{Y}} = c\hat{o}v(Y,\hat{Y}) = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(\hat{Y}_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}\sqrt{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y} - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{3.142}$$

– Coefficient of Multiple Determination R^2:

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}} \tag{3.143}$$

– Adjusted R^2:

$$R_\text{a}^2 = 1 - \frac{\text{MSE}}{\text{MST}} = 1 - \frac{n-1}{n-p-1}\frac{\text{SSE}}{\text{SST}} \tag{3.144}$$

Relation between r and R^2: Under Simple Linear Model, we have

$$R^2 = r^2 \tag{3.145}$$

Relation between R^2 and F-Statistic:

$$F = \frac{R^2}{1-R^2}\frac{n-p-1}{n-1} \sim F_{n-1,n-p-1} \tag{3.146}$$

Hypothesis testing for r:

$$t = \frac{r}{\sqrt{1-r^2}}\sqrt{n-2} \sim t_{n-p-1} \tag{3.147}$$

▷ R. **Code**

```r
cor.text(df$x,df$y)
```

- Coefficient of Partial Determination $R^2_{Yk|\wedge X_k}$ and Coefficient of Multiple Determination R^2: CMD reflects the interpretability of the model, to examine the interpretability of each variable, use coef. partial determination

$$R^2_{YX_k|X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_p} = R^2_{YX_k|\wedge X_k} = \frac{\mathrm{SSR}(X_k|X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_p)}{\mathrm{SSE}(X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_p)} = \frac{\mathrm{SSR}(X_k|\wedge X_k)}{\mathrm{SSE}(\wedge X_k)} \tag{3.148}$$

Note: Coef. Partial determination can also be used for X_i, X_j: $R^2_{X_iX_j|\wedge X_i,X_j}$

Sometimes we use $\eta^2_k = R^2_{YX_k|\wedge k} = R^2_{Yk.\wedge k}$

- Coefficient of Partial Correlation η_k: Measures the strength of linear relation, $\pm$ sign depend on posi./nega. correction.

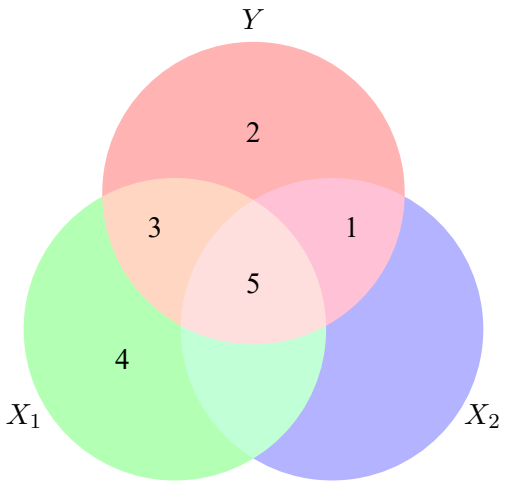
$$\eta_k = \pm\sqrt{\eta^2_k} \tag{3.149}$$

▷ R. **Code**

```r
library('heplots')
etasq(lmfit)
```

3.4.7 Diagnostics to Multi-colinearity

- Venn Diagram for Multi-Linear Regression: Used to show the interpretability of variables.



Explanation of each region:

  - 1/3: Variation in Y uniquely attributes to X_2/X_1;
  - 2: Variation in Y that cannot be explained by regression to X_1, X_2, corresponds to ε;

– 5: Cross term of X_1, X_2, **cannot** verify the orientation, corresponds to **Multi-colinearity**.

In the presence of multi-colinearity, i.e. X is column singular ($\frac{S_5}{S_1 \text{ or } S_3}$ large), the regression parameter

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{3.150}$$

Issue of multi-collinearity:

- Statistically:'better' prediction, worse interpretability;
- Numerically: Calculation of $(X'X)^{-1}$ becomes unstable/ill-posed/NAN.

□ **Use Variance Inflation Factor (VIF) to detect multi-collinearity.**

First construct R_k^2, $k = 1, 2, \dots p$: Regress X_k against other $p-1$ variable X_is and get corresponding R_k^2, and

$$\text{VIF}_k = (1 - R_k^2)^{-1} \tag{3.151}$$

$$\overline{\text{VIF}} = \frac{1}{p}\sum_{k=1}^{p} \text{VIF}_k \tag{3.152}$$

If $\|\text{VIF}_i\|_\infty > 10$ or $\overline{\text{VIF}} > 1$, then we identify an excessive multi-colinearity.[20]

▷ R. **Code**

```r
library('car')
vif(lmfit)
```

3.4.8 Diagnostics to Model Variable Selection

In Multi-variate regression, proper explanatory variables form a subset of all available variables.

Aim: Avoid over-fitting, get a simple explanatory model.

Comment: If we consider the model with all $p_{\max}$ variables as full, unbiased model, then model selection is a kind of **Bias-Variance Trade-Off**.

□ **Model Validation: k-Fold Cross Validation(CV):**

1. Separate the dataset size n into k parts;
2. pick the i^{th} part as test set y_i, and the other $k-1$ part as train set $y_{\wedge i}$(to conduct regression, etc); then conduct prediction of model $y_{\wedge i}$ to part y_i ane get MSE_i;
3. Take average of MSE_i as the measure of validity.

[20] Why $\text{VIF}_k = \frac{1}{R_k^2}$ is called 'variance inflation factor'? We can prove that

$$var(\hat{\beta}_k) = \frac{\sigma^2}{(n-1)S_{x_k}^2} \cdot \frac{1}{R_k^2} = \frac{\sigma^2}{(n-1)S_{x_k}^2} \cdot \text{VIF}_k \tag{3.153}$$

□ **Evaluation Criteria**

Useful model validation approach: To check a model with $p-1$ variable (this part p for 1+# variable)

- Traditional way: Test r, R^2, R_a^2, p-value, etc.
- Mallow's C_p: For a model with p variable:

$$\hat{Y}^p = X_p(X_p'X_p)^{-1}X_p'Y = H_pY \tag{3.154}$$

Denote:

$$\mathbb{E}(\hat{Y}^p) = H_p\mathbb{E}(Y) \equiv H_p\mu \qquad var(\hat{Y}^p) = H_p\sigma^2 I_n H_p' = \sigma^2 H_p \tag{3.155}$$

Recall the MSE expansion of bias-variance trade-off in equation 2.51 ~ page 44[21]

$$\sum_{i=1}^n \mathbb{E}[(\hat{Y}_i^p - \mu_i)^2] = \sum_{i=1}^n \mathbb{E}[(\hat{Y}_i^p) - \mu_i]^2 + \sum_{i=1}^n var(\hat{Y}_i^p) \tag{3.162}$$

$$\Rightarrow = \mathbb{E}(\text{SSE}(p)) - (n-2p)\sigma^2 \tag{3.163}$$

Sum Squared Prediction Error (SSPE):

$$\Gamma_0 \equiv \frac{\sum_{i=1}^n \mathbb{E}[(\hat{Y}_i^p - \mu_i)^2]}{\sigma^2} = \frac{\mathbb{E}(\text{SSE}(p))}{\sigma^2} - (n-2p) \tag{3.164}$$

And construct Mallow's C_p: Estimation of Γ_p

$$C_p = \hat{\Gamma}_p = \frac{\mathbb{E}(\text{SSE}(p))}{\hat{\sigma}^2} - (n-2p) \tag{3.165}$$

where $\text{SSE(p)} = Y'(I-H_p)Y$. When the model is unbiased, then we should have $\mathbb{E}(\text{SSE}(p))/\hat{\sigma}^2 \to n-p$. C_p-p plot could be used to pick a proper p:

- $C_p \approx p$: Model unbiased, then choose model with smaller C_p;

[21] Derivtion:

- Bias part: (Here use equation 4.67 ~ page 123 in 3rd line; use equation 3.63 ~ page 83 in 4th line.)

$$\sum_{i=1}^n [\mathbb{E}(\hat{Y}_i^p - \mu_i)]^2 = \mu'(H_p - I)'(H_p - I)\mu \tag{3.156}$$

$$= \mu(I - H_p)\mu' \tag{3.157}$$

$$= \mathbb{E}(Y'(I-H_p)Y) - tr[(I-H_p)\sigma^2] \tag{3.158}$$

$$= \mathbb{E}(\text{SSE}(p)) - (n-p)\sigma^2 \tag{3.159}$$

- Variance part:

$$\sum_{i=1}^n var(\hat{Y}_i^p) = tr(var(\hat{Y}^p)) = \sigma^2 tr(H_p) = p\sigma^2 \tag{3.160}$$

Then

$$\frac{\sum_{i=1}^n \mathbb{E}[(\hat{Y}_i^p - \mu_i)^2]}{\sigma^2} = \frac{\mathbb{E}(\text{SSE}(p))}{\sigma^2} - (n-2p) \tag{3.161}$$

  - $C_p \gg p$: Significant biased, miss some important predictors;
  - $C_p \ll p$: Overfitting.

- Akaike Information Criterion (AIC): Euivalent to Mallow's C_p for gaussion regression model.

$$\mathrm{AIC}(p) = -2\log(\hat{L}) + 2p \tag{3.166}$$

  where $\hat{L}$ is the maximum likelihood, for linear regression case

$$\mathrm{AIC}(p) = n\log\left(\frac{\mathrm{SSE}(p)}{n}\right) + 2p \tag{3.167}$$

  Select the model that minimizes $\mathrm{AIC}(p)$.

- Bayesian Information Criterion (BIC)/Schwarz's Bayesian Criterion (SBC):

$$\mathrm{BIC}(p) = -2\log(\hat{L}) + p\log n \tag{3.168}$$

  where $\hat{L}$ is the maximum likelihood, for linear regression case

$$\mathrm{BIC}(p) = n\log\left(\frac{\mathrm{SSE}(p)}{n}\right) + p\log n \tag{3.169}$$

  Select the model that minimizes $\mathrm{BIC}(p)$.

- PRESS Creterion (Predictive Residual Error Sum of Squares): A kind of within-model cross validation

$$\mathrm{PRESS}(p) = \sum_{i=1}^{n}(Y_i - \hat{Y}_{i(\wedge i)})^2 \tag{3.170}$$

  where

$$\hat{Y}_{i(\wedge i)} = (1, X_{i1}, \ldots, X_{ip})\hat{\beta}_{(\wedge i)} \tag{3.171}$$

$$\hat{\beta}_{(\wedge i)} = (X'_{(\wedge i)}X_{(\wedge i)})^{-1}X'_{(\wedge i)}Y_{(\wedge i)} \tag{3.172}$$

  where $\hat{\beta}_{(\wedge i)}$ as in EqaEstimatorWithWedgeX, is the estimated β with (X_i, Y_i) removed from X.[22]

  Select the model that minimizes $\mathrm{PRESS}(p)$.

▷ R. **Code**

```r
library('leaps')
predictor <- df[,c('...','...',...)]
response <- df[,...]
leapSet <- leaps(x=predictor, y=response, nbest = ...)
```

[22] A useful thm.: Deleted Residual

$$d_i := Y_i - \hat{Y}_{i(\wedge i)} = \frac{e_i}{1 - h_{ii}} \tag{3.173}$$

```
# method=c('Cp','adjr2','r2')
leapSet$which[which.min(leapSet$Cp),]
```

`nbest` for `NUMBER_OF_BEST_MODELS`

Section 3.5 Remedies

3.5.1 Variable Transformation

The goal of Transformation:

- Stablize Variance;
- Improve Normality;
- Simplify the Model.

☐ **Variance Stabilizing Transformations:**

With $\mathbb{E}(Y|X) = \mu_X$, variance of Y might be expressed by a function of expected value, i.e. $var(Y|X) := h(\mu_X)$, which we are trying to stablize.

Take transformation $Y \mapsto f(Y)$ such that variance is stablized: (with delta method approximation here)

$$var(f(Y)) \approx \big(f'(\mu_X)\big)^2 h(\mu_X) = \text{const}$$

which gives the stablizing transform:

$$f(\mu) = \int \frac{c\,\mathrm{d}\mu}{\sqrt{h(\mu)}} \tag{3.174}$$

Examples:

$$h(\mu) = \mu^2 \Rightarrow f(\mu) = \ln \mu \tag{3.175}$$

$$h(\mu) = \mu^{2\nu} \Rightarrow f(\mu) = \mu^{1-\nu} \tag{3.176}$$

☐ **Box-Cox Transformation:**

Take

$$Y^* = \frac{Y^\lambda - 1}{\lambda} \tag{3.177}$$

Examples:

$$\lambda = 1 \Rightarrow Y^* \sim Y \tag{3.178}$$

$$\lambda = 0.5 \Rightarrow Y^* \sim \sqrt{Y} \tag{3.179}$$

$$\lambda = 0 \Rightarrow Y^* \sim \ln Y \tag{3.180}$$

$$\lambda = -1 \Rightarrow Y^* \sim 1/Y \tag{3.181}$$

And conduct regression to model

$$Y^* = \beta_0 + \beta_1 X + \varepsilon_i \tag{3.182}$$

Likelihood Function

$$L(\beta,\sigma^2;\lambda)=\frac{1}{(2\pi\sigma^2)^{n/2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i^*-\beta_0-\beta_1X_i)^2\right)J(\frac{\partial Y^*}{\partial Y}) \tag{3.183}$$

where the Jacobi Matrix denoted in Geometric Mean $\mathrm{GM}(Y)=\prod_{i=1}^{n}Y_i^{1/n}$

$$J(\frac{\partial Y^*}{\partial Y})=\prod_{i=1}^{n}Y_i^{\lambda-1}=\mathrm{GM}(Y)^{n(\lambda-1)} \tag{3.184}$$

MLE Estiamtor:

$$\hat{\beta}^*=(X'X)^{-1}X'Y^* \tag{3.185}$$

$$\hat{\sigma}_n^2=\frac{1}{n}\mathrm{SSE}^* \tag{3.186}$$

$$\mathrm{SSE}^*=\sum_{i=1}^{n}(Y_i^*-\hat{Y}^*)^2 \tag{3.187}$$

And when β, σ^2 take MLE estimator, $L(\beta,\sigma^2;\lambda)$ can be regarded a function of λ:

$$\ln L(\beta,\sigma^2;\lambda)=l(\lambda)=-\frac{n}{2}\ln\frac{\hat{\sigma}_n^2}{\mathrm{GM}(Y)^{2(\lambda-1)}}+\text{const} \tag{3.188}$$

For simplification, denote $Z=Y^*/J^{1/n}$ and get

$$\ell(\lambda)=-n\ln\sigma_{n_Z}^2+\text{const} \tag{3.189}$$

where

$$Z_i^*=\begin{cases}\dfrac{Y_i^\lambda-1}{\lambda}\dfrac{1}{\prod_{k=1}^{n}Y_k^{\frac{\lambda-1}{n}}}, & \lambda\neq 0\\ \ln Y_i\prod_{k=1}^{n}Y_k^{\frac{1}{n}}, & \lambda=0\end{cases} \tag{3.190}$$

Plot $l(\lambda)$-λ to determine a proper λ and transform $Y^*=\dfrac{Y^\lambda-1}{\lambda}$:

- Selected λ should be closed to $\lambda_{\arg\max l}$, at least within CI[23]

$$\{\lambda|l(\lambda)\geq l(\lambda_{\arg\max l})-\frac{1}{2}\chi^2_{1,1-\alpha}\} \tag{3.191}$$

- Should pick a λ which is **Interpretable**. e.g. If $\lambda=1$ is within range $[0.94,1.08]$, then take $\lambda=1$ (does not transform).

▷ R. **Code**

```
library(MASS)
bctrans <- boxcox(y~x,df,lambda = seq(-1.5, 1.5, length = 15))
bctrans$x[which.max(bctrans$y)]
```

Note: we can transform on X or Y or simultaneously to get better regression model.

[23] Here CI can be derived using Wilk's Theorem

3.5.2 Weighted Least Squares Regression

To deal with heterogeneous variance, use Weighted Least Squares (WLS) instead of OLS: Minimizing

$$\sum_{i=1}^{n} e_i^2 \longrightarrow \sum_{i=1}^{n} w_i e_i^2 \tag{3.192}$$

And e.g. take weight for each case as

$$w_i = \frac{1}{\sigma_i^2} \tag{3.193}$$

Solution:

$$\hat{\beta}_W = (X'WX)^{-1}X'WY \tag{3.194}$$

▷ R. **Code**

```
1 Wlmfit <- lm(y~x,weights=WEIGHT_VECTOR,data=df)
```

3.5.3 Remedies for Model Variable Selection & More Regression Model

□ **Variable Selection Methods**

Several Algorithm to search for best variable set:

- Exhaustive Search and Test (usually through Mallow's C_p, see equation 3.165 ~ page 100): Used for $p \leq\sim 30$
- Greedy Search: Get a locally optimal solution.
  - Forward Selection: Start with $p = 0$, add one variable each times and conduct $t/F/p$-value test until a presupposed certain limit.
  - Backward Elimination: Start with $p_{\max}$, eliminate one variable each times and conduct $t/F/p$-value test until a presupposed certain limit.
  - Stepwise Regression: Alternate forward selection & backward elimination until no add/elimination.

□ **Regression with Penalty Term / Regularization**

Recall: OLS regression model: Minimize SSE[24]

$$\hat{\beta} = \arg\min \|Y - X\beta\|_2^2 \tag{3.195}$$

Idea: Add a penalty term in SSE, such that SSE increases with # of variables/value of variables. More about this 'loss + penalty' form optimization see section 9.1 ~ page 243 and section 9.4.5 ~ page 258.

- LASSO (Least Absolute Shrinkage and Selection Operator)

  Penalty term: $\lambda\|\beta\|_1$, where λ is a proper penalty parameter.

$$\hat{\beta} = \arg\min(\|Y - X\beta\|_2^2 + \lambda \|\beta\|_1) \tag{3.196}$$

[24] Here expressed in ℓ_p norm, definition see sec.4.1.2, Norm

or equivlantly expressed as [25]

$$\hat{\beta} = \arg\min \|Y - X\beta\|_2^2, \text{ with } \|\beta\|_1 \leq s \tag{3.197}$$

where s is a parameter correponding to λ. Select a proper value of λ(or equivlantly s) for expected model: Some $\hat{\beta}_i$ would be exactly 0.

- Ridge Regression/Tikhonov Regularization:

  Penalty term: $\lambda\|\beta\|_2^2$, where λ is penalty parameter.

$$\hat{\beta} = \arg\min(\|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2) \tag{3.198}$$

  or equivlantly expressed as

$$\hat{\beta} = \arg\min \|Y - X\beta\|_2^2, \text{ s.t. } \|\beta\|_2^2 \leq s \tag{3.199}$$

  Select a proper value of λ (or equivlantly s) for expected model. Generally Ridge regression **cannot** conduct variable selection, but usually used to avoid non-invertible $X'X$, or used to retain important but collinear variable.

  Solution of Ridge regression:[26]

$$\hat{\beta}_\lambda^{\text{Ridge}} = (X'X + \lambda I)^{-1}X'Y \tag{3.202}$$

  A Bayesian point of view for Ridge regreession see section 13.4.9 ~ page 357

- Mixed Model: Elastic Net

$$\hat{\beta} = \arg\min(\|Y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2) \tag{3.203}$$

  or equivalent form:

$$\hat{\beta} = \arg\min_\beta \|Y - X\beta\|^2 \tag{3.204}$$

$$s.t. \frac{\lambda_1}{\lambda_1 + \lambda_2}\|\beta\|_1 + \frac{\lambda_2}{\lambda_1 + \lambda_2}\|\beta\|_2^2 \leq s \tag{3.205}$$

  picking proper hyper-parameter $(s, \lambda = \dfrac{\lambda_2}{\lambda_1 + \lambda_2})$

[25] Constrained optimization theory intro see section 5.1.4 ~ page 147.

[26] Why Ridge regression can also fix the problem of colinearity, i.e. non-full rank XX':
Assume the SVD decomposition of X: $X = U\Sigma V'$, then

$$X'X + \lambda I = V\Sigma U'U\Sigma V' + \lambda I \tag{3.200}$$

$$= V\begin{bmatrix} \sigma_1^2 + \lambda & 0 & \dots & 0 \\ 0 & \sigma_2^2 + \lambda & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{p+1}^2 + \lambda \end{bmatrix} V' \tag{3.201}$$

then for $\lambda > 0$, we can get a positive-definite matrix $X'X + \lambda I$

▷ R. **Code**

```r
library('MASS')
Rfit <- lm.ridge(y~x,lambda=seq(0,0.1,0.001),data=df)
summary(Rfit)
whichLambda <- which.min(Rfit$GCV)
coef(fits)[whichLambda,]

library('lars')
Lfit <- lars(x,y,type='lasso')
summary(Lfit)
whichCp <- which.min(Lfit$Cp)
Lfit$Cp[whichCp]
Lfit$beta[whichCp,]
```

☐ Non-parametric Regression Model

Add smooth/penalty function. e.g. `loess` (Locally Regression), `lowess` (Locally Weighted ScatterPlot Smoother), Regression Tree.

☐ Other Regression Model

- Standardized Regression Model For regression model $Y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i,\ i = 1, 2, \ldots, n$, conduct Standardization (with an extra const $1/\sqrt{n-1}$) to Y and X.

$$Y_i^* = \frac{1}{\sqrt{n-1}} \frac{Y_i - \bar{Y}}{s_Y} \qquad X_{ij}^* = \frac{1}{\sqrt{n-1}} \frac{X_{ij} - \bar{X}_i}{s_{X_i}} \qquad \varepsilon_i^* = \frac{1}{\sqrt{n-1}} \frac{\varepsilon_i - \bar{\varepsilon}}{s_Y} \tag{3.206}$$

And the regreeesion model for standardized data:

$$Y_i^* = 0 + \sum_{j=1}^{n} X_{ij}^* \beta_j^* + \varepsilon_i^* \tag{3.207}$$

with

$$\beta_j^* = \frac{\beta_j s_{X_j}}{s_Y} \tag{3.208}$$

Note: set the const as $\sqrt{n-1}$ so that

$$r_{X^*X^*} = X^{*T} X^* \qquad r_{Y^*X^*} = X^{*T} Y^* \tag{3.209}$$

▷ R. **Code**

```r
scaledf <- data.frame(scale(df))
scalelmfit <- lm(~,scaledf)
summary(scalelmfit)
```

- Polynomial Regression Model

  ▷ R. **Code**

```
polfit <- lm(y~x+I(x^2),df)
polfit <- lm(y~polym(x1,x2,degree=),df)
```

- Interaction Model

  Example:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon \tag{3.210}$$

  Re-write as

$$Y = \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \varepsilon \tag{3.211}$$

$$Y = \beta_0 + \beta_1 X_2 + (\beta_2 + \beta_3 X_1) X_2 + \varepsilon \tag{3.212}$$

  test the regression coefficient dependence on another variable.

- Kernel Regression: see section 9.4.5 ~ page 258.

Section 3.6 Factor Analysis of Variance

Here are some basic introduction to factor model. For more knowledge see Chapter 8 ~ page 232 and Chapter 14 ~ page 360

3.6.1 Single Factor Model

Single factor, or one-way analysis of variance focuses on continuous $Y \sim$ categorical X (numeric-factor). Regression goal is the mean response of each category π_i: whether & how much they are different.

Basic assumptions: Normal within each categories + Equal variance + Independent

Model: See equation 3.10 ~ page 74 expression for single factor model

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2) \tag{3.213}$$

where τ_i for group effect, $\mu_i = \mu + \tau_i$ for factor effect. Originally only μ_i are estimatable.

▷ R. **Code**

`lm()` in R. uses cell means model, returns $\mu_i = \mu + \tau_i$ for each categories.

```
facfit <- lm(y~x,df) # where x should be as.factor() type
```

□ **Statistical Inference to Individual** μ, τ_i

Note: Initially we have $r+2$ variable $(\mu, \tau_{i=1}^r, \sigma^2) \Rightarrow$ estimator not unique. So we use a constraint to cancel the extra degree of freedom.

$$\sum_{i=1}^{r} c_i \tau_i = 0 \tag{3.214}$$

usually we take $c_i = \delta_{1i}$ (cell mean model) / $c_i = 1$ (factor effect model) / $c_i = n_i$.

- Cell means model solution for $c_i = \delta_{i1}$, i.e. $\tau_1 = 0$. In this case we directly express $\mu_i = \mu + \tau_i$

$$\hat{\mu}_i = \bar{Y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij}$$

- Factor effect solution for $c_i = 1$, i.e. $\sum_{i=1}^{r} \tau_i = 0$ (used most often, and is **used in the following parts in this chapter**).

$$\hat{\mu} = \frac{1}{r}\sum_{i=1}^{r} \bar{Y}_i = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n_i} \frac{Y_{ij}}{n_i} \tag{3.215}$$

$$\hat{\tau}_i = \bar{Y}_i - \hat{\mu} \tag{3.216}$$

- Factor effect solution for $c_i = n_i$, i.e. $\sum_{i=1}^{r} n_i\tau_i = 0$

$$\hat{\mu} = \bar{Y} = \frac{1}{n_T}\sum_{i,j} Y_{ij} \tag{3.217}$$

$$\hat{\tau}_i = \bar{Y}_i - \bar{Y} = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} - \hat{\mu} \tag{3.218}$$

□ One-Way ANOVA

ANOVA table in the form of $r = p + 1$ multivariate ANOVA in page 84

Source	dof	SS	MS	F-Statistic
SSRegression	$r-1$	$\sum_{i=1}^{r}(\hat{Y}_i - \bar{Y})^2$	SSR/dof_R	MSR/MSE
SSError	$n_T - r$	$\sum_{i=1}^{r}\sum_{j=1}^{n_i}(Y_{ij} - \hat{Y}_i)^2$	SSE/dof_E	
SSTotal	$n_T - 1$	$\sum_{i=1}^{r}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y})^2$	SST/dof_T	

Use MSE as estimator of σ^2:

$$\hat{\sigma^2} = \frac{1}{n_T - r}\sum_{i=1}^{r}\sum_{j=1}^{n_i}(Y_{ij} - \hat{Y}_i)^2 = \frac{1}{n_T - r}\left[\sum_{i=1}^{r}\sum_{j=1}^{n_i} Y_{ij}^2 - \sum_{i=1}^{r}\frac{\bar{Y}_i^2}{n_i}\right] \tag{3.219}$$

Also F-statistics for $H_0 : \tau_1 = \tau_2 = \ldots = \tau_r = 0$

$$F = \text{MSR/MSE} = \frac{\text{SSR}/(r-1)}{\text{SSE}/(n_T - r)} \sim F_{r-1, n_T - r}, \text{ under } H_0 \tag{3.220}$$

□ Statistical Inference to Group Difference

We usually focus on 'difference' between factor effects, general form

$$\phi = \sum_{i=1}^{r} \xi_i\tau_i, \qquad \sum_{i=1}^{r} \xi_i = 0 \tag{3.221}$$

where ϕ with $\sum_{i=1}^{r} \xi_i = 0$ is called a **contrast**. Assume there are m estimator ϕ_k, $k = 1, 2, \ldots, m$

$$\underset{m\times 1}{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{bmatrix} = \underset{m\times r}{\xi}\ \underset{r\times 1}{\tau} = \begin{bmatrix} \xi_{11} & \xi_{12} & \dots & \xi_{1r} \\ \xi_{21} & \xi_{22} & \dots & \xi_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{m1} & \xi_{m2} & \dots & \xi_{mr} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_r \end{bmatrix} \tag{3.222}$$

Sampling distribution of $\hat{\phi}_k = \sum_{i=1}^{r} \xi_{ki}\hat{\tau}_i$, with $\sum_{i=1}^{r} \xi_i = 0$:

$$\phi_k = \sum_{i=1}^{r} \xi_{ki}\tau_i \sim N(\sum_{i=1}^{r} \xi_{ki}\bar{Y}_i, \sigma^2 \sum_{i=1}^{r} \frac{\xi_{ki}^2}{n_i}) \tag{3.223}$$

Or use transform of multivariate normal in equation 4.66 ~ page 123

$$\phi \sim N_m(\xi\tau, \sigma^2\xi\xi') \tag{3.224}$$

with which we could coustruct corresponding interval. Here are some useful methods.

- Bonferroni's Confidence Region for $\underset{m\times 1}{\phi}$, using result in equation 4.110 ~ page 128

$$R(\phi) = \bigotimes_{k=1}^{m} \left(\sum_{i=1}^{r} \xi_{ki}\bar{Y}_i \pm \hat{\sigma} t_{n_T - r, \frac{\alpha}{2m}} \sqrt{\sum_{i=1}^{r} \frac{\xi_{ki}^2}{n_i}} \right) \tag{3.225}$$

- Scheffè's Confidence Region for $\underset{1\times 1}{\phi}$:

$$R(\phi) = \sum_{i=1}^{r} \xi_i\bar{Y}_i \pm \hat{\sigma}\sqrt{(r-1)F_{r-1,n_T-r,\alpha}} \sqrt{\sum_{i=1}^{r} \frac{\xi_{ki}^2}{n_i}} \tag{3.226}$$

- Tukey's HSD Confidence Region for $\underset{1\times 1}{\phi}$, under condition $n_1 = \ldots = n_r = n$: focus on estimating $\tau_i - \tau_j$
  - Def.: studentized range distribution: for $Z_1, \ldots, Z_n$ i.i.d. $\sim N(0,1)$, $mW^2 \sim \chi_m^2$, then

$$q = \frac{\max Z_i - \min Z_i}{W} \sim q_{n,m} \tag{3.227}$$

  Then confidence interval for $\phi = \tau_i - \tau_j$

$$R(\phi) = \bar{Y}_i - \bar{Y}_j \pm q_{r,n_T-r,\alpha}\frac{\hat{\sigma}}{\sqrt{n}} \tag{3.228}$$

  General case: $\phi = \sum_{i=1}^{r} \xi_i\tau_i$

$$R(\phi) = \sum_{i=1}^{r} \xi_i\bar{Y}_i \pm q_{r,n_T-r,\alpha}\frac{\hat{\sigma}}{2\sqrt{n}} \sum_{i=1}^{r} |\xi_i| \tag{3.229}$$

Comment: Scheffè is more conservative, i.e. shorter. If confidence interval does not include 0, we can say they are significantly different. More about theory behind these confidence region see section 14.1.3 ~ page 361-Multiple Comparison

▷ R. **Code**

```r
library('agricolae')
facaov <- aov(y~0+x,df)

LSD.test(facaov,trt='design',group=FALSE,console=TRUE)

scheffe.test(facaov,trt='design',group=FALSE,console=TRUE)

TukeyHSD(facaov,conf.level=0.95)
```

use `plot()` to view interval estimation

3.6.2 Double Factor Model

Double factor, or two-way analysis of variance, categories π_{ij}:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + e_{ijk} \tag{3.230}$$

OLS estimator with $\sum_{i=1}^{a} \alpha_i = 0$, $\sum_{j=1}^{b} \beta_j = 0$:

$$\hat{\mu} = \frac{1}{ab} \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n_{ij}} \frac{Y_{ijk}}{n_{ij}} \tag{3.231}$$

$$\hat{\alpha}_i = \frac{1}{b} \sum_{j=1}^{b} \sum_{k=1}^{n_{ij}} \frac{Y_{ijk}}{n_{ij}} - \hat{\mu} \tag{3.232}$$

$$\hat{\beta}_j = \frac{1}{a} \sum_{i=1}^{a} \sum_{k=1}^{n_{ij}} \frac{Y_{ijk}}{n_{ij}} - \hat{\mu} \tag{3.233}$$

MSE estimator of σ^2:

$$\hat{\sigma^2} = \frac{1}{n_T - a - b + 1} \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n_{ij}} (Y_{ijk} - \bar{Y}_{ij})^2 = \frac{1}{n_T - a - b + 1} \left[\sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n_{ij}} Y_{ijk}^2 - \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{\bar{Y}_{ij}^2}{n_{ij}} \right] \tag{3.234}$$

More about theory of Factor model or Dealing with complicted cases see Chapter 8 ~ page 232 and Chapter 14 ~ page 360.

Section 3.7 Generalized Linear Model

Recall: Linear model with normal assumption can be expressed as :

$$Y_i \sim N(\mu_i, \sigma_i^2) = N(x_i'\beta, \sigma_i^2) \tag{3.235}$$

Question: How to generalize the simple linear model?

- Generalize the distribution

- Generalize the dependent mode

□ **Distribution Generalize: Scaled Exponential Family** For different range and feature of Y we can use different distribution for regression. We usually use Exponential Family distribution $f(y;\vec{\theta},\vec{\phi})$ as in equation 2.18 ~ page 40, with some constraint on subfunctions for better distribution properties, written as linear scaled exponential family:

$$f(y;\vec{\theta},\vec{\phi}) = \exp\left\{\frac{y'\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\} \tag{3.236}$$

where $\vec{\theta}$ is the canonical parameter for location and $\vec{\phi}$ for scale(usually we take $a(\phi) \propto \phi$).

Properties of $f(y;\theta,\phi)$:

- Expectation

$$\mu \equiv \mathbb{E}(Y) = \int yf(y)\,\mathrm{d}y = \int \left(a(\phi)\frac{\partial}{\partial\theta} + \frac{\mathrm{d}b(\theta)}{\mathrm{d}\theta}\right) f(y)\,\mathrm{d}y = b'(\vec{\theta}) \tag{3.237}$$

- Variance

$$\sigma^2 \equiv var(Y) = \int yy^T f(y)\,\mathrm{d}y - \mathbb{E}(Y)\mathbb{E}(Y)^T \tag{3.238}$$

$$= \int \left(\frac{\partial^2}{\partial\theta\partial\theta^T} + (b'(\theta)y + yb'(\theta)) - b'(\theta)b'(\theta)^T + a(\phi)\frac{\mathrm{d}^2 b(\theta)}{\mathrm{d}\theta\mathrm{d}\theta^T}\right) f(y)\,\mathrm{d}y - \mathbb{E}(Y)\mathbb{E}(Y)^T \tag{3.239}$$

$$= a(\phi)\frac{\mathrm{d}^2 b(\vec{\theta})}{\mathrm{d}\theta\mathrm{d}\theta^T} = a(\phi)b''(\vec{\theta}) \tag{3.240}$$

- Examples: Normal, Binomial, Poisson
  - Normal $f(y) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp\left(-\frac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)\right)$ with $\Sigma = \sigma^2 I$:

$$f(y) = \exp\left(\frac{y'\mu - \frac{1}{2}\mu'\mu}{\sigma^2} - \frac{y'y}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right) \tag{3.241}$$

    Compare with equation 3.236 ~ page 111, $\theta = \mu$: $b(\theta) = \frac{1}{2}\mu'\mu$, $a(\phi) = \sigma^2$
    - $\mathbb{E}(Y) = b'(\theta) = \mu$
    - $var(Y) = a(\phi)b''(\theta) = \sigma^2$
  - Binomial $\mathbb{P}(y) = \binom{n}{y}\pi^y(1-\pi)^{n-y} \sim B(n,\pi)$:

$$f(y) = \exp\left(y\ln(\frac{\pi}{1-\pi}) + n\ln(1-\pi) + \ln\binom{n}{y}\right) \tag{3.242}$$

    Compare with equation 3.236 ~ page 111, $\theta = \ln(\frac{\pi}{1-\pi}) \Leftrightarrow \pi = \frac{1}{1+e^{-\theta}}$: $b(\theta) = -n\ln(1-\pi) = -n\ln\frac{1}{1+e^{\theta}}$, $a(\phi) = 1$
    - $\mathbb{E}(Y) = b'(\theta) = n\ln\frac{1}{1+e^{-\theta}} = n\pi$
    - $var(Y) = a(\phi)b''(\theta) = n\pi(1-\pi)$

– Poisson $\mathbb{P}(y) = \dfrac{\lambda^y}{y!}e^{-\lambda} \sim P(\lambda)$:

$$f(y) = \exp\left(y \ln \lambda - \lambda - \ln y!\right) \tag{3.243}$$

Compare with equation 3.236 ~ page 111, $\theta = \ln\lambda \Leftrightarrow \lambda = e^{\theta}$: $v(\theta) = \lambda = e^{\theta}$, $a(\phi) = 1$

* $\mathbb{E}(Y) = b'(\theta) = \lambda$
* $var(Y) = a(\phi)b''(\theta) = \lambda$

□ **Dependent Mode Generalize: Link Function**

Note that $Y_i \sim N(\mu_i, \sigma_i^2) = N(x_i'\beta, \sigma_i^2)$ contains the dependency of μ_i on $x_i\beta$ thus we can further generalize the regression model as $\mu_i = x'\beta$, here μ_i stands for $\mathbb{E}(Y)$ as in equation 3.237 ~ page 111. However for different distributions, $\mu = \mathbb{E}(Y)$ have specific range, e.g. $\mu \in [0, n]$ for $B(n, p)$, while $x'\beta \in \mathbb{R}$, thus use a **link function** g: $I_\mu \to I_{x'\beta}$ to adjust the range:

$$x_i'\beta = g(\mu_i) \Leftrightarrow \mu_i = g^{-1}(x'\beta) \tag{3.244}$$

Note: Link function should be monodrome & differentiable such that g^{-1} exists. And here $x'\beta$ term still exist (because it's still generalized linear model), thus we denote $\eta := x'\beta$ as a linear predictor/classifier

$$\eta := x'\beta \tag{3.245}$$

Regression Model:

$$\eta_i = g(\mu_i) \Leftrightarrow \mu_i = g^{-1}(\eta_i) \tag{3.246}$$

□ **Useful Generalized Linear Model:**

Important Question: how to choose proper generalization 'pair' : Distribution & Link Function pair?

Idea: Use the expectation transform:

$$\text{Distribution: } \mu = \mathbb{E}(Y) = b'(\theta) \tag{3.247}$$

$$\text{Link Function: } \mu = g^{-1}(x'\beta) \tag{3.248}$$

Thus

$$g^{-1}(x'\beta) = b'(\theta) \Rightarrow \eta = x'\beta = g(b'(\theta)) \tag{3.249}$$

For model simplification, we can choose $g(\,\cdot\,), b(\,\cdot\,)$ such that

$$g(b'(\,\cdot\,)) = \mathrm{Id}(\,\cdot\,) \Leftrightarrow g^{-1}(\,\cdot\,) = b'(\,\cdot\,) \tag{3.250}$$

such condition is called **Canonical Link** of generalized linear model, such choise of link function makes $x'\beta$ the canonical parameter in model.

$$\theta = \eta = x'\beta = g(\mu) \rightsquigarrow g^{-1}(\theta) = g^{-1}(\eta) = g^{-1}(x'\beta) = \mu = \mathbb{E}(Y) \tag{3.251}$$

- Simle linear model: $N(\mu, \sigma^2)$, $g(\,\cdot\,) = \mathrm{Id}(\,\cdot\,)$

$$\mu_i = \eta_i \tag{3.252}$$

- Logistic Model: $B(n,\pi)$, $g(x)=\text{logit}(x)=\ln\dfrac{x}{1-x}$ $\Leftarrow g^{-1}(y)=\text{logistic}(y)=\dfrac{1}{1+e^{-y}}$

$$n\pi_i=\mu_i=g^{-1}(\eta_i) \tag{3.253}$$

- Poisson Model: $P(\lambda)$, $g(\,\cdot\,)=\ln(\,\cdot\,)\Leftrightarrow g^{-1}(\,\cdot\,)=\exp(\,\cdot\,)$

$$\lambda_i=\mu_i=g^{-1}(\eta_i) \tag{3.254}$$

□ **Solution of Generalized Linear Model**

Using the distribution of Y_i dependent on $x_i'\beta$, we can use MLE maximizing to solve β. Algorithm for such maximizing task is called Iterative Re-weighted Least Squares, more specifically when using Newton-Raphson Method, this method is called Fisher's Scoring Method. Detail see section 5.4.3 ~ page 170.

Chapter. IV　多元统计分析部分

Instructor: Dong Li & Tianying Wang

Section 4.1　Multivariate Data

In this section, we consider a **Multivariate Statistic Model**. Sample comes from p dimension multivariate population $f(x_1, x_2, \ldots, x_p)$.

Notation : In this section, we still denote random variable in upper case and observed value in lower case, specially express random vector in bold font. **But** in this section we usually omit the vector symbol $\vec{\cdot}$. e.g. random vector with n **variable** is denoted as $\mathbf{X} = (X_{\cdot 1}, X_{\cdot 2}, \ldots, X_{\cdot p})$; sample of size n from the multivariate population is a $n \times p$ matrix $\{x_{ij}\}$, each sample item (a row in sample matrix) is denoted as x_i' or x_i^T.[1]

4.1.1　Matrix Representation

- Random Variable Representation
- Sample Representation
- Statistics Representation
- Sample Statistics Properties

□ **Random Variable Representation:**

- Random Vector: For a $p \times 1$ random vector $\vec{X} = (X_1, X_2, \ldots, X_p)^T$, denote (Marginal) expectation and variance, and covariance, correlation coefficient between X_i, X_j as follows:[2]

$$\mu_i = \mathbb{E}(X_i) \tag{4.1}$$

$$\sigma_{ii} = \sigma_i^2 = \mathbb{E}(X_i - \mu_i)^2 \tag{4.2}$$

$$\sigma_{ij} = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] \tag{4.3}$$

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{4.4}$$

[1] Here sample item (or sample case) $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]^T$ is a column vector.

[2] An intuition to avoid confusion of $\sigma_{\cdot\cdot}$: two subscripts means quadratic.

and we have covariance matrix (as defined in section 1.4.3 ~ page 26, equation 1.77 ~ page 27)

$$\Sigma = \mathbb{E}[(X-\mu)(X-\mu)^T] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1p} & \sigma_{p2} & \dots & \sigma_{pp} \end{bmatrix} \tag{4.5}$$

and Standard Deviation Matrix

$$V^{1/2} = diag\{\sqrt{\sigma_{ii}}\} \tag{4.6}$$

Based on $\vec{X} = (X_1, X_2, \dots, X_p)$, consider the linear combination:$Y = c'X = c_1X_1 + c_2X_2 + \dots c_pX_p$

$$\mathbb{E}(y) = c'\mu \qquad var(Y) = c'\Sigma c \tag{4.7}$$

and $Z_i = \sum_{j=1}^{p} c_{ij}X_j$ (i.e. $Z = CX$):

$$\mu_Z = \mathbb{E}(Z) = C\mu_X \qquad \Sigma_Z = C\Sigma_X C^T \tag{4.8}$$

and Correlation Matrix[3]

$$\varrho = \begin{bmatrix} \rho_{11} & \rho_{12} & \dots & \rho_{1p} \\ \rho_{21} & \rho_{22} & \dots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1p} & \rho_{p2} & \dots & \rho_{pp} \end{bmatrix} = V^{-1/2}\Sigma V^{-1/2} \tag{4.11}$$

- Random Matrix: Definition and basic properties of r.v. see section 1.3 ~ page 22. Now extend the definition to matrix $X = \{X_{ij}\}$.

$$X = \{X_{ij}\} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1n} & X_{n2} & \dots & X_{np} \end{bmatrix} \tag{4.12}$$

And we can further define $\mathbb{E}(X) = \{\mathbb{E}(X_{ij})\}$. For any const matrix A, B we have

$$\mathbb{E}(AXB) = A\mathbb{E}(X)B \tag{4.13}$$

Some more complex parameter can be expressed in language of tensors.

[3]Here the correlation matrix is the matrix of Pearson's Correlation Coefficients. Another frequently use correlation matrix called Cross Correlation Matrix is

$$\text{cross}(X, Y) = \mathbb{E}\left[X'Y\right] \tag{4.9}$$

and cross correlation matrix with $Y = X$:

$$\text{cross}(X, X) = \mathbb{E}\left[X'X\right] \tag{4.10}$$

□ **Sample Representation (for random vector):**

Sample of n items from population characterized by p variables

$$\begin{array}{c} \\ \text{item } 1 \\ \text{item } 2 \\ \vdots \\ \text{item } i \\ \vdots \\ \text{item } n \end{array} \begin{array}{c} \begin{array}{cccccc} \text{var } 1 & \text{var } 2 & \dots & \text{var } j & \dots & \text{var } p \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{ip} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nj} & \dots & x_{np} \end{pmatrix} \end{array} \tag{4.14}$$

Or represented in condense notation:

$$X = \{x_{ij}\} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} x_{\cdot 1} & x_{\cdot 2} & \dots & x_{\cdot p} \end{bmatrix} \tag{4.15}$$

□ **Statistics Representation**

- Unit 1 vector:

$$\mathbf{1}_k = \underbrace{(1, 1, \dots, 1)}_{k \text{ 1 in total}}^T \tag{4.16}$$

Unit 1 matrix: (Sometimes I also use notation $\mathcal{J}_n$)

$$\mathcal{I}_n = \{1\}_{n \times n} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}_{n \times n} \tag{4.17}$$

- Sample mean of the j^{th} variable:

$$\bar{x}_j = \frac{x_{1j} + x_{2j} + \ldots + x_{nj}}{n} = \frac{\mathbf{1}_n' x_{\cdot j}}{n}, \quad j = 1, 2, \dots, p \tag{4.18}$$

- Deviation of measurement of the j^{th} variable:

$$d_j = \begin{bmatrix} x_{1j} - \bar{x}_j \\ x_{2j} - \bar{x}_j \\ \vdots \\ x_{nj} - \bar{x}_j \end{bmatrix} = x_{\cdot j} - \bar{x}_j \mathbf{1}_n = (I - \frac{1}{n}\mathcal{I}_n) x_{\cdot j}, \quad j = 1, 2, \dots, p \tag{4.19}$$

- Covariance Matrix:

– Variance of $x_{\cdot j}$:

$$s_{jj} = s_j^2 = \frac{1}{n} d_j' d_j = \frac{1}{n} \sum_{k=1}^{n} (x_{kj} - \bar{x}_j)^2, \quad i = 1, 2, \dots p \tag{4.20}$$

$$= x_{\cdot j}' (I - \frac{1}{n} \mathcal{I}_n) x_{\cdot j}, \quad j = 1, 2, \dots, p \tag{4.21}$$

– Covariance between x_i and x_j:

$$s_{ij} = \frac{1}{n} d_i' d_j = \frac{1}{n} \sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j), \quad i, j = 1, 2, \dots p \tag{4.22}$$

$$= x_{\cdot i}' (I - \frac{1}{n} \mathcal{I}_n) x_{\cdot j}, \quad i, j = 1, 2, \dots, p \tag{4.23}$$

– Pearson's Correlation Coefficient between x_i and x_j:

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}}\sqrt{s_{jj}}} = \frac{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^{n} (x_{kj} - \bar{x}_j)^2}}, \quad i, j = 1, 2, \dots p \tag{4.24}$$

In condense notation, define Covariance Matrix from sample of size n:

$$S_n = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1p} & s_{p2} & \dots & s_{pp} \end{bmatrix} \tag{4.25}$$

and sample Correlation Coefficient Matrix:

$$R_n = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1p} \\ r_{21} & r_{22} & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1p} & r_{p2} & \dots & r_{pp} \end{bmatrix} \tag{4.26}$$

- Generalized sample variance: $|S_n| = \lambda_1 \lambda_2 \dots \lambda_p$, where λ_i are eigenvalues of S_n.
- 'Statistical Distance' between vectors: to measure the difference between two vectors $x = (x_1, x_2, \dots, x_p)$ and $y = (y_1, y_2, \dots, y_p)$.

  – Euclidean Distance:

$$d_E(x, y) = \sqrt{(x - y)^T (x - y)} \tag{4.27}$$

  – **Mahalanobis Distance**: Scale invariant distance, and include information about relativity position:

$$d_M(x, y) = \sqrt{(x - y)' S^{-1} (x - y)} \tag{4.28}$$

Remark: Mahalanobis distance is actually the normalized Euclidean distance in principal component space. So we can actually define the Mahalanobis distance for one sample case $\vec{x} = (x_1, x_2, \ldots, x_p)$ from distribution of $(\vec{\mu}, \Sigma)$

$$d_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})} \tag{4.29}$$

Note: the hyper-sruface $d_M(\vec{x}) =$ const forms a ellipsoid.

□ **Sample Statistics Properties**

Consider taking an n cases sample from r.v. population $\vec{X} = (X_1, X_2, \ldots, X_p)$, population mean μ and covariance matrix Σ. Basic statistics are sample mean and sample variance

$$\bar{X} = \frac{1}{n} X' \mathbf{1}_n, \qquad S_n = \frac{1}{n} \left(X - \frac{1}{n} \mathcal{I}_n X \right)' \left(X - \frac{1}{n} \mathcal{I}_n X \right) = \frac{1}{n} X' \left(I - \frac{1}{n} \mathcal{I}_n \right) X \tag{4.30}$$

Properties:

$$\mathbb{E}\left[\bar{X}\right] = \mu \qquad cov(\bar{X}) = \frac{1}{n}\Sigma \qquad \mathbb{E}\left[S_n\right] = \frac{n-1}{n}\Sigma \tag{4.31}$$

4.1.2 Review: Some Matrix Notation & Lemma

- Orthonormality: For square matrix P satisfies:

$$x_i^T x_j = \delta_{ij} \tag{4.32}$$

where x_i, x_j are columns of P.

- Eigenvalue and Eigenvector: For square matrix A, its eigenvalues λ_i and corresponding eigenvectors e_i satisfies:

$$Ae_i = \lambda_i e_i, \ \forall i = 1, 2, \ldots p \tag{4.33}$$

Denote $P = [e_1, e_2, \ldots, e_p]$, which is an orthonormal matrix. And denote $\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_p\}$.

$$A = \sum_{i=1}^{p} \lambda_i e_i e_i^T = P\Lambda P^T = P\Lambda P^{-1} \tag{4.34}$$

is called the Spectral Decomposition of A

- Squareroot matrix: Def. as

$$A^{1/2} = \sum_{i=1}^{p} \sqrt{\lambda_i} e_i e_i^T = P\Lambda^{1/2} P^T \tag{4.35}$$

Properties:

  - $A^{1/2} A^{1/2} = A$;
  - $A^{-1/2} = (A^{1/2})^{-1} = PL^{-1/2}P^T$;
  - $tr(A) = \sum_{i=1}^{n} \lambda_n$;
  - $|A| = \prod_{i=1}^{n} \lambda_n$.

- (Symmetric) Positive Definite Matrix: Say A a Positive Definite Matrix if

$$x^T Ax > 0, \forall x \in \mathbb{R}^p \tag{4.36}$$

where $x^T Ax$ is called a Quadric Form.

Properties:

  - Use the Spectral Decomposition of A, we can write the Quadric Form as

$$x^T Ax = x^T P \Lambda P^T x = y^T \Lambda y = \sum_{i=1}^{p} \lambda_i y_i^2 = \sum_{i=1}^{p} (\sqrt{\lambda_i} y_i)^2 \tag{4.37}$$

  - Eigenvalues $\lambda_i > 0, \forall i = 1, 2, \ldots, p$
  - A can be written as product of symmetric matrix: $A = Q^T Q$ (Q is symmetric);

Positive Semi-definite matrix is one with $\lambda_i \geq 0$

- Trace of Matrix: For $p \times p$ square matrix A

$$tr(A) = \sum_{i=1}^{p} a_{ii} \tag{4.38}$$

Properties:

  - $tr(AB) = tr(BA)$;
  - $x'Ax = tr(x'Ax) = tr(Axx')$
  - $tr(A) = \sum_i \lambda_i$

- Matrix Partition: partition square matrix $\underset{p \times p}{A}$ as

$$A = \begin{bmatrix} \underset{q_1 \times q_1}{A_{11}} & \underset{q_1 \times q_2}{A_{12}} \\ \underset{q_2 \times q_1}{A_{21}} & \underset{q_2 \times q_2}{A_{22}} \end{bmatrix} \tag{4.39}$$

where $p = q_1 + q_2$

Property:

$$|A| = |A_{22}||A_{11} - A_{12}A_{22}^{-1}A_{21}| = |A_{11}||A_{22} - A_{21}A_{11}^{-1}A_{12}| \tag{4.40}$$

- Matrix Differentiation

Calculus Notations: Take derivative of $y = (y_1, y_2, \ldots, y_q)^T$ over $x = (x_1, x_2, \ldots, x_p)^T$; or similarly of matrix A over scalar, etc.

We use 'Denominator-layout', which means the result follows the shape of denominator:

$$\frac{\partial y}{\partial x} := \frac{\partial y^T}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_q}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_p} & \frac{\partial y_2}{\partial x_p} & \cdots & \frac{\partial y_q}{\partial x_p} \end{bmatrix} \Leftrightarrow \left(\frac{\partial y}{\partial x}\right)_{ij} = \frac{\partial y_j}{\partial x_i} \tag{4.41}$$

Properties (under denominator-layout):[4]

- $\frac{\partial}{\partial x} Ax = A^T$;
- $\frac{\partial}{\partial x} x^T A = A$;
- $\frac{\partial}{\partial x} x^T x = 2x$;
- $\frac{\partial}{\partial x} x^T Ax = Ax + A^T x$;
- $\frac{\partial}{\partial x} \log(x^T Ax) = \frac{2Ax}{x^T Ax}$;
- $\frac{\partial |A|}{\partial A} = |A| A^{-1}$;
- $\frac{\partial tr(AB)}{\partial A} = B^T$;
- $\frac{\partial tr(A^{-1}B)}{\partial A} = -A^{-1}B^T A^{-1}$

- Kronecker Product: For matrix $\underset{m\times n}{A} = \{a_{ij}\}$, $\underset{p\times q}{B} = \{b_{ij}\}$. Their Kronecker product

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \tag{4.46}$$

[4] More matrix diffrenciation equation see book [9] P49. Or can be easily derivated using Einstein sumation notation.

An example:

$$\frac{\partial |A|}{\partial A} = \frac{\partial}{\partial A_{ij}} \mathrm{Ex}_{i_1,i_2,\dots,i_n} A_{1i_i} A_{2i_2} \dots A_{ni_n} \tag{4.42}$$

$$= \sum_{k=1}^{n} \mathrm{Ex}_{i_1,\dots,(\wedge i_k),\dots,i_n} \delta_{ki}\delta_{i_k j} A_{1i_1} \dots (\wedge A_{ki_k}) \dots A_{ni_n} \times (-1)^{(n-k)+(n-i_k)} \tag{4.43}$$

$$= (-1)^{i+j} \mathrm{Ex}_{i_1,\dots,(\wedge j),\dots,i_n} A_{1i_1} \dots (\wedge A_{ij}) \dots A_{ni_n} \tag{4.44}$$

$$= |A| A^{-1} \tag{4.45}$$

- Norm:
  - Vector Norm: for vector $x, y \in \mathbb{C}^m$, norm $\|\cdot\|$ is a function $\mathbb{C}^m \to \mathbb{R}$, with:

$$\text{Semi-definiteness: } \|x\| \geq 0, = \text{ for } x = 0 \tag{4.47}$$

$$\text{Absolute homogeneity: } \|kx\| = |k|\|x\|, \ k \in \mathbb{C} \tag{4.48}$$

$$\text{Triangle inequality: } \|x\| + \|y\| \geq \|x + y\| \tag{4.49}$$

    the ℓ_p-norm of x is

$$\|x\|_p \equiv \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{4.50}$$

    Useful norm:
    - ℓ_0-norm: # of none-0 elements in x;[5]
    - ℓ_1-norm: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$;
    - ℓ_2-norm/Euclidean norm: $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$;
    - ℓ_∞-norm: $\max |x_i|$.
  - Matrix Norm: for matrix $A, B \in \mathbb{C}^{m \times n}$, norm $\|\cdot\|$ is a function $\mathbb{C}^{m \times n} \to \mathbb{R}$, with:

$$\text{Semi-definiteness: } \|A\| \geq 0, = \text{ for } x = 0 \tag{4.51}$$

$$\text{Absolute homogeneity: } \|kA\| = |k|\|A\|, \ k \in \mathbb{C} \tag{4.52}$$

$$\text{Triangle inequality: } \|A\| + \|B\| \geq \|A + B\| \tag{4.53}$$

    further for $m = n$, i.e. $A, B \in \mathbb{C}^{m \times m}$, usually append

$$\text{Sub-multiplicative: } \|A\|\|B\| \geq \|AB\| \tag{4.54}$$

$$\text{Hermite: } \|A\| = \|A^*\| \tag{4.55}$$

    Matrix norm induced by vector norm:

$$\|A\| = \max \frac{\|Ax\|}{\|x\|} \tag{4.56}$$

    e.g. ℓ_p induced matrix norm:
    - ℓ_1-norm: $\|A\|_1 = \max\limits_{1 \leq j \leq n} \sum\limits_{i=1}^{m} |A_{ij}|$
    - ℓ_2-norm/Euclidean norm: $\|A\|_2 = \sigma_{\max}(A)$;
    - ℓ_∞-norm: $\|A\|_\infty = \max\limits_{1 \leq i \leq m} \sum\limits_{j=1}^{n} |A_{ij}|$.

    Non-induced matrix norm, e.g.
    - Frobenius norm: $\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2 \right)^{1/2} = \sqrt{tr(A^*A)}$
    - Weighted Frobenius norm: $\|A\|_W = \|W^{-1/2} A W^{-1/2}\|_F$(or some textbooks uses $\|W^{1/2} A W^{1/2}\|_F$)

[5] Note: actually triangle inequality is not satisfied for $\|\cdot\|_0$

* Max norm: $\|A\|_{\max} = \max_{i,j} |A_{ij}|$

- Sherman-Morrison Formula:

$$\left(A + u^T v\right)^{-1} = A^{-1} - \frac{A^{-1}u^T v A^{-1}}{1 + v^T A^{-1} u} \tag{4.57}$$

Or in matrix form:

$$(A + B)^{-1} = A^{-1} - \frac{A^{-1}BA^{-1}}{1 + tr(A^{-1}B)}, \quad \text{rank}(B) = 1 \tag{4.58}$$

Application instances see https://v1ncent19.github.io//texts/MahalanobisAndLeverage/ and https://v1ncent19.github.io//texts/DeletedResidual/.

- Woodbury Matrix Identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

4.1.3 Useful Inequalities

- Cauchy-Schwartz Inequality:

Let b, d any $p \times 1$ vectors.

$$(b'd)^2 \leq (b'b)(d'd) \tag{4.59}$$

- Extended Cauchy-Schwartz Inequality:

Let B be a positive definite matrix.

$$(b'd)^2 \leq (b'Bb)(d'B^{-1}d) \tag{4.60}$$

- Maximazation Lemma:

d be a given vector, for any non-zero vector x,

$$\frac{(x'd)^2}{x'Bx} \leq d'B^{-1}d \tag{4.61}$$

Take Maximum when $x = cB^{-1}d$.

Section 4.2 Statistical Inference to Multivariate Population

Statistics model: a n cases sample $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$, where each $\mathbf{X}_i$ i.i.d. from a multivariate population (usually consider a multi-normal). i.e.

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1p} \\ X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1n} & X_{n2} & \ldots & X_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1' \\ \mathbf{X}_2' \\ \vdots \\ \mathbf{X}_n' \end{bmatrix} \tag{4.62}$$

4.2.1 Multivariate Normal Distribution

Univariate Noraml Distribution: $N(\mu, \sigma^2)$

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{4.63}$$

Multivariate Normal Distribution: $X \sim N_p(\vec{\mu}, \Sigma)$[6]

$$f_{\mathbf{X}}(\vec{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{(\vec{x}-\vec{\mu})'\Sigma^{-1}(\vec{x}-\vec{\mu})}{2}\right) \tag{4.64}$$

Note: Here in the exp, the $(\vec{x}-\vec{\mu})'\Sigma^{-1}(\vec{x}-\vec{\mu})$ is the Mahalanobis Distance d_M defined in equation 4.29 ~ page 118

Remark: A n-dimension multivariate normal has $\frac{p(p+1)}{2}$ free parameters. Thus for a very high dimension, contains too many free parameters to be determined!

Properties: Consider $X \sim N_p(\mu, \Sigma)$

- Linear Transform:
  - For a $p \times 1$ vector a:

$$X \sim N_p(\mu, \Sigma) \Leftrightarrow a'X \sim N(a'\mu, a'\Sigma a),\ \forall a \in \mathbb{R}^p \tag{4.65}$$

    (Proof: use characteristic function.)

  - For a $q \times p$ const matrix A:

$$AX + a \sim N_q(A\mu + a, A\Sigma A') \tag{4.66}$$

  - For a $p \times p$ square matrix A:

$$\mathbb{E}(X'AX) = \mu'A\mu + tr(A\Sigma) \tag{4.67}$$

- Conditional Distribution: Take partition of $\underset{p\times 1}{X} \sim N(\underset{p\times 1}{\mu}, \underset{p\times p}{\Sigma})$ into $\underset{q_1\times 1}{X_1}$ and $\underset{q_2\times 1}{X_2}$, where $q_1+q_2=p$. Write in matrix form:

$$\underset{p\times 1}{X} = \begin{bmatrix} \underset{q_1\times 1}{X_1} \\ \underset{q_2\times 2}{X_2} \end{bmatrix} \qquad \underset{p\times 1}{\mu} = \begin{bmatrix} \underset{q_1\times 1}{\mu_1} \\ \underset{q_2\times 2}{\mu_2} \end{bmatrix} \qquad \underset{p\times p}{\Sigma} = \begin{bmatrix} \underset{q_1\times q_1}{\Sigma_{11}} & \underset{q_1\times q_2}{\Sigma_{12}} \\ \underset{q_2\times q_1}{\Sigma_{21}} & \underset{q_2\times q_2}{\Sigma_{22}} \end{bmatrix} \tag{4.68}$$

  i.e.

$$\underset{p\times 1}{X} = \begin{bmatrix} \underset{q_1\times 1}{X_1} \\ \underset{q_2\times 2}{X_2} \end{bmatrix} \sim N_{q_1+q_2}\left(\begin{bmatrix} \underset{q_1\times 1}{\mu_1} \\ \underset{q_2\times 2}{\mu_2} \end{bmatrix}, \begin{bmatrix} \underset{q_1\times q_1}{\Sigma_{11}} & \underset{q_1\times q_2}{\Sigma_{12}} \\ \underset{q_2\times q_1}{\Sigma_{21}} & \underset{q_2\times q_2}{\Sigma_{22}} \end{bmatrix} \right) \tag{4.69}$$

  Independence: $X_1 \parallel X_2 \Leftrightarrow \Sigma_{21} = \Sigma_{12}^T = 0$

[6] Detailed derivation see section 1.8 ~ page 32

And the conditional dictribution $X_1|X_2 = x_2$ is given by [7]

$$X_1|_{X_2=x_2} \sim N_p(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2),\ \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \tag{4.71}$$

- Multivariate Normal & χ^2

  Let $X \sim N_p(\mu, \Sigma)$, then

$$(X-\mu)^T\Sigma^{-1}(X-\mu) \sim \chi_p^2 \tag{4.72}$$

4.2.2 MLE of Multivariate Normal

Under the notation in equation 4.62 ~ page 122, i.e. each sample case $\mathbf{X}_i$ i.i.d. $\sim N_p(\mu, \Sigma)$, we can get the joint PDF of $\mathbf{X}$:

$$f_{\mathbf{X_1},\dots,\mathbf{X_n};\mu,\Sigma}(x_1,\dots,x_n) = \frac{1}{(2\pi)^{np/2}|\Sigma|^{n/2}} \exp\left(-\sum_{i=1}^{n}\frac{(x_i-\mu)'\Sigma^{-1}(x_i-\mu)}{2}\right) \tag{4.73}$$

and at the same time get likelihood function[8]:

$$L(\mu,\Sigma;x_1,\dots,x_n) = \frac{1}{(2\pi)^{np/2}|\Sigma|^{n/2}} \exp\left[-\frac{1}{2}tr\left(\Sigma^{-1}\left(\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})' + n(\bar{x}-\mu)(\bar{x}-\mu)'\right)\right)\right] \tag{4.75}$$

And we can get the MLE of μ and Σ as follows[9]:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}x_i = \bar{x} \tag{4.76}$$

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})' = \frac{n-1}{n}S \tag{4.77}$$

> Δ **Note:** In this section, S is used to denote $\hat{\Sigma}$, which is different from that in section 2.1.1 ~ page 38 (S^2 for $\hat{\Sigma}$)

And we can furthur construct MLE of function of μ, Σ (use invariance property of MLE), for example

$$\widehat{|\Sigma|} = |\hat{\Sigma}| \tag{4.78}$$

Note: $(\hat{\mu}, \hat{\Sigma})$ is sufficient statistic of multi-normal population.

[7] In equation 4.66 ~ page 123, take

$$\underset{p\times p}{A} = \begin{bmatrix} \underset{q\times q}{I} & -\underset{q\times(p-q)}{\Sigma_{12}\Sigma_{22}^{-1}} \\ \underset{(p-q)\times q}{0} & \underset{(p-q)\times(p-q)}{I} \end{bmatrix} \tag{4.70}$$

[8] Here we need to use the property of trace

$$x'Ax = tr(x'Ax) = tr(Ax'x) \tag{4.74}$$

[9] Detailed proof see '*Applied Multivariate Statistical Analysis*' P130

4.2.3 Sampling distribution of $\bar{X}$ and S

$\hat{\mu} = \bar{X}$ and $\hat{\Sigma} = \dfrac{n-1}{n}S$ are statistics, with sampling distribution.

□ **Sampling distribution of $\bar{X}$**

Similar to monovariate case:

$$\bar{X} \sim N_p(\mu, \frac{1}{n}\Sigma) \tag{4.79}$$

□ **Sampling distribution of S^2**

- Monovariate case: Consider $(X_1, X_2, \ldots, X_n)$ i.i.d. $\sim N(\mu, \sigma^2)$

  Then

  $$\frac{(n-1)S}{\sigma^2} \sim \chi^2_{n-1} \tag{4.80}$$

- Multivariate case: Consider $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ i.i.d. $\sim N_p(\mu, \Sigma)$

  Then

  $$(n-1)S \sim W_p(n-1, \Sigma) \tag{4.81}$$

  Where $W_p(n-1, \Sigma)$ is Wishart Distribution, details as follows:

  For r.v. $Z_1, Z_2, \ldots, Z_m$ i.i.d. $\sim N_p(0, \Sigma)$, def p dimensional **Wishart Distribution** with dof m as $W_p(m, \Sigma)$.[10]

  $$W_p = \sum_{i=1}^{n} Z_i Z_i' \tag{4.82}$$

  PDF of $W_p(m, \Sigma)$:

  $$f_W(w; p, m, \Sigma) = \frac{|w|^{\frac{m-p-1}{2}} \exp\left(-\frac{1}{2}tr(\Sigma^{-1}w)\right)}{2^{\frac{mp}{2}}|\Sigma|^{-1/2}\pi^{\frac{p(p-1)}{4}}\prod_{i=1}^{p}\Gamma(\frac{m-i+1}{2})} \tag{4.83}$$

  C.F.

  $$\phi(T) = |I_p - 2i\Sigma T|^{-\frac{m}{2}} \tag{4.84}$$

  Properties:

  – For independent $A_1 \sim W_p(m_1, \Sigma)$ and $A_2 \sim W_p(m_2, \Sigma)$, then

  $$A_1 + A_2 \sim W_p(m_1 + m_2, \Sigma) \tag{4.85}$$

  – For $A \sim W_p(m, \Sigma)$, then

  $$CAC' \sim W_p(m, C\Sigma C') \tag{4.86}$$

[10] $W_p(m, \Sigma)$ is a distribution defined on $p \times p$ matrix space.

– Wishart distribution is the matrix generization of χ_n^2. When $p=1, \Sigma=\sigma^2=1$, $W_p(m,\Sigma)$ naturally reduce to χ_m^2.

$$\chi_n^2 = W_1(n,1) \tag{4.87}$$

▷ R. **Code**

Distribution functions are in package `MCMCpack`, or use `rWishart()` function.

□ **Large sample $\bar{X}$ and S**

- $\sqrt{n}(\bar{X}-\mu) \xrightarrow{\text{d}} N_p(0,\Sigma)$;
- $n(\bar{X}-\mu)'S^{-1}(\bar{X}-\mu) \xrightarrow{\text{d}} \chi_p^2$

4.2.4 Hypothesis Testing for Normal Population

- **One-Population Hypothesis Testing**:

  Conduct hypothesis testing to μ:

$$H_0: \mu=\mu_0 \longleftrightarrow H_1: \mu \neq \mu_0 \tag{4.88}$$

□ **Hotelling's T^2 test**

– One-Dimensional case: t-test

$$T = \frac{\sqrt{n}(\bar{X}-\mu_0)}{S} \sim t_{n-1} \tag{4.89}$$

i.e.

$$T^2 = [\sqrt{n}(\bar{X}-\mu_0)]S^{-1}[\sqrt{n}(\bar{X}-\mu_0)] \sim t_{n-1}^2 = F_{1,n-1} \tag{4.90}$$

– Multi-Dimensional case: Hotelling's T^2

$$T^2 = [\sqrt{n}(\bar{X}-\mu_0)']S^{-1}[\sqrt{n}(\bar{X}-\mu_0)] \sim \frac{p}{n-p}(n-1)F_{p,n-p} \tag{4.91}$$

And we can get the distribution of **Hotelling's T^2**:

$$\frac{n-p}{p}\frac{T^2}{n-1} \sim F_{p,n-p} \tag{4.92}$$

Rejection Rule:

$$T^2 > \frac{p(n-1)}{n-p}F_{p,n-p,\alpha} \tag{4.93}$$

Property:

Invariant for X transform: For $Y=CX+d$, then

$$T_Y^2 = n(\bar{X}-\mu_0)'S^{-1}(\bar{X}-\mu_0) = T_X^2 \tag{4.94}$$

□ **LRT of $\hat{\mu}$**

Monovariate case see section 2.4.3 ~ page 60.

LRT uses the statistic:

$$\Lambda = \frac{\max_{H_0} L(\mu_0, \Sigma)}{\max_{H_0 \cup H_1} L(\mu, \Sigma)} = (1 + \frac{T^2}{n-1})^{-n/2} \tag{4.95}$$

where $T^2 = n(\bar{x} - \mu_0)' S^{-1} (\bar{x} - \mu_0)$

- **Two-Population Hypothesis Testing**:

Conduct hypothesis testing to $\delta = \mu_1 - \mu_2$:

$$H_0 : \delta = \delta_0 \longleftrightarrow H_1 : \delta \neq \delta_0 \tag{4.96}$$

Notation: The two sample of size n_1, n_2, each denoted as

$$X_{1,ij} \qquad X_{2,ij} \tag{4.97}$$

with mean μ_1, μ_2 and covariance matrix Σ_1, Σ_2

– Paired Samples: $n_1 = n_2$

For two paires samples $\{X_{1,ij}\}$, $X_{2,ij}$, take subtraction as

$$D_{ij} = X_{1,ij} - X_{2,ij} \tag{4.98}$$

denote $\bar{D} = \frac{1}{n} \sum_{j=1}^{n} D_j$, $S_D^2 = \frac{1}{n-1} \sum_{j=1}^{n} (D_j - \bar{D})'(D_j - \bar{D})$

and conduct test to

$$H_0 : \bar{D} = \delta_0 \longleftrightarrow H_1 : \bar{D} \neq \delta_0 \tag{4.99}$$

And the folloeing steps are as in One-population testing, test

$$T^2 = n(\bar{D} - \delta)'(S_D^2)^{-1}(\bar{D} - \delta) \sim \frac{(n-1)p}{n-p} F_{p,n-p} \tag{4.100}$$

– Under Equal Unknown Variance: $\Sigma_1 = \Sigma_2$

$$\bar{X}_1 = \frac{1}{n_1} \sum_{j=1}^{n_1} X_{1,j} \qquad \bar{X}_2 = \frac{1}{n_2} \sum_{j=1}^{n_2} X_{1,j} \tag{4.101}$$

$$S_1 = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1} (X_{1,j} - \bar{X}_1)(X_{1,j} - \bar{X}_1)' \quad S_2 = \frac{1}{n_2 - 1} \sum_{j=1}^{n_2} (X_{2,j} - \bar{X}_2)(X_{2,j} - \bar{X}_2)' \tag{4.102}$$

And denote pooled variance

$$S_{\text{pooled}} = \frac{1}{n_1 + n_2 - 2} ((n_1 - 1)S_1 + (n_2 - 1)S_2) \sim \frac{W_p(n_1 + n_2 - 2, \Sigma)}{n_1 + n_2 - 2} \tag{4.103}$$

Under H_0, we have

$$T^2 = \frac{1}{\frac{1}{n_1} + \frac{1}{n_2}} (\bar{X}_1 - \bar{X}_2 - \delta_0)' S_{\text{pooled}}^{-1} (\bar{X}_1 - \bar{X}_2 - \delta_0) \sim \frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1} \tag{4.104}$$

4.2.5 Confidence Region

Estimate the confidence region for μ of $X \sim N_p(\mu, \Sigma)$, Monovariate case see section 2.3.3 ~ page 54

- Confidence Region:

  Also use Hotelling's T^2

  $$\frac{n-p}{p}\frac{T^2}{n-1} \sim F_{p,n-p} \tag{4.105}$$

  And take $100(1-\alpha)\%$ confidence region of μ as

  $$R(x) = \{x | T(x)^2 \leq c^2\} \qquad c^2 = \frac{p}{n-p}(n-1)F_{p,n-p,\frac{\alpha}{2}} \tag{4.106}$$

  The shape of $R(x)$ is an ellipsoid.

- Individual Converage Interval

  Use the decomposition of S as a positive definite matrix $S = A^T A$, where A is some $p \times p$ matrix, then

  $$T^2 = [\sqrt{n}(\bar{X} - \mu_0)']S^{-1}[\sqrt{n}(\bar{X} - \mu_0)] = [A^{-1\prime}\sqrt{n}(\bar{X} - \mu_0)]'[A^{-1\prime}\sqrt{n}(\bar{X} - \mu_0)] \tag{4.107}$$

  Thus denote $Z = A^{-1\prime}(X - \mu_0) \sim N_p(0, A^{-1\prime}\Sigma A^{-1})$, the T^2 estimator of Z would be

  $$T_Z^2 = [\sqrt{n}\bar{Z}]'S_Z^{-1}[\sqrt{n}\bar{Z}] = n\bar{Z}'\bar{Z} = \frac{1}{n}\sum_{i=1}^{n}\bar{Z}_i^2 \sim F_{p,n-p} \tag{4.108}$$

  As a simplified case, we can take the **Individual Converage Interval** of Z_i, which is

  $$\frac{\sqrt{n}Z_i}{s_{Z_i}} \sim t_{n-1} \tag{4.109}$$

  And we can take the Confidence Region[11] as

  $$R(z) = \bigotimes_{i=1}^{n}(\bar{Z}_i \pm s_{Z_i}t_{n-1,\frac{\beta}{2}}) \tag{4.110}$$

  where β taken with Bonferroni correction

  $$1 - p\beta = 1 - \alpha \tag{4.111}$$

  Note: Consider that

  $$P(\text{all } Z_i \text{ in CI}_i) \geq 1 - m\beta = 1 - \alpha \tag{4.112}$$

  So the real CR for μ should be larger.

  The shape of $R(x)$ is an oblique cubold.

[11] The confidence region of Z can be transformed to that of X using $\hat{Z} = A^{-1\prime}(\hat{X} - \bar{X})$.

4.2.6 Large Sample Multivariate Inference

Basic point:

$$\bar{X} \xrightarrow{\text{d}} \mu \qquad S \xrightarrow{\text{d}} \Sigma \tag{4.113}$$

- One-sample Mean:

$$n(\bar{X}-\mu)S^{-1}(\bar{X}-\mu) \xrightarrow{\text{d}} \chi^2_p \tag{4.114}$$

- Unequal Variance Two-sample Mean:

$$\bar{X}_1-\bar{X}_2 \xrightarrow{\text{d}} N\left(\mu_1-\mu_2, \frac{1}{n_1}\Sigma_1+\frac{1}{n_2}\Sigma_2\right) \qquad \frac{1}{n_1}S_1+\frac{1}{n_2}S_2 \xrightarrow{\text{d}} \frac{1}{n_1}\Sigma_1+\frac{1}{n_2}\Sigma_2 \tag{4.115}$$

Test:

$$T^2=\left[(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)\right]'\left(\frac{1}{n_1}S_1+\frac{1}{n_2}S_2\right)^{-1}\left[(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)\right] \xrightarrow{\text{d}} \chi^2_p \tag{4.116}$$

Section 4.3 Principal Component Analysis

PCA and next subsection FA focus on data dimension reduction. Why?

☐ **'Curse of Dimensionality'**

- Difficulty in computation complexity: Many algorithms has complexity $O(n^2)$ or more, high dimension n cause high complexity.
- Hughes Phenomenon: As the number of feature dimension increases, the classifier's performance increases as well until an optimal dimension. Adding more features based on the same size as the training set will then degrade the classifier's performance. [a]

[a] Example: Volumn of unit sphere in n-dim space

$$V_n=\pi^{n/2}\frac{1}{\Gamma(1+n/2)} \to \left(\frac{2\pi e}{n}\right)^{n/2} \to 0 \tag{4.117}$$

i.e. data will naturally become 'sparse' in high dimension data → difficult to extract information.

Key Idea of PCA: Find the components most powerful in explaining variance. (Similar to the idea of ANOVA)

4.3.1 Population Principal Component

For population $\vec{X}=(X_1,X_2,\ldots,X_p)\sim(\mu,\Sigma)_p$, conduct spectrum decomposition to Σ such that

$$\Sigma P=P\Lambda \qquad P=\Big[e_1,e_2,\ldots,e_p\Big] \quad \Lambda=\text{diag}\{\lambda_1,\lambda_2,\ldots,\lambda_p\},\ \lambda_1\geq\lambda_2\geq\ldots\geq\lambda_p \tag{4.118}$$

where (λ_i,e_i) is the i^{th} eigenvalue-eigenvector pair of Σ, large λ_i suggests X is more 'extended' in e_i direction(large variance).

Then the **Principal Components** $Y = \{Y_i\}$

$$Y = P'X \sim (P'\mu, P'\Sigma P)_p = (P'\mu, \Lambda) \tag{4.119}$$

$$\begin{cases} Y_1 = e_1'X \sim (e_1'\mu, \lambda_1) \\ \vdots \\ Y_p = e_p'X \sim (e_p'\mu, \lambda_p) \end{cases} \tag{4.120}$$

Properties & Definitions:

- Trace of cov. matrix:

$$\sum_{i=1}^{p} \sigma_{ii} = \sum_{i=1}^{p} var(X_i) = \sum_{i=1}^{p} var(Y_i) = \sum_{i=1}^{p} \lambda_i \tag{4.121}$$

- $corr$ between Y_i, X_j:

$$\rho_{Y_i,X_j} = \frac{cov(Y_i, X_j)}{\sqrt{\lambda_i}\sqrt{\sigma_{jj}}} = \frac{(e_i)_j\sqrt{\lambda_i}}{\sqrt{\sigma_{jj}}} \tag{4.122}$$

- Factor Loading:

$$\text{FL}_{ij} = (e_i)_j\sqrt{\lambda_i} \tag{4.123}$$

- PC Score:

$$\text{PC Score}_i = Y_i = e_i'X \text{ or } Y_i = e_i'(X - \mu) \tag{4.124}$$

In practice, we pick the first several m PC such that

$$\sum_{i=1}^{m} \frac{\lambda_i}{\sum_{k=1}^{p} \lambda_k} \text{ large enough} \tag{4.125}$$

Note: Another important point for PCA is the **interpretability** of principal components.

A coutinuous version of PCA in stochastic process is Karhunen-Loève Expansion in **??** ~ page **??**.

□ Standardized Principal Component

To cancel out the influence due to scale, we can also obtain standardized PC from $Z = (V)^{-1/2}(X_\mu)$, where V is standard deviation matrix as def. in equation 4.6 ~ page 115.

And we have $\vec{Z} = (Z_1, Z_2, \ldots, Z_p) \sim N_p(0, V^{-1/2}\Sigma V^{-1/2\prime}) = N_p(0, \rho)$. Then obtain (λ_i, e_i) pairs[12] from ρ to form PC.

$$\rho P = P\Lambda \qquad P = \Big[e_1, e_2, \ldots, e_p\Big] \quad \Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_p\},\ \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \tag{4.126}$$

Then the Principal Components $W = \{W_i\}$

$$W = P'Z \sim (0, P'\rho P)_p = (0, \Lambda) \tag{4.127}$$

$$\begin{cases} W_1 = e_1'Z \sim (0, \lambda_1) \\ \vdots \\ W_p = e_p'Z \sim (0, \lambda_p) \end{cases} \tag{4.128}$$

Properties:

[12] The eigenvalue-eigenvector pairs obtained from ρ is generally **different** from Σ.

- Trace of cov. matrix:

$$\sum_{i=1}^{p} var(Z_i) = \sum_{i=1}^{p} var(W_i) = \sum_{i=1}^{p} \lambda_i = p \tag{4.129}$$

- $corr$ between Y_i, X_j:

$$\rho_{W_i,Z_j} = (e_i)_j \sqrt{\lambda_i} \tag{4.130}$$

4.3.2 Sample Principal Component

For sample matrix X denoted in equation 4.62 ~ page 122, with cov. matrix S in equation 4.25 ~ page 117. Then conduct the above spectrum decomposition to S to get sample PCs.

$$\hat{Y} = \hat{P}\hat{\Lambda}\hat{P}' \qquad \hat{P} = \left[\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_p\right] \quad \hat{\Lambda} = diag\{\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_p\},\ \hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_p \tag{4.131}$$

Properties and Definitions

- Trace of cov. matrix:

$$\sum_{i=1}^{p} s_{ii} = \sum_{i=1}^{p} \hat{\lambda}_i \tag{4.132}$$

- Sample corr & factor load:

$$\rho(\hat{y}_i, x_j) = \frac{(\hat{e}_i)_j \sqrt{\hat{\lambda}_j}}{\sqrt{s_{jj}}} \tag{4.133}$$

□ **Large Sample & Normal PCA**

Under normal assumption or large sample case, i.e.

$$X \sim N_p(\mu, \Sigma) \text{ or } X \xrightarrow{\text{d}} N_p(\mu, \Sigma) \tag{4.134}$$

We can examine the (asymptotic) distribution of $(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_p)$ and $(\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_p)$:

- $\hat{\lambda}$ distribution:

$$\sqrt{n}(\hat{\lambda} - \lambda) \sim N_p(0, 2\Lambda^2) \tag{4.135}$$

- $\hat{e}_i$ distribution:

$$\sqrt{n}(\hat{e}_i - e_i) \sim N_p(0, E_i), \quad E_i = \lambda_i \sum_{k \neq i} \frac{\lambda_k}{(\lambda_k - \lambda_i)^2} e_k e_k' \tag{4.136}$$

- Independence:

$$\hat{\lambda}_i \perp\!\!\!\perp \hat{e}_i \tag{4.137}$$

Section 4.4 Factor Analysis

Key idea of FA: For a model with p variable $X = (X_1, X_2, \ldots, X_p) \sim (\mu, \Sigma)_p$ (especially when p large and X_i interrelated), there would be some internal, latent **factors** F behind X determining the model structure.[13]

[13] As the most simplified case, here only consider X linear dependent on F.

4.4.1 Orthogonal Factor Model

$$\underset{p\times 1}{X-\mu}=\underset{p\times m}{L}\underset{m\times 1}{F}+\underset{p\times 1}{\varepsilon},\ m<p \tag{4.138}$$

where L is the const **loading matrix** ; F is r.v. **factor**; and ε is r.v. error.

$$X=\begin{bmatrix}X_1\\X_2\\\vdots\\X_p\end{bmatrix}\quad L=\begin{bmatrix}\ell_{11}&\ell_{12}&\dots&\ell_{1m}\\\ell_{21}&\ell_{22}&\dots&\ell_{2m}\\\vdots&\vdots&\ddots&\vdots\\\ell_{p1}&\ell_{p2}&\dots&\ell_{pm}\end{bmatrix}\quad F=\begin{bmatrix}F_1\\F_2\\\vdots\\F_m\end{bmatrix}\quad \varepsilon=\begin{bmatrix}\varepsilon_1\\\varepsilon_2\\\vdots\\\varepsilon_p\end{bmatrix} \tag{4.139}$$

Note: Intuitively, we cannot estimate $(m+p)$ (unobserable) r.v. from p r.v., so we need the following assumptions on F and ε

$$\begin{aligned}&\mathbb{E}(F)=0 && cov(F)=I_n\\&\mathbb{E}(\varepsilon)=0 && cov(\varepsilon)=\Psi=diag\{\psi_1,\psi_2,\dots,\psi_p\}\\&\varepsilon\perp\!\!\!\perp F\Leftrightarrow cov(F,\varepsilon)=0\end{aligned} \tag{4.140}$$

Derived Conclusions:

- Representation of Σ:

$$cov(X)=\Sigma=LL'+\Psi \tag{4.141}$$

  - Diagonal Elements:

$$var(X_i)=\sum_{k=1}^{m}\ell_{ik}^2+\psi_i=h_i^2+\psi_i \tag{4.142}$$

    where h_i^2 is Communality, ψ_i is Specific variance.

  - NonDiagonal Elements:

$$cov(X_i,X_j)=\sum_{k=1}^{m}\ell_{im}\ell_{jm} \tag{4.143}$$

- relation bet. X and F:

$$cov(X,F)=L \tag{4.144}$$

□ **Factor Rotation**

For any orthonormal rotation/reflection matrix $\underset{m\times m}{T}$, $\tilde{L}=LT$ satisfies the same factor model (with a different $\tilde{F}$):

$$\begin{aligned}X=&LF+\varepsilon=LTT'F+\varepsilon=\tilde{L}\tilde{F}+\varepsilon \qquad \tilde{L}=LT,\ \tilde{F}=T'F\\\Sigma=&LL'+\Psi=\tilde{L}\tilde{L}'+\Psi\end{aligned}$$

Comment: Factor rotation reflects the arbitrariness of selection of L, allowing us to choose an **interpretable** L for FA model.

4.4.2 Principal Component Approach

Origin: when $m = p$, factor decomposition reduces to spectrum (PC) decomposition.(At the same time Ψ can be taken 0.)

$$\begin{aligned} X =& LF + \varepsilon = PY \qquad \Rightarrow \Psi = 0 \\ \Sigma =& LL' + \Psi = P\Lambda P' \quad \Rightarrow L = P\Lambda^{1/2} \end{aligned} \tag{4.145}$$

Then take the first m eigenvectors to form L, and use $\psi_i = \sigma_{ii} - \sum_{k=1}^{m} \ell_{ik}^2$ as an approximation.

$$\Sigma = LL' + \Psi \qquad L = \left[\sqrt{\lambda_1}e_1, \sqrt{\lambda_2}e_2, \ldots, \sqrt{\lambda_m}e_m\right] \qquad \Psi = diag\{\psi_i\} \tag{4.146}$$

□ **Sample Factor Decomposition**

From sample cov. matrix S and eigenvalue-eigenvector pairs $(\hat{\lambda}_i, e_i)$, pick the first m paris to form $L = \{\ell_{ij}\}$:

$$\hat{L} = \{\hat{\ell}_{ij}\} = \left[\sqrt{\hat{\lambda}_1}\hat{e}_1, \sqrt{\hat{\lambda}_2}\hat{e}_2, \ldots, \sqrt{\hat{\lambda}_m}\hat{e}_m\right] \qquad \hat{\Psi} = diag\{s_{ii} - \sum_{k=1}^{m} \hat{\ell}_{ik}^2\} \tag{4.147}$$

- Selection of m: Construct Residual Matrix

$$\hat{E} = S - (\hat{L}\hat{L}' + \hat{\Psi}) \tag{4.148}$$

Residual matrix is trace 0, pick m such that

$$\text{Sum of All Elements in } \hat{E} < \sum_{k=m+1}^{p} \hat{\lambda}_k^2 \text{ small enough} \tag{4.149}$$

4.4.3 MLE Method

Assumption: Factor F and error ε are normal.(Then also $X \sim N_p(\mu, \Sigma)$ is normal)

$$F \sim N_m(0, I_m) \quad \varepsilon \sim N_p(0, \Psi) \quad X \sim N_p(\mu, \Sigma) \tag{4.150}$$

Likelihood Function:

$$L(\mu, \Sigma) = (2\pi)^{-np/2}|\Sigma|^{-n/2} \exp\left(-\frac{1}{2}tr\left[\Sigma^{-1}\left(\sum_{k=1}^{n}(x_k - \bar{x})(x_k - \bar{x})' + n(\bar{x} - \mu)(\bar{x} - \mu)'\right)\right]\right) \tag{4.151}$$

Maximize L to get $\hat{L}$ and $\hat{\Psi}$, usually for convenient (and to counteract the arbitrariness of factor rotation) we further assume

$$L'\Psi^{-1}L = \Xi \text{ (diagonal matrix)} \tag{4.152}$$

- Estimtor of communality variance h_i^2:

$$\hat{h}_i^2 = \sum_{k=1}^{m} \hat{l}_{ik}^2 \tag{4.153}$$

Section 4.5 Canonical Correlation Analysis

Key idea of CCA: For a model with two multivariate population $X^{(1)} = (X_1^{(1)}, X_2^{(1)}, \dots, X_p^{(1)})$, $X^{(2)} = (X_1^{(2)}, X_2^{(2)}, \dots, X_q^{(2)})$ with covariance

$$\underset{(p+q)\times(p+q)}{\Sigma} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{4.154}$$

find a few condensed variables to measure their similarity.

4.5.1 Canonical Variate Pair

By using the linear combination, we can construct a pair of vector $\underset{p\times1}{a}$ and $\underset{q\times1}{b}$ such that $corr(a'X^{(1)}, b'X^{(2)})$ large, i.e.

$$\{a, b\} = \arg\max_{a,b\neq0} \frac{a'\Sigma_{12}b}{\sqrt{a'\Sigma_{11}a}\sqrt{b'\Sigma_{22}b}} \tag{4.155}$$

where $U_1 = a'X^{(1)}$, $V_1 = b'X^{(2)}$ with $var(U_1) = var(V_1) = 1$ are the **(first) canonical variate pair**, and $\rho_1^* = corr(U_1, V_1)$ is the **(first) canonical correlation**.

Similarly, the k^{th} canonical pair (U_k, V_k) satisfy the same criterion as equation 4.155 ~ page 134 but with $a_k \in \text{span}\{a_1, \dots, a_{k-1}\}^\perp$, $b_k \in \text{span}\{b_1, \dots, b_{k-1}\}^\perp$, $k \leq \min\{p, q\}$.

Result: U_k, V_k can be expressed as

$$U_k = a_k'X^{(1)} = e_k'\Sigma_{11}^{-1/2}X^{(1)} \qquad V_k = b_k'X^{(2)} = f_k'\Sigma_{22}^{-1/2}X^{(2)} \tag{4.156}$$

where e_k is the k^{th} eigen vector of $\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1/2}$, f_k is the k^{th} eigenvector of $\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1/2}$. e_k and f_k satisfies:

$$f_k = \frac{1}{\rho_k^*}\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1/2}e_k \quad e_k = \frac{1}{\rho_k^*}\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2}f_k \tag{4.157}$$

4.5.2 Canonical Correlation based on Standardized Variables

Using standardized variable of X:

$$Z_k^{(\nu)} = \frac{X_k^{(\nu)} - \mu_k^{(\nu)}}{\sqrt{\sigma_{kk}^{(\nu)}}},\ k = 1, 2, \dots p \text{ or } q,\ \nu = 1, 2 \tag{4.158}$$

with covariance

$$\underset{(p+q)\times(p+q)}{\rho} = V^{-1/2}\Sigma V^{-1/2} = \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \tag{4.159}$$

And similarly, the CCA pair is

$$U_k = a_k'Z^{(1)} = e_k'\rho_{11}^{-1/2}Z^{(1)} \qquad V_k = b_k'Z^{(2)} = f_k'\rho_{22}^{-1/2}Z^{(2)} \tag{4.160}$$

with e_k is the k^{th} eigenvector of $\rho_1 1^{-1/2}\rho_{12}\rho_{22}^{-1}\rho_{21}\rho_{11}^{-1/2}$, $f_{[}k]$ is the k^{th} eigenvector of $\rho_{22}^{-1/2}\rho_{21}\rho_{11}^{-1}\rho_{12}\rho_{22}^{-1/2}$, and

$$f_k = \frac{1}{\rho_k^*}\rho_{22}^{-1/2}\rho_{21}\rho_{11}^{-1/2}e_k \quad e_k = \frac{1}{\rho_k^*}\rho_{11}^{-1/2}\rho_{12}\rho_{22}^{-1/2}f_k \tag{4.161}$$

4.5.3 Sample Canonical Correlation

Replacement:

$$\Sigma \longrightarrow S \qquad \rho \longrightarrow R \tag{4.162}$$

to get

$$\hat{U} = \hat{A}x^{(1)} \qquad \hat{V} = \hat{B}x^{(2)} \tag{4.163}$$

and we can use $\hat{U}, \hat{V},\ \hat{A}, \hat{B}$ to express S_{12} as

$$S_{12} = \hat{A}^{-1} \begin{bmatrix} \hat{\rho}_1^* & 0 & \dots & 0 \\ 0 & \hat{\rho}_2^* & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\rho}_p^* \end{bmatrix} (\hat{B}^{-1})' \tag{4.164}$$

When applying CCA, we pick the first r canonical variable, thus some infomation is lost. But we hope the first r canonical variables can contain enough information of $X^{(1)}$ and $X^{(2)}$.

Determine of r: consider the error if approximation by expressing

$$\hat{A}^{-1} = [\alpha_1, \alpha_2, \dots, \alpha_p] \qquad \hat{B}^{-1} = [\beta_1, \beta_2, \dots, \beta_p] \tag{4.165}$$

and

$$S_{12} = \sum_{i=1}^{p} \hat{\rho}_i^* \alpha_i \beta_i' \tag{4.166}$$

$$S_{11} = \hat{A}^{-1}(\hat{A}^{-1})' = \sum_{i=1}^{p} \alpha_i \alpha_i' \tag{4.167}$$

$$S_{22} = \hat{B}^{-1}(\hat{B}^{-1})' = \sum_{i=1}^{p} \beta_i \beta_i' \tag{4.168}$$

Total sample variance explained by the first r canonical variables:

$$\frac{\sum_{i=1}^{r} \alpha_i' \alpha_i}{tr(S_{11})} \qquad \frac{\sum_{i=1}^{r} \beta_i' \beta_i}{tr(S_{22})} \tag{4.169}$$

Section 4.6 Discriminant Analysis

Key idea of DA: for $\underset{n \times p}{X}$ with an extra column labeling the classification, we want to determine a rule to assign new objects. More specifically, determine the classification region R_i for each class π_i.

More on this topic see section 9.2 ~ page 246 and section 9.3 ~ page 251.

4.6.1 Classification Criterion

- Two-category classification case: Each row of X is labeled in π_1 or π_2, for two-category, only one of R_1, R_2 is needed.

  Some basic concept in classification model:

- Prior Possibility p_i, $i = 1, 2$;
- Penalty for misclassification $c(i|j)$, $i, j = 1, 2$: cost if a π_j object is classified in R_i.
- Conditional Probability $\mathbb{P}(i|j)$, $i, j = 1, 2$: probability that a π_j object falls in region R_i

□ **Determination Criterion:**

- Expected Cost of Misclassification (ECM) Criterion: Minimizing ECM,

$$\text{ECM} = c(2|1)\mathbb{P}(2|1)p_1 + c(1|2)\mathbb{P}(1|2)p_2 \tag{4.170}$$

For two-category problem, R_1, R_2 can be determined as

$$R_1 = \frac{f_{\pi_1}(x)}{f_{\pi_2}(x)} \geq \frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1} \tag{4.171}$$

$$R_2 = \complement_{R_x}^{R_1} = \underset{x \in R}{\arg} \frac{f_{\pi_1}(x)}{f_{\pi_2}(x)} < \frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1} \tag{4.172}$$

- Total Probability of Misclassification (TPM) Criterion: Minimizing TPM,

$$\text{TPM} = \mathbb{P}(\text{misclass}) = \mathbb{P}(2|1)p_1 + \mathbb{P}(1|2)p_2 \tag{4.173}$$

actually $\arg\min \text{TPM} = \underset{c(1|2)=c(2|1)}{\arg\min}\ \text{ECM}$

- Posterior Probability Criterion: Maximize posterior probability $P(\pi_i|x_0)$,

$$\mathbb{P}(X \in \pi_i | X = x_0) = \frac{p_i f_{\pi_i(x_0)}}{p_1 f_{\pi_1}(x_0) + p_2 f_{\pi_2}(x_0)},\ i = 1, 2 \tag{4.174}$$

Also equivalent to ECM for $c(1|2) = c(2|1)$

Here only introduce ECM: $\{R_i\} = \arg\min \text{ECM}$

$$\text{ECM}(i) = \sum_{j \neq i} c(j|i)\mathbb{P}(j|i) \tag{4.175}$$

$$\text{ECM} = \sum_{i=1}^{g} p_i \text{ECM}_i = \sum_{i=1}^{g}\sum_{j \neq i} c(j|i)p(j|i)p_i \tag{4.176}$$

4.6.2 Linear & Quadratic Discriminant Analysis

Now take two-category ECM criterion as example. An estimation to $\mathbb{P}(1|2)$, $\mathbb{P}(2|1)$, i.e. to f_{π_1}, f_{π_2} is needed.

Assumption: for $\pi_1 : X \sim N(\mu_1, \Sigma_1)$,$\pi_2 : X \sim N(\mu_2, \Sigma_2)$, further for

- $\Sigma_1 = \Sigma_2 = \Sigma$: Linear Discriminant Analysis (LDA).

$$f_{\pi_i}(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)'\Sigma^{-1}(x - \mu_i)\right),\ i = 1, 2 \tag{4.177}$$

then

$$R_1 = \underset{x\in R}{\arg}\ (\mu_1-\mu_2)'\Sigma^{-1}x - \frac{1}{2}(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2) \geq \ln\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right) \tag{4.178}$$

$$R_2 = \underset{x\in R}{\arg}\ (\mu_1-\mu_2)'\Sigma^{-1}x - \frac{1}{2}(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2) < \ln\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right) \tag{4.179}$$

Note that L.H.S. is a linear combination of x, thus called LinearDA.

Sample estimation to Σ: use pooled variance in equation 4.103 ~ page 127.

- $\Sigma_1 \neq \Sigma_2$: Quadratic Discriminant Analysis (QDA).

$$f_{\pi_i}(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right),\ i=1,2 \tag{4.180}$$

then

$$R_1 = -\frac{1}{2}x'(\Sigma_1^{-1}-\Sigma_2^{-1})x + (\mu_1'\Sigma_1^{-1}-\mu_2'\Sigma_2^{-1})x - \frac{1}{2}\ln\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \frac{1}{2}(\mu_1'\Sigma_1^{-1}\mu_1 - \mu_2'\Sigma_2^{-1}\mu_2) \geq \ln\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right) \tag{4.181}$$

$$R_2 = -\frac{1}{2}x'(\Sigma_1^{-1}-\Sigma_2^{-1})x + (\mu_1'\Sigma_1^{-1}-\mu_2'\Sigma_2^{-1})x - \frac{1}{2}\ln\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \frac{1}{2}(\mu_1'\Sigma_1^{-1}\mu_1 - \mu_2'\Sigma_2^{-1}\mu_2) < \ln\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right) \tag{4.182}$$

Note that L.H.S. is a quadric form of x, thus called QuadraticDA.

- Two extension: allow more flexible estimation to variance:
  - $\hat{\Sigma}_i(\alpha) = \alpha\hat{\Sigma}_i + (1-\alpha)\hat{\Sigma}$, shrink between QDA and LDA;
  - $\hat{\Sigma}_i(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\hat{\sigma}^2 I$, shrink toward scalar cov.

4.6.3 Fisher's Discriminant Analysis

Project X onto some hyperplane and conduct low-dimensional classification.

Project x onto some hyperplane by $y = a'x$, then we maximize $\psi = \frac{\text{mean of treatment}^2}{\text{variance}}$[14]. i.e.

$$\psi = \frac{\sum_{i=1}^{g}(\mu_{iY}-\mu_Y)^2}{\sigma_Y^2} = \frac{a'\left(\sum_{i=1}^{g}(\mu_i-\mu)(\mu_i-\mu)'\right)a}{a'\Sigma a} = \frac{a'B_\mu a}{a'\Sigma a} \tag{4.186}$$

[14]MANOVA Model: For g groups with same Σ, consider an MANOVA model: $X_{ij} = \mu + \tau_i + e_{ij}$. Then MANOVA table gives Sum of Squares and cross Products (SSP):

$$\text{Regression:}B = \sum_{i=1}^{g} n_i(\bar{x}_i-\bar{x})(\bar{x}_i-\bar{x})' \tag{4.183}$$

$$\text{Error:}W = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{ij}-\bar{x}_i)(x_{ij}-\bar{x}_i)' \tag{4.184}$$

$$\text{Total:}T = B + W = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{ij}-\bar{x})(x_{ij}-\bar{x})' \tag{4.185}$$

use B and W to measure the variance of sample.

Result: a is the largest eigen vector of $W^{-1}B$.

Relation between FDA and LDA: in FDA, take the first ξ eigenvectors to conduct classification, thus loses more information. But when $\xi = g-1$, FDA $\equiv$ LDA.[15]

4.6.4 Evaluation of Discriminant Model

□ **Judging Index:**

- Total Probability of Misclassification (TPM):

$$\text{TPM} = p_1\mathbb{P}(2|1) + p_2\mathbb{P}(1|2) = p_1\int_{R_2} f_{\pi_1}(x)\,\mathrm{d}x + p_2\int_{R_1} f_{\pi_2}(x)\,\mathrm{d}x \tag{4.187}$$

- APparent Error Rate (APER): used with cross validation (CV). The fraction of misclassification in training set.

Section 4.7 Clustering Analysis

Key idea of CA: Group a collection of data according to similarity and relation of objects.

More about this topic see section 9.5 ~ page 258.

4.7.1 Agglomerative Clustering Algorithm

□ **Clustring Algorithm**

Hierarchical clustering: start with individual points and combine them to form groups.

Algorithm *Hierarchical Clustering*

1. All $k = n$ points are individual clusters;
2. In each iteration step k:
   (a) Use a distance/dissimilarity matrix $\underset{k\times k}{D}$ to express distances between clusters; the 'distance' between clusters is diversified, choice of which see the following part;
   (b) merge the closest pair of clusters(or points) to form a larger cluster, and now number of clusters
   (c) $k = k-1$;
3. Only $k = 1$ cluster is left
4. Choose a proper threshold of distance to determine K

□ **Choice of between-cluster distance: To express distance between two clusters A and B,**

- Choice of distance functional $D(\cdot,\cdot)$:

[15] Because a is eigenvector of $W^{-1}B$, while $\text{rk}(B) = g-1$, thus there are $g-1$ non-zero eigenvalues at most.

  - Euclidean Distance D_{E};
  - Mahalanobis Distance D_{M};
  - Jaccard Distance $D_{\mathrm{J}} = 1 - \dfrac{|A \cap B|}{|A \cup B|}$;
  - etc.

- Location choice of cluster:
  - Complete link: $\max D(a \in A, b \in B)$;
  - Single link: $\min D(a \in A, b \in B)$;
  - Centroid distance: $D(A \text{ centroid}, B \text{ centroid})$;
  - Group average: $\langle D(a \in A, b \in B) \rangle$

- Note: pros-and-cons of agglomerative clustering algorithm
  - No assumptions for final k needed;
  - Intuitive display of relations;
  - Large computational requirement: $\sim O(n^3)$;
  - Sensitive to noise and outliers.

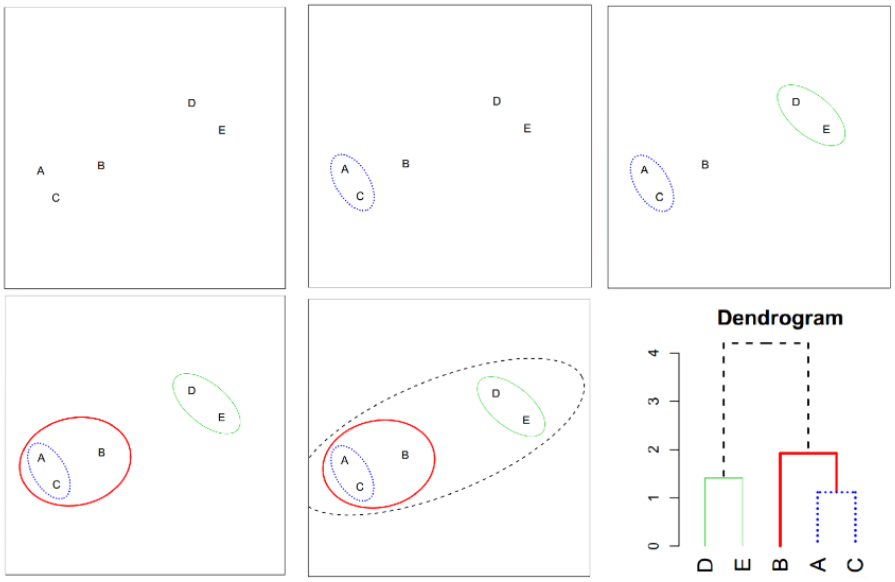


图 4.1: Illustration of Hierarchical Clustering

4.7.2 K-Means Clustering Algorithm

Assume we have a preset number K of clusters , we can use K-means clustering.

Algorithm *K-Means Clustering*

1. Choose/Preset number of clusters K;

2. Select K points as initial centroids, useful methods:
   - Randomly select;
   - Use Centroid of agglomerative algorithm;
   - Successively pick the farthest point from others.
3. In each iteration of centroids:
   (a) For all points i, calculate its distance from the l^{th} centroid $D(i,l)$
   (b) Classify each i point to the nearest centroid cluster;
   (c) Re-calculate the centroid of new K clusters;
4. Repeat until convergence.(Convergence criterion can be e.g. $\langle\sum_i D(i\in g_l,l)\rangle\to$ const)

Note: pros and cons of K-Means clustering algorithm:

- Efficient: $\sim O(n)$;
- Sensitive to outliers;
- Ineffective for non-convex shapes.

4.7.3 Gaussian Mixture Model with Expectation Maximization Algorithm

The Gaussian Mixture Model (GMM) for clustering assumes X is generated from a mixed distribution of K normal, i.e. X has probability π_l to be generated from corresponding normal $N(\mu_l,\Sigma_l)$:

$$X\sim\sum_{l=1}^{K}\pi_l N(\mu_l,\Sigma_l)=\sum_{l=1}^{K}\pi_l N(\theta_l),\quad \sum_{l=1}^{K}\pi_l=1,\ \pi_l\geq 0. \tag{4.188}$$

Use its likelihood function $L(\theta;x)$ and maximize posterior probability by $\dfrac{\partial\ell}{\partial\theta}$:

$$L(\{\pi_l\},\{\theta_l\};x)=\prod_{i=1}^{N}\sum_{l=1}^{K}\pi_l\frac{1}{(2\pi)^{p/2}|\Sigma_l|^{1/2}}\exp\left(-\frac{1}{2}(x_i-\mu_l)'\Sigma_l^{-1}(x_i-\mu_l)\right) \tag{4.189}$$

E-M Algorithm uses the ELBO maximizing method, detail see section 5.5 ~ page 183. For simplification express $\theta\equiv\{\cup\pi_l,\cup\mu_l,\cup\Sigma_l\}$. The maximizing function $Q(\theta|\theta^{(t)})$ for GMM model and corresponding iteration:

$$\theta^{(t+1)}=\arg\max_{\theta}Q(\theta|\theta^{(t)})=\arg\max_{\theta}\sum_{i=1}^{N}\sum_{l=1}^{K}\gamma_{il}^{(t)}\log\pi_l\phi(x_i|\mu_l,\Sigma_l),\quad \gamma_{il}^{(t)}\equiv\frac{\pi_l^{(t)}\phi(x_i|\mu_l^{(t)},\Sigma_l^{(t)})}{\sum_{j=1}^{K}\pi_j^{(t)}\phi(x_i|\mu_j^{(t)},\Sigma_j^{(t)})} \tag{4.190}$$

Lagrange Multiplier: Extreme value $\arg\max_{\theta}Q(\theta|\theta^{(t)})$ with constraint $\sum_{l=1}^{K}\pi_l=1$ requires

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial\mu_l}=0\quad\frac{\partial Q(\theta|\theta^{(t)})}{\partial\Sigma_l^{-1}}=0\quad\frac{\partial Q(\theta|\theta^{(t)})+\lambda(\sum_{l=1}^{K}\pi_l-1)}{\partial\pi_l}=0,\quad\forall l=1,2,\ldots,K \tag{4.191}$$

Result:

$$\begin{cases} \mu_l^{(t+1)} = \dfrac{\sum\limits_{i=1}^{N} \gamma_{il}^{(t)} x_i}{\sum\limits_{i=1}^{N} \gamma_{il}^{(t)}} \\ \Sigma_l^{(t+1)} = \dfrac{\sum\limits_{i=1}^{N} \gamma_{il}^{(t)} (x_i - \mu_l)(x_i - \mu_l)'}{\sum\limits_{i=1}^{N} \gamma_{il}^{(t)}} \\ \pi_l^{(t+1)} = \dfrac{1}{N} \sum_{i=1}^{N} \gamma_{il}^{(t)} \end{cases} \tag{4.192}$$

$$\gamma_{il}^{(t)} \equiv \frac{\pi_l^{(t)} \phi(x_i | \mu_l^{(t)}, \Sigma_l^{(t)})}{\sum\limits_{j=1}^{K} \pi_j^{(t)} \phi(x_i | \mu_j^{(t)}, \Sigma_j^{(t)})} \tag{4.193}$$

where γ_{il} is the posterior probability that the i^{th} object belongs to the l^{th} group.

The above constraint equations are difficult to solve, use iteration algorithm:

Algorithm *EM-Algorithm for Gaussian Mixture Model*

1. Use e.g. K-means method to set an initial estimation as $(\hat{\mu}_l^{(0)}, \hat{\Sigma}_l^{(0)})$, $\hat{\pi}_l^{(0)} = 1/K$;
2. Repeat Expectation & Maximization:
   (a) $\text{E}_{\text{xpectation}}$-Step: Compute posterior of latent variable on each point;

$$\hat{\gamma}_{il}^{(t)} = \frac{\pi_l^{(t)} \phi(x_i | \mu_l^{(t)}, \Sigma_l^{(t)})}{\sum\limits_{j=1}^{K} \pi_j^{(t)} \phi(x_i | \mu_j^{(t)}, \Sigma_j^{(t)})}, \quad 1 \le i \le N,\ 1 \le l \le K \tag{4.194}$$

   (b) $\text{M}_{\text{aximize}}$-Step: Re-calculate parameters $\{\mu_l, \Sigma_l, \pi_l\}$ by equation 4.192 ~ page 141.
3. Repeat until convergence.

Note: EM method for Gaussion Mixture Model is a greedy algorithm $\longrightarrow$ local maximum.

4.7.4 DBSCAN & OPTICS Density Clustering Algorithm

DBSCAN algorithm (Density-Based Spatial Clustering of Application with Noise) is a kind of density clustering algorithm. OPTICS algorithm (Ordering Point To Indentify the Cluster Structure) is its improved version.

□ **DBSCAN Algorithm** Key (preset) index in DBSCAN:

- Eps ε: Radius of neighbourhood of a point;
- MinPts M: Minimum number of points to be indentified as cluster core point, usually choose $M \ge \dim + 1$;
- (Also, a distance norm is needed, e.g. Euclidean D).

Notation:

- ε neighbourhood of point x_i:

$$\mathcal{N}_\varepsilon(x_i) \equiv \{y \in \mathbb{R}^n : 0 < D(y,x) < \varepsilon\} \tag{4.195}$$

- 'Density' (is actually an integer):

$$\rho_\varepsilon(x_i) \equiv \#x_j \in \mathcal{N}_\varepsilon(x_i) \tag{4.196}$$

- Three types of Points: X_c, X_{bd}, X_{noi}.
  - Core Point: label an x_i as core point if

$$\rho_\varepsilon(x_i) \geq M \tag{4.197}$$

    Denote the set of core point as X_c, and set of non-core point as X_{nc}
  - Border Point: label an $x_j \in X_{nc}$ as border point if

$$\exists(x_i \in X_c) \in \mathcal{N}_\varepsilon(x_j) \& x_j \in X_{nc} \tag{4.198}$$

    Denote the set of border point as X_{bd}
  - Noise Point: the set of noise point is

$$X_{noi} \equiv \complement_X^{X_c \cup X_{bd}} \tag{4.199}$$

- Point Relations: DDR, DR, DC
  - Directly Density Reachable: For $x_i, x_j \in X$, if $x_i \in X_c$, $x_j \in \mathcal{N}_\varepsilon(x_i)$, then say x_j is DDR from x_i;
  - Density Reachable: For point chain $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$, $m \geq 2$. If $x_{i_{\kappa+1}}$ is DDR from x_{i_κ}, $\forall 1 \leq \kappa \leq m-1$, then say x_{i_m} is DR from x_{i_1}.
  - Density Connected: For point x_{i_1}, x_{i_2}, x_{i_3}, if x_{i_2} and x_{i_3} are both DR from x_{i_1}, then say x_{i_2} and x_{i_3} are DC.

Note: DR is not symmetric for x_{i_1} and x_{i_m}; while DC is.

DBSCAN algorithm classify all points that are Density Connected to each other into a cluster $C \subset X$, i.e.

$$\text{Maximality:} x \in C \&\& y \text{ DR from } x \Rightarrow y \in C \tag{4.200}$$

$$\text{Connectivity:} x, y \in C \Rightarrow x, y \text{ DC.} \tag{4.201}$$

Pros and cons of DBSCAN:

- Insensitive to noise;
- Based on density, with no constraint on the shape of cluster;
- Suitable for clusters with uniformly densed data, otherwise difficult to choose proper Eps ε;
- Complexity $\sim O(n^2)$, at least $O(n \log n)$.

□ OPTICS Algorithm

OPTICS is based on DBSCAN and shares most of the basic concepts and ideas. Further define the following distance (preset ε and M):

- Core Distance: For $x_i \in X_c$, the smallest distance allowing x_i to become core point.

$$\text{CD}(x_i) = D(x_i, N_\varepsilon^M(x_i)),\ \rho_\varepsilon(x_i) \geq M \tag{4.202}$$

  where $N_\varepsilon^M(x_i)$ is the M^{th} closest point from x_i;

- Reachablity Distance: For $y \in X$, $x_i \in X_c \subset X$,

$$\text{RD}(y, x_i) = \max\{CD(x_i, D(y, x_i))\} \tag{4.203}$$

  Or equivlantly

$$\text{RD}(y, x_i) = \underset{\rho_d(x_i)\geq M, y\in\mathcal{N}_d(x_i)}{\arg\min} d \tag{4.204}$$

Algorithm flow:

Algorithm *OPTICS*

1. Construct X_c based on preset M, ε;
2. Pick an 'unprocessed' point $x_{n_i} \in X_c$ and calculate $\text{RD}(x_j, x_{n_i})$, $\forall$'unprocessed'$x_j \in \mathcal{N}_\varepsilon(x_{n_i}) \cap X_c$. Pick the $x_j \in X_c$ with smallest RD and label as $x_{n_{i+1}}$ processed;
3. Repeat step 2 until all points are processed. Output $\{x_{n_i}\} = (x_{n_1}, x_{n_2}, \ldots, x_{n_{|X_c|}})$. Each x_{n_i} is attached with a $\text{CD}(x_{n_i})$ and a $r(x_{n_i}) := \text{RD}(x_{n_{i-1}}, x_{n_i})$[16].

Then break the ordering sequence n_i according to $r(x_{n_i})$, .e.g. break n_i if $r(x_{n_i}) \geq \tilde{\varepsilon}$

Comment: OPTICS is more stable than DBSCAN, capable of dealing with multi-density clustering.

[16] For $i = 1$, just define as 0

Chapter. V　统计计算与软件部分

Instructor: Zaiying Zhou

Section 5.1　Algorithm Theory Introduction

5.1.1　Finite Precision Computation

An arbitrary real number $r \in \mathbb{R}$ is represented as (the nearest adjacent) float number v_r. A float is basically stored as (example take 32-bit float): 1 bit **S**ign + 8 bit **E**xponent + 23 bit **M**antissa.

$$v_r = (-1)^S \times 2^{E-127} \times \left(1 + \sum_{i=1}^{23} (M_i \times 2^{-i})\right) \tag{5.1}$$

Further, extreme value of (M, E) is used for some 'special value': denormalized number, NaN, inf, etc.

- Denormalized number: to fill the gap $[0, \pm 2^{-126}](E = 1)$, for $E = 0$ extremely small number, definition use

$$v_{\text{denormalized}} = (-1)^S \times 2^{1-127} \times \left(0 + \sum_{i=1}^{23} (M_i \times 2^{-i})\right) \tag{5.2}$$

  i.e. for $E = 0$, range $[2^{-127}, 2^{-126})_{\text{nor}} \to [0, 2^{-126})_{\text{denor}}$.

- NaN: $(E = 255, M \neq 0)$
- inf: $(E = 255, M = 0)$

	$E = 0$	$0 < E < E_{\max}$	$E = E_{\max}$
$M = 0$	± 0	$v_{\text{normalized}}$	$\pm\infty$
$M \neq 0$	$v_{\text{denormalized}}$		NaN

表 5.1: (De)Normalized Number

Use v_r to represent r: approximation $r \sim v_r$, the round-off error of r:

- Absolute rounding error:

$$\varepsilon = |r - v_r| \tag{5.3}$$

- Relative rounding error:

$$\varepsilon_{\text{machine}} = \frac{|r - v_r|}{|r|} = \text{const} \tag{5.4}$$

Note that for large $|r|$, the adjacency between floats $|r - v_r| = |r|\varepsilon_{\text{machine}}$ might be large, even cause some integer missing.

□ **Representation and arithmetic of floating-point number follows IEEE-754 standard**

- For 32-bit float (single precision float): 1 bit **S**ign + 8 bit **E**xponent + 23 bit **M**antissa. $\varepsilon_{\text{machine}} = 0.5 \times 2^{-23} = 2^{-24}$

$$v = (-1)^S \times 2^{E-127} \times \left(1 + \sum_{i=1}^{23}(M_i \times 2^{-i})\right) \in [-3.4 \times 10^{38}, 3.4 \times 10^{38}] \tag{5.5}$$

- For 64-bit float (double precision float): 1 bit **S**ign + 11 bit **E**xponent + 52 bit **M**antissa. $\varepsilon_{\text{machine}} = 0.5 \times 2^{-52} = 2^{-53}$

$$v = (-1)^S \times 2^{E-1023} \times \left(1 + \sum_{i=1}^{52}(M_i \times 2^{-i})\right) \in [-1.79 \times 10^{308}, 1.79 \times 10^{308}] \tag{5.6}$$

Key point for algorithm design: aviod plus/minus of numbers of significantly large magnitude difference.

5.1.2 Stability & Accuracy

- Forward/Backward Error:

  For a algorithm design $\tilde{f}$ of a problem f, with input x. Denote:

  - Expected output: $y \equiv f(x)$
  - Algorithm output: $\tilde{y} \equiv \tilde{f}(x)$
  - Forward Error: $\Delta_F = \tilde{f}(x) - f(x)$
  - Backward Error: $\Delta_B = \underset{f(\tilde{x})=\tilde{f}(x)}{\arg\min} |\tilde{x} - x|$

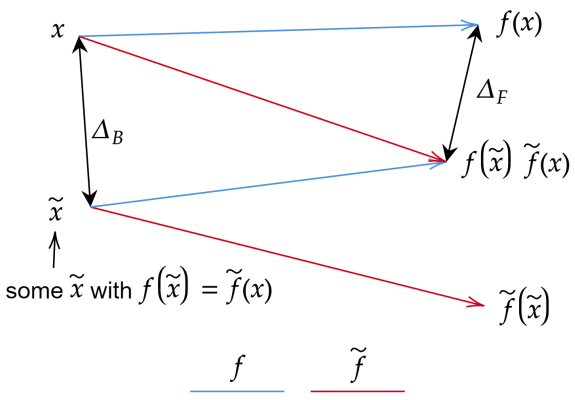


图 5.1: Illustration of Forward/Backward Error

- (Forward) Stability: An algorithm $\tilde{f}$ is stable if

$$\frac{\|\tilde{f}(x)-f(\tilde{x})\|}{\|f(\tilde{x})\|}=O(\varepsilon_{\text{machine}}),\ \forall\frac{\|\tilde{x}-x\|}{\|x\|}=O(\varepsilon_{\text{machine}}) \tag{5.7}$$

- Condition Number of problem f:
  - Absolute condition number:

$$\hat{\kappa}(x)=\lim_{\varepsilon\to 0}\sup_{\|\delta x\|<\varepsilon}\frac{\|\delta f(x)\|}{\|\delta x\|}=\left\|\frac{\partial f}{\partial x}\right\| \tag{5.8}$$

  - Relative condition number:

$$\kappa(x)=\lim_{\varepsilon\to 0}\sup_{\|\delta x\|<\varepsilon}\frac{\|\delta f\|/\|f\|}{\|\delta x\|/\|x\|} \tag{5.9}$$

(Relative) Condition Number of Matrix $\underset{m\times m}{A}$:

- $f(x)\equiv Ax$:

$$\kappa=\|A\|\frac{\|x\|}{\|Ax\|}\le\|A\|\|A^{-1}\| \tag{5.10}$$

- $f(b)\equiv$ solving $Ax=b$

$$\kappa=\|A^{-1}\|\frac{\|b\|}{\|x\|}\le\|A^{-1}\|\|A\| \tag{5.11}$$

Thus for matrix A, denote

$$\kappa(A)\equiv\|A\|\|A^{-1}\| \tag{5.12}$$

- For ℓ_2 norm $\|\cdot\|_2$: $\kappa(A)=\dfrac{\sigma_1}{\sigma_m}$ [1]

5.1.3 Iteration Algorithm

Iteration methods are used especially for problems without analytical solution, to obtain a numerical solution.

Iteration method: for problem f with solution x^* design an iteration function g: $X\to X$ so that

$$\lim_{n\to\infty}g^{\{n\}}(x)=\lim_{n\to\infty}\underbrace{g(g(g(\dots g(g}_{n}(x))\dots)))=x^* \tag{5.13}$$

then get solution by setting initial input value $x^{(0)}$ and calculate $x^{(t+1)}=g(x^{(t)})$ repeatedly until convergence as approximate solution.

□ **Three Steps for Iteration:**

Algorithm *General Steps for Iteration*

[1] Knowledge about matrix norm see section 4.1.2 ~ page 118

1. Starting: set $x^{(0)}$, more trials to initial value is recommended
2. Updating: $x^{(t+1)} = g(x^{(t)})$, $\forall t = 0, 1, 2, \ldots$
3. Stopping: when to stop, can choose various stopping criterion, e.g.
   - Absolute convergence criterion

$$|x^{(t+1)} - x^{(t)}| < \varepsilon \tag{5.14}$$

   - Relative convergence criterion

$$\frac{|x^{(t+1)} - x^{(t)}|}{|x^{(t)}|} < \phi \tag{5.15}$$

   - Relative convergence criterion (2), avoid $x^{(t)} = 0$

$$\frac{|x^{(t+1)} - x^{(t)}|}{|x^{(t)}| + \xi} < \phi \tag{5.16}$$

□ **Convergence Order and Convergence Rate**

For each iteration value $x^{(t)}$, define iteration error as $\varepsilon^{(t)} \equiv x^{(t)} - x^*$. Then an iteration method $\lim_{t\to\infty} \varepsilon^{(t)} = 0$ has convergence order α and convergence rate c as:

$$\lim_{t\to\infty} \frac{|\varepsilon^{(t+1)}|}{|\varepsilon^{(t)}|^\alpha} = c \tag{5.17}$$

A large α and small c declare a quick convergence.(Large α is needed more)

Comment: Actually convergence rate and order are generally dependent on specific problem, so we usually estimate α, c using some approximation/scaling to represent a generally case.

5.1.4 Constrained Optimize Theory

□ **Primal Problem**

For optimize problem in convex set $\mathcal{X}$

$$\begin{aligned} \arg\min_{x\in\mathcal{X}} \quad & f(x) && \text{(P)} \\ s.t. \quad & g_i(x) \le 0, \quad i = 1, 2, \ldots, k && (5.18) \\ & h_j(x) = 0, \quad j = 1, 2, \ldots, l && (5.19) \end{aligned}$$

which is called the **primal problem** for optimization.

The **generalized Lagrange function** for primal problem defined as

$$\begin{aligned} \mathcal{L}(x, \kappa, \lambda) \equiv & f(x) + \sum_{i=1}^{k} \kappa_i g_i(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \\ w.r.t. \quad & \kappa_i \ge 0, \quad i = 1, 2, \ldots, k \end{aligned} \tag{5.20}$$

---

and we could further define a function of x:

$$\theta_P(x) \equiv \max_{\kappa,\lambda:\kappa_i\geq 0} \mathcal{L}(x,\kappa,\lambda) = \begin{cases} f(x) & \text{constraint } g,\ h \text{ satisfied} \\ +\infty & \text{contraint unsatisfied} \end{cases} \tag{5.21}$$

which means we can give the solution value of primal problem (P) simply by minimizing $\theta_P(x)$, minimum denoted p^*

$$p^* \equiv \min_x \theta_P(x) = \min_x \max_{\kappa,\lambda:\kappa_i\geq 0} \mathcal{L}(x,\kappa,\lambda) \tag{5.22}$$

□ **Dual Problem**

Similar to primal problem, we can define a function of κ, λ:

$$\theta_D(\kappa,\lambda) \equiv \min_x \mathcal{L}(x,\kappa,\lambda) \tag{5.23}$$

and similarly get the **dual problem** of primal, value denoted d^*

$$d^* \equiv \max_{\kappa,\lambda:\kappa\geq 0} \theta_D(\kappa,\lambda) = \max_{\kappa,\lambda:\kappa\geq 0} \min_x \mathcal{L}(x,\kappa,\lambda) \tag{5.24}$$

it is obvious that

$$d^* = \max_{\kappa,\lambda:\kappa\geq 0} \min_x \mathcal{L}(x,\kappa,\lambda) \leq \min_x \max_{\kappa,\lambda:\kappa_i\geq 0} \mathcal{L}(x,\kappa,\lambda) = p^* \tag{5.25}$$

□ **Karush-Kuhn-Tucker Condition (KKT Condition)**

KKT condition to allow $d^* = p^*$ at $(x^*, \kappa^*, \lambda^*)$: in the case that

- $f(x)$ and $g_i(x)$ are convex
- $h_j(x)$ in the form of affine function $A_j x + b$
- $g_i(x)$ are feasible constraints

then KKT $\Leftrightarrow$ $p^* = d^* = \mathcal{L}(x^*, \kappa^*, \lambda^*)$.

the KKT conditions are:

$$\begin{aligned} &\nabla_x \mathcal{L}(x^*,\kappa^*,\lambda^*) = 0 \\ &\kappa_i^* g_i(x^*) = 0 && i = 1,2,\ldots,k \\ &g_i(x^*) \leq 0 && i = 1,2,\ldots,k \\ &\kappa_i \geq 0 && i = 1,2,\ldots,k \\ &\lambda_j(x^*) = 0 && j = 1,2,\ldots,l \end{aligned} \tag{5.26}$$

Section 5.2 Algebratic Problem in Statistics

Considering the data structure and algorithm implement, many fundamental problems in statistics are basically algebratic problem, e.g.

- Matrix multiplication:

$$y = Ax, \text{ solve } y \tag{5.27}$$

- Linear equation solution:

$$b = Ax = \sum_{i=1}^{n} x_i a_i, \text{ solve } x \tag{5.28}$$

- OLS solution:

$$\hat{\beta} = (X'X)^{-1}XY \tag{5.29}$$

Generally speaking matrix A can be constructed in an arbitrary form, so an algorithm implementation needs **matrix composition** so that we have a better form to handle.

5.2.1 Matrix Operation

- Inverse Matrix: Inverse matrix of $A = [a_1, \ldots, a_m]$ satisfies

$$A^{-1}A = AA^{-1} = I \tag{5.30}$$

then $Ax = b \Leftrightarrow x = A^{-1}b$

Or generally speaking, solve inverse matrix $A^{-1} = [\alpha_1, \ldots, \alpha_m]$ is solving linear equations

$$A\alpha_i = e_i \tag{5.31}$$

In the view of column space transfrom, A and A^{-1} are mappings between space span$\{e_1, \ldots, e_m\}$ and span$\{a_1, \ldots, a_m\}$, i.e.

$$\text{span}\{e_1, \ldots, e_m\} \underset{A^{-1}b}{\overset{Ax}{\rightleftharpoons}} \text{span}\{a_1, \ldots, a_m\} \tag{5.32}$$

- Unitary Matrix: Furthur for unitary A, denoted as Q with $QQ^* = I$, is an orthonormal transformation.
  - $|Q| = 1$ for rotation, $|Q| = -1$ for reflection.
  - $\lambda_Q = \pm 1$
  - Geometric structure preserved, e.g. inner product and norm.
- Projection:
  - Basic definition of projector P_X: idempotent matrix, project onto hyperplane X

$$P_X^2 = P_X \tag{5.33}$$

– Complementary projector $I - P_X$: onto the complementary space of X

$$(I-P)^2 = I - P \tag{5.34}$$

– Orthogonal Projection: Projector such that $Pv \perp (I-P)v$. Theorem: $Pv \perp (I-P)v \Leftrightarrow P^* = P$

Derivation: Projection of vector v on hyperplane X satisties (denoted as Xp)

$$0 = \langle Xp, Xp - v\rangle = p^*X^*(Xp - v) \Rightarrow p = (X^*X)^{-1}Xv \Rightarrow Xp = X(X^*X)^{-1}Xv = P_Xv \tag{5.35}$$

More Properties of orthogonal projector see section 3.3.2 ~ page 82.

– Orthogonal projector onto vector q:

$$P_q = q(q^*q)^{-1}q^* = \frac{qq^*}{\|q\|_2^2} \tag{5.36}$$

5.2.2 Projection and Least Square Problem

Recall: Linear model $Y = X\beta + \varepsilon$, basically solving linear equation $Y = X\beta$, however generally $Y \notin \text{span}(X)$, then we use OLS method to reach an estimation of β :

$$\hat{\beta} = \arg\min_{\beta} \|Y - X\beta\|^2 \tag{5.37}$$

where for $\|\cdot\| = \ell_2$-norm, $X\hat{\beta}$ is the projection of $X\beta$ onto hyperplane X:

$$X\hat{\beta} = X(X^*X)^{-1}X^*Y \equiv HY = P_XY \tag{5.38}$$

For non-full rank $A = X^*X$: use pseudoinverse $A^+ = (A^*A)^{-1}A^*$

□ **Task of OLS (Linear Model): Solve $\hat{\beta} = (X^*X)^{-1}X^*Y$, or equivalently solve $X^*X\hat{\beta} = X^*Y$**

Note: size of matrix denoted $X = \underset{m\times n}{X}$

- Cholesky decomposition algorithm: computation complexity $\sim mn^2 + \frac{n^3}{3}$
  1. Use Cholesky decomposition for X^*X:

$$A^*A = R^*R \Rightarrow R^*R\hat{\beta} = X^*Y \tag{5.39}$$

  2. Solve $\xi = \arg\{R^*\xi = X^*Y\}$:

$$R^*R\hat{\beta} = X^*Y = R^*\xi \Rightarrow R\hat{\beta} = \xi \tag{5.40}$$

  3. Solve $R\hat{\beta} = \xi$ to get $\hat{\beta}$
- QR decomposition algorithm: computation complexity $\sim 2mn^2 - \frac{2}{3}n^3$
  1. Use e.g. Householder Reflection algorithm to compute $X = QR$

2. use the orthonormal property of Q:

$$X^*X\hat{\beta} = X^*Y \Rightarrow R^*Q^*QR\hat{\beta} = R^*R\hat{\beta} = R^*Q^*Y \Rightarrow R\hat{\beta} = Q^*Y \tag{5.41}$$

3. Solve $R\hat{\beta} = Q^*Y$ to get $\hat{\beta}$

- SVD algorithm: computation complexity $\sim 2mn^2 + 11n^3$

1. Compute SVD of X: $X = U\Sigma V^*$

$$X^*X\hat{\beta} = X^*Y \Rightarrow V\Sigma^2V^*\hat{\beta} = V\Sigma U^*Y \Rightarrow \Sigma V^*\hat{\beta} = U^*Y \tag{5.42}$$

2. Solve $\hat{\beta} = V\Sigma^{-1}U^*Y$ to get $\hat{\beta}$

Algorithm comparison & trade-off: faster $\leftrightsquigarrow$ less stable.

5.2.3 Gaussian LU Decomposition & Cholesky Decomposition

□ **Gaussian Elimination Algorithm**

Gaussian Elimination decomposes matrix A as lower triangular matrix $\times$ upper triangular matrix

$$\underset{m\times m}{A} = \underset{m\times m}{L}\,\underset{m\times m}{U} = \begin{bmatrix} * & & & \\ * & * & & \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} \begin{bmatrix} * & * & \dots & * \\ & * & \dots & * \\ & & \ddots & \vdots \\ & & & * \end{bmatrix} \tag{5.43}$$

Conducted by continuously row transformation of A:

$$L_{m-1}\dots L_2L_1A = L^{-1}A = U \tag{5.44}$$

where each L_i corresponds to a gauss elimination operation such that $[L_i(L_{i-1}\dots L_2L_1A)]_{i+1:m,i} = 0$, with $[L_i(L_{i-1}\dots L_2L_1A)]_{1:i,:}$ fixed. L_i has the form as

$$L_i = I - l_ie_i^*, \qquad l_i = [0,\dots,l_{i+1,i},\dots,l_{m,i}]^T \quad l_{j,i} = A_{ji}/A_{ii} \tag{5.45}$$

Then we have $L = L_1^{-1}L_2^{-1}\dots L_{m-1}^{-1}U$, with $U = L_{m-1}\dots L_2L_1A$

If some pivot element $(L_{i-1}\dots L_1A)_{ii} = 0$, use a row transformation P_i such that $(P_iL_{i-1}\dots L_1A)_{ii} \neq 0$, thus LU decomposition is expanded as

$$L_{m-1}P_{m-1}\dots L_2P_2L_1P_1A = U \tag{5.46}$$

Good properties of $L_i = I - l_ie_i^*$: enable a quick algorithm implement of LU decomposition:

- Inverse of L_i:

$$L_i^{-1} = (I - l_ie_i^*)^{-1} = I + l_ie_i^* \tag{5.47}$$

- Multiplication of L_i^{-1}:

$$L_i^{-1}L_{i+1}^{-1} = (I + l_i e_i^*)(I + l_{i+1}e_{i+1}^*) = I + l_i e_i^* + l_{i+1}e_{i+1}^* \tag{5.48}$$

- Interchangeability of P_i and L_j :

$$L_{m-1}P_{m-1}\dots L_2P_2L_1P_1 = (\tilde{L}_{m-1}\dots\tilde{L}_2\tilde{L}_1)(P_{m-1}\dots P_2P_1), \qquad \tilde{L}_i = P_{m-1}\dots P_{i+1}L_iP_{i+1}^{-1}\dots P_{m-1}^{-1} \tag{5.49}$$

  where note that P_k only exchange row/column k and $\kappa > k$, thus $\tilde{L}_i$ is still left triangular.

Thus get expression of LU decomposition $PA = LU$:

$$PA = LU \quad \begin{cases} P = P_{m-1}\dots P_2P_1 \\ L = (\tilde{L}_{m-1}\dots\tilde{L}_2\tilde{L}_1)^{-1} \\ \tilde{L}_i = P_{m-1}\dots P_{i+1}L_iP_{i+1}\dots P_{m-1} \\ U = L_{m-1}P_m\dots L_2P_2L_1P_1A \end{cases} \tag{5.50}$$

Complexity of Gaussian Elimination:

$$\text{flops}_{\text{GE}} = \sum_{i=1}^{m-1}\sum_{k=i+1}^{m} 2(m-i+1) \sim \frac{2}{3}m^3 \tag{5.51}$$

□ **Cholesky Decomposition**

Hermitian positive-definite matrix A can LU decompose as

$$A = LU = R^*R \tag{5.52}$$

Algorithm: write A in partitioned matrix then conduct symmetric row/column transformation

$$A = \begin{bmatrix} 1 & w_1^* \\ w_1 & K \end{bmatrix} \tag{5.53}$$

$$= \begin{bmatrix} 1 & 0 \\ w_1 & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - w_1w_1^* \end{bmatrix} \begin{bmatrix} 1 & w_1^* \\ 0 & I \end{bmatrix} \tag{5.54}$$

$$= R_1^*K_1R_1 \tag{5.55}$$

Note that K_1 is still hermite positive-definite, we can repeat the above process

$$K_1 = \begin{bmatrix} 1 & 0 \\ 0 & K - w_1w_1^* \end{bmatrix} \tag{5.56}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & w_2 & I \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & K - w_1w_1^* - w_2w_2^* \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & w_2^* \\ 0 & 0 & I \end{bmatrix} \tag{5.57}$$

$$= R_2^*K_2R_2 \tag{5.58}$$

repeat untill $K_m = I$: $A = (R_m R_{m-1} \dots R_1)^* I (R_m R_{m-1} \dots R_1) = R^* R$

Complexity of Cholesky Decomposition:

$$\text{flops}_{\text{CD}} = \sum_{i=1}^{m} \sum_{k=i}^{m} 2(m-k+1) + 1 \sim \frac{1}{3} m^3 \tag{5.59}$$

5.2.4 QR Decomposition: Gram-Schmidt/Householder/Givens Method

QR Decomposition: Orthogonal Triangularization of matrix A

$$\underset{m\times n}{A} = \underset{m\times n}{Q}\ \underset{n\times n}{R} \tag{5.60}$$

$$A = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} = QR = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix} \tag{5.61}$$

Every $\underset{m\times n}{A} \in \mathbb{C}^{m\times n}$ $(m \geq n)$ has QR decomposition, specially:

- Full decomposition exists
- Reduced decomposition with $r_{ii} > 0$ is unique.

Here introduce 3 kinds of algorithm:

- Gram-Schmidt Orthogonalization: $\sim O(2mn^2)$, sequentially orthogonalizes the columns of A, traditional way

⋆ Householder Reflection: $\sim O(2mn^2 - \frac{2}{3}n^3)$, most commonly used, stable for ill & dense matrix

- Givens Rotation: $\sim O(3mn^2 - n^3)$, used for sparse matrix, e.g. Hessenberg matrix

□ **(Classical) Gram-Schmidt Orthogonalization**

Key idea: project a_i onto $\text{span}\{q_1, \dots, q_{i-1}\}^\perp$ as q_i, with q_1 initialized as $\hat{a}_1$, projection coefficient r_{ij} forms R.

For each projection, the projector matrix is

$$P_i = I - \sum_{k=1}^{i-1} q_k q_k^* \tag{5.62}$$

expression of q_i and r_{ij}:

$$r_{ij} = \begin{cases} q_i^* a_j & i \neq j \\ \|a_i - \sum_{k=1}^{i-1} r_{ki} q_k\| & i = j \end{cases} \qquad q_i = \frac{a_i - \sum_{k=1}^{i-1} r_{ki} q_k}{r_{ii}} = \frac{a_i - \sum_{k=1}^{i-1} q_k q_k^* a_i}{\|a_i - \sum_{k=1}^{i-1} q_k q_k^* a_i\|} \tag{5.63}$$

Note: Algorithm implementation of q_i is $(q_{<i}) \to r_{k<i,i} \to q_i \& r_{ii} \to (q_{>i})$

□ Modified Gram-Schmidt Orthogonalization Algorithm

In equation 5.62 ~ page 153, projection of G-S orthogonalization for each a_i is conducted 'simultaneously', while modified G-S decomposition is conducted step by step.

$$P_i = I - \sum_{k=1}^{i-1} q_k q_k^* = \prod_{k=1}^{i-1} (I - q_k q_k^*) \tag{5.64}$$

Decomsition result are the same, but modified algorithm is more stable for numerical computation, avoid problem of recursive q_i.

▷ R. **Code**

Algorithm of CGS/MGS

```r
GS <- function(A,MGS=FALSE){
    stopifnot(is.matrix(A))
    m <- dim(A)[[1]]
    n <- dim(A)[[2]]
    v=matrix(0,nrow = m,ncol=m)
    r=matrix(0,nrow = m,ncol=n)
    q=matrix(0,nrow = m,ncol=m)
    for(j in 1:n){
        v[,j] <- A[,j]
        if(j>1){
        for(i in 1:(j-1)){
          r[i,j] <- sum(q[,i]*ifelse(MGS,v[,j],A[,j]))#对MGS取v, CGS
              取a
          v[,j] <- v[,j]-r[i,j]*q[,i]
        }}
        r[j,j] <- sqrt(sum(v[,j]^2))
        q[,j] <- v[,j]/r[j,j]
    }
  return(list(q,r))
}
```

□ Householder Reflection

Key idea: Reflect $A_{i:m,i}$ onto $e_1 \in \mathbb{C}^{m-i+1}$ as a vector of the same length $\|A_{i:m,i}\|e_1 \in \mathbb{C}^{m-i+1}$ (later we denote the l^{th} unit vector $e_l \in \mathbb{C}^{m-i+1} \equiv e_{m-i+1,l}$), reflector F_i in $\mathbb{C}^{(m-i+1)\times(m-i+1)}$ and auxiliary vector v_i:[2]

$$\mathbb{C}^{(m-i+1)\times(m-i+1)} \ni F_i = I_{m-i+1} - 2\frac{vv^*}{\|v\|_2^2} \qquad v = \text{sgn}(A_{i,i})\|A_{i:m,i}\|e_{m-i+1,1} + A_{i:m,i} \tag{5.65}$$

[2] Here sgn() for reflecting toward $-e_1/e_1$.

where $\text{sgn}(\cdot)$ corresponds to reflection onto $\hat{e}$ or $-\hat{e}$. Reflector on $A \in \mathbb{C}^{m\times n}$:

$$Q_i = \begin{bmatrix} I_{i-1} & 0 \\ 0 & F_i \end{bmatrix} \tag{5.66}$$

and QR calculated by (note that $F^2 = I_{m-i+1}$)

$$R = Q_n \dots Q_2 Q_1 A \qquad Q = Q_1 Q_2 \dots Q_n \tag{5.67}$$

Householder Reflection is more stable than Gram-Schimidt Orthogonalization

Error of Householder Reflection $A = \tilde{Q}\tilde{R} + E$, residual is controlled by $\|E\| \le \|A\| O(\varepsilon_{\text{machine}})$

Mainly caused by stability and accuracy of orthogonal matrix $\tilde{Q}$.

▷ `R`. **Code**

`R`. uses Householder Reflection to conduct QR decomposition.

```
A.qr <- qr(A)
Q <- qr.Q(A.qr)
R <- qr.R(A.qr)
```

□ **Givens Rotation**

Key idea: use rotation

$$Rx = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} \cos\theta = \dfrac{x_i}{\sqrt{x_i^2 + x_j^2}} \\ \sin\theta = \dfrac{x_j}{\sqrt{x_i^2 + x_j^2}} \end{cases} \tag{5.68}$$

act on $A_{i-1:i,j:n}$ so that $A_{i,j} = 0$, each time use two rows to create 1 zero. Slow, used for special sparse matrix.

5.2.5 Eigenvalue Decomposition

For square matrix $A \in \mathbb{C}^{m\times m}$, its eigenvector is the vector x_i whose direction (subspace) is invariant under transform operator A.

$$Ax_i = \lambda_i x_i \tag{5.69}$$

Properties:

- Determinant and trace of A:

$$\det(A) = \prod_{i=1}^{m} \lambda_i \qquad tr(A) = \sum_{i=1}^{m} \lambda_i \tag{5.70}$$

- x_i for special kinds of A: if $\text{span}\{q_i\} = \mathbb{C}^m$, then (generally X is **not** orthogonal)

$$AX = X\Lambda \Rightarrow A = X\Lambda X^{-1} \tag{5.71}$$

Further for $AA^* = A^*A$ (Normal Matrix 规范矩阵. Includes: hermitian $A = A^*$, skew hermitian $A = -A^*$, unitary $A^{-1} = A^*$, circulant matrix[3], and such $A + kI$), orthonormality of $x_i \to q_i$:

$$\langle q_i, q_j \rangle = \delta_{ij} \tag{5.73}$$

Eigenvalue Decomposition/Spectrum Decomposition, $X \to Q$:

$$AQ = Q\Lambda \Rightarrow A = Q\Lambda Q^{-1} = Q\Lambda Q^* \tag{5.74}$$

- Eigenvalue decomposition and positive definite matrix (Gershgorin circle thm.), λ_i falls in neighbourhood of a_{ii}:

$$D(\lambda_i, a_{ii}) < \sum_{j=1, j\neq i}^{m} |a_{ij}| \tag{5.75}$$

- Rayleigh quotient:

$$\max R(A, q) \equiv \max \frac{q^* A q}{q^* q} = \lambda_1 \tag{5.76}$$

Eigenvactor Algorithm: Power method to find leading eigen pair.

for independent eigenvectors x_i and an arbitrary vector $\xi = \sum_{i=1}^{m} c_i x_i$:

$$A^k \xi = A^k \sum_{i=1}^{m} c_i x_i = \sum_{i=1}^{m} c_i \lambda_i^k x_i = c_1 \lambda_1^k \left[x_1 + \sum_{i=2}^{n} \frac{c_i}{c_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i \right] \to c_1 \lambda_1^k x_1 \tag{5.77}$$

Algorithm *Basic Eigen Decomposition*

1. pick a random q_0
2. compute normalized $\dfrac{Aq_i}{\|Aq_i\|} = q_{i+1}$
3. repeat until $\|q_{i-1} - q_{i-2}\| < \varepsilon_{\text{preset}}$
4. q_i as the eigenvector, $q_i^T A q_i \approx q_i^T \lambda_1 q_i = \lambda_1$

This algorithm requires $|\lambda_1| > |\lambda_2| \geq \dots$ for quick convergence.

[3]Circulant Matrix, or similarly Latin Square.

$$C = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ c_3 & c_0 & c_1 & c_2 \\ c_2 & c_3 & c_0 & c_1 \\ c_1 & c_2 & c_3 & c_0 \end{bmatrix} \tag{5.72}$$

See section 14.2.2 ~ page 369 for application details.

5.2.6 SVD Decomposition

□ **SVD (Singular Value Decomposition) Form:**

- Reduced Form:

$$\underset{m\times n}{A} = \underset{m\times n}{U}\ \underset{n\times n}{\Sigma}\ \underset{n\times n}{V^*} \tag{5.78}$$

$$A = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} = U\Sigma V^* = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_n^* \end{bmatrix} \tag{5.79}$$

- Full Form:

$$\underset{m\times n}{A} = \underset{m\times m}{U}\ \underset{m\times n}{\Sigma}\ \underset{n\times n}{V^*} \tag{5.80}$$

$$A = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} = U\Sigma V^* = \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_n^* \end{bmatrix} \tag{5.81}$$

Existence and uniqueness of SVD:

- Every $A \in \mathbb{C}^{m\times n}$ has SVD with $\{\sigma_i\}$ unique

$$A = U\Sigma V^* = \sum_{i=1}^{n} \sigma_i u_i v_i^* \tag{5.82}$$

- if A is squared, then U, V determined
- if $A \in \mathbb{R}^{m\times n}$, then $U, V \in \mathbb{R}$

□ **SVD Expression**

U, V are eigenvectors of AA^*, A^*A respectively

$$A^*A = V\Sigma^2 V^* \qquad AA^* = U\Sigma^2 U^* \qquad u_j = \frac{Av_j}{\sigma_j} \quad \sigma_j = \sqrt{\lambda_{A^*A}} = \sqrt{\lambda_{AA^*}} \tag{5.83}$$

□ **Properties of SVD:**

- rank of A: $r = \text{rk}(\underset{m\times n}{A}) = \#\text{ non-zero } \sigma_i$

- Space of A:

$$\mathcal{R}(A) = \text{span}\{u_1, \ldots, u_r\} \qquad \mathcal{C}(A) = \text{span}\{v_1, \ldots, v_r\} \qquad \mathcal{N}(A) = \text{span}\{v_{r+1}, \ldots, v_n\} \tag{5.84}$$

- Norm:
  - Euclidean Norm: $\|A\|_2 = \sigma_1$
  - Frobenius Norm: $\|A\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$
  - Nuclear Norm: $\|A\|_* = \sum_{i=1}^{r} \sigma_i$
- Square matrix:
  - if $A = A^*$, then $\sigma_j = |\lambda_A|_j$
  - $\det(A) = \prod_{i=1}^{m} \sigma_i$
- Low-rank Approximation of A using SVD:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^* = A - \sum_{j=k+1}^{r} \sigma_j u_j v_j^* \tag{5.85}$$

  is the 'nearest' rank k matrix from A

$$\min_{\text{rk}(\Xi)=k} \|A - \Xi\|_2 = \|A - A_k\|_2 = \sigma_{k+1} \tag{5.86}$$

□ **When A is positive definite, SVD and ED get the same result.**

$$A = Q\Lambda Q^* \Rightarrow A = Q\text{sgn}(\Lambda)|\Lambda|Q^* = U\Sigma Q^* = U\Sigma V^* \tag{5.87}$$

5.2.7 Schur Decomposition

Unitary Triangularization of matrix A (always exists in $\mathbb{C}^{m\times m}$):

$$A = QTQ^*,\ Q \text{ unitary},\ T \text{ upper-triangular} \tag{5.88}$$

for $A \in \mathbb{R}^{m\times m}$: T is quasi-triangular, diag of T is $\text{Re}(\lambda_i)$

Section 5.3 Numeric Optimization Algorithm I

Algorithm Optimization in Statistics: e.g.

- MLE Maximazation, e.g. section 5.4.3 ~ page 170.
- Clustering: minimizing within-cluster distance & maximizing between-cluster distance, see section 4.7 ~ page 138
- Box-Cox λ determining, see section 3.5.1 ~ page 102.
- Machine Learning Model training, minimizing loss function, e.g. section 9.4.5 ~ page 258.

□ **Duality of Optimization and Rooting:**

- Optimization: e.g. minimizing function $g(x)$:

$$\arg\min g(x) \leftrightharpoons \arg\{\nabla g(x) = 0\} \tag{5.89}$$

- Rooting: extract root $f(x) = 0$:

$$\arg\{f(x) = 0\} \leftrightharpoons \arg\min f(x)^T f(x) \tag{5.90}$$

More specific example: expand function to 2nd as $g(x) \approx \frac{1}{2}x^T Ax - bx + c$ (differnetiation of quadric $x^T Ax$ see section 4.1.2 ~ page 118)

$$\arg\min \frac{1}{2}x^T Ax - bx + c \leftrightharpoons \arg\{\frac{A + A^T}{2}x = b\} \tag{5.91}$$

Δ i.e. for optmizing task $\arg\min g(x)$, we can either minimizing $g(x)$, or rooting $f(x) \equiv \nabla g(x)$

□ **Algorithm Design Aiming:**

- Robustness: can be applied on various problems
- Accuracy: reach solution with great precision, at the same time insensitive to machine error
- Efficiency: computer time/storage not required

□ **Iteration in Optimization Problem**

Usually iteration is used in optmizing problem, by approximate solution x^* step by step.

- Bracketing method means the solution x^* is always within some iteration interval $I^{(t)} = [x_{\text{left}}, x^*, x_{\text{right}}]$, use convergence condition $m(I^{(t)}) < \varepsilon$ to obtain solution.
- Open method: Not necessarily $x^* \in I^{(t)}$, but convergence using $d(x^{(t)}, x^{(t-1)}) < \varepsilon$. Usually faster than bracketing, but less stable, and sensitive to initial value.
- Hybird Method: Mixture of bracketing and open according to iteration step feature

□ **Content**

- Golden Section & Fibonacci Section Search: Bracketing method direct search for minimizer;
- Bisection Search: Bracketing method direct search for root
- Interpolation Method: Include either bracketing/open method, approximate function to obtain root/minimizer
  - Regula Falsi: Bracketing linear interpolation for rooting
  - Secant Interpolation: Open linear interpolation for rooting
  - Parabolic Interpolation: Open parabolic interpolation for minimizing
  - Inverse Parabolic Interpolation (IQI): Open interpoation for rooting

- Hybrid Method: Combination of bracketing method and open interpolation method for rooting. Include Dekker's and Brent's, most used method
  - Dekker's Method: Hybrid of bisection and secant interpolation method for rooting
  - ⋆ Brent's Method: Hybrid of bisection, secant interpolation and IQI for rooting
- Fixed Point Iteration Method: Open method for rooting, including univariate and multivariate linear case.
  - Univariate Fixed Point Iteration
  - Jacobi Method
  - Gauss Seidel Method
  - Successive Over-Relaxation Method
- ⋆ Nelder-Mead Method/Simplex Method: Open method for minimizer based on simplex iteration

□ **Default Methods in R.**

- `optim(VEC_OF_INI_VAR,FUN)`: Nelder-Mead Simplex search method, use `method=c('Nelder-Mead','BFGS','L-BFGS-B','CG','SANN','Brent')` to choose different methods
- `uniroot(FUN,INTERVAL)`: Brent's Method;
- `optimize(FUN,INTERVAL)`: Golden Section+Parabolic Interpolation.

5.3.1 Golden Section/Fibonacci Section Search

Problem: minimizing univariate function $g(x)$, within a pre-estimated interval $[x_1^{(0)}, x_4^{(0)}]$. For f that is undiffernetiable/complicated to compute, this method is often used.

Basic idea: within a unimodal interval $I^{(0)} = [x_1^{(0)}, x_4^{(0)}]$ of $f(x)$, pick two symmetric points $x_2^{(0)}, x_3^{(0)}$ in I_0 so that

$$x_2^{(0)} - x_1^{(1)} = x_4^{(0)} - x_3^{(0)} = (1 - r^{(0)})(x_4^{(0)} - x_1^{(0)}) \quad r^{(t)} > 1/2 \tag{5.92}$$

then extreme point should falls in one of $[x_1^{(t)}, x_3^{(t)}]$ or $[x_2^{(t)}, x_4^{(t)}]$, iteration the interval by comparing $g(x_2)$ and $g(x_3)$: use one of them as the next interval. And for less computation, we hope that one of $g(x_2^{(t)})$ of $g(x_3^{(t)})$ can be used in step $t+1$ as $g(x_3^{(t+1)})$ or $g(x_2^{(t+1)})$, i.e.

$$\text{if } g(x_2^{(t)}) > g(x_3^{(t)}) : [x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t+1)}] := [x_2^{(t)}, x_3^{(t)}, x_4^{(t)}] \tag{5.93}$$

$$\text{if } g(x_2^{(t)}) \le g(x_3^{(t)}) : [x_1^{(t+1)}, x_3^{(t+1)}, x_3^{(t+1)}] := [x_1^{(t)}, x_2^{(t)}, x_3^{(t)}] \tag{5.94}$$

also use equation 5.92 ~ page 160, we have (here use $g(x_2^{(t)}) > g(x_3^{(t)})$ case for derivation)

$$r^{(t+1)} = \frac{x_4^{(t)} - x_2^{(t)}}{x_4^{(t)} - x_1^{(t)}} = \frac{x_4^{(t+1)} - x_2^{(t+1)}}{x_4^{(t+1)} - x_1^{(t+1)}} = \frac{x_4^{(t)} - x_3^{(t)}}{x_4^{(t)} - x_2^{(t)}} = \frac{1 - r^{(t)}}{r^{(t)}} \tag{5.95}$$

Algorithm *Golden Section/Fibonacci Section Search*

1. Initialize $I^{(0)} = [x_1^{(0)}, x_4^{(0)}]$ with $x^* \in I^0$
2. For each step $x^{(t)}$:
   (a) Calculate $r^{(t)}$, and then $g(x_2^{(t)})$, $g(x_3^{(t)})$
   (b) compare $g(x_2^{(t)})$ and $g(x_3^{(t)})$, and update interval

$$\text{if } g(x_2^{(t)}) > g(x_3^{(t)}) : [x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t+1)}] \equiv [x_2^{(t)}, x_3^{(t)}, x_4^{(t)}] \tag{5.96}$$

$$\text{if } g(x_2^{(t)}) \leq g(x_3^{(t)}) : [x_1^{(t+1)}, x_3^{(t+1)}, x_4^{(t+1)}] \equiv [x_1^{(t)}, x_2^{(t)}, x_3^{(t)}] \tag{5.97}$$

3. Repeat until convergence $m(I^{(t)}) < \varepsilon$

Choice of $r^{(t)}$: for algorithm robustness and avoid ill sequence, we will usually use some special $r^{(t)}$:

- Golden Section Search: use $r^{(t)} = r = \text{const}$, such r should satisfies

$$r = \frac{1-r}{r} \Rightarrow r = \frac{\sqrt{5}-1}{2} = \frac{1}{\phi} \approx 0.618 \tag{5.98}$$

Convergence at

$$m(I^{(t)}) = r^t m(I^{(0)}) < \varepsilon \tag{5.99}$$

- Fibonacci Section Search: choose for $t = 0$ as $r^{(0)} = \dfrac{F_{n-1}}{F_n}$, where $\{F_n\}$ is Fibonacci sequence, then

$$r^{(0)} = \frac{F_{n-1}}{F_n} \tag{5.100}$$

$$r^{(1)} = \frac{1-r^{(0)}}{r^{(0)}} = \frac{F_{n-2}}{F_{n-1}} \tag{5.101}$$

$$r^{(2)} = \frac{1-r^{(1)}}{r^{(1)}} = \frac{F_{n-3}}{F_{n-2}} \tag{5.102}$$

$$\vdots \tag{5.103}$$

$$r^{(t)} = \frac{F_{n-t-1}}{F_{n-t}} \tag{5.104}$$

$$\vdots \tag{5.105}$$

$$r^{(n-3)} = \frac{F_2}{F_3} = \frac{1}{2} \text{ (the last step of iteration)} \tag{5.106}$$

To determine the preset n, first use convergence condition

$$m(I^{(n-2)}) = \prod_{i=0}^{n-3} r^{(i)} m(I^{(0)}) = \frac{F_2}{F_n} m(I^{(0)}) < \varepsilon \Rightarrow \begin{cases} F_n > \dfrac{m(I^{(0)})}{\varepsilon} \\ F_{n-1} < \dfrac{m(I^{(0)})}{\varepsilon} \end{cases} \tag{5.107}$$

then conduct iteration, using $r^{(t)} = \dfrac{F_{n-t-1}}{F_{n-t}}$.

Basically the two methods have similar background, noticing that the eigen equation of Fibonacci sequence is $x^2 = x + 1$, and $\lim_{n\to\infty} \frac{F_{n-1}}{F_n} = \frac{\sqrt{5}-1}{2} = \frac{1}{\phi}$[4]

Can be proven: Golden section need one more iteration call than Fibonacci section:

$$n_{\text{GS}} = n_{\text{Fib}} + 1 \tag{5.109}$$

Convergence order $\alpha = 1$, rate $c = \frac{1}{\phi}$

5.3.2 Bisection Search Method

Problem: rooting univariate function $f(x)$, with a pre-estimated interval $I^{(0)} = [x_1^{(0)}, x_2^{(0)}]$, with $f(x_1^{(0)})f(x_2^{(0)}) < 0$

Idea: Intermediate value thm.: for continuous $f : [a, b] \to \mathbb{R}$, $f(a)f(b) < 0 \Rightarrow \exists x^*, s.t. f(x^*) = 0$.

Algorithm *Bisection Search*

1. Initialize $I^{(0)} = [x_1^{(0)}, x_2^{(0)}]$ satisfying $f(x_1^{(0)})f(x_2^{(0)}) < 0$
2. In each iteration $x^{(t)}$:
   (a) compute midpoint function value
   $$x_m^{(t)} = \frac{1}{2}\left(x_1^{(t)} + x_2^{(t)}\right) \tag{5.110}$$
   (b) update interval according sign of $f(x_m^{(t)})$:
   $$I^{(t+1)} = [x_1^{(t+1)}, x_2^{(t+1)}] := \begin{cases} [x_1^{(t)}, x_m^{(t)}], & f(x_1^{(t)})f(x_m^{(t)}) < 0 \\ [x_m^{(t)}, x_2^{(t)}], & f(x_m^{(t)})f(x_2^{(t)}) < 0 \end{cases} \tag{5.111}$$
3. Repeat until convergence $m(I^{(t)}) < \varepsilon$

Convergence order $\alpha = 1$, rate $c = \frac{1}{2}$.

5.3.3 Interpolation Methods: Linear/Quadratic/Lagrange Interpolation

Interpolation is an approximation to function, thus can get approximation to solution. Interpolation can be used for both minimizing or root finding.

□ **Regula Falsi/Linear Interpolation (Bracketing) for Root Finding:**

[4] General Formula of Fibonacci sequence:

$$F_n = \frac{1}{\sqrt{5}}\left((\phi)^n - (-\frac{1}{\phi})^n\right) \tag{5.108}$$

Idea of regula falsi linear interpolation: at root x^* of $f(x)$:

$$f(x) \approx \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x^*} (x - x^*) \tag{5.112}$$

Iterate by repeatedly constructing linear interpolation/secant and use the root as an approximation to x^*.

Algorithm *Regula Falsi Interpolation*

1. Initialize interval $I^{(0)} = [x_1^{(0)}, x_2^{(0)}]$ with $f(x_1^{(0)})f(x_2^{(0)}) < 0$
2. In each iteration $x^{(t)}$:
   (a) Compute linear interpolation of $(x_1^{(t)}, f(x_1^{(t)}))$, $(x_2^{(t)}, f(x_2^{(t)}))$, and compute the root of the straight line
   $$x_r^{(t)} = \frac{x_1^{(t)} f(x_2^{(t)}) - x_2^{(t)} f(x_1^{(t)})}{f(x_2^{(t)}) - f(x_1^{(t)})} \tag{5.113}$$
   compute $f(x_r^{(t)})$
   (b) update interval according to sign of $f(x_r^{(t)})$:
   $$I^{(t+1)} = [x_1^{(t+1)}, x_2^{(t+1)}] := \begin{cases} [x_1^{(t)}, x_r^{(t)}], & f(x_1^{(t)})f(x_r^{(t)}) < 0 \\ [x_r^{(t)}, x_2^{(t)}], & f(x_r^{(t)})f(x_2^{(t)}) < 0 \end{cases} \tag{5.114}$$
3. Repeat until convergence $m(I^{(t)}) < \varepsilon$

Note: for enough steps of iteration t_ξ, the iteration would be short enough such that $\operatorname{sgn}(f''(x)) = \text{const}$, in which case one of $x_1^{(t)}$ or $x_2^{(t)}$ would remain fixed for $t > t_\xi$.[5]

Convergence order $\alpha = 1$, rate $c = -\dfrac{f''(x^*)}{2f'(x^*)}(x^* - x_{\text{fixed}})$. Note that sign dependency of x_{fixed} on $f''(x)$ and $f'(x)$ ensures $c > 0$.

□ **Secant Interpolation/Linear Interpolation (Open) for Root Finding**

Instead of limiting $x^* \in [x_1^{(t)}, x_2^{(t)}]$ (bracketing) by ensuring $f(x_1^{(t)})f(x_2^{(t)}) < 0$, we can also remove the restrict, i.e. just use the latest two points to construct secant.

Algorithm *Secant Interpolation*

1. Initialize two points $x^{(-1)}$, $x^{(0)}$.(interval not necessarily include potential root $\hat{x}^*$)
2. In each iteration $x^{(t)}$:
   (a) Compute linear interpolation of $(x^{(t-1)}, f(x^{(t-1)}))$, $(x^{(t)}, f(x^{(t)}))$
   (b) Use the root to update $x^{(t+1)}$:
   $$x^{(t+1)} = \frac{x^{(t-1)} f(x^{(t)}) - x^{(t)} f(x^{(t-1)})}{f(x^{(t)}) - f(x^{(t-1)})} \tag{5.115}$$

[5] For $f''(x)f'(x) > 0$, x_2 fixed; $f''(x)f'(x) < 0$, x_1 fixed.

3. Repeat until convergence, .e.g. $|x^{(t)} - x^{(t-1)}| < \varepsilon$

Comment: For interval small enough such that $\operatorname{sgn}(f''(x)) = \text{const}$ and $f(x^{(t)})f(x^{(t-1)}) < 0$, this method might goes back to bracketing linear interpolation.

Convergence order $\alpha \approx 1.618$.

□ **Parabolic Interpolation for Minimizing**

Idea of parabolic interpolation: at extreme point x^*, function f has taylor series

$$g(x) \approx g(x^*) + \frac{1}{2}\left.\frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\right|_{x^*}(x - x^*)^2 \tag{5.116}$$

we can iteration by repeatedly construct parabola to approximate $f(x^*) + \frac{1}{2}\left.\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\right|_{x^*}(x - x^*)^2$ and use the exterme point of the parabola.

Algorithm *Parabolic Interpolation*

---

1. First initialize three point $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)})$,
2. In each iteration $x^{(t)}$:
   (a) Use $(x_1^{(t)}, x_2^{(t)}, x_3^{(t)})$ to compute corresponding $f(x)$, then use quadric fitting to obtain parabola $\Gamma^{(t)}$
   (b) Replace $\max\{x_1^{(t)}, x_2^{(t)}, x_3^{(t)}\}$ by extreme point of $\Gamma^{(t)}$ to update as $(x_1^{(t+1)}, x_2^{(t+1)}, x_3^{(t+1)})$
3. Repeat until convergence.

Convergence order $\alpha \approx 1.3247$.

□ **Lagrange Polynomial Interpolation**

Lagrange Polynomial is a function base set: Given $n+1$ point $(x_0, y_0), \ldots, (x_n, y_n)$ $(n \geq 1)$, Lagrange polynomial:

$$\ell_i = \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n \tag{5.117}$$

And Lagrange interpolation function: $L(x) = \sum_{i=0}^{n} y_i \ell_i$

$n = 1$ for linear interpolation, $n = 2$ for parabolic interpolation.

□ **Inverse Parabolic Interpolation (IQI): Open interpoation for rooting**

Note that general parabola $y = \frac{1}{2}ax^2 + bx + c$ might have 0 or 2 root simultaneously, thus use inverse quadric function $x = \frac{1}{2}ay^2 + by + c$, i.e. inverse quadric interpolation.

Algorithm *Inverse Parabolic Interpolation*

---

1. First initialize three point $C^{(0)} = \left(x^{(-2)}, x^{(-1)}, x^{(0)}\right)$
2. In each iteration $x^{(t)}$:

(a) Use $C^{(t)} = \left(x^{(t-2)}, x_2^{(t-1)}, x_3^{(t)}\right)$ to compute IQI function, and get root

$$s = \sum_{\text{cycle } x^{(t-2)}, x^{(t-1)}, x^{(t)}} \frac{x^{(t-2)} f(x^{(t-1)}) f(x^{(t)})}{\left[f(x^{(t-2)}) - f(x^{(t-1)})\right]\left[f(x^{(t-2)}) - f(x^{(t)})\right]} \tag{5.118}$$

(b) Update points

$$C^{(t+1)} = \left(x^{(t-1)}, x^{(t)}, x^{(t+1)}\right) = \left(x^{(t-1)}, x^{(t)}, s\right) \tag{5.119}$$

3. Repeat until convergence $|x^{(t)} - x^{(t-1)}| < \varepsilon$

5.3.4 Hybrid Method: Dekker's/Brent's

□ **Dekker's Method**

Dekker's method is a hybrid of open linear interpolation and bisection, in each step, use one of interpolation/bisection according to iteration condition to achieve both quick convergence and stability.

Algorithm *Dekker's Method*

1. Initialize three point $a^{(0)}$, $b^{(0)}$, $b^{(-1)} = a^{(0)}$, where interval bteween $a^{(0)}$, $b^{(0)}$ should include potential root $\hat{x}^*$, i.e. $f(a^{(0)}) f(b^{(0)}) < 0$

2. In each iteration $x^{(t)}$:

   (a) $a^{(t)}$, $b^{(t)}$ is labelled as follows: label ensure $|f(a^{(t)})| \geq |f(b^{(t)})|$, thus $b^{(t)}$ is the estimate of root, while $a^{(t)}$ is the 'contrapoint' of $b^{(t)}$

   (b) compute root s of linear interpolation of $(a^{(t)}, f(a^{(t)}))$, $(b^{(t)}, f(b^{(t)}))$, and compare with midpoint $m = \dfrac{a^{(t)} + b^{(t)}}{2}$

$$\tilde{b}^{(t+1)} = \begin{cases} s = \dfrac{a^{(t)} f(b^{(t)}) - b^{(t)} f(a^{(t)})}{f(b^{(t)}) - f(a^{(t)})}, & s \in [m, b^{(t)}] \text{ (or } [b^{(t)}, m]) \\ m = \dfrac{a^{(t)} + b^{(t)}}{2}, & s \notin [m, b^{(t)}] \text{ (or } [b^{(t)}, m]) \end{cases} \tag{5.120}$$

   (c) Then update $\tilde{a}^{(t+1)}$ as one of $a^{(t)}$ and $b^{(t)}$, such that $f(\tilde{a}^{(t+1)}) f(\tilde{b}^{(t)}) < 0$, then relabel $\tilde{a}^{(t+1)}$, $\tilde{b}^{(t)}$ to $a^{(t+1)}$, $b^{(t+1)}$ according to $|f(a^{(t+1)})| > |f(b^{(t+1)})|$

3. Repeat until convergence $|b^{(t)} - b^{(t-1)}| < \varepsilon$

Comment: In step 3, the choice between bisection and open interpolation take advantage of quick convergence of open method, also ensure stability by using bisection for ill secant root s. However for interval small enough, this method might also goes back to bracketing linear interpolation, then $b^{(t)}$ convergence very slow.

□ **Brent's Method**

Brent's Method is an improvement of Dekker's Method:

- Avoid convergence problem of $b^{(t)}$ in the case of bracketing linear interpolation by checking $|b^{(t)}-b^{(t-1)}|>\delta$ before linear interpolation, otherwise use bisection
- Further adding IQI interpolation if $a^{(t)}$, $b^{(t)}$, $b^{(t-1)}$ are distinct for quicker convergence, root for IQI:

$$s'=\sum_{\text{cycle } a^{(t)},b^{(t)},b^{(t-1)}}\frac{a^{(t)}f(b^{(t)})f(b^{(t-1)})}{\left[f(a^{(t)})-f(b^{(t)})\right]\left[f(a^{(t)})-f(b^{(t-1)})\right]} \tag{5.121}$$

▷ R. **Code**

```
uniroot()
```

5.3.5 Fixed Point Iteration: Univariate

Idea: Contraction mapping thm.: for function $f:X\to X$ satisfying

$$d(f(x),f(y))\le\beta d(x,y),\ \beta<1 \tag{5.122}$$

then such f has a unique fixed point x^* such that $f(x^*)=x^*$, and convergence is ensured:

$$d(f^{\{n\}}(x),x^*)\le\frac{\beta^n}{1-\beta}d(f(x),x^*) \tag{5.123}$$

For univariate function, requires $|f'(x)|<1$(at least at x near x^*)

To minimize $f(x)$, i.e. find root of $f'(x)=g(x)$, i.e. find fixed point of $G(x)\equiv\alpha f'(x)+x=x$, requires $|G'(x)|=|\alpha f''(x)+1|<1$.

Note: We can also use inverse function of $\alpha f'(x)+x$, and further use $G_1(x)=rG(x)+(1-r)x$ to find fixed point.

Iteration: use $\hat{x}^*=x^{(n)}=G^{\{n\}}(x)=\underbrace{G(G(G(\dots G(G}_{n}(x))\dots)))$, until $|x^{(n)}-x^{(n-1)}|<\varepsilon$

Basically, fixed point iteration is the same as parallel chord method: use the root of $y-g(x^{(t)})=-\frac{1}{\alpha}(x-x^{(t)})$ as $x^{(t+1)}$.

Convergence order is α in $G(x)=\alpha f'(x)_x$

5.3.6 Fixed Point Iteration: Multivariate Linear

For solution of $Ax=b$ using fixed point iteration, where $AA^*=A^*A$(normal matrix), requires:

$$\rho(A)=\max|\lambda|<1 \tag{5.124}$$

- Jacobi Method: Decompose $A=D+E$, where D is diagonal part

$$\underset{n\times n}{A}=\begin{bmatrix}a_{11}&a_{12}&\dots&a_{1n}\\a_{21}&a_{22}&\dots&a_{2n}\\\vdots&\vdots&\ddots&\vdots\\a_{n1}&a_{n2}&\dots&a_{nn}\end{bmatrix}=D+E=\begin{bmatrix}a_{11}&0&\dots&0\\0&a_{22}&\dots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\dots&a_{nn}\end{bmatrix}+\begin{bmatrix}0&a_{12}&\dots&a_{1n}\\a_{21}&0&\dots&a_{2n}\\\vdots&\vdots&\ddots&\vdots\\a_{n1}&a_{n2}&\dots&0\end{bmatrix} \tag{5.125}$$

Then fixed point iteration: using $(D+E)x = b \Rightarrow x^{(t+1)} = D^{-1}(b - Ex^{(t)})$

$$x_i^{(t+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(t)} \right) \tag{5.126}$$

- Gauss-Seidel Method: Decompose $A = L + U$

$$\underset{n \times n}{A} = L + U = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{5.127}$$

Then fixed point iteration: using $(L+U)x = b \Rightarrow Lx^{(t+1)} = (b - Ux^{(t)})$, iteration:

$$x_i^{(t+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(t+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(t)} \right) \tag{5.128}$$

- Successive Over-Relaxation Method (SOR Method): Decompose $A = D + L + U$

$$\underset{n \times n}{A} = D + L + U = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{5.129}$$

Then fixed point iteration: using $\omega(D+L+U)x = \omega b \Rightarrow (D+\omega L)x = \omega b - [\omega U + (\omega - 1)D])x$, move non-diagonal elements to R.H.S.

$$x_i^{(t+1)} = (1-\omega)x_i^{(t)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(t+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(t)} \right), \qquad \omega \in (0, 2) \tag{5.130}$$

Comment: SOR iteration step is the ω weighted average of $x^{(t)}$ and Gauss-Seidel iteration.

5.3.7 Nelder-Mead Method

For multivariate function $g(x)$, with $x \in \mathbb{R}^n$, usually use Nelder-Mead Method, or Simplex Search Method. Simplex is a generalization of triangle/tetrahedron to any arbitrary dimension, and Nelder-Mead method is conducted by iterating simplex.

Algorithm *Nelder-Mead Method*

1. First initialize simplex $C^{(0)}$ by preset p_0 and $\vec{\lambda}$:

$$C^{(0)} = \{x_0^{(0)}, x_1^{(0)}, \dots, x_n^{(0)}\}, x_0^{(0)} = p_0,\ x_i^{(0)} = p_0 + \lambda_i \hat{e}_i \tag{5.131}$$

2. In each iteration $x^{(t)}$:

(a) First sort $\{x_i^{(t)}\}$ according to $g(x_i^{(t)})$ as

$$g(x_{(0)}^{(t)}) \leq g(x_{(1)}^{(t)}) \leq \ldots \leq g(x_{(n)}^{(t)}) \tag{5.132}$$

(b) Compute centroid of $C^{(t)}_{x_{(n)}^{(t)}} = \{x_{(0)}^{(t)}, x_{(1)}^{(t)}, \ldots, x_{(n-1)}^{(t)}\}$

$$x_g^{(t)} = \frac{1}{n}\sum_{i=0}^{n-1} x_{(i)}^{(t)} \tag{5.133}$$

And compute the reflection point of $x_{(n)}^{(t)}$:

$$x_r^{(t)} := x_g^{(t)} + (x_g^{(t)} - x_{(n)}^{(t)}) \tag{5.134}$$

(c) Compute $g_{(0)}^{(t)} = g(x_{(0)}^{(t)})$, $g_{(n-1)}^{(t)} = g(x_{(n-1)}^{(t)})$, $g_{(n)}^{(t)} = g(x_{(n)}^{(t)})$, $g_r^{(t)} = g(x_r^{(t)})$ and compare:

- $g_r^{(t)} < g_{(0)}^{(t)}$: reflection point $x_r^{(t)}$ is a good trial for minimizing, further try a farther point

$$x_{2r}^{(t)} := x_g^{(t)} + 2(x_g^{(t)} - x_{(n)}^{(t)}) \tag{5.135}$$

then iteration according to $g_{2r}^{(t)}$:

$$C^{(t+1)} = \{x_0^{(t+1)}, x_1^{(t+1)}, \ldots, x_n^{(t+1)}\} \equiv \begin{cases} \{x_{(0)}^{(t)}, x_{(1)}^{(t)}, \ldots, x_{(n-1)}^{(t)}, x_{2r}^{(t)}\}, & g_{2r}^{(t)} < g_r^{(t)} \\ \{x_{(0)}^{(t)}, x_{(1)}^{(t)}, \ldots, x_{(n-1)}^{(t)}, x_r^{(t)}\}, & g_{2r}^{(t)} \geq g_r^{(t)} \end{cases} \tag{5.136}$$

- $g_{(0)}^{(t)} \leq g_r < g_{(n-1)}^{(t)}$: better simplex but not necessarily the best, just use

$$C^{(t+1)} = \{x_0^{(t+1)}, x_1^{(t+1)}, \ldots, x_n^{(t+1)}\} \equiv \{x_{(0)}^{(t)}, x_{(1)}^{(t)}, \ldots, x_{(n-1)}^{(t)}, x_r^{(t)}\} \tag{5.137}$$

- $g_{(n-1)}^{(t)} \leq g_r^{(t)}$: $x_r^{(t)}$ might not optmize the simplex, conduct shrinkage:

$$x_s^{(t)} := \begin{cases} x_g^{(t)} + 0.5(x_g^{(t)} - x_{(n)}^{(t)}), & g_{(n-1)}^{(t)} \leq g_r^{(t)} < g_{(n)}^{(t)} \\ x_g^{(t)} - 0.5(x_g^{(t)} - x_{(n)}^{(t)}), & g_{(n)}^{(t)} \leq g_r^{(t)} \end{cases} \tag{5.138}$$

if $g_s^{(t)} \leq g_{(n)}^{(t)}$, suggesting a successful shrinkage, use $g_s^{(t)}$ for iteration

$$C^{(t+1)} = \{x_0^{(t+1)}, x_1^{(t+1)}, \ldots, x_n^{(t+1)}\} \equiv \{x_{(0)}^{(t)}, x_{(1)}^{(t)}, \ldots, x_{(n-1)}^{(t)}, x_s^{(t)}\},\ g_s^{(t)} \leq g_{(n)}^{(t)} \tag{5.139}$$

otherwise we have to update the whole simplex:

$$x_0^{(t+1)} = x_0^{(t)},\ x_i^{(t+1)} = x_0^{(t)} + \frac{1}{2}(x_i^{(t)} - x_0^{(t)}) \tag{5.140}$$

5.3.8 Coordinate Descent Method*

Section 5.4 Numeric Optimization Algorithm II

To minimize some arbitrary function $f(x)$, the idea of gradient iteration method is to update $x^{(t)}$ based on (minus) gradient $-\nabla f(x)$, with some modification on direction $p^{(t)} = T\left(-\nabla f(x)\right)$ and step length $\alpha^{(t)}$

$$x^{(t+1)} = x^{(t)} + \alpha^{(t)} T\left(-\nabla f(x^{(t)})\right) = x^{(t)} + \alpha^{(t)} p^{(t)} \tag{5.141}$$

- Modifying Direction $p^{(t)}$:
  - Gradient Descent: $p^{(t)} = -\nabla f\left(x^{(t)}\right)$
  - Newton-Raphson Method: use Hessian matrix $p^{(t)} = -\left[H(x^{(t)})\right]^{-1} \nabla f\left(x^{(t)}\right)$
  - Fisher Scoring Method: for statistics problem, use fisher information $I\left(x^{(t)}\right) = -E_Y\left(H(x^{(t)})\right)$, $p^{(t)} = I\left(x^{(t)}\right)^{-1} \nabla f\left(x^{(t)}\right)$
  - Quasi-Newton Method: usually use secant condition to approximate Hessian $\hat{H}^{(t)} = M^{(t)}$ or $\hat{H^{-1}}^{(t)} = B^{(t)}$, with various updating SR-1/DFP/BFGS/L-BFGS/Broyden Class
  - Steepest Descent: general form based on various norm choice.
  - Stochastic Gradient Descent (SGD): modification for large sample
  - Conjugate Gradient Method: Use the 'perpendicular' property of conjugate vector for quick updating of p_k
- Modifying Step-Length / Learning Rate $\alpha^{(t)}$:
  - Fixed step-length: $\alpha^{(t)} = \alpha$
  - Backtracking line search: $\alpha^{(t)} = \dfrac{\alpha}{2^{n^{(t)}}}$
  - Exact line search: $\alpha^{(t)} = \arg\min_{\alpha} f\left(x^{(t)} + \alpha p^{(t)}\right)$
  - Trust Region Method: use Hessian matrix $H(x^{(t)})$, but restrict direction & step-length with trust region $\|\alpha^{(t)} p^{(t)}\| \le \Delta^{(t)}$

5.4.1 Gradient Descent Method

The simplest choice for $T(\cdot)$ is identity $p^{(t)} = -\nabla f\left(x^{(t)}\right)$, because negative gradient direction is the (local) descent direction. Iteration:

$$x^{(t+1)} = x^{(t)} - \alpha^{(t)} \nabla f\left(x^{(t)}\right) \tag{5.142}$$

Note: for such gradient method, step-length should be carefully specified, use proper fixed step-length or backtracking/exact line search.

Convergence order $\alpha_{\text{conv}} = 1$.

5.4.2 Newton-Raphson Method

Idea: For minimizing problem $x^* = \arg\min f(x)$[6], using iteration method with an initial value $x^{(0)}$, we hope to find iteration step $x^{(t+1)} - x^{(t)}$ such that $x^{(t+1)}$ can approach x^* quickly. We can try to use the taylor series at $x^{(t)}$ to $O(x^2)$ and try the minimizer of the quadric function:

$$f(x) \approx \tilde{f}_{x^{(t)}}(x) = f(x^{(t)}) + (x - x^{(t)})^T \nabla f(x)|_{x^{(t)}} + \frac{1}{2}(x - x^{(t)})^T \nabla\nabla f(x)|_{x^{(t)}} (x - x^{(t)}) \tag{5.143}$$

minimizer $\dfrac{\partial \tilde{f}(x)}{\partial x} = 0$:

$$\frac{\partial \tilde{f}}{\partial x} = \nabla \tilde{f}(x)\Big|_{x^{(t)}} + \nabla\nabla \tilde{f}(x)\Big|_{x^{(t)}} (x - x^{(t)}) = 0 \Rightarrow x^{(t+1)} - x^{(t)} = \left(\nabla\nabla \tilde{f}(x)\right)^{-1} \nabla \tilde{f}(x)\Big|_{x^{(t)}} \tag{5.144}$$

Use the above solution as the iteration step:

$$x^{(t+1)} = x^{(t)} - \left[H^{(t)}\right]^{-1} \nabla f(x^{(t)}) \tag{5.145}$$

where $H^{(t)}$ is the Hessian matrix $H^{(t)} \equiv \left.\dfrac{\partial^2 f(x)}{\partial x \partial x^T}\right|_{x^{(t)}}$

Convergence order $\alpha_{\text{conv}} = 2$.

□ **Main difficulties of Newton-Raphson method:**

- Calculation of $H(f(x^{(t)}))^{-1}$, a task of second derivative + marix inverse.
- As an open method, Newton-Raphson method is unstable and sensitive to initial value: more initial trials suggested
- Positive/Negative Definition of Hessian $\dfrac{\partial f}{\partial x \partial x^T}$ is not guarenteed, while positive/negative definition would lead to local minimum/maximum respectively, i.e. descent not guarenteed.

5.4.3 Fisher's Scoring Method in MLE

For MLE optmizing problem in statistics using Newton-Raphson Method, we can use properties of log-likelihood $l(\theta; \vec{x})$ to help overcome the difficulty of calculating H^{-1}. This method is called Fisher's Scoring Method/Iteratively Re-weighted Least Squares (IRLS).

Notation: for simplication, the following part uses $\nabla f(x) := f'(x)$ (a vector), $\nabla\nabla f(x) := f''(x)$ (a matrix)

□ **MLE Maxmizing ⇔ minus of MLE Minimizing**

MLE maximizing problem:

$$\theta^* = \arg\max l(\theta; \vec{x}) = \arg\max \ln \prod_{x_i} f(x_i; \theta) \tag{5.146}$$

Newton-Raphson iteration gives

$$\theta^{(t+1)} = \theta^{(t)} - l''(\theta^{(t)}; x)^{-1} l'(\theta^{(t)}; x) \tag{5.147}$$

[6]Here uses different notation from previous part to avoid confusion of $g(x)$ as link function.

Note that here $l'(\theta)$ is Score Function (equation 2.78 ~ page 48), and $l''(\theta)$ is relative to Fisher Information (equation 2.89 ~ page 48).[7]

Note that Fisher Information is the **expectation** of $-(l''(\theta)) := J(\theta)$, the idea of Fisher scoring method is the estimate $l''(x)$ using Fisher information:

$$\theta^{(t+1)} = \theta^{(t)} - l''(\theta; x)^{-1} l'(\theta; x) \longrightarrow \theta^{(t+1)} = \theta^{(t)} + I(\theta^{(t)})^{-1} l'(\theta^{(t)}; x) \tag{5.150}$$

How does Fisher Scoring improve Newton-Raphson method?

- Note that $l(\theta; \vec{x}) = \sum_{i=1}^{n} l(\theta; x_i) \Rightarrow l''(\theta; \vec{x}) = \sum_{i=1}^{n} l''(\theta; x_i)$, need much more computation for large n, while Fisher Information is a reasonable 'average' of $l''(\theta, x_i)$ and total Information is just the sum of each I_i

$$I(\theta) = nI_1(\theta) = n\mathbb{E}_\xi(\frac{\partial^2 l(\theta; \xi)}{\partial\theta\partial\theta^T}) \tag{5.151}$$

- Fisher Information $I(\theta)$ is always positive definite, thus improve stability.

□ **More Specific Case: Scaled Exponential Family** $f(y; \vec{\theta}, \phi) = \exp\left(\frac{y'\theta - b(\theta)}{a(\phi)} + c(y; \phi)\right)$

where θ is the canonical parameter, declaring location.

This form of exponential family distribution posses some good properties (when approaching expectation and variance), and is one of the basic distribution assumption in Generalized Linear Model, which is an important MLE task. Detail about GLM and scaled exponential family see section 3.7 ~ page 110.

Further note that here we demand θ as canonical parameter, which is not necessarily the parameter μ we use. Assume θ as function of μ as $\theta = g(\mu)$.[8]

Properties:

- Log-likelihood:

$$l(\theta, \phi; y) = \frac{y'\theta - b(\theta)}{a(\phi)} + c(y; \phi) \tag{5.152}$$

- Expectation: equation 3.237 ~ page 111

$$\mathbb{E}(Y) = b'(\theta) \tag{5.153}$$

- Variance: equation 3.238 ~ page 111

$$var(Y) = a(\phi)b''(\theta) \tag{5.154}$$

[7] Detail see section 2.2.3 ~ page 45 & section 2.2.4 ~ page 47, page 47

- Score Function

$$S(\theta; \vec{x}) = \frac{\partial \ln f(\vec{x}; \theta)}{\partial\theta} = \frac{\partial l(\theta; \vec{x})}{\partial\theta} = \sum_{i=1}^{n} \frac{\partial \ln f(x_i; \theta)}{\partial\theta} \tag{5.148}$$

- Fisher Information

$$I(\theta) = \mathbb{E}\left[\frac{\partial \ln f(\vec{x}; \theta)}{\partial\theta} \frac{\partial \ln f(\vec{x}; \theta)}{\partial\theta^T}\right] = \mathbb{E}\left[-\frac{\partial^2 \ln f(\vec{x}; \theta)}{\partial\theta\partial\theta^T}\right] \tag{5.149}$$

[8] Here use notation in GLM, where $\theta = \eta = g(\mu)$.

- Score function:

$$S(\theta;y)=\frac{\partial l(\theta,\phi;y)}{\partial\theta}=\frac{y-b'(\theta)}{a(\phi)} \tag{5.155}$$

$$S(\mu;y)=\frac{\partial l}{\partial\theta}\frac{\mathrm{d}g(\mu)}{\mathrm{d}\mu}=\frac{y-b'(g(\mu))}{a(\phi)}g'(\mu) \tag{5.156}$$

- Observed Information $J(\theta)$ or $J(\mu)$:

$$J(\theta)=-l''(\theta)=\frac{b''(\theta)}{a(\phi)} \tag{5.157}$$

$$J(\mu)=-\frac{\partial^2 l(\mu)}{\partial\mu\partial\mu^T}=\frac{\partial g}{\partial\mu}b''(g(\mu))\frac{\partial g}{\partial\mu^T}+\left(\frac{\partial}{\partial\mu}\otimes\frac{\partial}{\partial\mu^T}g\right)(b'(g(\mu))-y) \tag{5.158}$$

- Fisher Information:

$$I(\theta)=\mathbb{E}(J(\theta))=\frac{b''(\theta)}{a(\phi)} \tag{5.159}$$

$$I(\mu)=\mathbb{E}(J(\mu))=\frac{1}{a(\phi)}\frac{\partial g}{\partial\mu}b''(g(\mu))\frac{\partial g}{\partial\mu^T} \tag{5.160}$$

□ **Fisher Scoring and GLM: Iterative Re-weighted Least Square (IRLS)**

Recall in GLM in section 3.7 ~ page 110

$$\mu_i\sim g^{-1}(x_i'\beta)\text{ or }g(\mu_i)\sim x'\beta \tag{5.161}$$

where minimizing task is

$$\hat{\beta}=\arg\max\sum_i l(\mu;x_i,y_i)=\arg\max\sum_i l(\beta;x_i,y_i) \tag{5.162}$$

where $l(\mu;x,y)$ satisfies $y_i\sim f(\mu_{y_i}=g(^{-1}(x_i'\beta))$. Use $\mathbb{E}(Y)=b'(\theta)$ we have

$$\mu=\mathbb{E}(Y)=g^{-1}(\eta)=g^{-1}(x'\beta)=b'(\theta) \tag{5.163}$$

Note that in GLM model we should have chosen canonical link equation 3.250 ~ page 112 such that $g^{-1}=b'$, then

$$\theta=\eta=x'\beta=g(\mu)\rightsquigarrow g^{-1}(\theta)=g^{-1}(\eta)=g^{-1}(x'\beta)=\mu=E(Y) \tag{5.164}$$

i.e. we could get: (Here Y and X for sample matrix notation $\vec{y}$ and $\mathbf{X}$)

$$S(\beta;Y)=\frac{\partial l(\beta)}{\partial\beta}=\frac{X^TY-X^Tg^{-1}(X\beta)}{a(\phi)} \tag{5.165}$$

$$I(\beta)=\frac{1}{a(\phi)}\frac{\partial g}{\partial\beta}b''(\theta)\frac{\partial g}{\partial\beta^T}=\frac{1}{a(\phi)}X'WX \tag{5.166}$$

$$W(\theta):=b''(\theta)=\left.\frac{\partial g^{-1}(\theta)}{\partial\theta}\right|_{\theta=X\beta}=\frac{var(Y)}{a(\phi)} \tag{5.167}$$

Then we can use above result to modify Newton-Raphson Algorithm as

$$\beta^{(t+1)} = \beta^{(t)} + I(\beta^{(t)})^{-1}S(\beta) = \beta^{(t)} + (X'W^{(t)}X)^{-1}X'(Y - g^{-1}(X\beta^{(t)})) \tag{5.168}$$

where

$$W^{(t)} = b''(\xi)\big|_{\xi=X\beta^{(t)}} \tag{5.169}$$

$$g^{-1}(\xi) = b'(\xi) \tag{5.170}$$

Further comment: iteration can be written

$$\beta^{(t+1)} = (X'W^{(t)}X)^{-1}X'W^{(t)}\left(X\beta^{(t)} + W^{-1(t)}(Y - g^{-1}(X\beta^{(t)}))\right) \tag{5.171}$$

where $Z = X\beta^{(t)} + W^{-1}(Y - g^{-1}(X\beta^{(t)}))$ can be expressed as the taylor series of $Z = g(Y)$ at $\hat{Y} = g^{-1}(X\beta)$:

$$Z = g(Y) \approx g(g^{-1}(X\beta)) + \frac{\partial g}{\partial \mu}(Y - g^{-1}(X\beta)) \tag{5.172}$$

$$= X\beta + W^{-1}(Y - g^{-1}(X\beta)) \tag{5.173}$$

i.e. each step of iteration is a weighted generalized linear regression $Z \approx g(Y) \sim X\beta$

□ **Useful choise of General Linear Model and MLE iteration**

Note: for conciseness, the following part would use the most commonly used parameter, and canonical variable $\theta = \eta = x'\beta$

Regression data: $(y_i, x_i), i = 1, 2, \ldots, n$

- Simple Linear Regression: Normal Distribution

$$Y_i \sim N(x_i'\beta, \sigma^2) \qquad f(y; \mu, \sigma^2) = \exp\left\{\frac{y\mu - \frac{1}{2}\mu^2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right\} \tag{5.174}$$

  - Link function:

$$g(y) = y \Leftrightarrow g^{-1}(x'\beta) = x'\beta \tag{5.175}$$

  - Canonical variable $\theta = x'\beta = \mu$ and its function

$$b(\theta) = \frac{1}{2}\theta^2 \qquad a(\sigma^2) = \sigma^2 \tag{5.176}$$

$$\mathbb{E}(Y) = b'(\theta) = \theta \tag{5.177}$$

$$var(Y) = a(\phi)b''(\theta) = \sigma^2 \tag{5.178}$$

  - Log-likelihood:

$$l(\beta, \sigma^2; y, x) = \frac{yx'\beta - \frac{1}{2}\beta' xx'\beta}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2) \tag{5.179}$$

– Raphson Iteration:

$$\frac{\partial l}{\partial \beta} =\frac{1}{\sigma^2}\sum_{i=1}^{n}\left(x_i(y_i - x_i'\beta)\right) \tag{5.180}$$

$$\frac{\partial l}{\partial \beta \partial \beta^T} =-\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i x_i' \tag{5.181}$$

iteration step:

$$\beta^{(t+1)} = \beta^{(t)} + (X'X)^{-1}X'(Y - X\beta) \tag{5.182}$$

– Fisher' Scoring Iteration:

$$W(\beta) =b''(\mu) = I_{p+1} \tag{5.183}$$

$$I(\beta) =\frac{1}{a(\sigma^2)}X'WX = \frac{1}{\sigma^2}X'X \tag{5.184}$$

iteration step: the same as G-R method

$$\beta^{(t+1)} = \beta^{(t)} + (X'X)^{-1}X'(Y - X\beta) \tag{5.185}$$

- Logistic Regression: Binomial Distribution

$$Y_i \sim B(n_0, \text{logistic}(x'\beta)) \qquad f(y; n_0, \pi) = \exp\left\{y \ln\frac{\pi}{1-\pi} + n_0 \ln(1-\pi) + \ln\binom{n_0}{y}\right\} \tag{5.186}$$

– Link function:

$$g(y) = \ln\frac{y}{1-y} = \text{logit}(y) \Leftrightarrow g^{-1}(x'\beta) = \frac{1}{1+e^{-x'\beta}} = \text{logistic}(x'\beta) \tag{5.187}$$

– Canonical variable $\theta = x'\beta = \text{logit}(\pi)$

$$b(\theta) = n_0 \ln(1-\pi) = n_0 \ln\frac{1}{1+e^{\theta}} \qquad a(\phi) = 1 \tag{5.188}$$

$$\mathbb{E}(Y) = b'(\theta) = n_0\frac{1}{1+e^{-\theta}} = n_0\pi \tag{5.189}$$

$$var(Y) = a(\phi)b''(\theta) = n_0\frac{e^{-\theta}}{(1+e^{-\theta})^2} = n_0\pi(1-\pi) \tag{5.190}$$

– Log-likelihood:

$$l(n_0, \beta; y, x) = yx'\beta + n_0 \ln(1 - g^{-1}(x'\beta)) + \ln\binom{n_0}{y} \tag{5.191}$$

– Raphson Iteration:

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} x_i\left(y_i - n_0\text{logistic}(x_i'\beta)\right) \tag{5.192}$$

$$\frac{\partial l}{\partial \beta \partial \beta^T} = \sum_{i=1}^{n} x_i x_i' \frac{n_0 e^{-x_i'\beta}}{(1+e^{-x_i'\beta})^2} = \sum_{i=1}^{n} x_i x_i' n_0 g^{-1}(x'\beta)(1 - g^{-1}(x'\beta)) \tag{5.193}$$

iteration step:

$$\beta(t+1) = \beta^{(t)} - \left(\frac{\partial l}{\partial \beta \partial \beta^T}\right)^{-1}\frac{\partial l}{\partial \beta}\bigg|_{\beta^{(t)}} \tag{5.194}$$

– Fisher's Scoring Iteration:

$$W(\beta) = \frac{var(Y)}{a(\phi)} = n_0 g^{-1}(X\beta)(1 - g^{-1}(X\beta)) \tag{5.195}$$

$$I(\beta) = X'WX = X'\text{diag}\{n_0 g^{-1}(x_i'\beta)(1 - g^{-1}(x_i'\beta))\}X \tag{5.196}$$

- Poisson Regression: Poisson Distribution

$$Y_i \sim P(e^{x'\beta}) \qquad f(y;\lambda) = \exp\{y\ln\lambda - \lambda - \ln(y!)\} \tag{5.197}$$

– Link function:

$$g(y) = \ln y \Leftrightarrow g^{-1}(x'\beta) = e^{x'\beta} \tag{5.198}$$

– Canonical variable $\theta = x'\beta = \ln\lambda$

$$b(\theta) = \lambda = e^{\theta} \qquad a(\phi) = 1 \tag{5.199}$$

$$\mathbb{E}(Y) = b'(\beta) = e^{\theta} = \lambda \tag{5.200}$$

$$var(Y) = a(\phi)b''(\theta) = e^{\theta} = \lambda \tag{5.201}$$

5.4.4 Linear Modification to Step Length

In minimizing methods, the key idea is usually approximate original $g(x)$ with some $\tilde{g}(x)$, and the idea of restricting step length is to avoid severe deviation of $\tilde{g}$ from g, the most direct method is to adjust step length on a given direction:

$$x^{(t+1)} = x^{(t)} + \alpha^{(t)}p^{(t)} \tag{5.202}$$

we should choose proper scale of $a^{(t)}$ adapted to the craggedness of $g(x)$ for better convergence. In machine learning, $\alpha^{(t)}$ also refers to learning rate.

- Fixed step-length: Fix $\alpha^{(t)} = \alpha$ (usually $\alpha = 1$)
- Backtracking: Starting from e.g. $\alpha_0^{(t)} = 1$ and calculate corresponding $g\left(x^{(t)} + \alpha_i^{(t)}p^{(t)}\right)$, update $\alpha_{i+1}^{(t)} = \alpha_i^{(t)}/2$ until $g\left(x^{(t)} + \alpha_i^{(t)}p^{(t)}\right) < g\left(x^{(t)}\right)$, i.e.

$$\alpha^{(t)} = \max\frac{\alpha_0}{2^n},\ s.t. g\left(x^{(t)} + \alpha_i^{(t)}p^{(t)}\right) < g\left(x^{(t)}\right) \tag{5.203}$$

- Exact line search:

$$\alpha^{(t)} = \arg\min_{\alpha} g\left(x^{(t)} + \alpha p^{(t)}\right) \tag{5.204}$$

Special case for quadric form

Properties:

– For exact line search, contiguous direction step are perpendicular, i.e.

$$\left.\frac{\partial f\left(x^{(t)}+\alpha p^{(t)}\right)}{\partial \alpha}\right|_{\alpha^{(t)}} = 0 = \left.\nabla f^T\right|_{x^{(t+1)}} p^{(t)} \Rightarrow p^{(t+1)} \perp p^{(t)} \tag{5.205}$$

– $\alpha^{(t)}$ in special case for quadric form $f(x)=\frac{1}{2}x^TAx-b^Tx+c$, denote 'residual' $r^{(t)} \equiv Ax^{(t)}-b = \nabla f\left(x^{(t)}\right)$

$$\alpha^{(t)} = \arg\min f\left(x^{(t)}+\alpha p\right) = -\frac{p^T(Ax^{(t)}-b)}{p^TAp} = -\frac{p^Tr^{(t)}}{p^TAp} \tag{5.206}$$

$$\xlongequal{\text{for } p=-\nabla f^{(t)}=-r^{(t)}} \frac{r^{(t)T}r^{(t)}}{r^{(t)T}Ar^{(t)}} \tag{5.207}$$

More general modification based on quadric form see section 5.4.7 ~ page 179, Trust Region Method.

5.4.5 Quasi Newton Method

One of the main difficulty of Newton-Raphson method is calculation of Hessian $H\left(x^{(t)}\right)$ (as well as its inverse). We can use some estimation method for $M^{(t)} \equiv \hat{H}^{(t)}$, for equivalently for $B^{(t)} \equiv \left[\hat{H}^{(t)}\right]^{-1}$[9]

Updating:

$$x^{(t+1)} = x^{(t)} - \alpha^{(t)}\left[M^{(t)}\right]^{-1}\nabla f\left(x^{(t)}\right) = x^{(t)} - \alpha^{(t)}B^{(t)}\nabla f\left(x^{(t)}\right) \tag{5.208}$$

□ **Discrete Newton Method**

Numerical finite differential for $M^{(t)}$:

$$M_{ij}^{(t)} = \frac{f_i'\left(x^{(t)}+h_{ij}^{(t)}\hat{e}_j\right) - f_i'\left(x^{(t)}\right)}{h_{ij}^{(t)}} \tag{5.209}$$

This basic numeric method for Hessian has heavy calculation burden, and cannot ensure positive definition of Hessian, **Not** recommended.

□ **Quasi Newton Method: SR1, DFP, BFGS, L-BFGS, Broyden Class**

Instead of 'recalculating' $M^{(t+1)}$ (or $B^{(t+1)}$) in each step, we can 'update' $M^{(t+1)}$ based on known $M^{(t)}$, $x^{(t+1)}$, $x^{(t)}$, $\nabla f^{(t+1)}$, $\nabla f^{(t)}$. And Update of $x^{(t+1)}$ as

$$x^{(t+2)} = x^{(t+1)} - \left[M^{(t+1)}\right]^{-1}\nabla f^{(t+1)} \tag{5.210}$$

Calculation of second derivative is avoided. Note that in $M^{(t+1)}$, in total n^2 elements are needed, thus we usually has some basic assumptions/conditions for $M^{(t+1)}$ which should be inherited in iteration

- Symmetry:

$$M^{(t+1)} = \left(M^{(t+1)}\right)^T \Leftrightarrow B^{(t+1)} = \left(B^{(t+1)}\right)^T \tag{5.211}$$

[9] Notation different from lecture note. Here always use H for Hessian $H \equiv \nabla\nabla f$

- Secant Condition/Quasi-Newton Condition:

  Define

$$y^{(t)} \equiv \nabla f^{(t+1)} - \nabla f^{(t)} \qquad s^{(t)} \equiv x^{(t+1)} - x^{(t)} \tag{5.212}$$

  Secant condition:

$$y^{(t)} = M^{(t+1)} s^{(t)} \Leftrightarrow s^{(t)} = B^{(t+1)} y^{(t)} \tag{5.213}$$

- Curvature Condition/Strong Convex Condition (on function property)

$$\langle s^{(t)}, y^{(t)} \rangle \geq \xi > 0 \tag{5.214}$$

With these two constraint, degree of freedom of $M^{(t+1)}$ is reduced to $\dfrac{n(n-1)}{2}$

> In the following part in this subsubsection, we will usually ignore the superscipt $\cdot^{(t)}$ or use subscript $\cdot_t$ if necessary.

- SR-1 Method/Davidon Update: Rank-1 updated

$$M_{(t+1)} = M_{(t)} + \frac{(y - M_{(t)}s)(y - M_{(t)}s)^T}{(y - M_{(t)}s)^T s} \tag{5.215}$$

$$B_{(t+1)} = B_{(t)} + \frac{(s - B_{(t)}y)(s - B_{(t)}y)^T}{(s - B_{(t)}y)^T y} \tag{5.216}$$

  Note: SR-1 update cannot guarenteed the positive definition of $M_{(t+1)}$ and $B_{(t+1)}$. But this method can be used together with Trust Region method to avoid the disadvantage.

- DFP Method & BFGS Method:

  Idea: We want to pick the Hessian M nearest to $M_{(t)}$, with constraints above, i.e.

$$M_{(t+1)} = \arg\min_{M} \|M - M_{(t)}\| \qquad s.t. M = M^T,\ y = Ms \tag{5.217}$$

  where norm $\|\cdot\|$ can take different form, each giving a corresponding quasi-Newton update. Here we take weighted frobenius norm

$$\|A\|_W = \|W^{-1/2} A W^{-1/2}\|_F \qquad y = Ws \tag{5.218}$$

  Note: Here we take any W with secant condition for a scale-invariant norm (because W would also looks like some 'hessian'[10])

[10] One of possible form of W can take

$$W = \int_0^1 \nabla\nabla f\left(x_{(t)} + \tau s\right) \mathrm{d}\tau \tag{5.219}$$

Solution[11]:

$$M_{(t+1)} = \left(I - \frac{ys^T}{y^Ts}\right) M_{(t)} \left(I - \frac{sy^T}{y^Ts}\right) + \frac{yy^T}{y^Ts} \tag{5.225}$$

And inverse using Sherman-Morrison formula $\left(A + u^Tv\right)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}$ (Note: tough calculation, haven't tried) is the DFP updating:

$$B_{(t+1)} = M^{-1}_{(t+1)} = B_{(t)} - \frac{B_{(t)}yy^TB_{(t)}}{y^TB_{(t)}y} + \frac{ss^T}{y^Ts} \tag{DFP}$$

⋆ Similarly, using dual minimizing problem

$$B_{(t+1)} = \arg\min_B \|B - B_{(t)}\|_{W^{-1}} \qquad s.t. B = B^T,\ s = By \tag{5.226}$$

Solution:

$$B_{(t+1)} = \left(I - \frac{sy^T}{y^Ts}\right) B_{(t)} \left(I - \frac{ys^T}{y^Ts}\right) + \frac{ss^T}{y^Ts} \tag{BFGS}$$

Also we can inverse to get estimation of hessian in BFGS updating:

$$M_{(t+1)} = M_{(t)} - \frac{M_{(t)}ss^TM_{(t)}}{s^TM_{(t)}s} + \frac{yy^T}{y^Ts} \tag{5.227}$$

Note that our final goal is to evaluate $B_{(t+1)}$ to get step direction

$$p_{(t+1)} = -B_{(t+1)}\nabla f_{(t+1)} \tag{5.228}$$

$$B_{(t+1)} = \begin{cases} B_{(t)} + \dfrac{(s - B_{(t)}y)(s - B_{(t)}y)^T}{(s - B_{(t)}y)^Ty} & \text{(SR1)} \\ B_{(t)} - \dfrac{B_{(t)}yy^TB_{(t)}}{y^TB_{(t)}y} + \dfrac{ss^T}{y^Ts} & \text{(DFP)} \\ \left(I - \dfrac{sy^T}{y^Ts}\right) B_{(t)} \left(I - \dfrac{ys^T}{y^Ts}\right) + \dfrac{ss^T}{y^Ts} & \text{(BFGS)} \end{cases} \tag{5.229}$$

[11] Solution of minimizing problem using Lagrange multiplier: Note that weighted frobenius norm is

$$\|M - M_t\|^2_W = tr\left(W^{-1/2}(M - M_{(t)})W^{-1}(M - M_{(t)})W^{-1/2}\right) \tag{5.220}$$

with constraints $M = M^T$, $y = Ms$, given $y = Ws$, $M_{(t)} = M^T_{(t)}$. Minimizing Lagrange function taken as

$$\Xi(M, \lambda, \Lambda) = tr\left(W^{-1/2}(M - M_{(t)})W^{-1}(M - M_{(t)})W^{-1/2}\right) - 4\lambda^T(Ms - y) - 4tr\left(\Lambda(M - M^T)\right) \tag{5.221}$$

$$\arg\min \Xi(M, \lambda, \Lambda) \Rightarrow \begin{cases} \dfrac{\partial \Xi}{\partial M} = 2W^{-1}(H - H_{(t)})^TW^{-1} - 4\lambda s^T - 4(\Lambda^T - \Lambda) = 0 \\ \dfrac{\partial \Xi}{\partial 4\lambda} = Ms - y = 0 \\ \dfrac{\partial \Xi}{\partial 4\Lambda} = M^T - M = 0 \end{cases} \tag{5.222}$$

Solve: fisrt eliminate $\Lambda - \Lambda^T$ into $\lambda s^T - s\lambda^T$, then eliminate λs^T, finally eliminate $s\lambda^T$, solution:

$$M_{(t+1)} = M_{(t)} + \frac{y - M_{(t)}s}{y^Ts}y^T + \frac{y}{y^Ts}\left(\frac{s^TM_{(t)}sy^T}{y^Tsy^Ts} - \frac{s^TM_{(t)}}{y^Ts}\right) \tag{5.223}$$

$$= \left(I - \frac{ys^T}{y^Ts}\right) M_{(t)} \left(I - \frac{sy^T}{y^Ts}\right) + \frac{yy^T}{y^Ts} \tag{5.224}$$

Comment: DFP updating and BFGS updating can both update $(M^{(t+1)}, B^{(t+1)})$ from $(M^{(t)}, B^{(t)})$, with symmetry condition and secant condition. But such updating has $dof = \dfrac{n(n-1)}{2} > 0$, thus we can have multiple choice of updating, in which DFP and BFGS get such update from minimizing weight norm. In practical terms, **BFGS is usually more suitable than DFP in general optimization problem**.

A guess from their minimizing problem: BFGS is more 'direct' by minimizing $\|B - B^{(t)}\|_{W^{-1}}$, without the inverse of minimizer matrix as in DFP.

□ **More methods based on DFP and BFGS:**

- Broyden Class: linear combination of DFP and BFGS

$$B_{(t+1)} = M_{(t+1)}^{-1} \qquad M_{(t+1)} = (1-\phi_{(t)})M_{(t+1)}^{\text{BFGS}} + \phi_{(t)}M_{(t+1)}^{\text{DFP}} \tag{5.230}$$

  Set: $\phi = 1$ for DFP, $\phi = 0$ for BFGS, $\phi = \dfrac{s^T y}{s^T y - s^T M_{(t)} s}$ for SR-1

- L-BFGS Method: For high dimension $n = \dim(x) \gg 1$, storage of $M_{(t)}$ or $B_{(t)}$ take $\sim n^2$, which could be unacceptable. Thus instead of storing $B_{(t)}$, $y_{(t)}$ and $s_{(t)}$, we can store $y_{(t_i)}$, $s_{(t_i)}$ $\forall t_i < t$, or at least as more t_i as possible.

5.4.6 Steepest Descent*

Steepest Descent Method

5.4.7 Trust Region Method

Approximation quadric form $\tilde{f}$ at iteration $x^{(t)}$

$$\tilde{f}_{x^{(t)}}(x) = f(x^{(t)}) + \left(x - x^{(t)}\right)^T \nabla f^{(t)} + \frac{1}{2}\left(x - x^{(t)}\right)^T M^{(t)} \left(x - x^{(t)}\right) \tag{5.231}$$

Trust Region: within $\|x - x^{(t)}\| \le \Delta^{(t)}$, $\tilde{f}_{x^{(t)}}$ is similar enough to g, and we minimize $\tilde{f}_{x^{(t)}}$ within trust region.

□ **Iteration:**

- Preset parameters:

$$\text{Region Radius :} \Delta^{(t)} > 0 \tag{5.232}$$

$$\text{TR step quality measure :} \eta_\nu (= 0.9),\ \eta_s < \eta_\nu (= 0.1) \tag{5.233}$$

$$\text{region update :} \gamma_i \ge 1 (= 2),\ \gamma_d (= 0.5) \tag{5.234}$$

and approximation function (usually use quadric form)

$$\tilde{f}_{x^{(t)}}(x) \left(= f(x^{(t)}) + \left(x - x^{(t)}\right)^T \nabla f^{(t)} + \frac{1}{2}\left(x - x^{(t)}\right)^T M^{(t)} \left(x - x^{(t)}\right) \right) \tag{5.235}$$

- In each iteration step (t), solve constraint minimizing problem

$$x_{\text{cm}} = \arg\min_x \tilde{f}_{x^{(t)}}(x), \quad s.t. \|x - x^{(t)}\| \le \Delta^{(t)} \tag{5.236}$$

and the quality of reduction: $\rho^{(t)}$:

$$\rho^{(t)} = \frac{f^{(t)} - f(x_{\text{cm}})}{f^{(t)} - \tilde{f}(x_{\text{cm}})} \tag{5.237}$$

- Update $x^{(t+1)}$ and $\Delta^{(t+1)}$ based on quality $\rho^{(t)}$

$$\begin{cases} x^{(t+1)} = x_{\text{cm}},\ \Delta^{(t+1)} = \gamma_i \Delta^{(t)} & \rho^{(t)} \ge \eta_\nu \\ x^{(t+1)} = x_{\text{cm}},\ \Delta^{(t+1)} = \Delta^{(t)} & \eta_s \le \rho^{(t)} < \eta_\nu \\ x^{(t+1)} = x^{(t)},\ \Delta^{(t+1)} = \gamma_d \Delta^{(t)} & \rho^{(t)} < \eta_s \end{cases} \tag{5.238}$$

5.4.8 Conjugate Gradient Method

Note that in Gauss-Raphson method, our iteration step was obtained by minimizing the taylor series to $O(x^2)$ in equation 5.143 ~ page 170,

$$f(x) \approx \tilde{f}_{x^{(t)}}(x) = f(x^{(t)}) + (x - x^{(t)})^T \nabla f(x)|_{x^{(t)}} + \frac{1}{2}(x - x^{(t)})^T \nabla\nabla f(x)|_{x^{(t)}} (x - x^{(t)}) \tag{5.239}$$

or, as a more specific problem: get x^* by minimizing function

$$x^* = \arg\min_x f(x) = \frac{1}{2} x^T A x - b^T x + c \tag{5.240}$$

which has analytical solution $Ax^* = b$, and we could solve this equation using algebratic methods in section 5.2 ~ page 148. Here Conjugate Gradient Methods uses iteration method to solve it, which can be used in Newton-Raphson/Fisher Scoring etc. to help find $\hat{H}^{(t)} p^{(t)} = -\nabla f^{(t)}$. Or use some modified conjugate gradient method directly on $f(xx)$.

□ **Conjugate vectors of A**

Note: Here we assume $\underset{n\times n}{A}$ is symmetric positive definite (SPD). SPD of A allows us to define an inner product based on A:

$$\langle \xi_i, \xi_j \rangle_A = \xi_i^T A \xi_j \tag{5.241}$$

and conjugate vectors of A are vector set that are 'orthonormal' in the sense of $\langle \cdot, \cdot \rangle_A$:

$$\xi_i^T A \xi_j = \delta_{ij},\ \forall \xi_i, \xi_j \in \text{CV set} \tag{5.242}$$

Further if A is full-rank and conjugate vector set has n independent vector, it can span the whole space $\text{span}\{\xi_1, \xi_2, \ldots, \xi_n\} = \mathbb{R}^n$, thus we can expand any vector $x - x^{(0)}$ on $\{\xi_i\}$:

$$x = x^{(0)} + \sum_{i=1}^{n} c_i \xi_i \tag{5.243}$$

and express $f(x)$ as function of c_i, using orthonormal condition $\xi_i' A \xi_j = \delta_{ij}$

$$
\begin{aligned}
f(x) =& \frac{1}{2} x^T A x - b^T x + c && (5.244)\\
=& \frac{1}{2}(x^{(0)} + \sum_{i=1}^{n} c_i \xi_i)^T A (x^{(0)} + \sum_{i=1}^{n} c_i \xi_i) - b^T (x^{(0)} + \sum_{i=1}^{n} c_i \xi_i) + c && (5.245)\\
=& \frac{1}{2} \sum_{i=1}^{n} c_i^2 + \sum_{i=1}^{n} c_i (A x^{(0)} - b)^T \xi_i + f\left(x^{(0)}\right) && (5.246)\\
=& \left(\sum_{i=1}^{n} \frac{1}{2} c_i^2 + c_i (A x^{(0)} - b)^T \xi_i \right) + f\left(x^{(0)}\right) && (5.247)
\end{aligned}
$$

i.e. we can minimize the quadric form by minimizing on each direction separately.

□ **Conjugate Direction Construction**

General procedue: Using a linear-independent vector set $\{\nu_i\}$ and use a process similar to Gauss-Elimination to get $\{\xi_i\}$:

$$
\begin{aligned}
\xi_k =& \nu_k - \sum_{i=1}^{k-1} \frac{\xi_i^T A \nu_k}{\xi_i^T A \xi_i} \xi_i && (5.248)\\
=& \left(I - \sum_{i=1}^{k-1} \frac{\xi_i \xi_i^T A}{\xi_i^T A \xi_i} \right) \nu_k = \prod_{i=1}^{k-1} \left(I - \frac{\xi_i \xi_i^T A}{\xi_i^T A \xi_i} \right) \nu_k && (5.249)
\end{aligned}
$$

Note that here we only use the condition $\xi_i^T A \xi_j = \delta_{ij}$, and $\{\nu_i\}$ is arbitrary. To avoid the storage spend of $O(n^2)$, we could choose special way in descent such that conjugate perpendicular information of $\xi_{i<k}$ are automatically 'stored' in ξ_k, and we would only need storage $O(n)$:

- Conjugate Gradient for Quadric Form: In each descent steps k

$$
x_{k+1} = x_k + \alpha_k \xi_k, \qquad \xi_k = \nu_k - \sum_{i=1}^{k-1} \frac{\xi_i^T A \nu_k}{\xi_i^T A \xi_i} \xi_i \tag{5.250}
$$

choose α_k by exact line search $\alpha_k = -\dfrac{r_k^T \xi_k}{\xi_k^T A \xi_k} = -\dfrac{\nabla f_k^T \xi_k}{\xi_k^T A \xi_k}$, $r_k = A x_k - b = \nabla f_k$, and $\nu_k = -\nabla f(x_k)$[12], and using the fomula:

$$
\alpha_i A \xi_i = A(x_{i+1} - x_i) = r_{i+1} - r_i = \nabla f_{i+1} - \nabla f_i \tag{5.253}
$$

[12] Such that $\nu_k = -\nabla f_k \perp \text{span}\{\xi_1, \ldots, \xi_{k-1}\} = \text{span}\{\nu_1, \ldots, \nu_{k-1}\}$, then.

$$
\nu_k = -\nabla f_k \perp \nu_i, \ \forall i < k \tag{5.251}
$$

$$
\nu_k = -\nabla f_k \perp \xi_i, \ \forall i < k \tag{5.252}
$$

and the orthogonality of $\xi,\ \nabla f, \xi_k$ can be expressed as

$$\begin{aligned}\xi_k =& -\nabla f(x_k) - \sum_{i=1}^{k-1} \frac{(\nabla f_{i+1} - \nabla f_i)^T \nabla f_k}{(\nabla f_{i+1} - \nabla f_i)^T \xi_i} \xi_i && (5.254)\\ =& -\nabla f(x_k) + \frac{(\nabla f_k - \nabla f_{k-1})^T \nabla f_k}{\nabla f_{k-1}^T \nabla f_{k-1}} \xi_{k-1} && \text{(PR)}\\ =& -\nabla f(x_k) + \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2} \xi_{k-1} && \text{(FR)}\end{aligned}$$

For general minimizing problem, we can either use conjugate gradient just for solving $\hat{H}^{(t)} p^{(t)} = -\nabla f^{(t)}$ in each step (t), or more directly use the following conjugate method directly on the general $f(x)$: take different α and coefficient of vector as modification to the non-quadric part of f

$$\begin{aligned}x_{k+1}^{(t)} =& x_k^{(t)} + \alpha_k^{(t)} p_k^{(t)} && (5.255)\\ p_k^{(t)} =& -\nabla f(x_k^{(t)}) + \beta_k^{(t)} p_{k-1}^{(t)} && (5.256)\end{aligned}$$

where:

- General Form of Conjugate Gradient: In each sub-step $(t)k$, replace $A_k^{(t)} = A^{(t)}$ by $\nabla\nabla f\left(x_k^{(t)}\right)$, i.e.

$$\begin{aligned}\alpha_k^{(t)} =& -\frac{\nabla f(x_k^{(t)})^T p_k}{p_k^{(t)T} \nabla\nabla f(x_k^{(t)}) p_k^{(t)}} && (5.257)\\ \beta_k^{(t)} =& \frac{\nabla f(x_k^{(t)})^T \nabla\nabla f(x_k^{(t)}) p_k^{(t)}}{p_{k-1}^{(t)T} \nabla\nabla f(x_k^{(t)}) p_{k-1}^{(t)}} && (5.258)\\ k =& 1, 2, \ldots, n && (5.259)\end{aligned}$$

- Fletcher-Reeves Method:

$$\begin{aligned}\alpha_k^{(t)} =& \arg\min_{\alpha} f\left(x_k^{(t)} + \alpha p_k^{(t)}\right) && (5.260)\\ \beta_k^{(t)} =& \frac{\|\nabla f(x_k^{(t)})\|^2}{\|\nabla f(x_{k-1}^{(t)})\|^2} && (5.261)\\ k =& 1, 2, \ldots, n && (5.262)\end{aligned}$$

- Polak-Ribière Method:

$$\begin{aligned}\alpha_k^{(t)} =& \arg\min_{\alpha} f\left(x_k^{(t)} + \alpha p^{(t)}\right) && (5.263)\\ \beta_k^{(t)} =& \frac{\left(\nabla f(x_k^{(t)}) - \nabla f(x_{k-1}^{(t)})\right)^T \nabla f(x_k^{(t)})}{\|\nabla f(x_{k-1}^{(t)})\|^2} && (5.264)\\ k =& 1, 2, \ldots, n && (5.265)\end{aligned}$$

Section 5.5 Expectation Maximization Algorithm

Motivation: use MLE to estimate some model parameter θ for model $\{x_i\}$ i.i.d. $\sim f(x|\theta)$. Difficulty: for complex model

Main appication: Probability Generative Model, observed value x_i is generated from distributon $f(x|z_i, \theta_i, \theta_z)$ dependent on **unobserved** random $z \sim g(z|\theta_z)$(usually z is discrete, denoted $z_\nu = z_\alpha, \dots z_\gamma$). Where we know the form of $f(x, z|\theta_{z_\nu}, \theta_z)$, but form of $f(x|\theta_{z_\nu}, \theta)$ might be hard to solve, thus we use an iterative method to deal with the latent variable z so that we can use the known form $f(x, z|\theta_{z_\nu}, \theta)$.

5.5.1 Requisite Knowledge

- Kullback-Leibler Divergence: mearures the difference of distribution $p(x)$ from distribution $q(x)$

$$\mathrm{KL}(q||p) \equiv -\int q(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x \tag{5.266}$$

  Note: non-exchange for p, q.

- Jensen Inequality: For **concave** function $h(x)$ and random variable $X \sim f$

$$\mathbb{E}_f\left(h(X)\right) \leq h\left(\mathbb{E}_f(X)\right) \tag{5.267}$$

Then we have the property of non-negativity of $\mathrm{KL}(q\|p)$:

$$\mathrm{KL}(q||p) \geq 0, \forall p(x), \quad = \text{ for } p(x) = q(x) \tag{5.268}$$

A brief proof see section 1.7 ~ page 31.

5.5.2 Derivation

Notation: $\theta = (\theta_{z_\nu}, \theta_z)$, sample $X = (x_1, x_2, \dots, x_N)$. Expectation of function of ramdom variable $h(Y)$ on distribution $q(y)$ as $E_{(q_y)}\left(h(Y)\right)$.

Target: MLE of $l(\theta|X) \equiv \sum_{i=1}^{N} \log f(x_i|\theta)$. i.e. get $\theta^* = \arg\max_\theta l(\theta|X)$.

□ **Key Formula**

But due to the intractablility of $f(x|\theta)$, we have to expand to the full form $f(x, z|\theta)$, and use a mathematic trick of $E_q(\cdot)$, where $q(z)$ is any arbitrary distribution of z.

$$f(x|\theta) = f(x, z|\theta) f(z|x, \theta) \Rightarrow \tag{5.269}$$

$$\Rightarrow \log f(x|\theta) = E_{q(z)}\left(\log f(x|\theta)\right) = E_{q(z)}\left(\log f(x, z|\theta) f(z|x, \theta)\right) \tag{5.270}$$

$$= \int q(z) \log f(x, z|\theta) f(z|x, \theta) \,\mathrm{d}z \tag{5.271}$$

$$= \int q(z) \log \frac{f(x, z|\theta)}{q(z)} \,\mathrm{d}z + \mathrm{KL}\left(q||f(z|x, \theta)\right) \tag{5.272}$$

$$\geq \int q(z) \log \frac{f(x, z|\theta)}{q(z)} \,\mathrm{d}z, \quad \forall x = x_1, x_2, \dots, x_N \tag{5.273}$$

where $\int q(z)\log\frac{f(x,z|\theta)}{q(z)}\,\mathrm{d}z$ is also called ELBO (Evidence Lower Bound) of $\log f(x|\theta)$. And we could similarly get the ELBO of log-likelihood:

$$l(\theta|X)=\sum_{i=1}^{N}\log f(x_i|\theta)\geq\sum_{i=1}^{N}\int_z q_i(z)\log\frac{f(x_i,z|\theta)}{q_i(z)}\,\mathrm{d}z\equiv\mathrm{ELBO}(q,\theta),\quad q=\{q_i\}\tag{5.274}$$

i.e. ELBO provides a lower bound estimate for $l(\theta|X)$, thus we can instead maximize $\mathrm{ELBO}(q,\theta)$, using coordiante ascent is the Maximization-Maximization Algorithm:.[13]

$$q\text{ Maximum}:\ q^{(t+1)}=\arg\max_{q(z)}\mathrm{ELBO}(q,\theta^{(t)})=p(z|x,\theta^{(t)})\tag{5.275}$$

$$\theta\text{ Maximum}:\ \theta^{(t+1)}=\arg\max_{\theta}\mathrm{ELBO}(q^{(t+1)},\theta)\tag{5.276}$$

Further if we take can derive and use the form of $p(z|x,\theta)$ (sometimes this posterior is also intractable), then θ maximization step becomes

$$\theta^{(t+1)}=\arg\max_{\theta}\mathrm{ELBO}\left(p(z|x,\theta^{(t)}),\theta\right)=\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log\frac{f(x_i,z|\theta)}{p(z|x_i,\theta^{(t)})}\,\mathrm{d}z\tag{5.277}$$

$$=\arg\max_{\theta}\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log f(x_i,z|\theta)\,\mathrm{d}z\equiv Q(\theta|\theta^{(t)})\tag{5.278}$$

$$=\arg\max_{\theta}\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log f(x_i,z|\theta)\,\mathrm{d}z\tag{5.279}$$

and naturally q maximization Step becomes computing $Q(\theta|\theta^{(t)})=\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log f(x_i,z|\theta)\,\mathrm{d}z$, i.e. the Expectation of $f(x_i,z|\theta)$ on the posterior $p(z|x_i,\theta^{(t)})$, gather as Expectation-Maximization Algorithm:

Algorithm *Expectation-Maximization*

$$\mathrm{E_{xpectation}\text{-}Step}:Q(\theta|\theta^{(t)})=\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log f(x_i,z|\theta)\,\mathrm{d}z=\sum_{i=1}^{N}E_{p(z|x_i,\theta^{(t)})}\left[\log f(x_i,z|\theta)\right]\tag{5.280}$$

$$\mathrm{M_{aximization}\text{-}Step}:\theta^{(t+1)}=\arg\max_{\theta}Q(\theta|\theta^{(t)})=\arg\max_{\theta}\sum_{i=1}^{N}\int_z p(z|x_i,\theta^{(t)})\log f(x_i,z|\theta)\,\mathrm{d}z\tag{5.281}$$

E-M Algorithm can guarentee ascent of ELBO, and finally can ensure convergence (at least to a local maximum).

An application of E-M Algorithm is Gaussian Mixture Model for Clustering, detail see section 4.7.3 ~ page 140.

□ **Limitation and Improvement**

- Note that for generative model, we used a set of latent variable z, further we need an $\int_z \mathrm{d}x$ in $Q(\theta|\theta^{(t)})$, thus E-M requires low-dimensionality of z (e.g. in GMM, z is one-dimensional).

[13] where one of the 'coordinate' is the function space $q(z)$

- Slow convergence near extreme point, use acceleration improvement, e.g. Louis acceleration.
- In q-Maximization step, the form of q might be intractable (i.e. $p(z|x,\theta)$ intractable). For such function extreme value problem, use VEM (Variational Expectation Maximization) / VBEM(Variational Bayesian Expectation Maximization)

Section 5.6 Statistical Simulation

Statistic model inference problem can be solved using simulation, i.e. Monte-Carlo simulation. We can use the model-based random numbers to analyze model.

- Simulation is well-adapted, especially for high-dimensional problems
- Low-precision, usually sd $\sim O(\frac{1}{\sqrt{N}})$.
- Simulation method is also usually used for validation of model reliability.

An important application scenario of random simulation is Bayesian Statistics. More about it is introduced in section 13.3 ~ page 343, where there are variants of simulation methods and more example. This section would just introduce the basic ideas and methods of simulation.

▷ R. **Code**

Remember to sed random generator seed before simulation.

```
set.seed(INI_NUM)
```

5.6.1 Random Number Generation

Motivation: In many simulation models, we need to generate sets of random number with some distribution, **however** they are not totally 'random' because of repeatability need.

Idea: use a 'seed' to generate pseudo ranom number, where within each seed, numbers are random. The random number sequence can be repeated by setting the same seed.

□ **Linear Congruential Method for** $U(0,1)$

Linear Congruential Method (LCM) is the most commonly used method for generating uniform distribution $U(0,1)$, which is the basic for more complex distribution.

Algorithm *Linear Congruential for Uniform Distribution*

1. Set seed X_0 and pick proper a, c, m for LCM
2. Repeat for iterative i:
   (a) compute X_{i+1}

$$X_{i+1} \equiv aX_i + c (\mathrm{mod}\, m) \tag{5.282}$$

(b) Random number normalized to $(0,1)$

$$R_{i+1}=\frac{X_{i+1}}{m} \tag{5.283}$$

Choice of a,c,m: LCM sequence has period m thus m should large , and choice of a,c should avoid early period value, and let R_i distribute uniformly in $(0,1)$. Useful choice:

	a	c	m
Lehmer's	23		10^8+1
RANDU	$2^{16}+3$	1	2^{31}
IBM	16807		$2^{31}-1$

□ **Improvement of LCG**

Key problem is the periodically structure of generated X_i, i.e. when some $X_{\tilde{i}}$ return to X_0, then the following $X_{\tilde{i}+i}=X_i$ will repeat. Idea: modify the generation rule, e.g. use groups m of LCM X_{im} with differnet period P_m to generate R_i. Example: L'Ecuyer-CMRG Algorithm.

$$X_i=\left(\sum_{j=1}^{m}(-1)^{j+1}X_{ij}\right)\bmod m \quad R_i=\begin{cases}\dfrac{X_i}{m} & X_i>0\\ \dfrac{X_i}{m}+1 & X_i<0\\ 1-\dfrac{1}{m} & X_i=0\end{cases} \tag{5.284}$$

Note (Guess): why we want $X_i\in(0,1)$ rather than $[0,1]$. $(0,1)$ is homeomorphous with $\mathbb{R}$, which would be convenient for generate more distribution on $\mathbb{R}$.

More improvement: use general form

$$X_{i+1}=g(X_i,X_{i-1},\ldots)\bmod m \tag{5.285}$$

where in $g(\ldots)$ use more X_j, or take different function form.

□ **Random Variate Generation**

Further for any arbitrary distribution generation, which is 'variate' of uniform distribution[14]

Target: generate random number sequence with some distribution ($f(x)$ or $F(x)$ known). Denote random number sequence with $U(0,1)$ distribution as U_i

- Quantile Method/**Inverse Transform Method**: For distributions with traceable CDF $F(x)\in(0,1)$.

$$X_i=F_X^{-1}(U_i) \tag{5.286}$$

[14]关于 variate 的中译，笔者想到一个有趣的翻译是国际象棋术语"变例"variation，原指一类开局方法的衍生分支，这里或许可以指其他分布随机数可由均匀分布随机数衍生而来这一特点。

□ *Proof:*

$$\mathbb{P}\left(x < X < x + \mathrm{d}x\right) = \mathbb{P}\left(F(x) < U < F(x + \mathrm{d}x)\right) \tag{5.287}$$

$$= \frac{\partial F(x)}{\partial x}\mathrm{d}x = f(x)\,\mathrm{d}x \tag{5.288}$$

□

- **Acceptance-Rejection Sampling**: For $F(x)$ intractable, only $f(x)$ known,

  First decompose $f(x)$ as

$$f(x) = \frac{p(x)g(x)}{\int p(x)g(x)\,\mathrm{d}x} := \tilde{c}p(x)g(x) \Rightarrow p(x) = \frac{1}{\tilde{c}}\frac{f(x)}{g(x)} \tag{5.289}$$

  where $g(x)$ is some distribution that we can easily generate as **proposal distribution**. Then X_k sequence $\sim f(x)$ can be generated as follows:

  1. Propose a $\tilde{x} \sim g(x)$ and a $\tilde{u} \sim U(0,1)$
  2. Decide whether accept/reject $\tilde{x}$ to be X_k by:

$$\begin{cases} p(\tilde{x}) \geq \tilde{u} & \text{Reject} \\ p(\tilde{x}) < \tilde{u} & \text{Accept} \end{cases} \tag{5.290}$$

  □ *Proof:*

$$\mathbb{P}(\text{Accept}|x) = \frac{f(x)/g(x)}{\int p(\xi)g(\xi)\,\mathrm{d}\xi} \Rightarrow \mathbb{P}(\text{Accept}) = \int_x \mathbb{P}(\text{Accept}|x)g(x)\,\mathrm{d}x = \frac{1}{\int p(\xi)g(\xi)\,\mathrm{d}\xi} \tag{5.291}$$

  Using Bayesian Rule:

$$\mathbb{P}(x_k|\text{Accept}) = \frac{\mathbb{P}(\text{Accept}|x_k)g(x_k)}{\mathbb{P}(\text{Accept})} = f(X_k) \tag{5.292}$$

  □

  Note on Acceptance-Rejection Sampling:

  – Intuition: figure $f(x)$ lies under $\tilde{c}g(x)$ (where $\tilde{c} = \sup p(x)$). If for each x we accept it with probability $p(x) := \dfrac{f(x)}{\tilde{c}g(x)}$, then figure under $\tilde{c}g(x)$ is 'cut down' into $f(x)$, $\dfrac{1}{\tilde{c}} =$ acts as the normalize constant, which corresponds to 'accept rate' controlling generate frequency.

    We should choose a proper $g(x)$ which is similar to $f(x)$, so that $p(x)$ is close to 1 and the algorithm is efficient.

  – Expected accept ratio is

$$\mathbb{E}_{\tilde{x}\sim g(x)}\left[p(\tilde{x})\right] = \frac{1}{\tilde{c}} = \int p(x)g(x)\,\mathrm{d}x \tag{5.293}$$

  – Inefficient for high-dimentional problem: we have limited choices of sample-able distribution $g(\vec{x})$, and usually the expected accept rate $\int p(x)g(x)\,\mathrm{d}x$ is low.

  - Thick tail of $g(x)$ matters to control the bahaviour of $f(x)/g(x)$, otherwise we might not be able to find a $\tilde{c}$ such that $\tilde{c}g(x)$ bounds $f(x)$, making the sampling $\underset{x\to\infty}{}$ impossible.

- **Importance Resampling**: A method induced from importance sampling for integration, see Importance Sampling for detail.

▷ R. **Code**

Use the following command for all distributions supported in R. `stats::`. More distributions based on packages see https://CRAN.R-project.org/view=Distributions

```
?Distributions
```

5.6.2 Markov Chain Monte Carlo Method

Markov Chain Monte Carlo (MCMC) aims at solving integration and simulation problems by sampling from some distribution. MCMC can deal with complex distribution in high dimensional, an example is Gibbs distribution

$$\mathbb{P}(s) = \frac{e^{-\beta E(s)}}{\sum_\sigma e^{-\beta E(\sigma)}}, s \in \text{phase space} \tag{5.294}$$

In this case, partition function is almost impossible to calculate, what we could obtain is just the unnormalized distribution.

□ **Markov chain**

Detalied theory of Discrete Time Markov Chain (DTMC) see section 12.1.2 ~ page 310. Here are some brief precap.

Denote phase space $\mathcal{X} \ni x$. We could design such a process X_t to **transit from one state to another**, i.e. a conditional probability

$$\mathbb{P}(X_{t+1} = x | X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t) \tag{5.295}$$

a markov process is a memoryless one in which future only depends on the current one step, i.e.

$$\mathbb{P}(X_{t+1} = x | X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t) = \mathbb{P}(X_{t+1} = x | X_t = x_t) \tag{5.296}$$

for discrete version $x \in \{1, 2, \ldots\}$, we could denote it into a discrete-time stochastic process that is time-homogeneous

$$p_{ij} := \mathbb{P}(X_{t+1} = j | X_t = i), \quad \sum_j p_{ij} = 1 \tag{5.297}$$

which could be denoted in matrix form $P = \{p_{ij}\}$

Further, n-step transition denoted

$$p_{ij}^{(n)} := \mathbb{P}\left(X_{t+n} = j | X_t = i\right) \tag{5.298}$$
$$= \sum_k p_{ij}^{(n-1)} p_{kj} \tag{5.299}$$
$$= \sum_{k_1, k_2, \ldots, k_{n-1}} p_{ik_1} p_{k_1 k_2} \cdots p_{k_{n-1} j} \tag{5.300}$$
$$= P^n \tag{5.301}$$

Stationary Distribution / equilibrium distribution / invariant distribution π_∞ of a markov satisfies

$$\pi_\infty = \pi_\infty P = \pi_\infty P^n \tag{5.302}$$

Convergence and Ergodic Theorem: An ergodic markov chain converges to a unique stationary distribution π_∞[15]

$$\pi_\infty = \pi_0 \lim_{n \to \infty} P^n \tag{5.305}$$

where π_0 is an arbitrary initial distribution.

Detailed Balance Condition of stationart distribution π:[16]

$$\pi(i) p_{ij} = \pi(j) p_{ji}, \quad \forall i, j \tag{5.306}$$

is a sufficient condition for stationary distribution. Proof see section 12.1.2 ~ page 310.

MCMC aims at designing a proper chain p_{ij}, starting from some arbitrary state π_0, and after some (large enough) transitions t we would expect $\pi_{t+n} \to \pi_\infty$, $n = 1, 2, \ldots$.

□ **MCMC Algorithms for Unnormalized distribution**

To sample from an unnormalized distribution $\tilde{p}$, i.e. $p = \dfrac{\tilde{p}(x)}{\int \tilde{p}(\xi)\, \mathrm{d}\xi}$, but normalizer $Z = \int \tilde{p}(\xi)\, \mathrm{d}\xi$ is impossible to calculate, we could only get relative probability ratio of states.

- Metropolis-Hastings Algorithm:

Algorithm *MCMC*

---

[15] Ergodic = Irreducible + Aperiodic. Denote $i \rightsquigarrow j$ if $\exists n\, s.t. \mathbb{P}\left(X_n = j | X_0 = i\right) \neq 0$

- Irreducible:

$$i \rightsquigarrow j,\ j \rightsquigarrow i, \quad \forall i, j \tag{5.303}$$

All states of irreducible chain have the same period $T_i = T$.

- Aperiodic: if one of the state is aperiodic $T = 1$, then all states are, where

$$\text{Period } T_i = \gcd\{n : \mathbb{P}\left(X_n = i | X_0 = i\right) > 0\} \tag{5.304}$$

[16] Detailed balance condition has a similar correspondence in Quantum Mechanics, in which $\pi(i)$ is the state density at i, and p_{ij} is the transition probability.

1. A pre-selected conditional distribution $q(\cdot|x)$ is used as **proposal distribution**. In each step t, a new state is proposed as

$$Y \sim q(\cdot|X_t), \text{ i.e.} \mathbb{P}(Y = y|X_t) = q(y|X_t) \tag{5.307}$$

2. Acceptance ratio $\alpha_{Y|X_t}$ is the probability to accept the proposal as the new state

$$\alpha(Y|X_t) = \min\left\{1, \frac{\tilde{p}(Y)q(X_t|Y)}{\tilde{p}(X_t)q(Y|X_t)}\right\} \tag{5.308}$$

3. Increment of $t \to t+1$ if accept, else repeat the proposal-acceptance process.

Comment:

– Detailed balanced condition of M-H Algorithm

$$\begin{aligned}
p(x)p_{xy} =& p(x)q(y|x)\alpha(y|x) && (5.309)\\
=& p(x)q(y|x)\min\left\{1, \frac{p(y)q(x|y)}{p(x)q(y|x)}\right\} && (5.310)\\
=& \min\{p(x)q(y|x), p(y)q(x|y)\} && (5.311)\\
=& p(y)q(x|y)\min\left\{\frac{p(x)q(y|x)}{p(y)q(x|y)}, 1\right\} && (5.312)\\
=& p(y)p_{yx} && (5.313)
\end{aligned}$$

i.e. $p_{xy} = q(y|x)\alpha(y|x)$ is the transition matrix to generate the stationary distribution as $\pi_\infty = p(x)$

– Choice of proposal $q(\cdot|x)$ is flexible, but should be properly chosen for higher acceptance to increase efficiency.

- More variants of MCMC see section 13.3.5 ~ page 346 and section 13.3.6 ~ page 348.

5.6.3 Numerical Integration With Simulation

Motivation: In Bayesian statistics we usually use the following expression to calculate some posterior:

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{\int_\theta f(y|\theta)f(\theta)\,\mathrm{d}\theta} \tag{5.314}$$

in which θ is the parameter, y is the observd data. We might further construct come statistics using the posterior, usually in an integration form. But a key difficulty is calculation of the normalize integration $\int_z f(x|z)f(z)\,\mathrm{d}z = E_{f(z)}[f(x|z)]$, where $f(z)$ is the prior of z. Usually such integration needs numeric calculation. Statistical simulation using sampling is one of the methods.

Target: some kind of integration calculation:

$$I(h) = \int_{x\in\mathcal{X}} h(x)\,\mathrm{d}x \tag{5.315}$$

- Hit-and-Miss Method: if $\mathcal{X} \otimes h(x)$ is bounded in e.g. $[a,b] \otimes [0,M]$. We can generate uniform distribution (x,y) in the region, and count # points under $h(x)$, proportion of accept denoted $\hat{p}$, then

$$\hat{I} = M(b-a)\hat{p} \tag{5.316}$$

such estimation is guarenteed by CLT:

$$\hat{I}_{\mathrm{H}} \xrightarrow{\mathrm{d}} N\left(I, \frac{[M(b-a)]^2 p(1-p)}{N}\right) \tag{5.317}$$

- Mean Value Method: generate uniform distribution in e.g. $\mathcal{X} = [a,b]$, and calculate function value at each sample item $h(u_i)$, estimator

$$\hat{I} = \frac{N}{a-b}\sum_{i=1}^{N} h(u_i), \qquad w.r.t.\, u_i \sim U(a,b) \tag{5.318}$$

with CLT:

$$\hat{I}_{\mathrm{M}} \xrightarrow{\mathrm{d}} N\left(I, \frac{(b-a)^2 var(h(U))}{N}\right) \tag{5.319}$$

Note: $var(\hat{I}_{\mathrm{H}}) > var(\hat{I}_{\mathrm{M}})$. Intuitively, more points are used in mean value method, thus is more precise.

Random simulation has good performance for high-dimensional case by avoiding curse of dimensionality.

□ **Importance Sampling Estimator**

Improvement of mean value estimator: Note that in mean value with uniform distribution, variance

$$var(\hat{I}_{\mathrm{M}}) = \frac{(b-a)^2}{N} var(h(U)) \tag{5.320}$$

could be large if $h(x)$ varies dramatically. To avoid the disadvantage, we could use some other distribution of $x_i \sim p(x)$ instead of $x_i \sim U(a,b)$, to get the integration

$$I = \int_{x\in\mathcal{X}} h(x)\,\mathrm{d}x = \int_{x\in\mathcal{X}} \frac{h(x)}{p(x)} p(x)\,\mathrm{d}x = \mathbb{E}_{p(x)}\left[\frac{h(x)}{p(x)}\right] \tag{5.321}$$

Estimator use

$$\hat{I}_{g(x)} = \frac{1}{N}\sum_{i=1}^{N} \frac{h(x_i)}{g(x_i)}, \qquad w.r.t.\, x_i \sim g(x) \tag{5.322}$$

Variance

$$var(\hat{I}_{g(x)}) = \frac{1}{N} var\left(\frac{h(X)}{g(X)}\right) \tag{5.323}$$

i.e. if $\dfrac{h(x)}{g(x)} \approx$ const, the estimator can be more precise.

- An application of importance sampling: estimating expectation of function of r.v. $E_{f(z)}(\phi(z))$, where r.v. with $f(z)$ distribution is hard to generate. We can generate another random number series $x_i \sim q(x)$:

$$I(\phi) = \int \phi(z) f(z)\,\mathrm{d}z = \int \phi(x)\frac{f(x)}{q(x)} q(x)\,\mathrm{d}x = \mathbb{E}_{q(x)}\left(\phi(x)\frac{f(x)}{q(x)}\right) \tag{5.324}$$

$$= \int \phi(x) W(x) q(x)\,\mathrm{d}x, \quad W(x) \equiv \frac{f(x)}{q(x)} \tag{5.325}$$

Use Estimator:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)W(x_i) \tag{5.326}$$

As a worse case where we only have an unnormalized $\tilde{f}(x)$, with the normalize integration $f(x) = \dfrac{\tilde{f}(x)}{\int \tilde{f}(\xi)\,\mathrm{d}\xi} = \frac{1}{c}\tilde{f}(x)$ incomputable. Use property of weight $\tilde{W}(x) \equiv \dfrac{\tilde{f}(x)}{g(x)}$:

$$\int \tilde{W}(x)g(x)\,\mathrm{d}x = \int \tilde{f}(\xi)\,\mathrm{d}\xi = c \Rightarrow \hat{c} = \frac{1}{N}\sum_{i=1}^{N}\tilde{W}(x_i) \tag{5.327}$$

Estimator:

$$\hat{I} = \frac{\sum_{i=1}^{N}\phi(x_i)\tilde{W}(x_i)}{\sum_{i=1}^{N}\tilde{W}(x_i)}, \quad \tilde{W}(x_i) = \frac{\tilde{f}(x_i)}{g(x_i)} \tag{5.328}$$

- Effective Sample Size (Number of independent sample unit to get equivalent precision):

$$n_{\text{effect}} \equiv \frac{N}{\mathbb{E}\left[W(x)^2\right]} \approx \frac{N^2}{\sum_{i=1}^{N}W(x_i)^2} \tag{5.329}$$

- Importance Resampling: Importance sampling could be used to obtain random number series with distribution $f(x)$, but it is not efficient. Here's the process:

  First obtain $\tilde{x}_i \sim g(x)$ the proposal distribution, $i = 1, 2, \ldots, N$. then compute importance $\tilde{W}(\tilde{x}_i) = \dfrac{f(\tilde{x}_i)}{g(\tilde{x}_i)}$. By sampling from $\{\tilde{x}_i\}_{i=1}^{N}$ a relative small subset with probability weight $\tilde{W}(x_i)$, we can get a new random number series $\{x_j\}_{j=1}^{n}$ with distribution $x \sim f(x)$.

Comment: Idea of importance sampling estimation is to put more point at where $h(x)$ has large function value to get better fit of integration, i.e. smaller variance.

5.6.4 Bootstrap

In statistic inference for distribution $x \sim f(x;\theta)$, $\theta \in \Theta$, we want to estimate some statistic ϕ by estimator $\hat{\phi}$, including e.g. mean $E(\hat{\phi})$, standard error SE $= \sqrt{var(\hat{\phi})}$. In section 2.3 ~ page 52 we used pivot variable method to estimate statistics: parametric method, model required. Difficluty: strange distribution/strange statistics $\rightarrow$ use non-parametric method, e.g. bootstrap method.

□ **Bootstrap Method**

Conduct bootstrap given sample $X = (X_1, X_2, \ldots, X_N)$, X_i i.i.d. $\sim f(x;\theta)$.

1. Use sample X to estimate population distribution as $\hat{f}(x)$. e.g. empirical CDF.
2. Repeatedly sample from $\hat{f}(x)$ to get B samples of size n:

$$X^{(b)} = (X_1^{(b)}, X_2^{(b)}, \ldots, X_n^{(b)}), \quad b = 1, 2, \ldots, B \tag{5.330}$$

3. For each sample $X^{(b)}$ estimate a statistic $\hat{\phi}^{(b)}$

4. $\{\hat{\phi}^{(b)}\}, b = 1, 2, \ldots, B$ is the distribution estimation of $\hat{\phi}$ based on $\hat{f}(x)$, i.e. sample of statistics. We could use this sample of statistics to estimate e.g. SE$(\hat{\phi})$, or get interval estimation of $\hat{\phi}$.

$$\hat{\phi}_{\text{boot}} = \frac{1}{B}\sum_{b=1}^{B}\hat{\phi}^{(b)} \tag{5.331}$$

□ **Bias Correction**

The above estimator is the unbiased estimator for $\hat{\phi}$. However in the sense of minimizing MSE, usually $\tilde{\phi} \equiv \hat{\phi} - \hat{bias}(\phi)$ is a better estimator. Bias $b = \hat{\phi} - \phi$ can be estimated as

$$\hat{b} = \hat{\phi}_{\text{boot}} - \hat{\phi} \tag{5.332}$$

where $\hat{\phi}$ is calculated by using the original sample X. And MSE estimator is:

$$\tilde{\phi} = \hat{\phi} - \hat{b} = 2\hat{\phi} - \hat{\phi}_{\text{boot}} = 2\hat{\phi} - \frac{1}{B}\sum_{b=1}^{B}\hat{\phi}^{(b)} \tag{5.333}$$

Chapter. VI　数据科学导论部分

Instructor: Sheng Yu

This section contains basic data acquisition, data cleaning, data processing, date visualization methods. Details for data analysis are not covered.

□ **Road to Data Scientist**

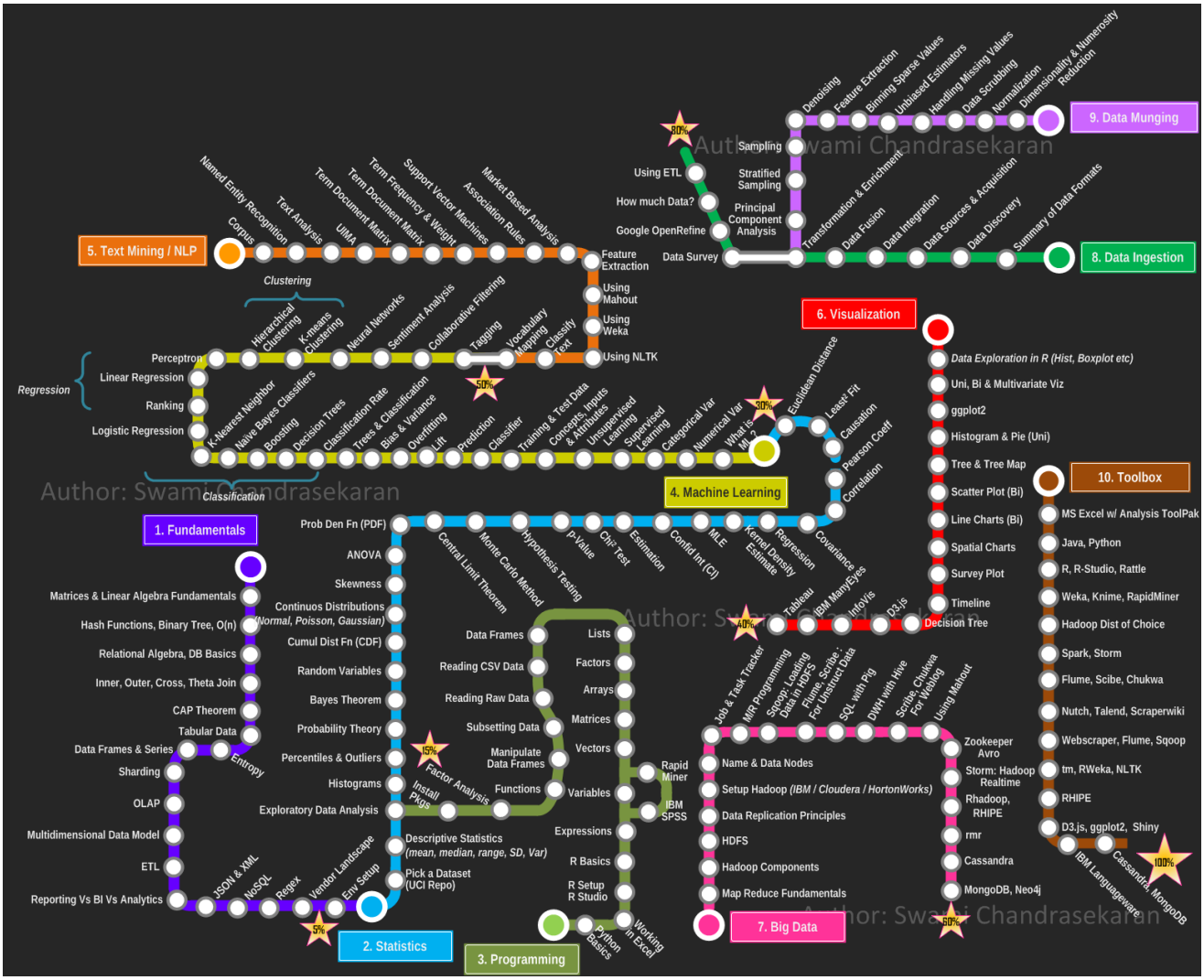


图 6.1: Road to Data Scientist, http://nirvacana.com/thoughts/2013/07/08/becoming-a-data-scientist/. More tutorial see https://github.com/MrMimic/data-scientist-roadmap

Comparison of R, python:focus on different aspects of 'Statistics':

- Differnece in programming philosophy: R for data analysis and python for data processing
- Difference in operating domain: R for statistical programming while python for general programming.

Section 6.1 Basic R. Manipulation

6.1.1 Installation and Maintenance of R.

☐ **Installing and Updating**

R.: update by delete old version and install new version.

- In CRAN (The Comprehensive R Archive Network): https://cran.r-project.org
- In Mirror@TUNA: https://mirrors.tuna.tsinghua.edu.cn/CRAN

RStudio: https://www.rstudio.com

☐ **Running R. command :**

- In R. GUI;
- In R. command line terminal;
- R. CMD BATCH;
- Rscript;
  - Use > to redirect output(overwrite);
  - Use >> to append output.

☐ **R. package library: packages are collection of R. functions (as well as test data and sample code).**

- .libPaths() show package library location[1] ;
- library('PACKAGE_NAME1','PACKAGE_NAME2',...) load packages.
- install.packages('PACKAGE_NAME1','PACKAGE_NAME2',...) install package from CRAN/mirrors;
- installed.packages() show all installed packages;
- updata.packages(checkBuilt = TRUE, ask = FALSE) update installed packages;

☐ **Working directory manipulation:**

- getwd() get current working directory;
- setwd('TARGET_PATH') set working directory (as an existing path).
- dir() show current directory.

[1]Unlike in C or python where . is an operator, . in R. is just a common character, without special meaning.

This feature can be used in naming self-defined functions: use .FUN_NAME1 for within-project function while FUN_NAME2 for external interface.

□ **Recommended R. Project Organization : working directory organized like**

- `data/` folder for structured original dataset;
- `result/` folder for output result;
- `presentation/` folder for result representing slides/reports/etc.;
- `.r` project file $\times n$.

□ **Looking for Help/Example of function:**

- `?FUN_NAME();`
- `help('FUN_NAME');`

6.1.2 Data Structure and Basic Manipulation in R.

□ **Atomic Classes**

- Character: `'abc'`;
- Integer: `3L`;
- Numeric: `2.4`;
- Logical: `TRUE,FALSE,T,F`;
- Special types: `NA`, `NaN`, `NULL`, `Inf`

□ **Operators**

- Numerical Operators: `+`, `-`, `*`(multiply by column), `/`, `%*%`(matrix multiply), `^`, `%%`(remainder operate);
- Logical Operators: `==`,etc.; `&` and `|` for common operator, `&&` and `||` for comparing the first element;
- Round a numeric:
  - `as.integer()`, round towards 0
  - `trunc()`
  - `ceiling()`
  - `floor()`
  - `round(NUMBER_TO_ROUND,digits = DIGITS)`

□ **Type Conversion**

- First need to meet the need of

Key Criterion: when converting mixed type in to the same type, use the type with more compatibility.

- Logical $\rightarrow$ Numeric:

□ **Data Structure**

- **Atomic Vector** : Column vector is the **basic** data structure in R. (scalar is length=1 vector).

Only data of the same class can be held in one vector.

Initialization:

- Ordinary way:
  - `c(1,2,3)`, `c(T,FALSE,TRUE)`, `c('a',NA,'b')`
  - `vector(mode = MODE,length = LENGTH)`
  - `logical(LENGTH)` return `FALSE` vector

  where `c()` for 'combine';

  `c()` combines all 'vector-like objects' into one vector, e.g. `c(c(1,2,3),c(1,2))>> c(1,2,3,1,2)`.
- Sequence vector:
  - `1:3.5>> c(1,2,3)`, `3:1 >> c(3,2,1)`
  - `seq(from, to , by, length.out)`, `length.out` for total vector length;
  - `rep(SEQ_TO_REP, times, lenght.out, each)`, used in k-fold cross validation labelling.

Operations:

- between vectors of different length `SHORT` and `LONG`: First `SHORT <- rep(SHORT, length.out=length(LONG))`. Then operate `SHORT` and `LONG`.

  e.g. `c(1,2)+ c(1,2,3)>> c(1,2,1)+ c(1,2,3)>> c(2,4,4)`
- Element access: `a[i]`, `i` starts from `1`

> **Vectorized Operation**: All operation in `R.` are based on vector, and vectorized operation is Parallel Arithmetic, which is **much faster** than loop such as `for`
>
> **Δ** Consider using vectorized opertion when writing code for **Speed**! Detail see section 6.1.4 ~ page 200.

- **Factor** : A special kind of 'vector' in `R.`, used to label discrete categorical data.[2]

  Initialization:

  `factor(FACTOR_SEQ, levels = FACTOR_LEVEL, labels = ...)`, `FACTOR_LEVEL` is the 'rank' of each factor, `labels` is the 'tag' of levels.

  A quick way to factorize a numeric vector `x` by interval division:

  `cut_number(x, NUM_OF_LEVELS)`

- **Matrix** : Only data of the same class can be held in one matrix.

  Initilaization: `matrix(DATA_SEQ, nrow, ncol, byrow = FALSE, dimnames = NULL)`. Default `byrow = FALSE` because matrix data is stored as combination of column vectors.

  If `length(DATA_SEQ)< nrow*ncol`, then `DATA_SEQ` is repeated with `length.out=nrow*ncol`.

  Operation:

[2] Factor vector is stored as integer vector.

  - Common operators `+-*/^` etc. operate in column-by-column mode (vectorized operation).
  - Binding matrix: `cbind` for `[A,B]` and `rbind` for `[A;B]`
  - Transpose: `t()`
  - Matrix multiplication: `%*%`
  - Inverse matrix: `solve()` (The essence of inversion is solving linear equations)
  - Diagonal matrix:
    * `diag(VECTOR)` returns a matrix diag{VECTOR}
    * `diag(MATRIX)` returns the diagonal element vector
  - Element access: `a[i,j]`, `a$OBJECT_NAME`
  - Dimension: `dim()`, `nrow()`, `ncol()`
  - Rank: `qr(MATRIX)$rank`

- **List** : A pack containing various datatype, generally also a kind of vector(but not atomic vector)

  Initialization: `list(OBJECT1,OBJECT2,...)`

  Element access: `a[[i]]`, `a$OBJ_NAME`

- `data.frame`: 'Mixture' of matrix and list. `data.frame` is actually a kind of list(with some constraint), organized in the shape of matrix (but allowing different datatype for different columns, each column is a list object).

  Each column of `data.frame` has name: `names(DATA_FRAME)`, `colnames(DATA_FRAME)`

  Element access: `a[i,j]`, `a[[i]]`, `a$COL_NAME`

□ **Data Read & Write**

- Common R&W: `read.` / `write.`
  - `read.table(FILE_NAME,header = FALSE,sep,colClasses,stringAsFactors = FALSE)`
  ⋆ `read.csv()` basically the same as `read.table`
  ⋆ `write.table(DF,FILE_NAME,sep,row.names=FALSE)`
  - `readxl::read_xlsx(FILE_NAME,sheet = SHEET_NUM,range = 'RANGE')`

  Some relative arguments:

  - `quote="'"`, use `'` to quote/identify string, set `quote=''` to avoid misread strings such as 'Levene's Test'
  - `encoding='UTF-8'`, char encoding system, used especially for dataset containing CJK char.
  - `nrows=LINE_NUM` read first `LINE_NUM` lines

- Large Data Read & Write:
  - preset `colClasses`

```r
temp.dat <- read.table(FILE_NAME, nrows = 100)
classes <- sapply(temp.dat, class)
dat <- read.table(FILE_NAME, colClasses = classes)
```

  - `readr::read_delim(FILE_NAME,delim=SEP)` can speed up

- Text Write: `sink(FILE_NAME,append=FALSE)`, write output into a file, the same as > in terminal.
- `.RData` Binary Format Read & Write: RW in `.RData` format, fast to load.
  - `save(DF,file = FILENAME)`
  - `load(FILE_NAME)`

6.1.3 Functions and Control Flow

☐ **Program Speed:**

`system.time({COMMAND})`

☐ **Function Call**

- `FUN_NAME(ARGUs)`
- `do.call('FUN_NAME',LIST_OF_ARGUs)`, look for a function naming `FUN_NAME` in `R.` and call.
- `a % NEW_OPRTR % b` to call self-defined binary operator.
- `'*'` etc. used in `apply(FUN = '*')`
- `R.` allows auto-completion to `ARGUs`, e.g. `rep(0,length.out = 10)` » `rep(0,length = 10)`

☐ **Function Definition**

▷ `R.` **Code**

Basic function definition in `R.`

```r
FUNC_NAME <- function(ARG1 = ARG1_VALUE , ARG2, ...) {
    FUNCTION_BODY
}
```

More key elements in `funtion{}`

- `return(RETURN_OBJ)` at the end of function, without `return()`, output the last line
- `stopifnot(COND1,COND2,...)` at the beginning of function, used to test `ARG` class
- `stop(ERROR_MESG)` output error message
- `...` as a special argument
  - Pass `...` to another func in this function
  - Handle arbitrary number of input

- Function can be defined within function
- Function is a kind of variable $\longrightarrow$ used in `apply`, `sapply` etc. for vectorized programming.
- Anonymous function: used in `sapply(X,FUN=function(){STATs})` for quick definition
- `FUNC_NAME` can be used for new-defined binary operator as `'%NEW_OPRTR%'<- function()`

□ **Flow Control**

- `if` and `else if`, example:

```
if(COND1) {
    STATEMENT
} else if(COND2) {
    STATEMENT
} else {
    STATEMENT
}
```

- `ifelse(COND,IF_YES_STAT,IF_NO_STAT)` a vectorized version of `for + if else`.
- `for`: Loop in `R.` is **Extremely Slow**, avoid loop, use **vectorized operation**.

```
for(VAR in SEQ) {
    STATEMENT
}
```

- `switch(TEST_EXPR,CASE1= RETN1,CASE2= RETN2,...)`

6.1.4 Vectorized Operation

- `apply()` function series:
  - `apply(MAT,MARGIN,FUN)` for matrix apply, `MARGIN=1` for each row, `2` for each column

    Example:

```
apply(matrix(c(1,2,3,4),2,2), 1, sum) >> c(4,6)
apply(matrix(c(1,2,3,4),2,2), 2, sum) >> c(3,7)
```

  - `lapply(LIST,FUN)` for list/`data.frame`, apply `FUN` on each list elements,`list` returned
  - ⋆ `sapply(X,FUN)` for list/`data.frame` apply+simplify, `vector`/`matrix`/`list` returned
  - `tapply(X,INDEX,FUN)`: for each index, use `FUN` respectively.
  - `mapply(FUN,ARGU_OF_FUN)`, use argument name to label `ARGU_OF_FUN`, or causes bad readability.

    Example:

```
mapply(function(x,y,z,k){(x+k)^(y+z)} , x = ,y = ,z = ,k = )
```

- `Vfunc <- Vectorize(FUNC_NAME)`: define vectorize version of function.
- `with()` and `within()`:
  - `with(DF,aggregate(PART,by,FUN))`
  - `with(DF,STATE),within(DF,STATE)`, `within` allows new column append
- `outer(VEC1,VEC2,FUN)`: A Two-variate extension of `mapply()`, output wedge of two vectors.
- `ifelse(COND,YES_STAT,NO_STAT)`, vectorization supported.

6.1.5 Subsetting

- By position: `x[RANGE]`
  - `x[4]`
  - `x[-4]`: `x` without the 4^{th} item (which is different from `python`, where selects the reciprocal 4^{th} element).
  - `x[2:4]`
  - `x[c(1,2,5)]`
- By name: `x['COL_NAMEs']`, `x['COL_NAME1':'COL_NAME2']`
- By condition: basically, `x[LOGI_VEC]`
  - `x[x==10]`
  - `x[x %in% c(1,3,4)]`, linear search, not based on hash algorithm[3].

  Usually used for conditional selection of `data.frame`
- Subsetting for `data.frame` and list: `x[[RANGE]]`

  Simplified / Preserved subsetting: whether preserved datatype, e.g. df → df (preserved) v.s. df → vector (simplified).

DataType	Simplified	Preserved
vector	`x[[1]]` / `x[1]`	
list	`x[[1]]`	`x[1]`
factor	`x[1:4,drop=T]`	`x[1:4]`
matrix	`x[,1]`	`x[,1,drop=F]`
`data.frame`	`x[,1]`,`x[[1]]`	`x[,1,drop=F]`, `x[1]`

表 6.1: Simplified/Preserved subsetting

[3] If really needed, use `env()` to reset environment.

- Other subsetting:
  - `%in%`
  - `unique()`, return with each element appears only one times
  - `duplicated()`, `TRUE` when appear the $n > 1$ times
  - `which(x==4)`, return position of matched element
  - `which.min()`, `which.max`, `min()`, `max()`
  - `grep(REGEX,X,value)`, search for elements with `REGEX` pattern: `value=F` returns position, `value=T` returns elements, `grepl(REGEX,X)` returns logical vector
  - `match(TO_BE_MATCHED,TARGET)`, returns the index of elements of `TO_BE_MATCHED` in `TARGET`

    ▷ `R`. **Code**

    Example:

```
vec1 <- c('a','a','b','b','d','d','b')
vec2 <- c('d','a','b')
match(vec1,vec2)
> [1] 2 2 3 3 1 1 3
```

  - `subset(X,...)`, `...` a series of select criterion. **not** allowed: `subset(X,...)<-`
- Use subsetting to sample: `DATA[sample(1:nrow(DATA),NUM_OF_SAMPLE,replace),]`, `replace=T` for with replacement

6.1.6 Data Manipulation With dplyr. And tidyr.

`dplyr` and `tidyr` are two useful package for data cleaning & manipulation. Use package `tidyverse` include both of them.

`tidyverse` for `tidyuniverse`, includes `dplyr`, `tidyr`, `readr`, `ggplot2`, `stringr`, etc.

☐ **`%>%`: pipe in `tidyverse`, so that functions in `tidyverse` with format `FUNC(DF,...)` can pass on `DF` results along the pipeline.**

Some examples see section 8.1.5 ~ page 239.

☐ `dplyr` **Package.**

- Cheet Sheet: https://nyu-cdsc.github.io/learningr/assets/data-transformation.pdf
- `select(DF,...)`, where `...` can use column index/name range as in subsetting, or some helper function for advanced subsetting:
  - matching position:
    - `everything()`
    - `last_col()`
  - matching column name:

    * `start_with('PATTERN')`, `end_with('PATTERN')`, `contains('PATTERN')`
    * `match('REGEX')`, column name with `REGEX` pattern
    * `num_range('x',1:4)` delect column name `c('x1','x2','x3','x4')`
    * `any_of(CHR_VEC)` select column from `CHR_VEC`
  - `where(FUN)`, select those `FUN(COL_NAME)` returns `TRUE`

- `filter(DATA,CONDs)`, select elements with `CONDs` conditions
- `arrange(DATA,COL)`, sort by `COL`, `arrange(DATA,desc(COL))` for descending order
- `mutate(DATA,...)`, append new columns according to `...` definition; `transmute()` drops original columns.

  `...` definition can use advanced window function:

  - `lead(COL)`,`lag(COL)`, e.g. `lead(COL)[i]=COL[i+1]`, can use `...=COL-lead(COL)` for differnetial
  - `dense_rank(COL)`, `percent_rank(COL)` rank number
  - `ntile(COL,N)` break into `N` groups labeling `1:N`
  - `cume_dist(COL)`, `cummean(COL)`, `cumsum(COL)`, `cummax(COL)`, `cummin(COL)`, etc. cumulative value

- `summarise(data,...)`, `...` for summarise function.
- Row selection:

  - `slice(DF,ROW_RANGE)`
  - `distinct(DF)` remove duplicated rows
  - `sample_frac(DF,FRAC,replace)`, sample `FRAC` fraction from `DF`
  - `sample_n(DF,N,replace)`, sample `N` cases from `DF`
  - `top_n(DF,AMOUNT,RANK_COL)` select `AMOUNT` top ranking by `RANK_COL` cases

- Data combining see slides.

☐ `tidyr` **Package**

- Cheet Sheet: https://leadousset.github.io/intro-to-R/cheatsheet_tidy.pdf
- `gather(DF,key='KEY_NAME',value='VALUE_NAME',...,na.rm)`, melt a `data.frame`.

e.g. `gather(df,'KEY','VALUE',c('COL1','COL2','COL3'))` transfers ... as:

$$\begin{array}{cccc} \text{ID} & \text{COL1} & \text{COL2} & \text{COL3} \\ 1 & a_1 & b_1 & c_1 \\ 2 & a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots & \vdots \end{array} \quad \rightarrow \quad \begin{array}{cc|c} \text{ID} & \text{KEY} & \text{VALUE} \\ 1 & \text{COL1} & a_1 \\ 2 & \text{COL1} & a_2 \\ \vdots & \vdots & \vdots \\ 1 & \text{COL2} & b_1 \\ 2 & \text{COL2} & b_2 \\ \vdots & \vdots & \\ 1 & \text{COL3} & c_1 \\ 2 & \text{COL3} & c_2 \\ \vdots & \vdots & \end{array} \tag{6.1}$$

- `spread(DF,key='KEY_NAME',value='VALUE_NAME')`, inverse of `gather()`
- `separate(DF,COL,into=SET_VEC,sep='REGEX')`, separate `COL` into columns with name in `SET_VEC`, sep according to `sep`
- `unite(DF,COL,SET_VEC,sep='')` inverse of `separate()`

Section 6.2 Text Processing & Text Mining

- Data cleaning
- Data manipulation
- Information extraction: mode identifying/relation extraction
- Text mining: anaylzing token distribution, ignore word order
- NLP: concept identifying based on sentence; untimate goal: 'understand' sentence meaning.

Tools for Text processing:

- `R.`: suitable for easy task
- `python.`: best
- `java`: strong, but not suitable for deep learning
- `c++`: fast, inadequate package
- `Notepad++` / `Vim` / `VSCode`, etc.

6.2.1 Basic Text Manipulation With stringr.

□ `R. base` & `stringr` **package:**

The prior one is used more often

- Cheet Sheet: http://edrub.in/CheatSheets/cheatSheetStringr.pdf
- `str_length(STRING)`, `nchar(STRING)`
- `paste(...,collapse=NULL)`,`str_c(...)`, both are vectorized operation

  Argument:

  - `sep`: sep between each `...` corresponding elements, with `collapse=NULL`, return a char vector
  - `collapse`: sep when combining `collapse=NULL` vector elements, `NULL` for not combining
  - Special character: `\t` tab, `\r` & `\n` & `\r\n` new line, `\xad` '-' at end on line for word-connecting
- `str_split(STRING,pattern='REGEX')`/`strsplit()`, split string at `REGEX` pattern fitted, list returned
- `str_sub(STRING,start,end)`, `substr()`. The `start` char to `end` char of string, use negative index as in python.

  Can be used to replace: `str_sub(...)<- REP_STR`
- `str_locate_all('STRING',pattern='REGEX')`/`str_match_all('STRING',pattern='REGEX')`

  `grep(pattern='REGEX',x='STRING',value=T)`, search for elements with `REGEX` pattern: `str_locate_all()` or `value=F` returns position, `str_match_all()` or `value=T` returns elements.
- `str_replace_all('STRING',pattern='REGEX',replacement='REP')` `grepl(REGEX,X)` returns logical vector, include or not. `str_extract_all('STRING',pattern='REGEX')`
- `gsub(pattern='REGEX',replacement='REP',x='STRING')`, replace `REGEX` field with `REP`
- `str_trim(...,side = )`, trim extra white space at `side='both'/'left'/'right'`

6.2.2 Regular Expression

Regular expression is a text pattern/mode. abbr. regex/regexp. Regex is supported in most common language, same syntax used.

Tutorial: https://www.runoob.com/regexp/regexp-tutorial.html

□ Key Elements

- Literal: common char, e.g. `a`. Include most char on keyboard. Upper/Lower case sensitive.
- Metacharacters: `\^$.|?*+()[]{}`, use e.g. `\.` to escape meaning.

  Note: when typing regex in programming language, sometimes use `\\.`: `\\.` $\xrightarrow{\text{language interpreter}}$ `\.` $\xrightarrow{\text{regex interpreter}}$ identifying `.`
- Character Class: `[]`, identify one of elements in `[]`. `^` within `[]` for ∁.
  - e.g. `gr[ae]y` identifies `grey` and `gray`.
  - e.g. `[0-9]` numbers, `[a-zA-z]` letter
  - e.g. `q[^x]` matches `question`, not matchs `qxestion`, not matches `Iraq`

character class shorthand

ShortHand	Meaning	Equivalant REGEX
\d	numeric digit	[0-9]
\D	Not numeric digit	[^\d]
\w	a word character	[a-zA_Z0-9_]
\s	white space	[\t\r\n\f]

- Wildcard（通配符）：. matches any single character except line break \r,\n
- Anchor（词边界/定位符）：match 'word boundary' (not the space at the start/end of string).

  ^ string start, $ string end, \b word boundary, \B not-a-word-boundary position
- Repetition/Quantifier: here X for some regex pattern like CHAR, [] etc.

Greedy	⋆ Reluctant	Possessive	Freq of Occurrence
X?	X??	X?+	$0, 1$
X+	X+?	X++	≥ 1
X*	X*?	X*+	$0, > 1$
X{n}	X{n}?	X{n}+	n
X{n,}	X{n,}?	X{n,}+	$\geq n$
X{n,m}	X{n,m}?	X{n,m}+	$[n, m]$

Example: Search 'foo' in 'xfooxxxxxxfoo':

  – Greedy: 'xfooxxxxxxfoo' found at index 0-13

  ⋆ Reluctant: 'xfoo' found at index 1-4, 'xxxxxxfoo' found at index 4-13

  – Possessive: no match found (not usually used)

Example: regex match 'aaaa'

  –

- Alternation & Grouping & Back Reference: XA|XB identify XA or XB, use grouping () to set boundary of XA,XB.

  Use \n for back reference the n^{th} group. ▷ R. **Code**

  Example: search for immediate repeat word in a sentence

```
(\b[a-zA-Z]+\b) \1
```

- Lookaround:

  - LookAhead: `(?<=X)q`
  - LookBehind: `q(?=X)`

6.2.3 Web Scraping

Basic elements of web page:

- HTML (HyperText Markup Language): structure and content of page
- CSS (Cascading Style Sheet): page style.
- JavaScript: functionality, interaction

Basic `html` document format:

▷ `R`. **Code**

```
<!DOCTYPE html> # an html document
<html> # html page begin
<head> # head elements declare
<meta charset="utf-8">
<title> TITLE OF WEB PAGE </title>
</head>
<body> # html body begin

<h1> HEADING 1 </h1>
<p class='TEST_TEXT'> PARAGRAPH 1 </p>

</body>
</html>
```

We can use elements like `<p>` or `class` to extract page information.

□ **Web Scraping with** `rvest`.

- `pge <- read_html('URL')`: page read

  Proxy set: `Sys.setenv(https_proxy='http://127.0.0.1:7890')`

- `pge %>% html_elements(css='.CSS_CLASS_NAME')%>% html_text()`: basic scraping. use SelectGadget tool for help finding proper css label.

Section 6.3 Graphic in R.

6.3.1 R::base Plotting

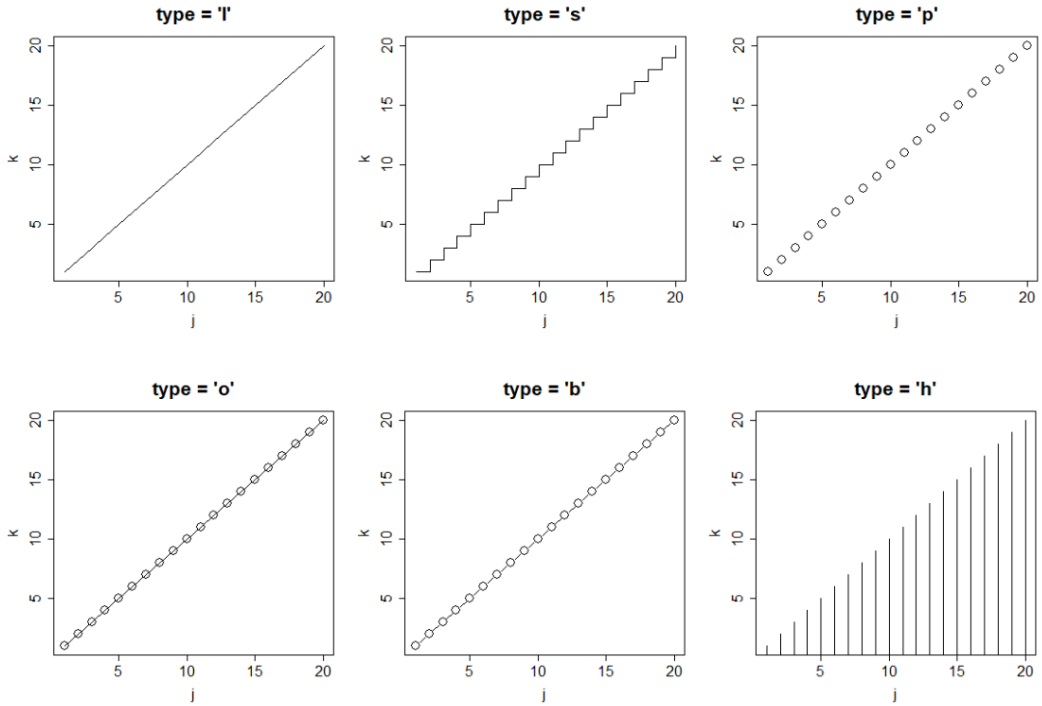
Plot function in `R::base`:

```
plot(X,Y) # scatter/line plot of Y-X
plot(FUNC_OBJ, from = , to = ) # function plot ranging in c(from, to
    )
plot(FACTOR) # barplot of factors
plot(FACTOR, Y) # boxplot of numeric v.s. levels of factor
plot(DATA.FRAME) # correlation plot
plot(ANY_PLOTTABLE_OBJ) # plot any plottable object
```

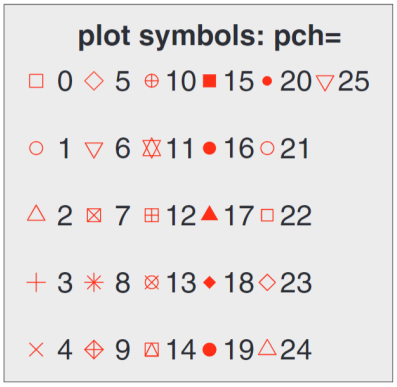
- Plot saving: first open a plotting device, then make plot and close the device

```
pdf("PLOT_FILE_NAME.pdf", FIG_HEIGHT, FIG_WIDTH)
plot(PLOT_PARAM)
dev.off()
```

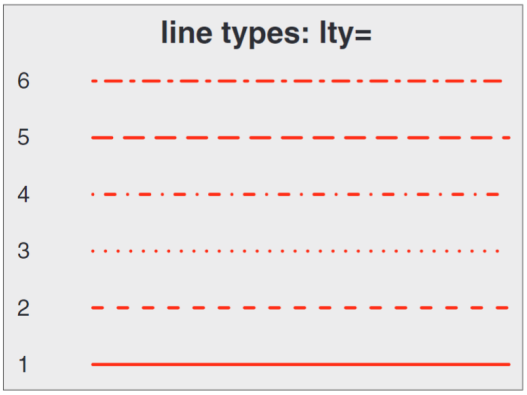
- `plot()` plotting parameters:
  - `main =` string for title; or use `title('TITLE')` as the next command
  - `sub =` string for subtitle;
  - `xlab = , ylab =` string axis labels;
  - adding LaTeXexpression as text: use `main = expression(PLOTMATH_EXPRESSION)`, use `?plotmath` to look for possible symbols
  - `xlim = , ylim =` axis range, e.g. use `xlim = c(0,100)`
  - `type =` value taken in `c('p', 'l', 'b', 'o', 's', 'h')` for plot **type**



- `pch =` **p**oint **ch**aracter, value taken in `0:25` for defulat point charaters listed below, or use (vector of) charater to specify, e.g. `pch = c(' ')`



- `lty =` **l**ine **ty**pe, value taken in `1:6` (0 for not shown)



- `cex =` **c**haracter **ex**pansion, relative size with 1 as baseline and default.

  Some derivative function to control size of other plotting elements:

  * `cex.axis =` relative size of axis node text
  * `cex.lab =` relative size of labels
  * `cex.main =` relative size of title
  * `cex.sub =` relative size of subtitle

- `lwd =` **l**ine **wid**th, relative width of line with 1 as baseline and default
- `col =` **color** of elements in plot, value examples for color white:

  * Index: `col = 1` predefined color in `R`.
  * Color name: `col = 'white'`, use `colors()` to see all available color names
  * Hexadecimal code: `col = '#FFFFFF'`
  * RGB code: `col = rgb(1,1,1)`, `col = rgb(255,255,255, maxColorValue = 255)`
  * HSV code: `col = hsv(0,0,1)`

  `col =` can accept vector for various colors, or acccept some function for continuous colors:

  * Discrete color: `col = c('red','blue')`, or use `col = df$GROUP` to color different groups

    * Continuous color function: `rainbow(NUM_OF_COLORS)`, `heat_colors()`, `terrain.colors()`, `topo.colors()`, `cm.colors()`

  Some derivative function to control color of other plotting elements:

    * `col.axis =` color of axis node text
    * `col.lab =` color of labels
    * `col.main =` color of title
    * `col.sub =` color of subtitle
    * `bg =` color of background

  – `font =` **font** used in plot, with 1 = plain, 2 = bold, 3 = italic, 4 = bold italic

  Some derivative function to control font of other plotting elements:

    * `font.axis =` font of axis node text
    * `font.lab =` font of labels
    * `font.main =` font of title
    * `font.sub =` font of subtitle
    * `ps =` baseline font **p**oint **s**ize, i.e. text size = `ps*cex`
    * `family =` extra text type, value taken in `c('serif', 'sans', 'mono')`m etc. use `names(pdfFonts())` to see possible font families

  – `bty =` **b**ox **ty**pe of the box surrounding the figure. Value taken in `c('o', '7', 'L', 'U', 'C', 'n')`

  – `las =` relation btw. **l**able **a**nd axi**s**. Value taken `c(0,1,2,3)`.

- `axis()` parameters for axis settings: after using `xaxt = 'n'` or `yaxt = 'n'` to remove correponding axis when executing `plot()`, other variation of axis could be made by using `axis()`

  – `axis(1)` for creating x aixs, `axis(2)` for creating y aixs. Here we would use x axis in the following parts.

  – `aixs(1, at = )` to specify ticks.

  – `plot(las = )` to specify rotation of ticks, value taken in `c('Parallel', 'Horizontal', 'Perpendicular', 'Vertical')`

  – `plot(xlim = c( , ), ylim = c( , ))` for axis limits

  – `plot(log = )` for log transfrom on axis, value taken in `c('x', 'y', 'xy')`.

- `legend()` parameters:

  – `x =` position of legend, value taken in `c("top", "bottom", "topleft", "topright", "bottomleft", "bottomright")`

  – `inset =`

  – other parameters are set following the setting in `plot`. An example:

```
legend("bottomright", legend = c("red", "green"), lty = c
    (2,4), lwd = 3, col = c("red", "green"))
```

- `text(X_COOR, Y_COOR, labels = TEXT)` parameters for adding text in figure. An application is `text(df$X, df$Y, labels = df$Z)` to label each point.
  - `pos =` **pos**ition of text around the coordinate point, value taken in `c(1,2,3,4)`
- `lines()` to put an extra line on existing figure (device). Parameters are similarly set as `plot()`
- `par()` to set global **par**ameters. An example to put 3 different figure in the same device:

```
opar <- par(no.readonly = TRUE) # copy original setting
par(mfrow = c(1,3))
plot()
plot()
plot()
par(opar)
```

□ **More Charts**

- `barplot(counts, horiz, besides, ...)` for bar plot. Data should be first prepared by `counts <- table(Y_TO_COUNT)`.
- `hist(x, breaks, freq, ...)` for histogram.
- `plot(density(df, kernel = ), ...)` for density plot.
- `boxplot(x, ...)` for box plot. use `boxplot(x ~ GROUP, data = , ...)` to plot grouped boxplot
- `dotchart(x, labels, groups, ...)` to compare x value for categories

6.3.2 R::ggplot2 Plotting

ggplot2: **G**rammar of **G**raphics **plot** (2nd edi). It provides a convenient way to produce fancy plots. Reference see https://ggplot2.tidyverse.org/reference/

Basic steps for ggplot2:

1. Specify data and arsthetic mapping
2. Adding 'layers' with geom_
3. Adding labels

An example:

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
    geom_point(pch=17, color="blue", size=2) +
```

```
    geom_smooth(method="lm", color="red", linetype=2) +
    labs(title="Automobile Data", x="Weight", y="Miles Per Gallon")
```

Elements in ggplot2:

- aes() to specify **aes**thetic mapping, e.g. aes(x = , y = , col = , ...). Used in ggplot() as global setting, in geom_() as local override (different geom_() may need different local settings). Examples:

```
aes(x = mpg ^ 2, y = wt / cyl, col = am)
#> Aesthetic mapping:
#> * x -> mpg^2
#> * y -> wt/cyl
#> * color -> am
```

- geom_ layer to specify statistical figure you want. Some useful plot:

geom_() Function	Charts	Options
geom_bar()	bar plot	color, fill, alpha
geom_boxplot()	box plot	color, fill, alpha, notch, width
geom_density()	density plot	color, fill, alpha, linetype
geom_histogram()	histogram	color, fill, alpha, linetype, binwidth
geom_hline()	horizontal line	color, alpha, linetype, size
geom_vline()	vertical line	color, alpha, linetype, size
geom_line()	line gragh	color, alpha, linetypem size
geom_point()	scatter plot	color, alpha, shape, size
geom_smooth()	fitted line	method, formula, color, fill, linetype, size
geom_violin()	violin plot	color, fill, alpha, linetype
geom_text()	text annotation	see functon help

- labs(title, x, y) to specify labels and title
- facet_grid() and facet_wrap() to plot multiple plot, with factor levels as categories, parameters:

  - `facets =` facet variable. For `facet_wrap()` use `~VAR1` (one variable); `facet_grid()` use `.~VAR1` or `VAR1~.` or `VAR1~VAR2` (allow two variable)
  - `nrow = , ncol =` grid shape
  - `shrink =` whether adjust ticks, set `TRUE` or `FALSE`
  - `drop =` whether drop levels with censored data, set `TRUE` or `FALSE`

- `theme()` to set fonts, backgrouds, gridlines, etc.

  There are some pre-defined theme: `theme_grey()`, `theme_bw()`, `theme_linedraw()`, `theme_light()`, `theme_dark()`, `theme_minimal()`, `theme_classic()`, `theme_void()`, `theme_test()`.

  Detailed elements in a plot is adjust by passing `element_()`:

  - `element_line()` set some line element
  - `element_rect()` set some rectangular element
  - `element_text()` set some text element

  Some useful command:

  - `plot.title = element_text(hjust = 0.5)` adjust position of title to mid. Other similar parameters: `plot.background`, `plot.title.position`, `plot.subtitle`, `plot.caption`, `plot.caption.position`, `plot.tag`, `plot.tag.position`, `plot.margin`
  - `panel.background = element_rect(fill = 'white', color = 'blue')` adjust figure background and border. Other similar parameters: `panel.grid.major/minor.x/y`
  - `aspect.ratio =` height:width
  - `legend.position = 'none'` to remove automatic legend

- `ggsave('FILE_NAME', PLOT, WID, HEI)`, or use `ggsave('FILE_NAME')` to save the active device.

Chapter. VII　可靠性数据与生存分析部分

Instructor: Jiangdian Wang

Key focus of reliability data and survival analysis: Study the 'survival time' T before some 'failure event'. Basically the research problem is the distribution of T, including topics on descriptive statistics, estimation and hypothesis testing. Further for actual cases, T might be censored, i.e. the observe time is not exact; and we may also wonder the influence of covariants z.

Section 7.1　Reliability Data

The main feature of reliability data is **censoring**, to be distinguished from the exact numbers in usual statistical inference. Censor means we cannot observe the exact **event time** T. Instead, a **censoring time** C is observed, together with a censoring type, e.g.

$$\text{Right Censoring: } T_{\text{actual}} > C \tag{7.1}$$

$$\text{Left Censoring: } T_{\text{actual}} < C \tag{7.2}$$

$$\text{Interval Censoring: } C_l < T_{\text{actual}} < C_r \tag{7.3}$$

$$\cdots \tag{7.4}$$

7.1.1　Right Censor Data and Representation

In most parts of this course we focus on right censor data, i.e. dataset contains both event time T and right censor time T^+:

$$\text{Event Time: } T_1, \ldots, T_{n_1} \tag{7.5}$$

$$\text{Right Censor: } T_1^+, \ldots, T_{n_r}^+ \tag{7.6}$$

$$\tag{7.7}$$

Or we could use an indicator δ to express whether a time is event (1) or right censored (0):

$$(T_i, \delta_i; z_i),\ i = 1, 2, \ldots n_1 + n_r \tag{7.8}$$

where z_i for covariants.

Usually we assume that event and censor are independent $T \perp\!\!\!\perp C$

7.1.2 Life Table Data

Life table collect survival data at ordinal, uniformly-spaced time points, where each row contains # items at risk, # events, ...

Section 7.2 Survival Model and Statistical Inference

7.2.1 Survival Function and Hazard

Key focus of survival analysis problem is the distribution of T (note that in actual cases we need to make use of both event time T_i and censored data T_i^+ to estimate the distribution of T). The distribution feature can be described in various approaches: PDF $f(t)$, CDF $F(t)$, Survival Function $S(t)$, Hazard Function $\lambda(t)$, Cumulative Hazard Function $\Lambda(t)$:

- Continuous Case: $t \in \mathbb{R}^+$
  - Survival Function $S(t)$:

$$S(t) \equiv 1 - F(t) = \int_t^\infty f(\tau)\,\mathrm{d}\tau, \qquad f(t) = -\frac{\mathrm{d}S(t)}{\mathrm{d}t} \tag{7.9}$$

  - Hazard Function $\lambda(t)$ (or in some materials denoted $h(t)$): mortality at t:

$$\lambda(t) = \lim_{h\to 0} \frac{\mathbb{P}[t \le T < t+h | T \ge t]}{h} = \frac{f(t)}{S(t)} = -\frac{\mathrm{d}\log S(t)}{\mathrm{d}t} \tag{7.10}$$

  - Cumulative Hazard Function $\Lambda(t)$ (or in some materials denoted $H(t)$):

$$\Lambda(t) = \int_0^t \lambda(\tau)\,\mathrm{d}\tau = -\log S(t) \tag{7.11}$$

$$S(t) = e^{-\Lambda(t)} = e^{-\int_0^t \lambda(\tau)\,\mathrm{d}\tau} \tag{7.12}$$

- Discrete Case: $t \in \{t_1, t_2, \ldots, t_n\}$
  - PMF: $p(t)$ is defined on

$$t \in \mathcal{T}, \quad p(t) \in (\mathcal{T} \to [0,1]^n) \tag{7.13}$$

  - Survival Function: Note that CDF $F(t)$ is right continuous, then $S(t) = 1 - F(t)$ is left continuous:

$$S(t) = \mathbb{P}(T > t) = \sum_{t_i > t} p(t_i), \qquad p(t_i) = S(t_{i-1}) - S(t_i) = \lambda(t_i) S(t_{i-1}) \tag{7.14}$$

Decomposition of survival function into hazard production

$$\begin{aligned} S(t) =& \mathbb{P}(T > t) = P(T > t \cap T > t_j), \quad \forall t_j < t \\ =& \mathbb{P}(T > t | T > t_j) \cdot \mathbb{P}(T > t_j) \\ =& \mathbb{P}(T > t | T > t_j) \cdot \mathbb{P}(T > t_j | T > t_{j-1}) \cdot \mathbb{P}(T > t_{j-1}) \\ =& \mathbb{P}(T > t | T > t_j) \cdot \frac{S(t_j)}{S(t_{j-1})} \cdot \mathbb{P}(T > t_{j-1}) \\ =& \prod_{0<t_j \le t} \frac{S(t_j)}{S(t_{j-1})} \\ =& \prod_{0<t_j \le t} [1 - \lambda(t_j)] \end{aligned} \tag{7.15}$$

– Hazard Function $\lambda(t)$:

$$\lambda(t_i) = \mathbb{P}(T = t_i | T \ge t_i) = \frac{p(t_i)}{S(t_{i-1})} = 1 - \frac{S(t_i)}{S(t_{i-1})} \tag{7.16}$$

□ **Properties of survival function and hazard function & More concepts and definition**

- Mean Survival Time:

$$\mu \equiv \mathbb{E}(T) = \begin{cases} \int_0^\infty \tau f(\tau)\, \mathrm{d}\tau = \int_0^\infty S(\tau)\, \mathrm{d}\tau \\ \sum_{i=1}^n t_i p(t_i) \end{cases} \tag{7.17}$$

- Mean Residual Life Time (mrl):

$$\mathrm{mrl}(t) = \mathbb{E}[T - t | T \ge t_0] = \frac{\int_t^\infty S(\tau)\, \mathrm{d}\tau}{S(t)} \tag{7.18}$$

- Considering that $T > 0$ and $\lim_{t\to\infty} F(t) \to 0$, $S(t)$ has following properties

$$S(0) = 1 \qquad S(\infty) = 0 \tag{7.19}$$

- For independent survival time $T_1, , T_2$, define $T = \min\{T_1, T_2\}$, then

$$\lambda_T(t) = \lambda_1(t) + \lambda_2(t) \tag{7.20}$$

- Hazard Rate: for two survival r.v. T_1, T_2, the hazard rate at t

$$\text{hazard ratio}(t) = \frac{\lambda_1(t)}{\lambda_2(t)} \tag{7.21}$$

7.2.2 Parametric Statistical Inference to Survival Function

Usually the parametric inference is based on a hypothetical distribution, then we conduct estimation using the parametric distribution, or conduct hypothesis testing on parameter(s).

□ **Common Survival Distribution Prior**

In parametric model, there are some commonly used distribution models

- Exponential $T \sim \varepsilon(\lambda)$

$$f(t) = \lambda e^{-\lambda t} \tag{7.22}$$
$$F(t) = 1 - e^{-\lambda t} \tag{7.23}$$
$$S(t) = e^{-\lambda t} \tag{7.24}$$
$$\lambda(t) = -\frac{\mathrm{d}\log S(t)}{\mathrm{d}t} = \lambda \tag{7.25}$$
$$H(t) = \lambda t \tag{7.26}$$
$$\mathbb{E}(T) = \frac{1}{\lambda} \tag{7.27}$$
$$var(T) = \frac{1}{\lambda^2} \tag{7.28}$$

- Weibull $T \sim W(p, \lambda) = [\varepsilon(\lambda^p)]^{1/p}$, degrade to exponential $\varepsilon(\lambda)$ when $p = 1$[1]

$$f(t) = p\lambda^p t^{p-1} e^{-(\lambda t)^p} \tag{7.29}$$
$$F(t) = 1 - e^{-(\lambda t)^p} \tag{7.30}$$
$$S(t) = e^{-(\lambda t)^p} \tag{7.31}$$
$$\lambda(t) = p\lambda^p t^{p-1} \tag{7.32}$$
$$H(t) = (\lambda t)^p \tag{7.33}$$
$$\mathbb{E}(T) = \frac{1}{\lambda}\Gamma(1 + \frac{1}{p}) \tag{7.34}$$
$$var(T) = \frac{1}{\lambda^2}\left[\Gamma(1 + \frac{2}{p}) - (\Gamma(1 + \frac{1}{p}))^2\right] \tag{7.35}$$
$$t_{0.5} = \left[\frac{\log 2}{\lambda^p}\right]^{1/p} \tag{7.36}$$

- Gamma $T \sim \Gamma(\alpha, \lambda)$. Degrade to exponential $\varepsilon(\lambda)$ when $\alpha = 1$, to $2\lambda T \sim \chi^2_{2\alpha}$ when $2\alpha \in \mathbb{N}$

$$F(t) = \frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda t} \tag{7.37}$$
$$\mathbb{E}(T) = \frac{\alpha}{\lambda} \tag{7.38}$$
$$var(T) = \frac{\alpha}{\lambda^2} \tag{7.39}$$

- Log-Normal $T \sim \mathrm{LN}(\mu, \sigma^2) = \exp\left[N(\mu, \sigma^2)\right]$.

$$f(t) = \frac{\phi(\frac{\log(t) - \mu}{\sigma})}{t\sigma} \tag{7.40}$$
$$F(t) = \Phi\left(\frac{\log(t - \mu)}{\sigma}\right) \tag{7.41}$$
$$S(t) = 1 - \Phi\left(\frac{\log(t - \mu)}{\sigma}\right) \tag{7.42}$$
$$\mathbb{E}(T) = e^{\mu + \frac{\sigma^2}{2}} \tag{7.43}$$
$$var(T) = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1) \tag{7.44}$$

[1] Weibull distribution could also be parameterized as $W(p, \gamma)$, where $\gamma = 1/\lambda$ is the scale factor.

- Generalized Gamma $T \sim \text{GG}(\alpha, p, \lambda)$, degrade to Weibull when $\alpha = p$, to Gamma when $p = 1$

$$f(t) = p\lambda(\lambda t)^{\alpha-1} e^{-(\lambda t)^p} \Big/ \Gamma(\frac{\alpha}{p}) \tag{7.45}$$

$$\mathbb{E}(T) = \frac{\Gamma((\alpha+1)/p)}{\lambda\Gamma(\alpha/p)} \tag{7.46}$$

□ **Likelihood for Censored Data**

When dealing with censored data, we put a basic assumption that $T \parallel C$ so that we can consider their distribution separately:

$$\text{General Notation: } \begin{cases} f(t) \\ F(t) \\ S(t) \\ \lambda(t) \end{cases} \quad T: \begin{cases} f_T(t) \\ F_T(t) \\ S_T(t) \\ \lambda_T(t) \end{cases} \quad C: \begin{cases} f_C(t) \\ F_C(t) \\ S_C(t) \\ \lambda_C(t) \end{cases} \tag{7.47}$$

Probability that we observe either T or C, or equivalently observe $(\tilde{T}, \delta)$:

$$\mathbb{P}(\tilde{T}, \delta) = \begin{cases} f_T(t)S_C(t), & \text{case event} \\ f_C(t)S_T(t), & \text{case censor} \end{cases} \tag{7.48}$$

$$= [f_T(t)S_C(t)]^{\delta} [f_C(t)S_T(t)]^{1-\delta} \tag{7.49}$$

$$\propto f_T(t)^{\delta} S_T(T)^{1-\delta} = \lambda_T(t)^{\delta} S_T(t) \tag{7.50}$$

Likelihood for estimating survival $S(t)$ can be taken as the part of T:

$$L(\theta; \tilde{t}, \delta) = \prod_{i=1}^{n} f_T(\tilde{t}_i; \theta)^{\delta_i} S_T(\tilde{t}_i; \theta)^{1-\delta_i} = \prod_{i=1}^{n} \lambda_T(\tilde{t}_i; \theta)^{\delta_i} S_T(\tilde{t}_i; \theta) \tag{7.51}$$

$$= \prod_{e \in \mathcal{E}} f(\tilde{t}_e; \theta) \prod_{r \in \mathcal{R}} S(\tilde{t}_r; \theta) \tag{7.52}$$

where $\mathcal{E}$ denotes indices of event data, $\mathcal{R}$ for indices of right censored data. This form can be generalized to more kinds of censoring, e.g. left censor $\mathcal{L}$, interval censor $\mathcal{I} = \{(t_{i,l}, t_{i,r})\}_{i=1}^{n_{\mathcal{I}}}$:

$$L(\theta; \tilde{t}, \delta) = \prod_{e \in \mathcal{E}} f(\tilde{t}_e; \theta) \prod_{r \in \mathcal{R}} S(\tilde{t}_r; \theta) \prod_{l \in \mathcal{R}} [1 - S(\tilde{t}_l; \theta)] \prod_{(t_{i,l}, t_{i,r}) \in \mathcal{I}} [S(\tilde{t}_{i,l}; \theta) - S(\tilde{t}_{i,r}; \theta)] \tag{7.53}$$

then use proper methods to maximize the Likelihood / conduct hypothesis testing. Following are some knowledeg recap for inference concerning likelihood:

□ **Likelihood Function**

Knowledge on likelihood function see section 2.2.4 ~ page 47. Some recap:

$$\text{Likelihood: } L(\theta; X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} f(X_i; \theta) \tag{7.54}$$

$$\text{Log-Likelihood: } \ell(\theta; X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} \log\{f(X_i; \theta)\} \tag{7.55}$$

$$\text{Score: } U(\theta; X_1, X_2, \ldots, X_n) = \frac{\partial \ell(\theta; X_1, X_2, \ldots, X_n)}{\partial \theta} = \sum_{i=1}^{n} \frac{\partial \log\{f(X_i; \theta)\}}{\partial \theta} \tag{7.56}$$

$$\text{Fisher Information: } I(\theta) = -\mathbb{E}\left[\frac{\partial^2 \log f(\vec{X}; \theta)}{\partial \theta \partial \theta^T}\right] = -n\mathbb{E}_{\vec{X}}\left[\frac{\partial^2 \log f(X_i; \theta)}{\partial \theta \partial \theta^T}\right] = n\bar{I}(\theta) \tag{7.57}$$

$$\bar{I} = I_i(\theta) = -\mathbb{E}\left[\frac{\partial^2 \log f(X_i; \theta)}{\partial \theta \partial \theta^T}\right] \tag{7.58}$$

$$\text{Observed Information: } I_n(\theta) = J(\theta) = -\sum_{i=1}^{n} \frac{\partial^2 \log f(X_i; \theta)}{\partial \theta \partial \theta^T} \tag{7.59}$$

Note: Fisher is an expectation of function of r.v., not random.

Properties:

$$\mathbb{E}_{\vec{X}}\left[U(\theta; \vec{X})\right] = 0 \tag{7.60}$$

$$I(\theta) = -\mathbb{E}_{\vec{X}}\left[\frac{\partial^2 \log f(\vec{X}; \theta)}{\partial \theta \partial \theta^T}\right] \tag{7.61}$$

$$= \mathbb{E}_{\vec{X}}\left[\frac{\partial \log f(\vec{X}; \theta)}{\partial \theta} \frac{\partial \log f(\vec{X}; \theta)}{\partial \theta^T}\right] = \mathbb{E}_{\vec{X}}\left[U(\theta; \vec{X}) U(\theta; \vec{X})^T\right] \tag{7.62}$$

$$var_{\vec{X}}\left[U(\theta; \vec{X})\right] = \mathbb{E}_{\vec{X}}\left[\left(U(\theta; \vec{X}) - \mathbb{E}_{\vec{X}}\left[U(\theta; \vec{X})\right]\right)\left(U(\theta; \vec{X}) - \mathbb{E}_{\vec{X}}\left[U(\theta; \vec{X})\right]\right)^T\right] \tag{7.63}$$

$$= \mathbb{E}_{\vec{X}}\left[U(\theta; \vec{X}) U(\theta; \vec{X})^T\right] = I(\theta) \tag{7.64}$$

$$\tag{7.65}$$

By CLT, considering U as a function of r.v.: (for a given θ and the data generated from the distribution with **this** parameter θ, i.e. $U(\theta) = U\left(\theta; \vec{X}(\theta)\right)$)

$$\sqrt{n}\left\{\bar{U}(\theta) - \mathbb{E}(U(\theta))\right\} = \frac{1}{\sqrt{n}} U(\theta) \xrightarrow{\text{d}} N(0, \frac{I(\theta)}{n}) \tag{7.66}$$

and by taking MLE estimation $\hat{\theta}^{MLE} \xrightarrow{p} \theta^*$, we can estimate the distribution (Note that MLE Estimator requires $U(\theta) = 0$)

$$J(\hat{\theta})^{-1/2}\left(U(\hat{\theta}) - \mathbb{E}(U(\hat{\theta}))\right) = J(\hat{\theta})^{-1/2} U(\hat{\theta}) \xrightarrow{\text{d}} N(0, 1) \tag{7.67}$$

□ **Statistical Inference on Parameter θ**

Statistical Inference concerning θ can be conducting using the above functions of θ

- Score Test: Use the distribution of score function directly: we can construct

$$J(\theta_0)^{-1/2} U(\theta_0; \vec{X}(\theta)) \xrightarrow[H_0]{\mathscr{L}} N(0, 1) \tag{7.68}$$

explanation: under $H_0 : \theta = \theta_0$, we should have $\hat{\theta} \to \theta = \theta_0$, which would lead to

$$J(\theta_0)^{-1/2} U(\theta_0; \vec{X}(\theta_0)) \xrightarrow{\text{d}} N(0,1) \tag{7.69}$$

however if $\hat{\theta} \to \theta \neq \theta_0$, then

$$\mathbb{E}\left[U(\theta_0; \vec{X}(\theta))\right] \neq 0 \tag{7.70}$$

which would lead to a different distribution, thus we can test the assumption $H_0 : \theta = \theta_0$ using equation 7.68 ~ page 219. Conduct hypothesis testing utilizing the fractiles of $N(0,1)$

- Wald Test: Use the Taylor Series of $U(\theta)$ to the 1^{st} order

$$U(\theta) \approx -J(\hat{\theta})(\theta - \hat{\theta}) \Rightarrow J(\hat{\theta})^{1/2}(\hat{\theta} - \theta) \approx J(\hat{\theta})^{-1/2} U(\hat{\theta}) \xrightarrow{\text{d}} N(0,1) \tag{7.71}$$

i.e.

$$\hat{\theta} \xrightarrow{\text{d}} N(\theta, J(\hat{\theta})^{-1}) \tag{7.72}$$

which can be utilized to construct testing statistics/interval estimations.

- Likelihood Ratio Test: Use the Taylor Series of $\ell(\theta)$ to the 2^{nd} order, and take $\hat{\theta} = \hat{\theta}^{MLE}$ so that $\ell'(\hat{\theta}) = 0$

$$\ell(\theta) \approx \ell(\hat{\theta}) - \frac{1}{2}(\theta - \hat{\theta})^T J(\theta)(\theta - \hat{\theta}) \Rightarrow 2(\ell(\hat{\theta}) - \ell(\theta)) \approx (\theta - \hat{\theta})^T J(\theta)(\theta - \hat{\theta}) \xrightarrow{\text{d}} \chi^2_p \tag{7.73}$$

where p is the dimension of θ

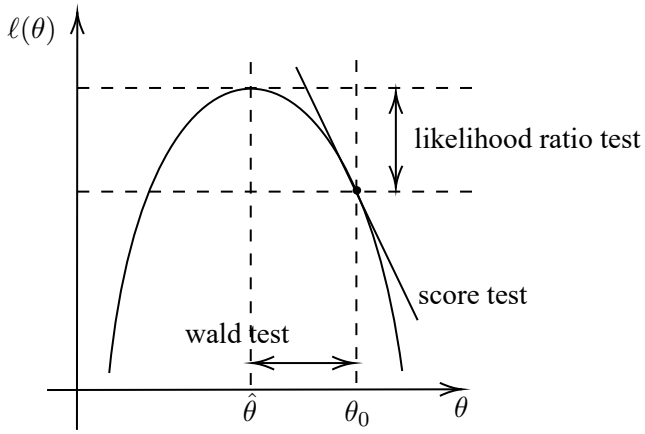


图 7.1: Illustration of Tests on $\ell(\theta)$-θ Plot

7.2.3 Non-Parametric Estimation to Survival Function

In this part we only focus on right censor data $(\tilde{T}_i, \delta_i)$, $\delta_i = 0$ for right censoring.

□ **Kaplan-Meier Estimator**

Idea of KM Estimator: Separate time into segments by censor/event time t_i, and decompose survival function

into products of hazard within segments, using equation 7.15 ~ page 216 which is:

$$\hat{S}(t) = \hat{\mathbb{P}}(T > t) = \prod_{t_i \le t} \hat{\mathbb{P}}\left(T > t_i | T > t_{i-1}\right) \tag{7.74}$$

$$= \hat{\mathbb{P}}\left(T > t | T > t_i\right) \prod_{t_i \le t} \left[1 - \hat{\mathbb{P}}\left(t_i \ge T > t_{i-1} | T > t_{i-1}\right)\right] \tag{7.75}$$

$$= \left(1 - \hat{\lambda}(t_i)\right) \prod_{t_i \le t} \left(1 - \hat{\lambda}(t_{i-1})\right) \tag{7.76}$$

$$= \prod_{t_i \le t} \left[1 - \hat{\lambda}(t_i)\right] \tag{7.77}$$

where $\hat{\lambda}(t_i)$ are relatively easy to estimate with censoring considered. r_i for # at risk: not censored/event till t_i, d_i for # event (death). We can model $\hat{\lambda}_i$ as

$$d_i | r_i \sim B(r_i, \lambda_i) \xrightarrow{\text{d}} N(r_i\lambda_i, r_i\lambda_i(1-\lambda_i)) \tag{7.78}$$

and obtain the MLE estimation of $\hat{\lambda}_i | r_i, d_i$[2]

$$\hat{\lambda}_i = \frac{d_i}{r_i} \tag{7.80}$$

$$var(\hat{\lambda}_i) = var(\frac{d_i}{r_i}) = \frac{\hat{\lambda}_i(1-\hat{\lambda}_i)}{r_i} \tag{7.81}$$

$$\hat{S}(t) = \prod_{t_i \le t} \left[1 - \hat{\lambda}(t_i)\right] = \prod_{t_i \le t} \left[1 - \frac{d_i}{r_i}\right] \qquad \text{(KM Estimator)}$$

$$var(\hat{S}(t)) = var\left\{\exp\left[\log \hat{S(t)}\right]\right\} \tag{7.82}$$

$$\approx [\hat{S}(t)]^2 var\left[\log \hat{S}(t)\right] \tag{7.83}$$

$$= [\hat{S}(t)]^2 \sum_{t_i \le t} var\left[\log(1-\hat{\lambda}_i)\right] \tag{7.84}$$

$$= [\hat{S}(t)]^2 \sum_{t_i \le t} \frac{1}{(1-\hat{\lambda}_i)^2} var(\hat{\lambda}_i) \tag{7.85}$$

$$= [\hat{S}(t)]^2 \sum_{t_i \le t} \frac{d_i}{r_i(r_i - d_i)} \qquad \text{(Greenwood's Formula)}$$

$$= [\hat{S}(t)]^2 var(\hat{\Lambda}(t)) \tag{7.86}$$

Interval Estimation of $\hat{S}(t)$ can be conducted using pointwise interval/confidence band:

- Plain pointwise approach:

$$\hat{S}(t) \pm N_{1-\frac{\alpha}{2}} \sigma[\hat{S}(t)] \tag{7.87}$$

- Log-Log pointwise approach (R. default): using $\hat{L}(t) = \log\left[-\log \hat{S}(t)\right] = \log\left[\hat{\Lambda}(t)\right]$

$$\hat{S}(t) \times e^{\pm N_{1-\frac{\alpha}{2}} \sigma(\hat{L}(t))} \tag{7.88}$$

[2]Here we use the Δ method for estimating the variance of function of r.v.: if $X \sim f(\mu, \sigma^2)$:

$$g(X) \approx g(\mu) + g'(\mu)(X-\mu) \Rightarrow var(g(X)) \approx [g'(\mu)]^2 var(X) \leftarrow [g'(X)]^2 var(X) \tag{7.79}$$

where

$$\sigma(\hat{L}(t)) = \sqrt{\frac{1}{[\log \hat{S}(t)]^2}} \sum_{t_i \le t} \frac{d_i}{r_i(r_i - d_i)} \tag{7.89}$$

- EP confidence band approach
- HW confidence band approach

Estimator of mean survival time:

$$\hat{\mu}_\tau = \int_0^\tau \hat{S}(t)\,\mathrm{d}t \tag{7.90}$$

$$var(\hat{\mu}_\tau) = \sum_{t_i} \left[\int_{t_i}^\tau \hat{S}(t)\,\mathrm{d}t\right]^2 \frac{d_i}{r_i(r_i - d_i)} \tag{7.91}$$

□ **Nelson-Aalen Estimator**

Idea of NA Estimator: estimate $\hat{\Lambda}(t)$ first, then obtain Fleming-Harrington Estimator $\hat{S}_{FH}(t) = e^{-\hat{\Lambda}(t)}$:

$$\hat{\Lambda}(t) = \sum_{t_i \le t} \hat{\lambda}(t_i) = \sum_{t_i \le t} \frac{d_i}{r_i} \tag{7.92}$$

$$var(\hat{\Lambda}(t)) = \sum_{t_i \le t} \frac{d_i(r_i - d_i)}{r_i^2(r_i - 1)} \tag{7.93}$$

$$\hat{S}_{FH}(t) = \exp\left[-\hat{\Lambda}(t)\right] \tag{7.94}$$

□ **Survival Function of Life Table**

A key difference of survival data of life table is that we cannot know the exact event time/censor time, locating in $[t_{i-1}, t_i)$, in this case we usually estimate $d_i,\ r_i$ using

$$d_i' = d_i \tag{7.95}$$

$$r_i' = r_i - \frac{c_i}{2} \tag{7.96}$$

where c_i is # censor in $[t_i, t_{i+1})$, r_i is # censoring at the begeinning of interval, i.e. t_{i-1}. And construct KM/NA estimator:

$$\hat{S}_{KM}(t) = \prod_{t_i \le t} \left(1 - \frac{d_i}{r_i'}\right) \tag{7.97}$$

$$var(\hat{S}_{KM}(t)) = \left[\hat{S}_{KM}(t)\right]^2 \sum_{t_i \le t} \frac{d_i}{r_i'(r_i' - d_i)} \tag{7.98}$$

$$\hat{\lambda}(t_{\text{mid}\,i}) = = \frac{\hat{f}(t_i)}{\hat{S}(t_i)} = \frac{d_i}{(t_i - t_{i-1})(r_i' - \frac{d_i}{2})} = \frac{d_i}{(t_i - t_{i-1})(r_i - \frac{c_i + d_i}{2})} \tag{7.99}$$

where mid i means the mid point of $[t_{i-1}, t_i)$, i.e. $\dfrac{t_{i-1} + t_i}{2}$

7.2.4 Hypothesis Testing to Group Comparison

Key focus: how to judge the difference between two survival function $S_1(t),\ S_2(t)$, or even when there are more than two groups.

□ **Mantel-Haenszel Logrank Test** [3]

[3] Note: Log means 'time record' here, rather than logarithm.

Idea of logrank test: adapt contingency table to censor table

表 7.1: 2×2 contingency table

	Event δ		
Group	Yes(1)	No(0)	Total
0	d_0	$r_0 - d_0$	r_0
1	d_1	$r_1 - d_1$	r_1
Total	d	$r - d$	r

- Recap: Pearson's χ^2 test: assign n sample into k groups, and conduct test on p_i, $i = 1, 2, \ldots, k$, denote that v_i samples are assigned to the i^{th} groups, then

$$K_n = \sum_{i=1}^{k} \frac{(v_i - np_i)^2}{np_i} \xrightarrow{\text{d}} \chi^2_{df} \tag{7.100}$$

In the above example, $df = k - 1$. In 2×2 contingency table, $df = 1$ because we assume d, r, r_0, r_1 are fixed. Pearson's χ^2 statistics for 2×2 contingency table:

$$\chi^2_P = \sum_{4 \text{ grids}} \frac{(\text{obs} - \text{expe})^2}{\text{expe}} \tag{7.101}$$

$$= \frac{(d_0 - r_0 \frac{d}{r})^2}{r_0 \frac{d}{r}} + \text{etc} \tag{7.102}$$

$$= \frac{\left[d_0 - r_0 \frac{d}{r}\right]^2}{r_0 r_1 d(r-d)/r^3} \sim \chi^2_1 \tag{7.103}$$

- Recap: Mental-Haenszel test, based on the Hypergeometric distribution that

$$d_0 \sim H(r_0, d, r) \Rightarrow \begin{cases} \mathbb{E}(d_0) = r_0 \dfrac{d}{r} \\ var(d_0) = \dfrac{r_0 r_1 d(r-d)}{r^2(r_0 - 1)} \end{cases}, \quad d_1, r_0 - d_0, r_1 - d_1 \text{ similar} \tag{7.104}$$

and construct

$$\chi^2_{MH} = \frac{\left(\sum_{4 \text{ grids}} \text{obs} - \mathbb{E}(\text{obs})\right)^2}{\sum_{4 \text{ grids}} var(\text{obs})} = \frac{\left[d_0 - r_0 \frac{d}{r}\right]^2}{\frac{r_0 r_1 d(r-d)}{r^2(r-1)}} \sim \chi^2_1 \tag{7.105}$$

χ^2_{MH} and χ^2_P are equal for lare r.

$$\chi^2_{MH} = \frac{r-1}{r} \chi^2_P \tag{7.106}$$

- Cochran-Mantel-Haenszel log-rank test

For survival data $t_1, t_2, \ldots, t_K$, we can construct a contingency table $\mathcal{C}_i$ at each time, and test on the $K \times 2 \times 2$ contingency table sequence:

表 7.2: 2×2 contingency table, $j = 1, 2, \ldots, K$

	Event δ		
Group	Yes(1)	No(0)	Total
0	d_{0j}	$r_{0j} - d_{0j}$	r_{0j}
1	d_{1j}	$r_{1j} - d_{1j}$	r_{1j}
Total	d_j	$r_j - d_j$	r_j

and get the CMH statistics for testing $H_0 : \theta_{t_1} = \theta_{t_2} = \ldots = \theta_{t_K} = 1, \theta$ for odds ratio between group 0/1.

$$\chi^2_{CMH} = \frac{\left[\sum_{j=1}^{K}(d_{0j} - r_{0j}\frac{d_j}{r_j})\right]^2}{\sum_{j=1}^{K}\frac{r_{0j}r_{1j}d_j(r_j-d_j)}{r_j^2(r_j-1)}} \sim \chi^2_1 \tag{7.107}$$

where the K contingency tables are treated independent, but they are still ordinal beacuse r_j contains information of history $d_{t_i<t_j}$, $c_{t_i<t_j}$

Properties & Special Cases & Extension of CMH logrank test:

- No tied death $d_j = 1$:

$$\chi^2_{CMH} = \frac{\left[\sum_{j=1}^{K}(d_{0j} - r_{0j}\frac{d_j}{r_j})\right]^2}{\sum_{j=1}^{K}\frac{r_{0j}r_{1j}d_j(r_j-d_j)}{r_j^2(r_j-1)}} = \frac{\left[\sum_{j=1}^{K}(d_{0j} - r_{0j}\frac{d_j}{r_j})\right]^2}{\sum_{j=1}^{K}\frac{r_{0j}r_{1j}}{r_j^2}} \sim \chi^2_1, \quad d_{0j} \in \{0, 1\} \tag{7.108}$$

- Intuition of obs $-\ \mathbb{E}\,(\text{obs})$:

$$\begin{aligned} \text{obs} - \mathbb{E}\,(\text{obs}) \approx & d_{0j} - d_j\frac{r_{0j}}{r_j} && (7.109)\\ = & \frac{r_{0j}r_{1j}}{r_j}\left(\hat{\lambda}_{0j} - \hat{\lambda}_{1j}\right) && (7.110) \end{aligned}$$

- Attach weight $w_i \geq 0$, $i = 1, 2, \ldots, K$ to $\mathcal{C}_i$:

$$\chi^2_{CMH,w} = \frac{\left[\sum_{j=1}^{K} w_j(d_{0j} - r_{0j}\frac{d_j}{r_j})\right]^2}{\sum_{j=1}^{K} w_j^2\frac{r_{0j}r_{1j}d_j(r_j-d_j)}{r_j^2(r_j-1)}} \sim \chi^2_1 \tag{7.111}$$

by choosing differnet kinds of weight $\vec{w}$ we could get variants of CMH test.

  - $w_i = 1$ for log-rank test. Focus more on difference at large t
  - $w_i = r_i$ for generalized Wilcoxon rank sum test. Focus more on difference at small t.

Note: weighted log-rank test should be used when **no cross** btw. $S_1(t)$ and $S_2(t)$. Kink-of-Weight to choose depends on H_1.

□ **Generalized Wilcoxon Rank Sum Test**

- Wilcoxon Two-Sample Rank Sum Test: Knowledge of Wilcoxon two-sample rank sum test see section 2.4.6 ~ page 65.

  Recap: to test the distribution difference of $\vec{X} = (X_1, X_2, \dots, X_n)$ and $\vec{Y} = (Y_1, Y_2, \dots, Y_m)$, we mix them together and rank as $\vec{Z} = (Z_{(1)}, Z_{(2)}, \dots, Z_{(m+n)})$. Rank of X_i:

$$R_i \equiv \text{rank}(X_i) \text{ in } \vec{Z}, \quad i = 1, 2, \dots, n \tag{7.112}$$

$$R \equiv \sum_{i=1}^{n} R_i \tag{7.113}$$

  A rank sum statistic to test:

$$\frac{R - \mathbb{E}(R)}{\sqrt{var(R)}} \sim N(0,1) \tag{7.114}$$

$$\begin{cases} \mathbb{E}(R) = \dfrac{n(m+n+1)}{2} \\ var(R) = \dfrac{mn(m+n+1)}{12} \end{cases} \tag{7.115}$$

  Rank sum statistic can be written in a Mann-Whitney form that can be generalized:

$$U_{ij} = U(X_i, Y_j) \equiv \begin{cases} +1 & \text{, case } X_i > Y_j \\ 0 & \text{, case } X_i = Y_j \\ -1 & \text{, case } X_i < Y_j \end{cases}, \qquad U = \sum_{i,j}^{n,m} U_{ij} \tag{7.116}$$

$$R = \frac{n(m+n+1)}{2} + \frac{U}{2} \tag{7.117}$$

- Mann-Whitney-Wilcoxon rank sum test for censored data:

  Notation: we still mix $X = \{(\tilde{t}_{1i}, \delta_{1i})\}_{i=1}^{n}$ and $Y = \{(\tilde{t}_{2j}, \delta_{2j})\}_{j=1}^{m}$ to get:

$$Z_{\text{mix}} = \left\{(\tilde{t}_i, \delta_i)\right\}_{i=1}^{m+n} \tag{7.118}$$

  and the Mann-Whitney form for Z_{mix}:

$$U_{ij} = U(Z_i, Z_j) \equiv \begin{cases} +1 & \text{, case } \tilde{t}_i > \tilde{t}_j,\ \delta_j = 1 \\ 0 & \text{, case } \tilde{t}_i = \tilde{t}_j \text{ or } \delta_j = 0 \\ -1 & \text{, case } \tilde{t}_i < \tilde{t}_j,\ \delta_j = 1 \end{cases}, \quad i = 1, 2, \dots, m+n.\ j = 1, 2, \dots, m+n. \tag{7.119}$$

  and the Extended Wilcoxon rank sum statistic:

$$W = \sum_{i \text{ if } Z_i \in X}^{m+n} \sum_{j=1}^{m+n} U_{ij} \tag{7.120}$$

  Under $H_0 : X \sim Y$, distribution features

$$\mathbb{E}(W) = 0 \tag{7.121}$$

$$\hat{var}(W) = \frac{mn}{(m+n)(m+n-1)} \sum_{i=1}^{m+n} \left(\sum_{j=1}^{m+n} U_{ij} \right)^2 \tag{7.122}$$

– choose $w_i = r_i$ in weighted log-rank test, and nominator becomes

$$\sum_{j=1}^{K} r_j(d_{0j} - r_{0j}\frac{d_j}{r_j}) = \sum_{j=1}^{K}[(r_{1j} - d_{1j})d_{0j} - (r_{0j} - d_{0j})d_{1j}] \tag{7.123}$$

$$= \sum_{j=1}^{K}\left[\#_{Y>t_j} \times \#_{X=t_j} - \#_{X>t_j} \times \#_{Y=t_j}\right] \tag{7.124}$$

$$= \#_{Y>X} - \#_{Y<X} \tag{7.125}$$

$$= -W \tag{7.126}$$

in which $\chi^2_{w_i=r_i,CMH}$ test is the same as generalized Wilcoxon rank sum test.

Section 7.3 Survival Model with Covariants

To research on the dependence of T with regard to covariants z. Survival data with covariants: $X = (\tilde{t}_i, \delta_i, z_i)$

7.3.1 Cox's Proportion Hazard Model

Basic assumption on dependence form: T hazard part and covariants part are Separatable:

$$\lambda(t|z) = \lambda_0(t)g(z) \Leftrightarrow S(t|z) = [S_0(t)]^{g(z)}, \quad S_0(t) = e^{-\int_0^t \lambda_0(\tau)\,\mathrm{d}\tau} \tag{7.127}$$

further a linear form $g(z) = \beta^T z$ is used;

$$\lambda(t|z) = \lambda_0(t)\exp\left[\beta^T z\right] \tag{7.128}$$

Basic Assumptions of Cox's PH Model:

- constant regression coefficient β;
- linear dependent of covariants $\beta' z$;
- exponential link function $e^{\cdot}$

in this proportion hazard model, the ratio of hazard only depend on β:

$$\log\left\{\frac{\lambda_{z_i}(t)}{\lambda_0(t)}\right\} = \beta^T z_i \parallel t \tag{7.129}$$

The unknown components are $\lambda_0(t), \beta$, where the $\lambda_0(t)$ lies in the $dim \to \infty$ space, and causes difficulty in conducting inference. Solution: decompose full likelihood into two parts, in which one of them, **Partial Likelihood** $L_{\mathrm{PH}}(\beta; X)$ is only function of β:

$$L(\beta, \lambda_0(\cdot); X) = \prod_i \left[\left(\lambda_0(t_i)e^{\beta^T z_i}\right)_i^{\delta}\left(e^{-\int_0^{t_i}\lambda_0(\tau)\,\mathrm{d}\tau}\right)^{e^{\beta^T z_i}}\right] \tag{7.130}$$

$$= L_{\mathrm{PH}}(\beta; X)L_{\mathrm{res}}(\beta, \lambda_0; X) \tag{7.131}$$

and we could focus on L_{PH} for further inference.

Note: the feasibility of partial likelihood comes from the form of proportion hazard.

□ **Partial Likelihood without Tie**

Derivation: First we assert t_i in ascending order and without tie: $t_1 < t_2 < \ldots < t_n$, and we use an discrete estimated form of $\lambda_0(t_i) = \lambda_i$

$$\int_0^{t_i} \lambda_0(\tau)\,\mathrm{d}\tau \approx \sum_{j=1}^{i} \lambda_j \tag{7.132}$$

then we could use a trick to reformulate $\ell(\beta, \lambda_1, \ldots, \lambda_n; X)$ as[4]

$$\ell(\beta, \lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \left\{ \delta_i(\log \lambda_i + \beta' z_i) - \sum_{j=1}^{i} \lambda_j e^{\beta' z_i} \right\} \tag{7.134}$$

$$= \sum_{i=1}^{n} \left\{ \delta_i(\log \lambda_i + \beta' z_i) - \lambda_i \sum_{j=i}^{n} e^{\beta' z_j} \right\} \tag{7.135}$$

and use MLE with regard to λ_i to get an estimate to λ_i:

$$\frac{\partial \ell(\beta, \lambda_1, \ldots, \lambda_n)}{\partial \lambda_i} = 0 \Rightarrow \lambda_i(\beta) = \frac{\delta_i}{\sum_{j=1}^{n} e^{\beta' z_j}} \quad \forall i \tag{7.136}$$

[4] Illustration for $\sum_{i=1}^{n} \lambda_i \sum_{j=i}^{n} e^{\beta' z_j} = \sum_{i=1}^{n} \sum_{j=1}^{i} \lambda_j e^{\beta' z_i}$ (Abel's Lemma for Summation by Parts)

$$\begin{pmatrix} \lambda_1 e^{\beta' z_1} & & & & \\ \lambda_1 e^{\beta' z_2} & \lambda_2 e^{\beta' z_2} & & & \\ \lambda_1 e^{\beta' z_3} & \lambda_2 e^{\beta' z_3} & \lambda_3 e^{\beta' z_3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \lambda_1 e^{\beta' z_n} & \lambda_2 e^{\beta' z_n} & \lambda_3 e^{\beta' z_n} & \cdots & \lambda_n e^{\beta' z_n} \end{pmatrix} \begin{matrix} \leftarrow \sum_{j=1}^{1} \lambda_j e^{\beta' z_1} \\ \leftarrow \sum_{j=1}^{2} \lambda_j e^{\beta' z_2} \\ \leftarrow \sum_{j=1}^{3} \lambda_j e^{\beta' z_3} \\ \leftarrow \vdots \\ \leftarrow \sum_{j=1}^{n} \lambda_j e^{\beta' z_n} \end{matrix} \tag{7.133}$$

$$\begin{matrix} \uparrow & \uparrow & \uparrow & \therefore & \uparrow \\ \lambda_1 \sum_{j=1}^{n} e^{\beta' z_j} & \lambda_2 \sum_{j=2}^{n} e^{\beta' z_j} & \lambda_3 \sum_{j=3}^{n} e^{\beta' z_j} & & \lambda_n \sum_{j=n}^{n} e^{\beta' z_j} \end{matrix} \quad \sum_{i=1}^{n} \lambda_i \sum_{j=i}^{n} e^{\beta' z_j} = \sum_{i=1}^{n} \sum_{j=1}^{i} \lambda_j e^{\beta' z_i}$$

then we could get the partial likelihood

$$L(\beta,\lambda_1(\beta),\ldots,\lambda_n(\beta)) = \prod_{i=1}^{n}\lambda_i(\beta)^{\delta_i}e^{\delta_i\beta' z_i}e^{-\sum_{j=1}^{n}e^{\beta' z_i}} \tag{7.137}$$

$$= e^{-\sum_i \delta_i}\prod_{i=1}^{n}\left(\frac{e^{\beta' z_i}}{\sum_{j:t_j\geq t_i}e^{\beta' z_j}}\right)^{\delta_i} \tag{7.138}$$

$$PL(\beta) \equiv \prod_{i=1}^{n}\left(\frac{e^{\beta' z_i}}{\sum_{j:t_j\geq t_i}e^{\beta' z_j}}\right)^{\delta_i} \tag{7.139}$$

$$P\ell = \sum_{i=1}^{n}\delta_i\left[\beta' z_i - \log\left(\sum_{j:t_j\geq t_i}e^{\beta' z_j}\right)\right] \tag{7.140}$$

$$U(\beta) = \sum_{i=1}^{n}\delta_i\left[z_i - \frac{\sum_{j:t_j\geq t_i}z_j e^{\beta' z_j}}{\sum_{j:t_j\geq t_i}e^{\beta' z_j}}\right] \tag{7.141}$$

$$J(\beta) = \sum_{i=1}^{n}\delta_i\left[\sum_{j:t_j\geq t_i}^{n}\frac{e^{\beta' z_j}}{\sum_{l:t_l\geq t_j}e^{\beta' z_l}}\left(z_j - \frac{\sum_{l:t_l\geq t_j}z_l e^{\beta' z_l}}{\sum_{l:t_l\geq t_j}e^{\beta' z_l}}\right)^2\right] \tag{7.142}$$

The above statistics can be use for further inference.

$$J(\beta_0)^{-1/2}U(\beta_0) \xrightarrow{\text{d}} N(0,1) \tag{7.143}$$

$$(\hat{\beta}-\beta_0) \xrightarrow{\text{d}} N(0, J(\hat{\beta})^{-1}) \tag{7.144}$$

$$2(\ell(\hat{\beta}) - \ell(\beta)) \xrightarrow{\text{d}} \chi^2_p \tag{7.145}$$

□ **Modification for Partial Likelihood with Tie**

There are various modification for tied data case. In PL without tie, the $\frac{e^{\beta' z_i}}{\sum_{j:t_j\geq t_i}e^{\beta' z_j}}$ term are usually changed to adapt for the case of. Intuition:

$$\frac{e^{\beta' z_i}}{\sum_{j:t_j\geq t_i}e^{\beta' z_j}} = \frac{\lambda(t_i|z_i)}{\sum_{j:t_j\geq t_i}\lambda(t_j|z_j)} \approx \mathbb{P}\left(i^{\text{th}}\text{event}|1\text{out of}\#\{j:t_j\geq t_i\}\right) \tag{7.146}$$

Notation: $\mathcal{R}_i$ for all datapoints at risk at time t_i, $\mathcal{D}_i$ for event cases at time t_i, $\mathcal{D}_i\subset\mathcal{R}_i$

- Cox's modification:

$$\mathbb{P}\left(\mathcal{D}_i\text{ events}\middle||\mathcal{D}_i|\text{out of}\#\{j:t_j\geq t_i\}\right) = \frac{e^{\sum_{l\in\mathcal{D}_i}\beta' z_l}}{\sum_{\text{all possible }|\mathcal{D}_j|=|\mathcal{D}_i|}e^{\sum_{l\in\mathcal{D}_j}\beta' z_j}} \tag{7.147}$$

drawback: $\sim O(|\mathcal{D}_i|!)$ complexity

$$PL(\beta) = \prod_{i=1}^{n}\left\{\frac{e^{\sum_{l\in\mathcal{D}_i}\beta' z_l}}{\sum_{\text{all possible }|\mathcal{D}_j|=|\mathcal{D}_i|}e^{\sum_{l\in\mathcal{D}_j}e^{\sum_{l}\in\mathcal{D}_j}\beta' z_l}}\right\} \tag{7.148}$$

- Breslow's approximation:

$$\mathbb{P}\left(\mathcal{D}_i\text{event}\middle||\mathcal{D}_i|\text{out of}\#\{j:t_j\geq t_i\}\right) \approx \frac{e^{\sum_{l\in\mathcal{D}_i}\beta' z_l}}{\left(\sum_{l\in\mathcal{R}_i}e^{\beta' z_l}\right)^{|\mathcal{D}_i|}} \tag{7.149}$$

or directly write the PL as

$$PL(\beta) = \prod_{i=1}^{n}\left\{\prod_{j\in\mathcal{D}_i}\frac{e^{\beta' z_j}}{\sum_{l\in\mathcal{R}_i}e^{\beta' z_l}}\right\} \tag{7.150}$$

- Efron's approximation: usually better than Breslow's, default method in `coxph()`

$$PL(\beta)=\prod_{i=1}^{n}\left\{\frac{e^{\sum_{l\in\mathcal{D}_i}\beta' z_l}}{\prod_{j=1}^{|\mathcal{D}_i|}\left(\sum_{l\in\mathcal{R}_i}e^{\beta' z_l}-\frac{j-1}{|\mathcal{D}_i|}\sum_{l\in\mathcal{D}_i}e^{\beta' z_l}\right)}\right\} \tag{7.151}$$

□ **Extension for Time-Dependent Variable**

Model:

$$\begin{cases}\lambda(t)= & \lambda_0(t)e^{\beta' z(t)}\\ \lambda(t)= & \lambda_0(t)e^{\beta(t)' z}\end{cases} \tag{7.152}$$

□ **Diagnostic Methods for PH Assumption**

- log-log plots: for categorical z_1, z_2, use relation

$$\log\left[-\log S(t,z_1)\right]-\log\left[-\log S(t,z_2)\right]=\beta'(z_1-z_2)\perp\!\!\!\perp t \tag{7.153}$$

Plot of $\log\left[\log\hat{S}(t,z)\right]$ should be 'parallel' curves.

- Check the coherence bet. observed data v.s. expected data.
- Goodness-of-fit using Schoenfeld residuals

$$\hat{r}_i=z_i-\sum_{j\in\mathcal{R}_i}z_k\cdot p(\hat{\beta},z_k)=z_i\bar{z}_i \tag{7.154}$$

$$p(\beta,z_k):=\frac{e^{\beta' z_k}}{\sum_{j\in\mathcal{R}_k}e^{\beta' z_j}} \tag{7.155}$$

- (Generalized) Cox-Snell Residual for overall goodness-of-fit:

Recall: for r.v. $T\sim f(t)$, $S(t)=\int_t^\infty f(\tau)\,\mathrm{d}\tau$. function of r.v. has distribution:

$$S(T)\sim U(0,1)\Rightarrow\Lambda(T)\sim\varepsilon(1) \tag{7.156}$$

define Cox-Snell Residual:

$$\hat{\Lambda}(z_i)=-\log\hat{S}(z_i) \tag{7.157}$$

the set $\{\hat{\Lambda}(z_i)\}$ could be viewed as a sample from $\varepsilon(1)$, we could test on the distribution, e.g. plot the cumulative hazard **of residual** v.s. residual to check $\Lambda(e)=e$.

- Delta-Beta Residual for infulential: for $\beta=(\beta_0=1,\beta_1,\ldots)$, define

$$\hat{\Delta}_{ij}=\hat{\beta}_j-\hat{\beta}_{j(\wedge i)} \tag{7.158}$$

where $\wedge i$ for estimator with the i^{th} subject removed. Plot the scatter plot of $\hat{\Delta}_{ij}$ to locate influential.

□ **Experiment Design for Log-rank Test under PH Assumption**

Question: how many events are needed for the testing $H_0 : \beta = 0 \nleftrightarrow H_a : \beta = \beta_a$?

Using log-rank statistics equation 7.108 ~page 224 in z-test form, under condition 1. no ties $d_j = 0, 1, 2$. β_a is small enough for taylor expansion:[5]

$$T_{CMH} = \frac{\sum_{j=1}^{K}\left(d_{0j} - r_{oj}\frac{d_j}{r_j}\right)}{\sqrt{\sum_{j=1}^{K}\frac{r_{0j}r_{1j}d_j(r_j-d_j)}{r_j^2(r_j-1)}}} \xrightarrow{\text{d}} N(\beta_a\sqrt{d\theta(1-\theta)}, 1) \tag{7.160}$$

where $d = \sum_{j=1} d_j$, θ is the prevalence of group 1.

Power of the test: denote γ for probability of type II error

$$\mathbb{P}\left(T_{CMH} > N_{\alpha/2}|H_a\right) = 1 - \gamma \Rightarrow \mu := \beta_a\sqrt{d\theta(1-\theta)} \approx N_{\alpha/2} + N_\gamma \tag{7.161}$$

Minimum number of events:

$$d = \frac{(N_{\alpha/2} + N_\gamma)^2}{\beta_a^2\theta(1-\theta)} \tag{7.162}$$

7.3.2 Accelerated Failure Time Model

Basic form of AFT Model (Accelerated Failure Tome Model) for categorical covariants:

$$S(t; z = 1) = S(\gamma t; z = 2) \Leftrightarrow \mathbb{P}(T_1 > t) = \mathbb{P}(T_2 > \gamma t) \tag{7.163}$$

Usually we attach some assumptions on function form of $S(t, z)$, usually take (parameter denoted α):

- Exponential:

$$S(t) = e^{-\lambda t}, \quad \lambda(t) = \lambda \tag{7.164}$$

$$\Rightarrow t = -\frac{1}{\lambda}\log S(t) \tag{7.165}$$

$$\Rightarrow \gamma := e^{\alpha' z} = \frac{1}{\lambda} = e^{-\beta' z} \tag{7.166}$$

  i.e. Exponential AFT model in which $\gamma = e^{\alpha' z}$ is equivalent to PH model with $\lambda = e^{\beta' z}$, and $\beta = -\alpha$

- Weibull:

$$S(t) = e^{-\lambda t^p}, \quad \lambda(t) = \lambda p t^{p-1} \tag{7.167}$$

$$\Rightarrow t = -\frac{1}{\lambda^{1/p}}\log S(t) \tag{7.168}$$

$$\Rightarrow \gamma := e^{\alpha' z} = \frac{1}{\lambda^{1/p}} = e^{-\beta' z/p} \tag{7.169}$$

  i.e. Weibull AFT model with $\gamma = e^{\alpha' z}$ is equivalent to PH model with $\lambda = e^{\beta' z}$, and $\beta = -\alpha p$

[5] Proof key:

$$d_{0j} \sim B(p_{0j}), \quad p_{0j} = \frac{r_{0j}\lambda_0}{r_{0j}\lambda_0 + r_{1j}\lambda_0 e^{\beta_a}} \tag{7.159}$$

and at small β_a, take approximation $\theta \approx r_{1j}/r_j$

- General Case: In different groups z, survivial time

$$T_i = T_0 e^{\alpha' z_i + \varepsilon_i / p}, \quad \varepsilon_i \sim \varepsilon(1) \tag{7.170}$$

$$S_i(t) = \mathbb{P}(T_i \geq t) \tag{7.171}$$

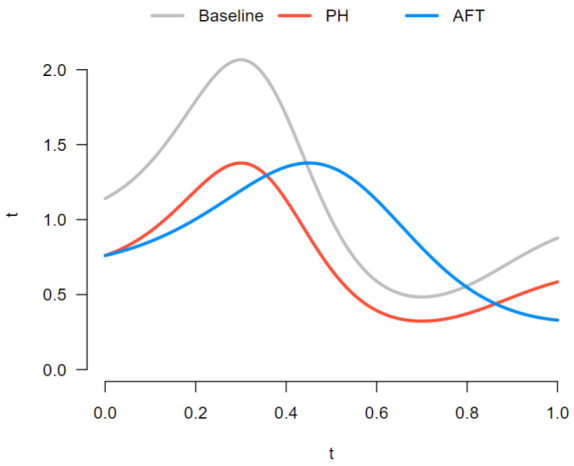
$$= \mathbb{P}\left(\log T_0 + \alpha' z_i + \frac{\varepsilon_i}{p} \geq \log t\right) \tag{7.172}$$

$$= S_{\varepsilon(1)}\left(p(\log t - \log T_0 - \alpha' z_i)\right) \tag{7.173}$$

□ AFT Model and PH Model

An intuition for parameters in AFT model and PH model:

$$\text{PH}: \lambda_i(t) = \lambda_0(t) e^{\eta_i}$$
$$\text{AFT}: \lambda_i(t) = \lambda_0(e^{-\eta_i} t) e^{-\eta_i}$$



Usually AFT model depends on a parametric model, whlie PH model only depends on the PH assumption.

▷ R. Code

An example:

```
coxph(formula = Surv(start, stop, event) ~ rx + number + size +
    factor(enum), data = bladder2)
```

Chapter. VIII　生物统计学概论部分

Instructor: Tianying Wang

Biostatistics is discipline to apply statistical methods to biological problems, including medicine, biology experiment, public health, etc. This section would focus on basic quantative skills to be used in advanced biostatistics research.

Section 8.1　Factor Model and ANOVA

A major question in biostatistics is to study the difference between groups, i.e. explanatory variable X is categorical. A 'way' to conduct grouping is called a **factor** , e.g. $\{\alpha_i\}$ where each i correponds to a **level** of the factor.

To compare groups, e.g. to determine whether there is significant difference between Y of each group, ANOVA is used. The key thought is to analyze difference value and variance and see whether the difference is large enough to 'exceed' variance.

□ **Factor Notation**

Response Y is denoted by its subsript to declare its group and index in this group, e.g. Y_{ijkl} indicates it is the l^{th} sample in group (i, j, k)

8.1.1　Single Factor Model and One-Way ANOVA

□ **Cell Means Model**

$$Y_{ij} = \mu_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \text{ i.i.d. } \sim N(0, \sigma^2) \tag{8.1}$$

Estimation target: $\mu_1, \ldots, \mu_r, \sigma^2$

Hypothesis testing H_0: $\mu_1 = \ldots = \mu_r = \mu$, v.s. H_1 : at least 1 μ_i is different.

Estimation:

$$\hat{\mu}_i = \bar{Y}_{i\cdot} = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} \tag{8.2}$$

$$s_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{i\cdot})^2 \tag{8.3}$$

$$s^2 = \frac{\sum_{i=1}^{r}(n_i-1)s_i^2}{\sum_{i=1}^{r}(n_i-1)} = \frac{\sum_{i=1}^{r}(n_i-1)s_i^2}{n_T - r} \tag{8.4}$$

Key of ANOVA: Decomposition of variation SS:

$$\text{SST} = \sum_{i=1}^{r}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{\cdot\cdot})^2 = \sum_{i=1}^{r}\sum_{j=1}^{n_i}\left(Y_{ij} - \bar{Y}_{i\cdot} + \bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}\right)^2 \tag{8.5}$$

$$= \sum_{i=1}^{r} + \sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{i\cdot})^2 + \sum_{i=1}^{r}(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2 \tag{8.6}$$

$$= \text{SSE} + \text{SSR} \tag{8.7}$$

□ **Effect Model**

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \varepsilon_{ij} \text{ i.i.d. } \sim N(0, \sigma^2) \tag{8.8}$$

Estimation target: $\mu, \alpha_1, \ldots, \alpha_r, \sigma^2$, w.r.t. $\sum_{i=1}^{r}\alpha_i = 0$.

Hypothesis tesing: $H_0: \alpha_1 = \ldots = \alpha_r = 0$, v.s. H_1 : at least 1 $\alpha_i \neq 0$

Estimation:

$$\hat{\mu} = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n_i}\frac{Y_{ij}}{n_i} \tag{8.9}$$

$$\hat{\alpha}_i = \bar{Y}_{i\cdot} - \hat{\mu} = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} - \hat{\mu} \tag{8.10}$$

$$s_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(Y_{ij} - \bar{Y}_{i\cdot}\right)^2 \tag{8.11}$$

$$s^2 = \frac{\sum_{i=1}^{r}(n_i-1)s_i^2}{n_T - r} \tag{8.12}$$

8.1.2 Fixed Effect and Random Effect

When divided into groups/naturally assigned in groups, we need to specify whether the factor levels are specially chosen (fixed effect) of randomly chosen from a 'population of levels' (random effect).

- Fixed Effect: whether there is a difference between / estimating the value of mean value μ_i of each specific levels
- Random Effect: whether the overall behaviour of μ_i comes from a 'random distribution'

Comment on fised / random in actual model building and statistical inference:

- whether a factor is fixed or random should be determined by how the data are obtained and the research problem to be studied, i.e. determining fixed / random model does **not** come from mathematics.
- for effect of interaction term, say $(\alpha\beta)_{ij}$ as the interaction effect of factor α_i and β_j, then $(\alpha\beta)_{ij}$ would be random once one of α_i or β_j is random.

Here use a one-way factor model as example:

□ **Fixed Effect:**

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \varepsilon_{ij} \text{ i.i.d. } \sim N(0, \sigma^2) \tag{8.13}$$

Estimation target: $\mu, \alpha_1, \ldots, \alpha_r, \sigma^2$, w.r.t. $\sum_{i=1}^r \alpha_i = 0$.

Hypothesis tesing: $H_0: \alpha_1 = \ldots = \alpha_r = 0$, v.s. H_1 : at least 1 $\alpha_i \neq 0$

Estimation (the same):

$$\hat{\mu} = \frac{1}{r} \sum_{i=1}^{r} \sum_{j=1}^{n_i} \frac{Y_{ij}}{n_i} \tag{8.14}$$

$$\hat{\alpha}_i = \bar{Y}_{i\cdot} - \hat{\mu} = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} - \hat{\mu} \tag{8.15}$$

$$s_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(Y_{ij} - \bar{Y}_{i\cdot}\right)^2 \tag{8.16}$$

$$s^2 = \frac{\sum_{i=1}^r (n_i - 1) s_i^2}{n_T - r} \tag{8.17}$$

ANOVA table:

Source of Var	SS	dof	MS	$\mathbb{E}$ (MS)
α_i	$\text{SS}\alpha = \sum_{i=1}^r n_i \left(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}\right)^2$	$r-1$	$\frac{\text{SS}\alpha}{r-1}$	$\sigma^2 + \frac{\sum_{i=1}^r n_i \alpha_i^2}{r-1}$
σ^2	$\text{SSE} = \sum_{i=1}^r \sum_{j=1}^{n_i} \left(Y_{ij} - Y_{i\cdot}\right)^2$	$n_T - r$	$\frac{\text{SSE}}{n_T - r}$	σ^2

F statistics for $H_0: \alpha_1 = \ldots = \alpha_r = 0$:

$$F = \frac{\text{MS}\alpha}{\text{MSE}} \sim F_{r-1, n_T - r} \tag{8.18}$$

□ **Random Effect:**

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \alpha_i \text{ i.i.d. } \sim N(0, \sigma_\alpha^2), \quad \varepsilon_{ij} \text{ i.i.d. } \sim N(0, \sigma^2) \tag{8.19}$$

Estimation target: $\mu, \sigma_\alpha^2, \sigma^2$

Hypothesis testing $H_0: \sigma_\alpha^2 = 0$, v.s. $H_1: \sigma_\alpha^2 \neq 0$

Estimation:

$$\hat{\mu} = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n_i}\frac{Y_{ij}}{n_i} \tag{8.20}$$

$$s_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(Y_{ij} - \bar{Y}_{i\cdot}\right)^2 \tag{8.21}$$

$$s^2 = \frac{\sum_{i=1}^{r}(n_i - 1)s_i^2}{n_T - r} \tag{8.22}$$

$$\hat{\sigma}_\alpha^2 = \frac{1}{r}\left(\frac{\text{SS}\alpha}{r-1} - \frac{\text{SSE}}{n_T - r}\right) \tag{8.23}$$

$$\tag{8.24}$$

ANOVA table:

Source of Var	SS	dof	MS	$\mathbb{E}$ (MS)
σ_α^2	$\text{SS}\alpha = \sum_{i=1}^{r} n_i\left(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}\right)^2$	$r-1$	$\frac{\text{SS}\alpha}{r-1}$	$\sigma^2 + n\sigma_\alpha^2$
σ^2	$\text{SSE} = \sum_{i=1}^{r}\sum_{j=1}^{n_i}\left(Y_{ij} - Y_{i\cdot}\right)^2$	$n_T - r$	$\frac{\text{SSE}}{n_T - r}$	σ^2

F statistics for $H_0 : \sigma_\alpha^2 = 0$:

$$F = \frac{\text{MS}\alpha}{\text{MSE}} \sim F_{r-1, n_T - r} \tag{8.25}$$

8.1.3 Two Factor Model and Two-Way ANOVA

Two factor model with interation term:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk} \tag{8.26}$$

$$Y_{ijk} - \bar{Y}_{\cdots} = \left(\bar{Y}_{i\cdot\cdot} - \bar{Y}_{\cdots}\right) + \left(\bar{Y}_{\cdot j\cdot} - \bar{Y}_{\cdots}\right) \tag{8.27}$$

$$+ \left(\bar{Y}_{ij\cdot} - \bar{Y}_{i\cdot\cdot} - \bar{Y}_{\cdot j\cdot} + \bar{Y}_{\cdots}\right) + \left(Y_{ijk} - \bar{Y}_{ij\cdot}\right) \tag{8.28}$$

$$\alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk} = \left((\mu + \alpha_i) - \mu\right) + \left((\mu + \beta_j) - \mu\right) \tag{8.29}$$

$$+ \left((\mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}) - (\mu + \alpha_i) - (\mu + \beta_j) + \mu\right) + (\varepsilon_{ijk}) \tag{8.30}$$

Here for convenience and clarity, when applying model with more factors, we use terms like $(\alpha\beta)_{ij}$ to avoid confusion of too many symbols.

8.1.4 General Case for Factor Model

e.g. three factors model

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijkl} \tag{8.31}$$

□ **Montgomery's Method for Restricted Model**

Montgomery describe a useful trick to form the ANOVA table and to find correponding $\mathbb{E}$ (MS) (EMS), and finally help construct proper F^* statistics. Here an explicit example of three factor (1F+2R) model is provided to illustrate the procedure.

Model we use here as example:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijkl} \tag{8.32}$$

$$i = 1, 2, \ldots, a \tag{8.33}$$

$$j = 1, 2, \ldots, b \tag{8.34}$$

$$k = 1, 2, \ldots, c \tag{8.35}$$

$$l = 1, 2, \ldots, n \tag{8.36}$$

where a is for fixed effect, b and c are for random effect.

model parameter:

$$\theta = \{\mu, \alpha_i^{i=1,\ldots,a}, \sigma^2_\beta, \sigma^2_\gamma, \sigma^2_{\alpha\beta}, \sigma^2_{\alpha\gamma}, \sigma^2_{\beta\gamma}, \sigma^2_{\alpha\beta\gamma}, \sigma^2\} \tag{8.37}$$

1. Prepare the framework of the EMS table, including:
   - column: list groups, and their random/fixed, and their number of levels.
   - row: terms in the model
   - error term written as $\varepsilon_{(ijk)l}$, i.e. random term index excluded from the bracket.

Random/Fix	F	R	R	R	
# level	a	b	c	n	
Index	i	j	k	l	$\mathbb{E}$ (MS)
α_i					
β_j					
γ_k					
$(\alpha\beta)_{ij}$					
$(\alpha\gamma)_{ik}$					
$(\beta\gamma)_{jk}$					
$(\alpha\beta\gamma)_{ijk}$					
$\varepsilon_{(ijk)l}$					

2. For each row, copy the number of observations under each column subscripts, if the column subscript does not appear in the index subscripts of the term. e.g. $(\alpha\beta)_{ij}$ does not contain, k, l so fill in the grid $((\alpha\beta)_{ij}, k)$ with c, and fill $((\alpha\beta)_{ij}, l)$ with n.

Random/Fix	F	R	R	R	
# level	a	b	c	n	
Index	i	j	k	l	$\mathbb{E}$ (MS)
α_i		b	c	n	
β_j	a		c	n	
γ_k	a	b		n	
$(\alpha\beta)_{ij}$			c	n	
$(\alpha\gamma)_{ik}$		b		n	
$(\beta\gamma)_{jk}$	a			n	
$(\alpha\beta\gamma)_{ijk}$				n	
$\varepsilon_{(ijk)l}$					

3. 1 is filled in the row of error term $(\varepsilon_{(ijk)l}, \cdot)$

Random/Fix	F	R	R	R	
# level	a	b	c	n	
Index	i	j	k	l	$\mathbb{E}$ (MS)
α_i		b	c	n	
β_j	a		c	n	
γ_k	a	b		n	
$(\alpha\beta)_{ij}$			c	n	
$(\alpha\gamma)_{ik}$		b		n	
$(\beta\gamma)_{jk}$	a			n	
$(\alpha\beta\gamma)_{ijk}$				n	
$\varepsilon_{(ijk)l}$	1	1	1	1	

4. for remaining grids, fill 1 if the column is Fixed, or 0 if the column is Random

Random/Fix	F	R	R	R	
# level	a	b	c	n	
Index	i	j	k	l	$\mathbb{E}$ (MS)
α_i	0	b	c	n	
β_j	a	1	c	n	
γ_k	a	b	1	n	
$(\alpha\beta)_{ij}$	0	1	c	n	
$(\alpha\gamma)_{ik}$	0	b	1	n	
$(\beta\gamma)_{jk}$	a	1	1	n	
$(\alpha\beta\gamma)_{ijk}$	0	1	1	n	
$\varepsilon_{(ijk)l}$	1	1	1	1	

5. Now the L.H.S. of the table is finished. To get the $\mathbb{E}$ (MS), we will need the coefficients in front of the variance term[1]. The approach is as follows: use the fourth row $(\alpha\beta)_{ij}$ as example:

   * (e.g. focus on row $(\alpha\beta)_{ij}$)

   (a) ignore columns with the same indexes, here it would be column i and j

   (b) select rows with the same or more extra indexes, here it would be row $(\alpha\beta)_{ij}$, $(\alpha\beta\gamma)_{ijk}$, $\varepsilon_{(ijk)l}$

   (c) now the grids to be used are colored brown

   (d) for each row, multiply all used grids to form the correponding coefficient (of the variance of this row), here it would be

$$\mathbb{E}\left(\mathrm{MS}_{(\alpha\beta)}\right) = c \times n\sigma^2_{\alpha\beta} + 1 \times n\sigma^2_{\alpha\beta\gamma} + 1 \times 1\sigma^2 = \sigma^2 + cn\sigma^2_{\alpha\beta} + n\sigma^2_{\alpha\beta\gamma} \tag{8.38}$$

Random/Fix	F	R	R	R	
# level	a	b	c	n	
Index	i	j	k	l	$\mathbb{E}$ (MS)
α_i	0	b	c	n	$\sigma^2 + cn\sigma^2_{\alpha\beta} + bn\sigma^2_{\alpha\gamma} + n\sigma^2_{\alpha\beta\gamma} + bcn\frac{\sum_i \alpha_i^2}{a-1}$
β_j	a	1	c	n	$\sigma^2 + an\sigma^2_{\beta\gamma} + acn\sigma^2_{\beta}$
γ_k	a	b	1	n	$\sigma^2 + an\sigma^2_{\beta\gamma} + abn\sigma^2_{\gamma}$
$(\alpha\beta)_{ij}$	0	1	c	n	$\sigma^2 + cn\sigma^2_{\alpha\beta} + n\sigma^2_{\alpha\beta\gamma}$
$(\alpha\gamma)_{ik}$	0	b	1	n	$\sigma^2 + bn\sigma^2_{\alpha\gamma} + n\sigma^2_{\alpha\beta\gamma}$
$(\beta\gamma)_{jk}$	a	1	1	n	$\sigma^2 + an\sigma^2_{\beta\gamma}$
$(\alpha\beta\gamma)_{ijk}$	0	1	1	n	$\sigma^2 + n\sigma^2_{\alpha\beta\gamma}$
$\varepsilon_{(ijk)l}$	1	1	1	1	σ^2

[1] Note the variance term is what we already know: for fixed effect it would be $\frac{\sum_i \alpha_i^2}{a-1}$, for random effect it would be σ^2_β

6. Now we can use $\mathbb{E}(\text{MS})$ to construct correponding F^*. e.g. to test $H_0 : \alpha_1 = \alpha_2 = \ldots = \alpha_a$, we use:

$$\mathbb{E}(\text{MS}_\alpha) = \sigma^2 + cn\sigma^2_{\alpha\beta} + bn\sigma^2_{\alpha\gamma} + n\sigma^2_{\alpha\beta\gamma} + bcn\frac{\sum_i \alpha_i^2}{a-1} \tag{8.39}$$

$$\mathbb{E}(\text{MS}_{\alpha\beta} + \text{MS}_{\alpha\gamma} - \text{MS}_{\alpha\beta\gamma}) = \sigma^2 + cn\sigma^2_{\alpha\beta} + bn\sigma^2_{\alpha\gamma} + n\sigma^2_{\alpha\beta\gamma} \tag{8.40}$$

$$F^*_{\alpha_i} = \frac{\text{MS}_\alpha + \text{MS}_{\alpha\beta\gamma}}{\text{MS}_{\alpha\beta} + \text{MS}_{\alpha\gamma}} \sim F_{(a-1)+(a-1)(b-1)(c-1),\,(a-1)(b-1)+(a-1)(c-1)} \tag{8.41}$$

8.1.5 Diagnosis

Some useful diagnosis to check assumptions:

- Levene's Test for homogeneity of variance: ▷ R. **Code**

```
dat %>% group_by(cat_1) %>% rstatix::levene_test(y ~ group)
```

- Shapiro-Wilk Test for Normality: ▷ R. **Code**

```
dat %>% group_by(cat_1) %>% rstatix::shapiro_test(y)
```

- Outlier test: ▷ R. **Code**

```
dat %>% group_by(cat_1) %>% rstatix::identify_outliers(y)
```

8.1.6 Miscellaneous Topics

Some miscellanea in design of experiment and about some advanced models:

□ **Crossed and Nested Factors**

In multi-factor studies, we may not be able to go through all possible factor settings.

- Crossed factor: all level combinations are covered in the experiment.
- Nested factor: the levels of one factor are unique to a particular level of another factor.

□ **Longitudinal Study**

When discrete **time** is used as factors, say $\tau_t^{t=\{t_1,\ldots,t_T\}}$ in Y_{ijt} where i for treatment, j for individuals, we may notice that response Y_{ijt} is effected by individual baseline, in such case we cannot use the ordinary factor model to study the difference of trent. Instead we would use **longitudinal study** to construct model and study the trend.. e.g.

$$Y_{ijt} = \mu + \alpha_i + \beta_{j(i)} + \tau_t + \varepsilon_{ijt} \tag{8.42}$$

where $\beta_{j(i)}$ stands for indivudual difference (say, with assumption $\beta_{j(i)} \sim N(0, \sigma^2_\beta)$)

Section 8.2 Statistical Inference on Contingency Table

Contingency table is an easy way to display categorical variables, an example:

表 8.1: A 2×2 contingency table

	Variable Z		
Variable Y	D	$D^{\complement}$	Total
E	n_{11}	n_{12}	$n_{1\cdot}$
$E^{\complement}$	n_{21}	n_{22}	$n_{2\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	$n_{\cdot\cdot}$

8.2.1 Quantities and Statistics from Contingency Table

□ **Prospective Study and Retrospective Study**

Contingency table itself is symmetric w.r.t. Y, Z, but in experimental design we usually first specify and divide groups, and then conduct experiment (prospective) or conduct survey (retrospective), which would cause different conditional probability. An example in studying the effect of medicine

- Prospective Study: say, $Y = E/E^{\complement}$ for drug/placebo group is assigned before experiment, and then $Z = D/D^{\complement}$ for medicine effect is studied after treatment.

  In this case $n_{1\cdot}$, $n_{2\cdot}$ are pre-determined fixed number.

  Such design is a well-controlled experiment to study the effct, but sometimes faced with problem concerning survival analysis, see Chapter 7 ~ page 214 for detail. And for some problems like, e.g. Z is related to rare disease, this method is **low-efficient**.

- Retrospective Study: say, some $Z = D/D^{\complement}$ for medicine effect patients are selected, and then their history of taking drug or not is collected.

  In this case $n_{\cdot 1}$, $n_{\cdot 2}$ are pre-determined fixed number.

  This method is quick and convenient to conduct study, but usually we cannot control the exposure status Y accurately (because they are collected by, e.g. questionnaire)

Statistics and tests should be selected **based on the data collection design** (prospective/retrospective) because of different probability condition.

□ **Statistics and Estimation**

With respective probabilities in two groups E, $E^{\complement}$ denoted as

$$p_1 = \mathbb{P}\left(D|E\right), \qquad p_2 = \mathbb{P}\left(D|E^{\complement}\right) \tag{8.43}$$

we usully focus on the 'difference' between group E and $E^{\complement}$, there are some quantities to help measure the group difference:

$$\text{Risk difference: } \Delta = p_1 - p_2 \tag{8.44}$$

$$\text{Relative risk: } \phi = p_1/p_2 \tag{8.45}$$

$$\text{Odds ratio: } \theta = \frac{p_1/(1-p_1)}{p_2/(1-p_2)} \tag{8.46}$$

Their estimation:

- Respective probability p_1, p_2:

$$\text{Prospective: } \begin{cases} \hat{p}_1 = \dfrac{n_{11}}{n_{1\cdot}} \\ \hat{p}_2 = \dfrac{n_{21}}{n_{2\cdot}} \end{cases} \tag{8.47}$$

$$\text{Retrospective: } \begin{cases} \hat{p}_1 = \dfrac{\rho\frac{n_{11}}{n_{\cdot 1}}}{\rho\frac{n_{11}}{n_{\cdot 1}} + (1-\rho)\frac{n_{12}}{n_{\cdot 2}}} \\ \hat{p}_2 = \dfrac{\rho\frac{n_{21}}{n_{\cdot 1}}}{\rho\frac{n_{21}}{n_{\cdot 1}} + (1-\rho)\frac{n_{22}}{n_{\cdot 2}}} \end{cases} \tag{8.48}$$

$$\text{where } \rho \text{ is the prevalence btw} D, D^{\complement} \text{ in natural condition} \tag{8.49}$$

- Relative Risk ϕ:

$$\text{Prospective: } \hat{\phi} = \frac{n_{11}/n_{1\cdot}}{n_{21}/n_{2\cdot}} \tag{8.50}$$

$$\text{Retrospective: } \hat{\phi} = \frac{\hat{p}_1}{\hat{p}_2} \tag{8.51}$$

- Odds Ratio θ:

$$\text{Prospective\&Retrospective: } \hat{\theta} = \frac{n_{11}n_{22}}{n_{21}n_{12}} \tag{8.52}$$

which is the same in either cases.

variance of $\hat{\theta}$: estimated at $(n_{11}, n_{12}, n_{21}, n_{22}) \sim \text{Multinomial}(n_{\cdot\cdot}, \pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$:

$$\hat{var}(\log\hat{\theta}) = \frac{1}{n_{11}} + \frac{1}{n_{12}} + \frac{1}{n_{21}} + \frac{1}{n_{22}} \tag{8.53}$$

□ **Hypothesis Testing**

The mostly used hypothesis is the dependence assumption: $p_1 = p_2$, or more generally speaking for $m \times n$ table:

$$H_0 : \pi_{ij} = \pi_{i\cdot}\pi_{\cdot j}, \quad \forall i, j \tag{8.54}$$

Denote $O_{ij} = n_{ij}$ as the **O**bserved value, $E_{ij} = n_{\cdot\cdot}\pi_{ij}$ as the **E**xpected value.[2] Expected value is calculated for the model used, under null hypothesis H_0. Example for independence test $\pi_{ij} = \pi_{i\cdot}\pi_{\cdot j}$:

$$\hat{\pi}_{ij} = \hat{\pi}_{i\cdot}\hat{\pi}_{\cdot j} = \frac{n_{i\cdot}}{n_{\cdot\cdot}}\frac{n_{\cdot j}}{n_{\cdot\cdot}} \Rightarrow E_{ij} = n_{\cdot\cdot}\hat{\pi}_{ij} = \frac{n_{i\cdot}n_{\cdot j}}{n_{\cdot\cdot}} \tag{8.61}$$

Statistics:

- **Pearson's χ^2 Test**:

$$\chi^2_P = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(O_{ij} - E_{ij})^2}{E_{ij}} \xrightarrow{\mathscr{L}} \chi^2_{(I-1)(J-1)} \tag{8.62}$$

[2] E_{ij} is calculated based on data and the model you choose, thus can be applied to more complexed cases, e.g. Hardy-Weinberg

- **Likelihood Ratio Test**:

$$G^2 = -2\log(\Lambda) = 2\sum_{i=1}^{I}\sum_{j=1}^{J} O_{ij}\log\frac{O_{ij}}{E_{ij}} \xrightarrow{\mathscr{L}} \chi^2_{(I-1)(L-1)} \tag{8.63}$$

Some other useful tests:

- McNemar test on $\pi_{12} = \pi_{21}$ for matched pairs:

$$z^2 = \frac{(n_{12} - n_{21})^2}{n_{12} + n_{21}} \xrightarrow{\mathscr{L}} \chi^2_1 \tag{8.64}$$

Section 8.3 Clinical Trial Design*

Section 8.4 GWAS*

proportions with X^a gene grequency p

$$\mathbb{P}\left(X^aX^a;\text{Female}\right) = p^2 \tag{8.55}$$

$$\mathbb{P}\left(X^AX^a;\text{Female}\right) = 2p(1-p) \tag{8.56}$$

$$\mathbb{P}\left(X^AX^A;\text{Female}\right) = (1-p)^2 \tag{8.57}$$

$$\mathbb{P}\left(X^aY;\text{Male}\right) = p \tag{8.58}$$

$$\mathbb{P}\left(X^AY;\text{Male}\right) = (1-p) \tag{8.59}$$

In such complex case, parameter should be estimated using e.g. MLE estimation.And then calculate E_{ij}s

$$L(p) = \left[p^2\right]^{O_{a,\text{F}}}\left[1-p^2\right]^{O_{A,\text{F}}}[p]^{O_{a,\text{M}}}[1-p]^{O_{A,\text{M}}} \tag{8.60}$$

Chapter. IX　统计学习导论部分

Instructor: Sheng Yu

In this course, some key formulations/theorem in machine learning are deduced, together with core principles illustrated.

□ **What is Machine Learning?**

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to ”learn” with data, without being explicitly programmed.

Examples of Machine Learning:

- Linear/Logistic Regression (Linear Model), Chapter 3 ~ page 71, section 9.1 ~ page 243;
- Decision Tree, section 9.6 ~ page 261;
- Support Vector Machine, section 9.3 ~ page 251;
- Clustering, section 4.7 ~ page 138, section 9.5 ~ page 258;
- Bayesian Network, section 11.4 ~ page 299;
- Neural Network, section 9.7 ~ page 264;
- Conditional Random Field
- etc.

This section will cover some of the methods above in a machine learning perspective.

Section 9.1　Linear Model

Linear model is the basic model in statistics, see Chapter 3 ~ page 71.

9.1.1　Linear Model in Machine Learning Perspective

In machine learning field, key feature of linear model is its affine form of variable dependence:

$$Y = f(X) + \varepsilon = \tilde{f}_\beta(X'\beta) + \varepsilon \tag{9.1}$$

where usually $X = (1, X_1, X_2, \ldots, X_p)$, $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$.[1] Some example of linear model:

[1] Some materials use $X = (X_1, X_2, \ldots, X_p)$, $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$, and the affine dependence is $\tilde{f}(\beta_0 + X'\beta)$

- Linear Regression:

$$Y = \beta_0 + \beta_1 x_1 + \dots \beta_p x_p + \varepsilon = X'\beta + \varepsilon \tag{9.2}$$

- Generalized Linear Model:

$$Y \sim f(\theta(X'\beta)) \tag{9.3}$$

in this framework,

  - Linear regression:

$$Y \sim N(X'\beta, \sigma^2) \tag{9.4}$$

  - Logistic regression:

$$Y \sim \text{Bernoulli}(\text{logistic}(X'\beta)) \tag{9.5}$$

9.1.2 Linear Regression

Linear Regression:

$$Y = \beta_0 + \beta_1 x_1 + \dots \beta_p x_p + \varepsilon = X'\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \tag{9.6}$$

usually use Squared Error Loss to eatimate (β, σ^2)

$$\mathcal{L}(Y, \hat{f}(X)) = \Big(Y - \hat{f}(X)\Big)^2 = \Big(Y - X\hat{\beta}\Big)^2 \tag{9.7}$$

LSE estimator (where Y and X imply corresponding sample vector/matrix), more detail see section 3.3 ~ page 81:

$$\frac{\partial \mathcal{L}}{\partial \beta} = 0 \Rightarrow \hat{\beta} = (X'X)^{-1}X'Y \tag{9.8}$$

- Predict:

$$\hat{Y} = X\beta = X(X'X)^{-1}X'Y \tag{9.9}$$

- Hat Matrix:

$$H = P_X \equiv X(X'X)^{-1}X' \tag{9.10}$$

idempotent and symmetry

$$H^2 = H, \quad H = H' \tag{9.11}$$

- Properties of $\hat{\beta}$, $\hat{\sigma^2}$:[2]

$$cov(\hat{\beta}) = cov\left((X'X)^{-1}X'(X\beta + \varepsilon)\right) = (X'X)^{-1}\sigma^2 \tag{9.12}$$

$$var(\hat{\beta}_j) = \frac{\sigma^2}{(n-1)S_{x_j}^2} \cdot \text{VIF}_j \tag{9.13}$$

$$cov(e) = cov(Y - \hat{Y}) = (I - H)\sigma^2 \tag{9.14}$$

$$var(\hat{\sigma^2}) = var(\text{MSE}) = \frac{Y'(I-H)Y}{n - (p+1)} \tag{9.15}$$

[2] Definition of VIF_j see section 3.4.7 ~ page 98

9.1.3 Regularization Methods

In machine learning topic we would focus more on model generalization ability, so that the model can perform better on reality problems. In linear regression, we usually use normalization methods.

Basically linear model uses SE loss:

$$\mathcal{L} = \sum_{i=1}^{n}(y_i - \beta_0 - \beta' x_i)^2 = \sum_{i=1}^{n}(y_i - x_i'\beta)^2 \tag{9.16}$$

we can put various normalize term (penalty) in loss or put constraint on β: (these two methods are equivalent in many cases)

- Ridge Regression/ℓ_2 Penalty/Tikhonov Regularization:[3]

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda\|\beta\|_2^2 \tag{9.18}$$

  or equivalent form

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{i=1}^{n}(y_i - x_i'\beta)^2 \tag{9.19}$$

$$s.t.\ \|\beta\|_2^2 \le s \tag{9.20}$$

  in either case, λ or s is hyper-parameter.

  Ridge regression has closed form solution

$$\hat{\beta}^{\text{ridge}} = (X'X + \lambda I)^{-1}X'Y \tag{9.21}$$

  Intuitively speaking, ridge regression help shrink $\hat{\beta}$ by an non-zero factor.

  A Bayesian point of view for Ridge regreession see section 13.4.9 ~ page 357

- LASSO/ℓ_1 Penalty:

$$\hat{\beta}^{\text{LASSO}} = \arg\min_{\beta} \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda\|\beta\|_1 \tag{9.22}$$

  or equivalent form

$$\hat{\beta}^{\text{LASSO}} = \arg\min_{\beta} \sum_{i=1}^{n}(y_i - x_i'\beta)^2 \tag{9.23}$$

$$s.t.\ \|\beta\|_1 \le s \tag{9.24}$$

  LASSO help shrink significantly large coefficients and truncate small coefficients.

[3] Recall for ℓ_p norm: for n-dim vector $\vec{v} = (v_1, v_2, \ldots, v_n)$

$$\|v\|_p = \left(\sum_{i=1}^{m}|v_i|^p\right)^{1/p} \tag{9.17}$$

- Generalized ℓ_p norm penalty:

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \|\beta\|_2^2 \tag{9.25}$$

or equivalent form

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 \tag{9.26}$$

$$s.t. \ \|\beta\|_2^2 \leq s \tag{9.27}$$

- Elastic Net:

$$\hat{\beta} = \arg\min_{\beta} \|Y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \tag{9.28}$$

equivalent form:

$$\hat{\beta} = \arg\min_{\beta} \|Y - X\beta\|^2 \tag{9.29}$$

$$s.t. \ \frac{\lambda_1}{\lambda_1 + \lambda_2} \|\beta\|_1 + \frac{\lambda_2}{\lambda_1 + \lambda_2} \|\beta\|_2^2 \leq s \tag{9.30}$$

picking proper hyper-parameter $(s, \lambda = \dfrac{\lambda_2}{\lambda_1 + \lambda_2})$

A note on elastic net: the boundary of elastic net $\lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 = \text{const}$ is between ℓ_1 boundary and ℓ_2 boundary. Both the variable selection feature of ℓ_1 and the differnetiable feature of ℓ_2 are partially maintained.

- Adaptive LASSO:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_u - x_i'\beta\right)^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j^{\text{OLS}}|} \tag{9.31}$$

- Non-negative Garrote method*
- SCAD*

Section 9.2 Basic Classification Model

Denote: Dataset $\mathcal{D} = \{(x_i, y_i),\ i = 1, 2, \ldots, N\}$, $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]$, with reponse $y_i \in \mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ as a K-classification problem. When $K = |\mathcal{C}| = 2$ for binary classification, in this case we usually denote $\mathcal{C}_{01} = \{0, 1\}$.

Target is to predict/classify Y from X

$$\hat{Y} = \hat{f}(X) \rightsquigarrow Y \tag{9.32}$$

9.2.1 Classification Metrics

- Accuracy

$$\mathbb{P}\left(\hat{Y}=Y\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=y_i)}{N} \tag{9.33}$$

- Error Rate/ Misclassification Rate

$$\mathbb{P}\left(\hat{Y}\neq Y\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i\neq y_i)}{N} \tag{9.34}$$

- Prevalence for binary classification

$$\mathbb{P}\left(Y=1\right) \hat{\to} \frac{\sum_{i=1}^{N} y_i}{N} \tag{9.35}$$

□ **Confusion Matrix and Metrics for Binary Classification**

表 9.1: Confusion matrix for binary classification

	Predicted Value $\hat{Y}$	
Ground Truth Y	1	0
1	n_{11}	n_{10}
0	n_{01}	n_{00}

Metrics:

- True Positive Rate (TPR)/ Sensitivity/ Recall:

$$\mathbb{P}\left(\hat{Y}=1|Y=1\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)\cdot\mathbb{I}(y_i=1)}{\sum_{i=1}^{N}\mathbb{I}(y_i=1)}=\frac{n_{11}}{n_{11}+n_{10}} \tag{9.36}$$

- False Positive Rate (FPR):

$$\mathbb{P}\left(\hat{Y}=1|Y=0\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)\cdot\mathbb{I}(y_i=0)}{\sum_{i=1}^{N}\mathbb{I}(y_i=0)}=\frac{n_{01}}{n_{01}+n_{00}} \tag{9.37}$$

- True Negatie Rate (TNR)/ Specific (SPC):

$$\mathbb{P}\left(\hat{Y}=0|Y=0\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)\cdot\mathbb{I}(y_i=0)}{\sum_{i=1}^{N}\mathbb{I}(y_i=0)}=\frac{n_{00}}{n_{01}+n_{00}} \tag{9.38}$$

- False Negative Rate (FNR):

$$\mathbb{P}\left(\hat{Y}=0|Y=1\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)\cdot\mathbb{I}(y_i=1)}{\sum_{i=1}^{N}\mathbb{I}(y_i=1)}=\frac{n_{10}}{n_{11}+n_{10}} \tag{9.39}$$

-

- Positive Predictive Value (PPV)/ Precision:

$$\mathbb{P}\left(Y=1|\hat{Y}=1\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)\cdot\mathbb{I}(y_i=1)}{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)}=\frac{n_{11}}{n_{11}+n_{01}} \tag{9.40}$$

- False Discovery Rate (FDR):

$$\mathbb{P}\left(Y=0|\hat{Y}=1\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)\cdot\mathbb{I}(y_i=0)}{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=1)}=\frac{n_{01}}{n_{11}+n_{01}} \tag{9.41}$$

- Negative Predictive Value (NPV):

$$\mathbb{P}\left(Y=0|\hat{Y}=0\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)\cdot\mathbb{I}(y_i=0)}{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)}=\frac{n_{00}}{n_{10}+n_{00}} \tag{9.42}$$

- False Omission Rate (FOR):

$$\mathbb{P}\left(Y=1|\hat{Y}=0\right) \hat{\to} \frac{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)\cdot\mathbb{I}(y_i=1)}{\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i=0)}=\frac{n_{10}}{n_{10}+n_{00}} \tag{9.43}$$

F_1 Score:

$$F_1=2\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \tag{9.44}$$

Receive Operating Characteristic Curve (ROC Curve) is used to examing performance of a model with threshold s:

$$\hat{Y}=\begin{cases}1, & \text{case } \hat{f}(X)>s\\ 0, & \text{case } \hat{f}(X)\le s\end{cases} \tag{9.45}$$

for each s, the model gives a corresponding $\text{TPR}(s)$ (recall) and $\text{FPR}(s)$, all $(\text{TPR}(s),\text{FPR}(s))$ forms the ROC curve. Area Under ROC Curve (AUC) is used also as a measure of model performance.

9.2.2 Cross-Validation

In general process of train & validate, we split the data into train set and validation set, which causes insufficient usage of data. k-fold Cross-validation (CV) is proposed to oversome the problem.

1. Divide $\mathcal{D}$ into k folds
2. For each time $i=1,2,\ldots,k$, pick the i^{th} fold as validation set, others as train set, train the model and calculate the metric m_i
3. Average over all folds is used as final performance

$$m=\frac{\sum_{i=1}^{k}m_i}{k} \tag{9.46}$$

CV could help ease the problem of overfitting.

9.2.3 Bayes Optimal Classifier

Due to the randomness of class distribution, no classifier could reach 100% accuracy, but there is an optimal classifier (if we really know the underlying distribution) to minimizie the expected loss:

$$\mathbb{E}_{Y,X\sim p_{Y,X}}(\mathcal{L}) = \mathbb{E}_X\left(\sum_{k=1}^{K}\mathcal{L}(k,\hat{y}(X))\right)\cdot\left\|Y=k|X\right\| \tag{9.47}$$

$$\Rightarrow \hat{y}(x)_{\text{optimal}} = \arg\min_j \mathcal{L}(k,j)\cdot\mathbb{P}(Y=k|X=x) \tag{9.48}$$

$$(\text{if 0/1 loss}) = \arg\max_j \mathbb{P}(Y=j|X=x) \tag{9.49}$$

which is the Bayes Optimal Classifier $\hat{y}(x)_{\text{optimal}}$, its error rate is Bayes optimal rate.

9.2.4 k-Nearest Neighbours Approach

The k-nearest neighours (KNN) fit with threshold s:

$$\hat{f}(x) = \frac{1}{k}\sum_{i:x_i\in\mathcal{N}_k(x)} y_i \tag{9.50}$$

$$\hat{Y} = \begin{cases} 1, & \text{case } \hat{f}(X) > s \\ 0, & \text{case } \hat{f}(X) \le s \end{cases} \tag{9.51}$$

where $\mathcal{N}_k(x)$ is the nearest k datapoints of x, various distance measure $\|\cdot\|$ could be used. k-NN method is faced with the problem of curse of dimensionality (see section 4.3 ~ page 129) in high dimension case. Calculation cost is at $O(N)$.

9.2.5 Density Based Classification

An intuition: samples from the same class k should be clustered, we use some distribution to represent it as $f_k(x)$. Bayes optimal criterion with prior π_k:

$$\hat{y}(x) = \arg\max_k \mathbb{P}(Y=k|X=x) = \arg\max_k \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l} = \arg\max_k f_k(x)\pi_k \tag{9.52}$$

□ Discriminant Analysis

Detail about discriminant analysis could be found in section section 4.6 ~ page 135. Here are some recaps:

Discriminant analysis assume a gaussian distribution

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}}\exp\left\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\right\} \tag{9.53}$$

- Linear Discriminant Analysis (LDA): Assume $\Sigma_k = \Sigma, \forall k$

$$\log\frac{\mathbb{P}(k|x)}{\mathbb{P}(l|x)} = \log\frac{f_k(x)\pi_k}{f_l(x)\pi_l} \tag{9.54}$$

$$= \log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)'\Sigma^{-1}(\mu_k-\mu_l) + x'\Sigma^{-1}(\mu_k-\mu_l) \tag{9.55}$$

Classification function:

$$\hat{y}(x) = \arg\max_k \delta_k(x) = \arg\max_k \log \hat{\pi}_k + x'\hat{\Sigma}^{-1}\hat{\mu}_k - \frac{1}{2}\hat{\mu}_k'\hat{\Sigma}^{-1}\hat{\mu}_k \tag{9.56}$$

$$\hat{\pi}_k = \frac{N_k}{N} \tag{9.57}$$

$$\hat{\mu}_k = \frac{\sum_{i:y_i=k} x_i}{N_k} \tag{9.58}$$

$$\hat{\Sigma} = \frac{\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)'}{N-K} \tag{9.59}$$

- Quadratic Discriminant Analysis (QDA): Allow different Σ_k, Classification function:

$$\hat{y}(x) = \arg\max_k \delta_k(x) = \arg\max_k \log \hat{\pi}_k - \frac{1}{2}\log|\hat{\Sigma}_k| - \frac{1}{2}(x-\hat{\mu}_k)'\hat{\Sigma}_k^{-1}(x-\hat{\mu}_k) \tag{9.60}$$

$$\hat{\pi}_k = \frac{N_k}{N} \tag{9.61}$$

$$\hat{\mu}_k = \frac{\sum_{i:y_i=k} x_i}{N_k} \tag{9.62}$$

$$\hat{\Sigma}_k = \frac{\sum_{i:y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)'}{N_k-1} \tag{9.63}$$

□ **Naïve Bayes Classifier**

Distribution is estimated as (which is a naïve decomposition)

$$f_k(\vec{x}) = f_k(x_1)f_k(x_2)\dots f_k(x_p) \tag{9.64}$$

Classification function:

$$\hat{y}(x) = \arg\max_k \hat{\pi}_k \prod_{i=1}^{p} \hat{f}_k(x_i) = \arg\max_k \sum_{i=1}^{k} \pi_k \log \hat{f}_k(x_i) \tag{9.65}$$

9.2.6 Logistic Regression

Logistic Regression calculates $\mathbb{P}(Y|X)$ directly. Detail theory see section 3.7 ~ page 110. Here are some recaps:

$$y|x \sim \text{Binom}\left(1, \frac{e^{x'\beta}}{1+e^{x'\beta}}\right) \tag{9.66}$$

$$\mathbb{P}(Y=1|X=x) = \frac{e^{x'\beta}}{1+e^{x'\beta}} := \text{logit}(x'\beta) \tag{9.67}$$

Classify with thres hold s.

□ **Multiple Classification**

$$\mathbb{P}(Y=k|X=x) = \frac{e^{x'\beta_k}}{1+\sum_{l=1}^{K-1} e^{x'\beta_l}}, \quad k=1,2,\dots,K-1 \tag{9.68}$$

$$\mathbb{P}(Y=K|X=x) = \frac{1}{1+\sum_{l=1}^{K-1} e^{x'\beta_l}} \tag{9.69}$$

Comment on Logistic Regression:

- Classification core $x'\beta$ is linear, so logistic regression is still a linear classifier.
- Classification paramter βs are usually obtained using MLE. Detail see section 5.4.3 ~ page 170.

$$\begin{aligned}\beta^{(t+1)} =& \beta^{(t)} - \left(\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta'}\right)\frac{\partial \ell(\beta)}{\partial\beta} && (9.70)\\ =& \beta^{(t)} + (X'WX)^{-1}X'(Y - \text{logit}(X,\beta^{(t)})), \quad W = \text{diag}\left\{\text{logit}(X,\beta^{(t)}) \odot (1 - \text{logit}(X,\beta^{(t)}))\right\} && (9.71)\end{aligned}$$

▷ R. **Code**

```r
library(glmnet)
glmnet(x, y, family="binomial") # two-class
glmnet(x, y, family="multinomial") # multi-class
glmnet(x, y, family="binomial", alpha, lambda) # with penalty
```

□ **Logistic Regression as Loss-Penalization Method**

Logistic Regression with ℓ_2 norm regularized term is

$$\begin{aligned}&\arg\min_{\beta} \sum_{i=1}^{N} \log \mathbb{P}\left(Y \neq y_i | X = x_i; \beta\right) + \frac{\lambda}{2}\|\beta\|^2 && (9.72)\\ =&\arg\max_{\beta} \sum_{i=1}^{N} \log[1 + e^{y_i f(x_i)}] + \frac{\lambda}{2}\|\beta\|^2, \quad y_i \in \{+1, -1\} && (9.73)\end{aligned}$$

where $f(\cdot)$ is classification function, $\beta_0 + x'\beta$ for linear classification.

Section 9.3 Support Vector Machine

Support vector machine (SVM) classifier was one of the most successful classification model in 2010±, mainly because of the kernel trick method in extending feature space.

First we will consider the linear classification case, i.e. dataset $\mathcal{D} = \{(\vec{x}_i, y_i), i = 1, 2, \ldots, N\}$ are devided by a linear boundary $x'\beta + \beta_0$, where label $y_i \in \{1, -1\}$.

9.3.1 Derivation of Basic Optimize Problem

□ **Hard Margin SVM**

The intuition of SVM is to determine the classification boundary by ensuring all the points are 'far away enough' from the boundary.

$$\begin{aligned}\arg\max_{\beta,\beta_0,M} \quad & M\\ s.t. \quad & \frac{1}{\|\beta\|} y_i(x_i'\beta + \beta_0) \geq M \quad i = 1, 2, \ldots, N\end{aligned}$$

where M for 'Margin', which indicates the distance of point from boundary. L.H.S. of inequality is the distance from x_i to boundary.[4]

However note that the *dof* of this problem is 1, i.e. all $(\beta_0, \beta) \propto (\beta_0^*, \beta^*)$ give the same result. We could omit this *dof* by putting an extra constraint, here a convenient one is used: $\|\beta\| = \frac{1}{M}$. i.e.

$$\begin{aligned} \underset{\beta,\beta_0:M=1/\|\beta\|}{\arg\min} \quad & \frac{1}{2}\|\beta\|^2 \\ s.t. \quad & y_i(x_i'\beta + \beta_0) \geq 1 \quad i = 1, 2, \ldots, N \end{aligned}$$

□ Soft Margin SVM

To tackle the case when $y_i(x_i'\beta + \beta_0) \geq 1$ cannot always been satisfied, use soft margin by inducing a 'slack variable' ξ_i for each point, indicating the proportion of distance that the point enters the margin, see figure 9.1 ~ page 252

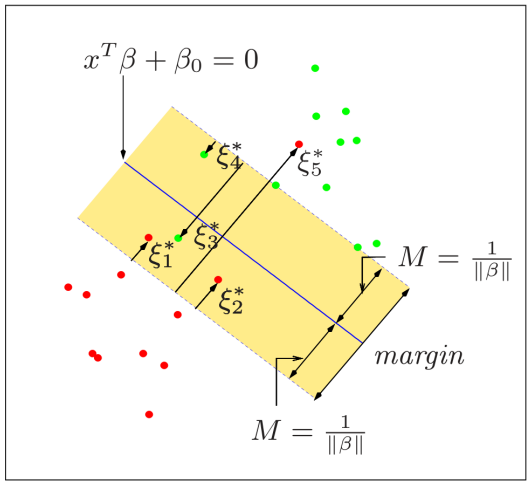


图 9.1: Support Vector Machine Illustration

[4] □ *Proof:* denote some point on $x'\beta + \beta_0 = 0$ as $x_\perp$ (i.e. $x_\perp'\beta + \beta_0 = 0$), then the distance of x to boundary is the projection of $x - x_\perp$ on unit normal vector $\frac{\beta}{\|\beta\|}$:

$$d = \left|(x - x^*)'\frac{\beta}{\|\beta\|}\right| = \frac{1}{\|\beta\|}|x'\beta + \beta_0| \tag{9.74}$$

further because y_i varies at different sides of boundary:

$$y_i = 1 : x'\beta + \beta_0 > 0 \tag{9.75}$$

$$y_i = -1 : x'\beta + \beta_0 < 0 \tag{9.76}$$

we can replace the $|\cdot|$ using label:

$$d = \frac{1}{\|\beta\|}y(x'\beta + \beta_0) \tag{9.77}$$

Primal θ_P:

$$\begin{aligned} \underset{\beta,\beta_0:M=1/\|\beta\|}{\arg\min} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(x_i'\beta+\beta_0) \geq 1-\xi_i \quad i=1,2,\ldots,N \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad i=1,2,\ldots,N \end{aligned}$$

write the generalized lagrange function as defined in equation 5.20 ~ page 147:

$$\mathcal{L}(\beta,\beta_0,\xi_i;\alpha,\mu) = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i\left[1-\xi_i-y_i(x_i'\beta+\beta_0)\right] - \sum_{i=1}^{N}\mu_i\xi_i \tag{9.78}$$

$$s.t. \quad \alpha_i \geq 0, \quad \mu_i \geq 0, \qquad i=1,2,\ldots,N \tag{9.79}$$

dual problem is given when $\dfrac{\partial\mathcal{L}}{\partial\beta,\beta_0,\xi_i} = 0$:

$$\frac{\partial\mathcal{L}}{\partial\beta} = 0 : \hat{\beta} = \sum_{i=1}^{N}\alpha_i y_i x_i \tag{9.80}$$

$$\frac{\partial\mathcal{L}}{\partial\beta_0} = 0 : \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{9.81}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = 0 : C = \alpha_i + \mu_i, \quad i=1,2,\ldots,N \tag{9.82}$$

Dual θ_D:

$$\theta_D(\alpha,\mu) = \min_{\beta,\beta_0,\xi_i}\mathcal{L} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i'x_j + \sum_{i=1}^{N}\alpha_i \tag{9.83}$$

$$s.t. \quad 0 \leq \alpha_i \leq C \tag{9.84}$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{9.85}$$

we can maximize θ_D to obtain $\hat{\alpha}_i$, $\hat{\mu}_i = C - \hat{\alpha}_i$. And $(\hat{\beta},\hat{\beta}_0,\xi_i)$ are given utilizing KKT condition for $d^* = \max\limits_{\alpha,\mu}\theta_D = \min\limits_{\beta,\beta_0,\xi_i}\theta_P = p^*$:

$$\hat{\alpha}_i\left[1-\hat{\xi}_i-y_i(x_i'\hat{\beta}+\hat{\beta}_0)\right] = 0 \tag{9.86}$$

$$(C-\hat{\alpha}_i)\hat{\xi}_i = 0 \tag{9.87}$$

$$1-\hat{\xi}_i-y_i(x_i'\hat{\beta}+\hat{\beta}_0) \leq 0 \tag{9.88}$$

$$0 \leq \hat{\alpha}_i \leq C \tag{9.89}$$

$$\hat{\xi}_i \geq 0 \tag{9.90}$$

$$\hat{\beta} = \sum_{i=1}^{N}\hat{\alpha}_i y_i x_i \tag{9.91}$$

discussion on different cases of α_i, ξ_i:

$$\hat{\alpha}_i = 0 : \hat{\xi}_i = 0 \tag{9.92}$$

$$\hat{\alpha}_i = C : y_i(x_i\hat{\beta} + \hat{\beta}_0) = 1 - \hat{\xi}_i \tag{9.93}$$

$$0 < \hat{\alpha}_i < C : \hat{\xi}_0 = 0,\ y_i(x_i\hat{\beta} + \hat{\beta}_0) = 1 \tag{9.94}$$

where all points $\mathcal{I}^{\text{sv}} := \{i^{\text{sv}} \big| 0 < \hat{\alpha}_{i^{\text{sv}}} < C, \hat{\xi}_{i^{\text{sv}}} = 0\}$ are called '**support vector**', that can be used to determine β_0:

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i = \sum_{i \in \mathcal{I}^{\text{sv}}} \hat{\alpha}_i y_i x_i \tag{9.95}$$

$$\hat{\beta}_0 = y_{i^{\text{sv}}} - x'_{i^{\text{sv}}}\hat{\beta} \tag{9.96}$$

9.3.2 Support Vector Machine as Loss-Penalization Method

SVM Primal can be express in equivalent form with $f(x_i)$ as prediction function, e.g. $f(x_i) = \beta_0 + x_i'\beta$ for linear SVM:

$$\begin{cases} \xi_i \geq 0 \\ \xi_i \geq 1 - y_i f(x_i) \end{cases} \Rightarrow \xi_i \geq \max\{0, 1 - y_i f(x_i)\} = [1 - y_i f(x_i)]_+ \tag{9.97}$$

in which $[\,\cdot\,]_+ \equiv \max\{0, \cdot\}$ is hinge loss:

$$\arg\min_{\beta, \beta_0} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2, \qquad \lambda = \frac{1}{C}, \quad f(x_i) = \beta_0 + x_i'\beta \tag{9.98}$$

which is naturally in an $\arg\min_f \sum_{i=1}^{N} \mathcal{L}\left(x_i, y_i, f(x_i)\right) + \frac{\lambda}{2}\mathcal{P}(f(\,\cdot\,))$ Loss+Penalty form.

Section 9.4 Feature Expansion and Kernel Methods

Motivation: Map the data point $x \in \mathcal{X}(e.g.\ = \mathbb{R}^p)$ to another feature space $\mathcal{F}(e.g.\ = \mathbb{R}^M)$ (not necessarily a linear transform, usually $M > p$, or just proper to describe the features). The mapping function lies in a Hilbert space $\mathcal{H}$ of function:

$$h(\,\cdot\,) = (h_1(\,\cdot\,), h_2(\,\cdot\,), \ldots, h_M(\,\cdot\,))' \in \mathcal{H} :\ \mathcal{X} \to \mathcal{F} \tag{9.99}$$

and we can construct model in feature space.

9.4.1 Reproducing Kernel Hilbert Space and The Representer Theorem

Based on the idea of feature space, make a step forward: the key focus of model is actually 'measuring space structure by similarity between points' rather than having to define a feature space. i.e. describe similarity by a bi-linear **Kernel Function**

$$K(x, x') \in \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{9.100}$$

In intuition for Kernel is an 'inner product kernel'. The Kernel corresponds a kind of inner product structure on $\mathcal{H}$, the kernel should satisfies the following properties:

1. Positive Semi-Definition:

$$\iint K(x,y)g(x)g(y)\,\mathrm{d}x\mathrm{d}y \geq 0, \quad \forall g(\,\cdot\,) \tag{9.101}$$

or an equivalent form:

$$\sum_{i,j=1}^{n} K(x_i,x_j)a_i a_j \geq 0, \quad \forall \{x_i\}_{i=1}^{m}, \{a_i\}_{i=1}^{n}, \quad \forall n \in \mathbb{Z}^+ \tag{9.102}$$

2. Symmetry:

$$K(x,y) = K(y,x) \tag{9.103}$$

Eigenvalue γ_i and eigen function $\phi_i(x)$ of Kernel:

$$\int_x K(x,y)\phi_i(y)\,\mathrm{d}y = \gamma_i\phi_i(x) \tag{9.104}$$

In Hilbert space, the eigen functions are orthonormal:

$$\langle \phi_i, \phi_j \rangle = \int_x \phi_i(x)\phi_j(x)\,\mathrm{d}x = \delta_{ij} \tag{9.105}$$

And Kernel $K(x,y)$ could be represented from its eigen value and eigen function:

$$K(x,y) = \sum_i \gamma_i\phi_i(x)\phi_i(y) \tag{9.106}$$

which is Mercer's Theorem: Semi-positive definite symmetric kernel could be expressed as an inner product form. Such a form is also called the kernel trick because it usually avoid calculating inner product in high dimensional space.

□ **Reproducing Kernel Hilbert Space (RKHS)** Now use set $\{\phi_i\}$ as the orthonormal base to form a Hilbert space $\mathcal{H}_K = \mathrm{span}\{\phi_i\}$ i.e. any function $f \in \mathcal{H}_K$ could be expressed as expansion

$$\mu(x) = \sum_i \mu_i\phi_i(x) \tag{9.107}$$

The inner product defined for this Hilbert space is[5]

$$\left\langle \sum_i \mu_i\phi_i(x), \sum_i \nu_i\phi_i(x) \right\rangle_{\mathcal{H}_K} = \sum_i \frac{\mu_i\nu_i}{\gamma_i} \tag{9.108}$$

and norm induced by inner product

$$\|f\|_{\mathcal{H}_K} = \sum_i \frac{f_i^2}{\gamma_i}, \qquad f(x) = \sum_i f_i\phi_i(x) \tag{9.109}$$

[5]Hilbert space is complete linear space with inner product defined.

Note: when x is fixed, $f_x(y) = K(x,y)$ is a function of y, and vice versa. Use the above expansion and inner product:

$$K(x,y) = \sum_i \gamma_i \phi_i(x)\phi_i(y) \tag{9.110}$$

$$= \sum_i \sqrt{\gamma_i}\phi_i(x)\sqrt{\gamma_i}\phi_i(y) \tag{9.111}$$

$$= \sum_i \frac{(\gamma_i\phi_i(x))(\gamma_i\phi_i(y))}{\gamma_i} \tag{9.112}$$

$$= \left\langle \sum_i \gamma_i\phi_i(x)\phi_i(\xi), \sum_i \gamma_i\phi_i(y)\phi_i(\xi) \right\rangle_{\mathcal{H}_K} \tag{9.113}$$

$$= \langle K(x,\xi), K(\xi,y)\rangle_{\mathcal{H}_K} \tag{9.114}$$

which is the reproducing property of Kernel $K(\,\cdot\,,\,\cdot\,)$ and its corresponding Hilbert space $\mathcal{H}_K$

□ **Representer Theorem for RKHS**

With Kernel and its corresponding RKHS defined, we could write a optimization problem as loss+penalty form:

$$\arg\min_{f\in\mathcal{H}_K} \sum_{i=1}^N \mathcal{L}(y_i, f(x_i)) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} \tag{9.115}$$

Representer Theorem: Solution to above optimization has a **finite** form

$$\hat{f}(x) = \sum_{i=1}^N \hat{\alpha}_i K(x, x_i) \tag{9.116}$$

i.e. we can optimize over $\{\hat{\alpha}_i\}_{i=1}^N$, instead of optimizing over $\{f_i\}_{i=1}^\infty$.

norm of $\hat{f}$ is represented as

$$\left\|\hat{f}\right\|^2_{\mathcal{H}_K} = \left\langle \sum_{i=1}^N \hat{\alpha}_i K(x,x_i), \sum_{i=1}^N \hat{\alpha}_i K(x,x_i) \right\rangle_{\mathcal{H}_K} \tag{9.117}$$

$$= \sum_{i=1}^N \sum_{j=1}^N \hat{\alpha}_i\hat{\alpha}_j K(x_i,x_j) \tag{9.118}$$

Optimization problem equation 9.115 ~ page 256 is parameterized by $\{\hat{\alpha}_i\}_{i=1}^N$:

$$\arg\min_{\{\hat{\alpha}_i\}_{i=1}^N\in\mathbb{R}^N} \sum_{i=1}^N \mathcal{L}(y_i, \sum_{j=1}^N \hat{\alpha}_j K(x_i,x_j)) + \frac{\lambda}{2}\sum_{i=1}^N\sum_{j=1}^N \hat{\alpha}_i\hat{\alpha}_j K(x_i,x_j) \tag{9.119}$$

Or written in matrix form $y = (y_1, y_2, \ldots, y_N)$ $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$, $K = \{K(x_i,x_j)\}_{i,j=1}^N$:

$$\arg\min_{\alpha\in\mathbb{R}^N} \sum_{i=1}^N \mathcal{L}(y, K\alpha) + \frac{\lambda}{2}\alpha' K\alpha \tag{9.120}$$

Classification criterion

$$\hat{f}(x) = \sum_{i=1}^N \hat{\alpha}_i K(x, x_i) \tag{9.121}$$

9.4.2 Useful Kernel

Some useful Kernel for numeric vector x:

- Linear Kernel (identity):

$$K(x,y) := \langle x,y\rangle \tag{9.122}$$

- d^{th} Degree Polynomial Kernel:

$$K(x,y) := (1+\langle x,y\rangle)^d \tag{9.123}$$

- Radical Base Function Kernel:

$$K(x,y) := \exp\left[-\frac{\|x-y\|^2}{\sigma^2}\right] \tag{9.124}$$

- Sigmoid Kernel:

$$K(x,y) = \tanh\left(1+\langle x,y\rangle\right) \tag{9.125}$$

Note that equation 9.119 ~ page 256 includes Kernel $K(\cdot,\cdot)$ only, thus Kernel trick could be applied to various scenarios once we could define a proper Kernel. e.g. Substring Kernel for string sequence.

9.4.3 Kernel Support Vector Machine

Replace the inner produce term in Dual problem of SVM equation 9.83 ~ page 253 into Kernel function to obtain Kernel SVM:

$$\arg\max_{\alpha} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{9.126}$$

$$s.t.\ 0\le\alpha_i\le C, \qquad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{9.127}$$

$$\hat{f}(x) = \sum_{i\in\mathcal{I}^{\text{sv}}}\alpha_i y_i K(x,x_i) \tag{9.128}$$

Or use the loss+penalization primmal form of SVM:

$$\arg\min_{\hat{\alpha}} \sum_{i=1}^{N}\left[1-y_i\sum_{j=1}^{N}\hat{\alpha}_j K(x_i,x_j)\right]_+ + \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{\alpha}_i\hat{\alpha}_j K(x_i,x_j), \quad \lambda=\frac{1}{C} \tag{9.129}$$

$$\hat{y}(x) = \begin{cases} 1 & ,\hat{f}(x)=\sum_{i=1}^{N}\hat{\alpha}_i K(x,x_i)\ge s \\ -1 & ,\hat{f}(x)=\sum_{i=1}^{N}\hat{\alpha}_i K(x,x_i)< s \end{cases} \tag{9.130}$$

Note: Here α_i and $\hat{\alpha}_i$ are not the same set of number, but the optimization problems are the same (if $\{K(x_i,x_j)\}$ is non-singular).

9.4.4 SMO Algorithm for Kernel SVM*

Idea: reduction from N-dim optimization to many 2-dim optimization.

9.4.5 Kernel Regression

□ **Kernel Regression with Squared Error Loss**

Recall linear regression with Squared loss and penalty

$$\arg\min_{\beta_0,\beta} \sum_{i=1}^{N} \left[y_i - \beta_0 - x_i'\beta\right]^2 + \frac{\lambda}{2}\|\beta\|_2^2 \tag{9.131}$$

replace linear classification $f(x) = \beta_0 + x'\beta$ by Kernel $K\alpha$:

$$\arg\min_{\hat{\alpha}} (y - K\hat{\alpha})'(y - K\hat{\alpha}) + \frac{\lambda}{2}\hat{\alpha}' K\hat{\alpha} \tag{9.132}$$

Solution is similar to ridge regression form:

$$\hat{\alpha} = (K + \lambda I)^{-1} y \tag{9.133}$$

□ **Kernel Logistic Regression**

In logistic regression, the loss function is binomial deviance $\log\left[1 + e^{-yf(x)}\right]$

$$\arg\min_{\hat{\alpha}} \sum_{i=1}^{N} \log\left[1 + e^{y_i \sum_{j=1}^{N} \hat{\alpha}_j K(x_i, x_j)}\right] + \frac{\lambda}{2} \sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\alpha}_i \hat{\alpha}_j K(x_i, x_j) \tag{9.134}$$

$$\hat{y}(x) = \begin{cases} 1 & , \hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i K(x, x_i) \geq s \\ -1 & , \hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i K(x, x_i) < s \end{cases} \tag{9.135}$$

Section 9.5 Clustering

Clustering is an important scenario of unsupervised learning $\mathcal{D} = \{x_i\}_{i=1}^N$, to cluster 'similar' data points into the same group.

9.5.1 Proximity Matrix

For separation concern, we should first define some metric to measure similarity between data

$$d_{ij} = D(x_i, x_j) \tag{9.136}$$

common usage of distance measure see section 4.7 ~ page 138

And form the proximity matrix W:

$$D = \{d_{ij}\}_{i,j=1}^N \tag{9.137}$$

Usually some clustering algorithm would claim some properties:

- Non-negative element and non-zero diagonal:

$$d_{ij} \geq 0, \forall i, j. \qquad d_{ii} = 0, \forall i \tag{9.138}$$

- Symmetry

$$D = D^T \tag{9.139}$$

Overall dissimilarity:

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} D(x_i, x_j) \tag{9.140}$$

□ **Optimizing Goal of Clustering**

With similarity/dissimilarity defined, clustering target could be expressed as maximizing within cluster scatter/ minimizing between cluster scatter, with respect to clustering group $C(\cdot)$

$$\arg\max_{C(\cdot)} \frac{1}{2} \sum_{k=1}^{K} \sum_{i:C(x_i)=k} \sum_{j:C(x_j)=k} D(x_i, x_j) \tag{9.141}$$

$$\arg\min_{C(\cdot)} \frac{1}{2} \sum_{k=1}^{K} \sum_{i:C(x_i)=k} \sum_{j:C(x_j)\neq k} D(x_i, x_j) \tag{9.142}$$

The two forms are equivalent due to a fixed sum:

$$\frac{1}{2} \sum_{k=1}^{K} \sum_{i:C(x_i)=k} \sum_{j:C(x_j)=k} D(x_i, x_j) + \frac{1}{2} \sum_{k=1}^{K} \sum_{i:C(x_i)=k} \sum_{j:C(x_j)\neq k} D(x_i, x_j) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} D(x_i, x_j) := T = \text{const} \tag{9.143}$$

Usually search for cluster assignment is based on iterative greedy descent search.

Some frequently used clustering methods were included in section 4.7 ~ page 138

- Hierarchical Method
- K-Means
- EM-Gaussian Mixture Model
- DBSCAN & OPTICS Density Method

In this section, an extra model based on spectrum is introduced

9.5.2 Spectrum Clustering

Express the dataset as a Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = \{v_i\}_{i=1}^N$ for vertex, $\mathcal{E} = \{e_{ij}\}_{i,j=1}^N$, $\mathcal{W} = \{w_{ij}\}_{i,j=1}^N$ for edges and weights. In this case cluster is a graph partition problem.

□ **Graph Laplacian**

Some definition:

- Degree of vertex:

$$d_i = \sum_{j=1}^{N} w_{ij} \tag{9.144}$$

- Degree matrix

$$D = \text{diag}\{d_1, d_2, \ldots, d_N\} \tag{9.145}$$

- Unnormalized graph Laplacian:

$$L := D - W \tag{9.146}$$

is symmetric and semi-positive definite

$$\xi' L \xi = \sum_{i,j=1}^{N} w_{ij}(\xi_i - \xi_j)^2 \geq 0, \quad \forall \xi \in \mathbb{R}^N \tag{9.147}$$

Spectrum is based on studying the eigen vector and eigen value of L.

- For any graph Laplacian $\underset{m\times m}{L}$, $\mathbf{1}_m$ is a eigen vector with eigen value 0
- In the case that $\mathcal{G}$ is **not** fully connected, with K subgraph $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_K\}$, i.e. W and L could be written in diagonal form (usually need some row/column transformation)

$$L = \begin{bmatrix} L_1 & 0 & \ldots & 0 \\ 0 & L_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & L_K \end{bmatrix} \tag{9.148}$$

the multiplicity of eigen value 0 is K, with each eigen vector as

$$\mathbf{1}_{\mathcal{G}_k} = [\mathbb{I}(v_1 \in \mathcal{G}_k), \ldots, \mathbb{I}(v_N) \in \mathcal{G}_k], \quad k = 1, 2, \ldots, K \tag{9.149}$$

- In real world case, the graph could probably expressed as a small deviance from a graph with subgraph:

$$L = \begin{bmatrix} L_1 & 0 & \ldots & 0 \\ 0 & L_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & L_K \end{bmatrix} + \underset{\delta}{N \times N} \tag{9.150}$$

where we would expect the smallest K eigen value $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_K$ corresponds to the K cluster we want.

Algorithm *Spectral Clustering*

1. Compute $\underset{N\times N}{L}$

---

2. Determine the K smallest eigen values $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_K$ with eigen vector u_i, $i = 1, 2, \ldots, K$

$$U = [u_1, u_2, \ldots, u_K] = \begin{bmatrix} u_{11} & u_{12} & \ldots & u_{1K} \\ u_{21} & u_{22} & \ldots & u_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \ldots & u_{NK} \end{bmatrix} = [z_1, z_2, \ldots, z_N]^T \tag{9.151}$$

$$z_i = [u_{i1}, u_{i2}, \ldots, u_{iK}]^T, \quad i = 1, 2, \ldots, N \tag{9.152}$$

3. Cluster $\{z_i\}_{i=1}^N$ with e.g. K-Means.

Choice of normalized graph Laplacian, would cause different cluster results:

- Ratio Cut $L = I - D^{-1}W$

$$\underset{\{\mathcal{G}_1, \ldots, \mathcal{G}_K\}}{\arg\min} \frac{1}{2} \sum_{i=1}^{K} \frac{\text{Bet}(\mathcal{G}_i, \mathcal{G}_i^{\complement})}{|\mathcal{G}_i|} \tag{9.153}$$

- Normalized Cut $L = I - D^{-1/2}WD^{-1/2}$

$$\underset{\{\mathcal{G}_1, \ldots, \mathcal{G}_K\}}{\arg\min} \frac{1}{2} \sum_{i=1}^{K} \frac{\text{Bet}(\mathcal{G}_i, \mathcal{G}_i^{\complement})}{\sum_{i \in \mathcal{G}_i} \sum_{j \in \mathcal{G}} d_{ij}} \tag{9.154}$$

Section 9.6 Tree-Based Classification Model

Idea of tree: divide the space $\mathcal{X}$ into grids R_m and assign prediction into the most frequent class

$$\hat{f}(x \in R_m) = \underset{k}{\arg\max} \sum_{x_i \in R_m} \mathbb{I}(y_i = k) \tag{9.155}$$

But such method is not practical in high dimensional due to curse of dimensionality. Nore practical method would be a greedy search, each step along one variable.

9.6.1 Tree-Based Classification

□ Branch Growing Process

Grow branch on a node

Algorithm *Classification Tree*

In each branck growing on a node:

1. Look for a splitting variable x_j and split value s:

$$\underset{j,s}{\arg\min} \left[N_{\text{left}} \text{ImPu}(x_i \in R_{\text{left}}(j, s)) + N_{\text{right}} \text{ImPu}(x_i \in R_{\text{right}}(j, s)) \right] \tag{9.156}$$

$$R_{\text{left}}(j, s) = \{x : x_j \leq s\}, \quad R_{\text{right}}(j, x) = \{x : x_j > s\} \tag{9.157}$$

useful impurity measure $\text{ImPu}(\{x\})$ with $p_k(X = \{x\})$ defined

$$p_k(X) = \frac{\sum_{x \in X} \mathbb{I}(C(x) = k)}{|X|} \tag{9.158}$$

- Misclassification rate

$$1 - \max_k p_k \tag{9.159}$$

- Gini impurity

$$\sum_{k=1}^{K} p_k(1-p_k) = \sum_{k=1}^{K} \sum_{k' \neq k} p_k p_{k'} \tag{9.160}$$

Gini impurity with category weight $\underset{K\times K}{W_K} = \{w_{kk'}\}_{k,k'=1}^{K}$

$$\sum_{k=1}^{K} \sum_{k' \neq k} w_{kk'} p_k p_{k'} \tag{9.161}$$

- Entropy

$$-\sum_{k=1}^{K} p_k \log p_k \tag{9.162}$$

2. usually the process ends when

$$|\text{node}| \leq \text{const}, \quad \forall \text{node} \tag{9.163}$$

3. Apply cost complexity pruning strategy

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m \text{ImPu}(R_m) + \alpha|T| \tag{9.164}$$

where T is tree, $|T|$ for number of nodes in the tree.

Comment:

- Tree methods is well-interpreted, especially similar to a natural desicion making process
- Handle non-linear classification pattern
- Unstable to data.

Performance of tree classification could be largely improved with bagging method and boosting method.

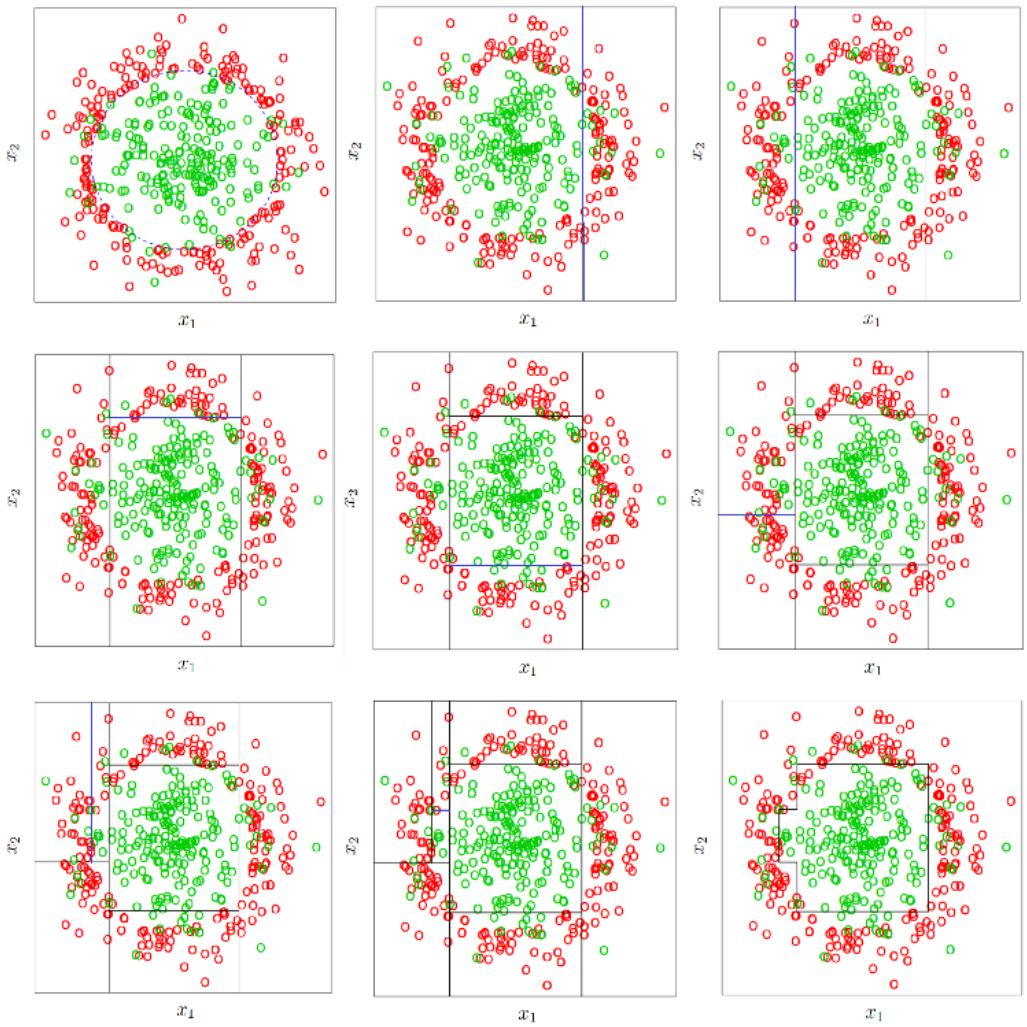


图 9.2:

9.6.2 Bagging and Boosting

□ **Bagging**

Bagging is short for **B**ootstrap **Agg**regation. Idea: for B boostrapped training data, the boostrapping result

$$\hat{f}_{\text{boot}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x) \text{ or } = \arg\max_k \sum_{b=1}^{B}\mathbb{I}(\hat{f}_b(x)=k) \tag{9.165}$$

□ **Random Forest**

Random Forest aims at decorrelating trees to reduce variance when averaging trees.

Algorithm *Random Forest Bagging*

1. Generate B different boostrapped training data. (*random 1* by bootstrap sampling)
2. For each sample, grow a tree. In each split of tree (i.e. a branch growth), $q \approx \sqrt{p}$ variable components are randomly selected for classification. (*random 2* by randomizing components)
3. Take average or vote of all B trees as the final result

Comment: A prune is usually needed, cause variance is reduced by averaging.

□ **Boosting**

Idea: Fitting result of previous trees could be used to modify following trees. The error rate of each tree would influence the vote weight when bagging the results.

Algorithm *Adaboost*

1. Each observant is given weights $w_i^{(0)} = \dfrac{1}{N}$, $i = 1, 2, \ldots, N$
2. For $m = 1 : M$, M for loops of boosting:
   (a) Grow a tree $T^{(m)}(x)$ with weight $w_i^{(m)}$
   (b) Compute **error rate**

$$\text{err}^{(t)} := \frac{\sum_{i=1}^{N} w_i^{(m)} \mathbb{I}(y_i \neq T^{(m)}(x_i))}{\sum_{i=1}^{N} w_i^{(m)}} \tag{9.166}$$

   and define

$$\alpha^{(m)} = \log\left[(1 - \text{err}^{(m)})/\text{err}^{(m)}\right] \tag{9.167}$$

   (c) Reset weights by

$$w_i^{(m+1)} = w^{(m)} \cdot \exp\left[\alpha^{(m)} \mathbb{I}(y_i \neq T^{(m)}(x_i))\right] \tag{9.168}$$

3. Output

$$\hat{f}(x) = \text{sgn}\left[\sum_{m=1}^{M} \alpha^{(m)} T^{(m)}(x)\right] \tag{9.169}$$

Section 9.7 Neural Network

□ **Linear Perceptron with Activate Function**

Usually linear perceptron is used as a neuron in neutral network:

$$y = g(w_0 + w_1 x_1 + \ldots + w_p x_p) = g(x'w), \quad x_0 \equiv 1 \tag{9.170}$$

where $g(\cdot)$ is activate function. Such Perceptron could be optimized by gradient

Some useful activate function:

- Linear Threshold Unit (LTU)

$$g(\xi) = \begin{cases} 0, & \xi < 0 \\ 1, & \xi \geq 0 \end{cases} = \eta(\xi) \tag{9.171}$$

- Logistic Function

$$g(\xi) = \frac{1}{1 + e^{-\xi}} \tag{9.172}$$

- Hyperbolic Tangent Function

$$g(\xi) = \tanh \xi = \frac{e^{2\xi} - 1}{e^{2\xi} + 1} \tag{9.173}$$

- Rectified Linear Unit (ReLU)

$$g(\xi) = \begin{cases} 0, & \xi < 0 \\ \xi, & \xi \geq 0 \end{cases} \tag{9.174}$$

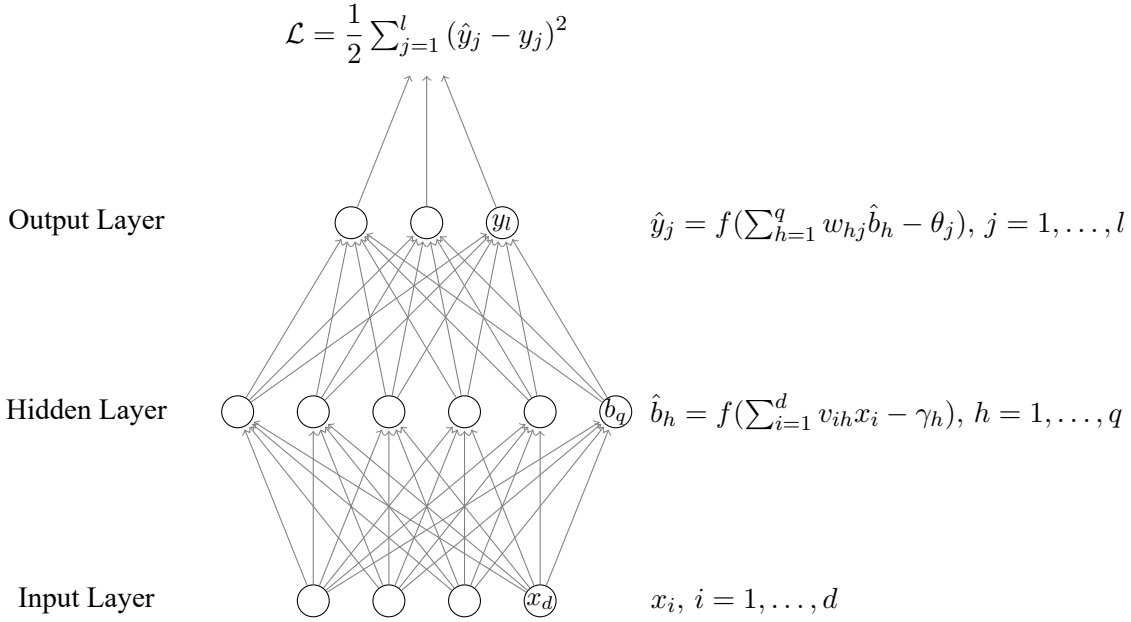


图 9.3: Structure of Feed-Forward Neural Network (1 Layer)

A MonoLayer perceptron with enough neurons (hidden units) could represent any coutinuous function. MultiLayer Perceptron (MLP) could even represent discontinuous functions.

9.7.1 Back Propagation

Perceptron system is usually optimized by back propagation (of gradient).

An example to optimize v_{ih}, γ_h in figure 9.3 ~ page 265:

$$\frac{\partial \mathcal{L}}{\partial v_{ih}} = \sum_{j=1}^{l} \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \hat{b}_h} \frac{\partial \hat{b}_h}{\partial v_{ih}} \tag{9.175}$$

$$= \sum_{j=1}^{l} \hat{y}_j (\hat{y}_j - y_j) \cdot \frac{\partial f(u)}{\partial u}\Big|_{u=\sum w_{hj}\hat{b}_h - \theta_j} w_{hj} \cdot \frac{\partial f(v)}{\partial v}\Big|_{v=\sum_{i=1}^{d} v_{ih} x_i - \gamma_h} x_i \tag{9.176}$$

$$\frac{\partial \mathcal{L}}{\partial \gamma_h} = \sum_{j=1}^{l} \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \hat{b}_h} \frac{\partial \hat{b}_h}{\partial \gamma_h} \tag{9.177}$$

$$= \sum_{j=1}^{l} \hat{y}_j (\hat{y}_j - y_j) \cdot \frac{\partial f(u)}{\partial u}\Big|_{u=\sum w_{hj}\hat{b}_h - \theta_j} w_{hj} \cdot \frac{\partial f(v)}{\partial v}\Big|_{v=\sum_{i=1}^{d} v_{ih} x_i - \gamma_h} \cdot (-1) \tag{9.178}$$

9.7.2 Neural Tangent Kernel*

Chapter. X　应用时间序列部分

Instructor: Dong Li

Section 10.1　Time Series Data and Model

10.1.1　Time Series Data and Tasks

Time Series : a sequential r.v. indexed in time order.

$$\{Y_t\},\ t \in \mathcal{T} \quad \mathcal{T} \text{ is index set} \tag{10.1}$$

and actual data of time series, i.e. times series data is called a **Realization** of time series, denoted[1]

$$\{y_t\},\ t \in T \subset \mathcal{T} \tag{10.2}$$

e.g. in forecasting task, T encodes history. In this chapter we usually focus on easier case of arithmetic progression $T = \{1, 2, \ldots, N\}$, or at least numeric orderal sequence.

Time Series Analysis (TSA): Analysis on time series data to extract meaningful statistics/other characteristics. Task of TSA includes:

- Describing and Explanaing the machanism of time series
- Forecasting
- Guiding the intervention of Time Series

In this section several modelling/forecasting methods would be included.

10.1.2　Time Series Model

There are plenty of useful modelling methods:

- Regression Model: View y as function of t, regression on some model $y = f(t)$ with loss $\mathcal{L}$. e.g. linear regression

$$y = \beta_0 + \beta_1 t + \varepsilon, \quad \mathcal{L} = \sum_{t \in T} (y_t - \hat{y}_t)^2 \tag{10.3}$$

Modelling strategy is similar to that introduced in linear regression, see Chapter 3 ~ page 71

[1] A note on $T \subset \mathcal{T}$: actually T has to be discrete beacuse it is a sample of $\mathcal{T}$. while $\mathcal{T}$ is not necessarily defined as discrete.

- STL Method: Seasonal and Trend decomposition using Loess. A decomposition of time series into 'TS = Trend + Season + Random', i.e.

$$Y_\tau = T_\tau + S_\tau + X_\tau \tag{10.4}$$

  and we could model T_τ, S_τ, X_τ separately. The focus is the modelling of random term X_t, which we expect to be 'stationarily random' through time. (Usually we model this part also by ARMA model)

- Exponential Smoothing Model: Use weighted average over history to predict future.
- ARIMA Model: The main focus of this chapter.

Section 10.2 Stochastic Process and Statistics

10.2.1 Basic Knowledge of Stochastic Process

A stochastic process can be denoted:

$$\{X_t : t \in \mathcal{T}\} : \Omega \to \mathcal{T} \times \mathcal{E} \tag{10.5}$$

i.e. the random 'variable' of stochastic process is a function $X(t) \in L^2(\mathcal{T})$

□ **Some important cases of stochastic process:**

- i.i.d. sequence: ε_t i.i.d. $\sim \varepsilon$
- White Noise: uncorrelated for different subscript t in the sense of 2nd moment, $\varepsilon_t \sim \text{WN}(\mu, \sigma^2)$. where

$$\mathbb{E}(\varepsilon_t) = \mu \tag{10.6}$$

$$cov(\varepsilon_t, \varepsilon_s) = \sigma^2 \delta_{t,s} \tag{10.7}$$

  Further we can append more constraints on WN:

  + $\{\varepsilon_t\}$ independent: independent white noise $\varepsilon_t \sim \text{IWN}(\mu, \sigma^2)$
  + $\mu = 0$: zero-mean white noise $\varepsilon_t \sim \text{WN}(0, \sigma^2)$
  + $\mu = 0$, $\sigma^2 = 1$: standard white noise $\varepsilon_t \sim \text{WN}(0, 1)$
  + $\varepsilon \sim N(\mu, \sigma^2)$: normal white noise.

- Martingale difference sequence (MDS): zero expectation given history information: $\varepsilon_t \sim$ MDS, where

$$\mathbb{E}(|\varepsilon_t|) < \infty \tag{10.8}$$

$$\mathbb{E}(\varepsilon_t | \mathcal{F}_{t-1}) = 0 \tag{10.9}$$

  where $\mathcal{F}_\tau$ denotes the history until time τ:

$$\mathcal{F}_\tau \equiv \sigma(\varepsilon_s,\ s \le \tau)\{\varepsilon_s, \varepsilon_{s-1}, \varepsilon_{s-2}, \ldots\} \tag{10.10}$$

Relation: i.i.d. $>$ MDS $>$ WN $>$ Stationary

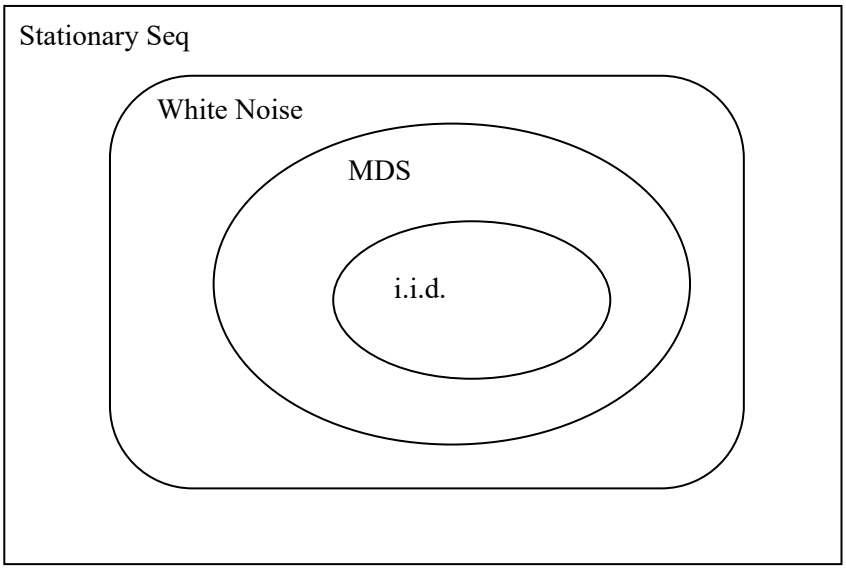


图 10.1: Relation bet. Sequences

□ **Measure of dependence within stochastic process**

Given a stochastic process $\{X_t : t \in \mathcal{T}\}$

- Mean Function

$$\text{Mean: } \mu_t \equiv \mathbb{E}(X_t), \qquad \mathcal{T} \mapsto \mathbb{R} \tag{10.11}$$

- AutoCovariance Function (ACVF) and AutoCorrelation Function (ACF):

$$\text{ACVF: } \gamma_{t,s} \equiv cov(X_t, X_s), \qquad \mathcal{T} \times \mathcal{T} \mapsto \mathbb{R} \tag{10.12}$$

$$\text{ACF: } \rho_{t,s} \equiv corr(X_t, X_s) = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}}, \qquad \mathcal{T} \times \mathcal{T} \mapsto [-1, 1] \tag{10.13}$$

- Stationarity: Stationarity is a measure that the 'correlation structure of stochastic process looks the same' at any time t, i.e. is stationary through time.
  - Weakly Stationary (WS): given $\mathbb{E}\left(X_t^2\right) < \infty$, has const $\mathbb{E}[]$ and cov (independent of time)

$$\mathbb{E}(X_t) = \mu_t = \mu \tag{10.14}$$

$$cov(X_t, X_t + k) = \gamma_{t,t+k} = \gamma_k \perp\!\!\!\perp t \tag{10.15}$$

  - Strictly Stationary (SS): joint distribution invariant through time. For any given $\{t_1, t_2, \ldots, t_n\} \subset \mathcal{T}$

$$f_{X_{t_1}, X_{t_2}, \ldots, X_{t_n}} = f_{X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_n+h}}, \quad \forall h \tag{10.16}$$

Some note on WS and SS:

  - Generally speaking, WS and SS are not equivalant, WS $\nLeftrightarrow$ SS (note that SS does not put constraint on $\mathbb{E}\left(X_t^2\right)$)

– equivalent for gaussian stochastic process.

– ACF and ACVF of WS:

$$\gamma_{t,t+k} = \gamma_k = \gamma_{-k}, \quad \forall t \in \mathcal{T} \tag{10.17}$$

$$\rho_{t,t+k} = \rho_k = \frac{\gamma_k}{\gamma_0}, \quad \forall t \in \mathcal{T} \tag{10.18}$$

Notation of ACVF matrix:

$$\Gamma_k = \{\gamma_{i-j}\}_{i,j=1}^{k} = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{k-2} & \gamma_{k-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{k-3} & \gamma_{k-2} \\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \gamma_{k-4} & \gamma_{k-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{k-2} & \gamma_{k-3} & \gamma_{k-4} & \cdots & \gamma_0 & \gamma_1 \\ \gamma_{k-1} & \gamma_{k-2} & \gamma_{k-3} & \cdots & \gamma_1 & \gamma_0 \end{bmatrix}_{k\times k} \tag{10.19}$$

Γ_k is semi-positive definite.

$$\sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_i \alpha_j \gamma_{|t_i - t_j|} \geq 0, \quad \forall k, \{t_1, \ldots, t_k\}, \vec{\alpha} \tag{10.20}$$

- Partial Autocorrelation (PACF): correlation given information between two time points, orginal definition

$$\phi_{11} = \phi_1 \tag{10.21}$$

$$\phi_{kk} = corr\big(X_t - L(X_t|X_{t+1}, \ldots, X_{t+k-1}), X_{t+k} - L(X_{t+k}|X_{t+1}, \ldots, X_{t+k-1})\big), \quad k \geq 2 \tag{10.22}$$

where $L(X_\tau|X_{t+1}, \ldots, X_{t+k-1})$ is the **Best Linear Estimation** of linear model

$$X_\tau = \beta_0 + \beta_1 X_{t+1} + \ldots + \beta_{k-1} X_{t+k-1} + \epsilon \tag{10.23}$$

deduction:

– Best MMSE linear estimation $\hat{X}_\tau \equiv L(X_\tau|X_{t+1}, \ldots, X_{t+k-1})$ satisfies [2]

$$\{\beta_0, \beta\} = \underset{\beta_0,\beta}{\arg\min}\, \mathbb{E}\left(\hat{X}_\tau - \beta_0 - \sum_{j=1}^{k-1} \beta_j X_{t+j}\right)^2 \tag{10.24}$$

solution: denote $X = (X_{t+1}, \ldots, X_{t+k-1})$, $\beta = (\beta_1, \ldots, \beta_{k-1})$

$$\hat{\beta} = \Sigma_X^{-1}\Sigma_{X,X_\tau} \tag{10.25}$$

$$\hat{\beta}_0 = \mathbb{E}(X_\tau) - \mathbb{E}(X)'\hat{\beta} \tag{10.26}$$

i.e.

$$L(X_\tau|X_{t+1}, \ldots, X_{t+k-1}) = \mathbb{E}(X_\tau) + \Sigma_{X_\tau,X}\Sigma_X^{-1}(X - \mathbb{E}(X)) \tag{10.27}$$

Simplified case for zero-mean Weakly Stationary $\mathbb{E}(X_t) = \mu$; γ_k, Γ_k

$$L(X_{t+k}|X_{t+1}, \ldots, X_{t+k-1}) = \mathbb{E}\big(X_{t+k}\big) + \Sigma_{X_{t+k},X}\Sigma_X^{-1}(X - \mathbb{E}(X)) \tag{10.28}$$

$$= \gamma_{k-1}'\Gamma_{k-1}^{-1}X_{t+k-1:t+1} \tag{10.29}$$

[2] Detailed theory about MMSE and linear estimator see section 12.4.1 ~ page 331, Linear MMSEstimator.

Calculation formula for zero-mean Weakly Stationary:

– using determinant form

$$\phi_{11} = \rho_1 \tag{10.30}$$

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & \rho_k \end{vmatrix}_{k\times k}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & 1 \end{vmatrix}_{k\times k}} \tag{10.31}$$

– Levinson-Durbin's recursive formula

$$\phi_{11} = \rho_1 \tag{10.32}$$

$$\phi_{k+1,k+1} = \frac{\rho_{k+1} - \sum_{j=1}^{k} \phi_{k,j}\rho_{k+1-j}}{1 - \sum_{j=1}^{k} \phi_{k,j}\rho_j}, \quad k \geq 1 \tag{10.33}$$

$$\phi_{k+1,j} = \phi_{k,j} - \phi_{k+1,k+1}\phi_{k,k+1-j}, \quad j = 1, 2, \ldots, k \tag{10.34}$$

where $\phi_{k+1,j}$ here is a formal notation for recursion. But we will see its meaning in AR(p) model (equation 10.92 ~ page 276)

- Wold Decomposition: zero-mean weakly stationary time series can be decomposed as :

$$X_t = \sum_{j=-\infty}^{\infty} \phi_j \varepsilon_{t-j} + V_t \tag{10.35}$$

where

$$\phi_0 = 1 \tag{10.36}$$

$$\varepsilon_t \sim \text{WN}(0, \sigma^2) \tag{10.37}$$

- Spectrum of zero-mean weak stationary time series $\{X_t\}$:

$$X_t = \int_{\lambda} \xi(\lambda) e^{i\lambda t} \, \mathrm{d}\lambda \tag{10.38}$$

We can use this form to construct ACF, ACVF, etc.

– Spectrum and ACVF: the fourier expansion of γ_k is denoted

$$\gamma_k = cov(X_t, X_{t+k}) \equiv \int_{\lambda} e^{i\lambda k} \nu(\lambda) \, \mathrm{d}\lambda \tag{10.39}$$

and here a function $F(\lambda) = \int \nu(\lambda) \, \mathrm{d}\lambda$ is the **spectrum** of γ_k, and $\nu(\lambda)$ is the **spectrum density**.

For $k = 0, 1, 2, \ldots$ (discrete time)

$$\gamma_k = \int_{-\pi}^{\pi} \nu(\lambda) e^{i\lambda k} \, \mathrm{d}\lambda \tag{10.40}$$

and also use inverse fourier transform: for weak stationary TS $X_t = \sum_{j=-\infty}^{\infty} \phi_j \varepsilon_{t-j}, \varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$

$$\nu(\lambda) = \frac{1}{2\pi} \int_{\mathbb{R}} \gamma_k e^{-i\lambda k} \, \mathrm{d}k = \frac{\sigma^2}{2\pi} \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \phi_j \phi_{j-k} e^{-i\lambda k} = \frac{\sigma^2}{2\pi} \left| \sum_{j=-\infty}^{\infty} \phi_j e^{i\lambda j} \right|^2 \tag{10.41}$$

10.2.2 Statistics

To estimate the above $\mu_t = \mu, \gamma_k, \rho_k, \phi_{kk}$ given a realization of $\{X_t\}$, say we have $\{x_t\}_{t=1}^n$, we can construct:

- Sample mean μ:

$$\hat{\mu} = \hat{x}_n = \frac{1}{n} \sum_{t=1}^{n} x_t \tag{10.42}$$

$\hat{\mu}$ is the unbiased, consistent estimator, with

$$\sqrt{n}(\hat{\mu} - \mu) \xrightarrow{\mathrm{d}} N(0, \sigma^2) \tag{10.43}$$

an estimator using spectrum:

$$\sqrt{n}(\hat{\mu} - \mu) \xrightarrow{\mathrm{d}} N(0, 2\pi\nu(0)) \tag{10.44}$$

$$2\pi\nu(0) = \gamma_0 + 2\sum_{j=1}^{\infty} \gamma_j = \sum_{j=-\infty}^{\infty} \gamma_j \tag{10.45}$$

- ACVF γ_k:

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (x_t - \hat{\mu})(x_{t+k} - \hat{\mu}) \tag{10.46}$$

$$\hat{\hat{\gamma}}_k = \frac{1}{n-k} \sum_{t=1}^{n-k} (x_t - \hat{\mu})(x_{t+k} - \hat{\mu}) \tag{10.47}$$

Note for actual usage:

  - We usually avoid estimation for $k \sim n$ due to large error when $n - k$ is small
  - In most cases we use $\hat{\gamma}_k$ rather than $\hat{\hat{\gamma}}_k$, for two reasons:
    * We often estimate γ_k for small k, which means $\hat{\gamma}_k \approx \hat{\hat{\gamma}}_k$
    * $\hat{\gamma}_k$ could guarantee the semi-positive-definition of $\hat{\Gamma}_k$:

$$\hat{\Gamma}_k = \{\hat{\gamma}_{i-j}\}_{i,j=1}^{k} \succeq 0 \tag{10.48}$$

asymptotic distribution: denote i.i.d. standard normal time series $W_t \sim$ i.i.d. $N(0, 1)$

$$\sqrt{n}(\hat{\gamma}_0 - \gamma_0, \hat{\gamma}_1 - \gamma_1, \ldots, \hat{\gamma}_h - \gamma_h) \xrightarrow{\mathrm{d}} (\xi_0, \xi_1, \ldots, \xi_h) \tag{10.49}$$

$$\xi_j = (\frac{\sqrt{\mathbb{E}(\varepsilon^4) - \sigma^4}}{\sigma^2} \gamma_j) W_0 + \sum_{t=1}^{\infty} (\gamma_{t+j} + \gamma_{t-j}) W_t, \quad j \geq 0 \tag{10.50}$$

- ACF ρ_k:

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=1}^{n-k}(x_t - \hat{\mu})(x_{t+k} - \hat{\mu})}{\sum_{t=1}^{n-k}(x_t - \hat{\mu})^2} \tag{10.51}$$

asymptotic distribution: denote i.i.d. standard normal time series $W_t \sim$ i.i.d. $N(0,1)$

$$\sqrt{n}(\hat{\gamma}_0 - \gamma_0, \hat{\gamma}_1 - \gamma_1, \ldots, \hat{\gamma}_h - \gamma_h) \xrightarrow{\text{d}} (R_0, R_1, \ldots, R_h) \tag{10.52}$$

$$R_j = \sum_{t=1}^{\infty}(\phi_{t+j}\rho_{t-j} - 2\rho_t\rho_j)W(t), \quad j \geq 1 \tag{10.53}$$

- PACF ϕ_{kk}: take $\hat{\rho}_k$ in the calculation equation of ϕ_{kk}.

Section 10.3 ARMA Model

Two of the basic modeling methods for time series: Auto-Regression (AR) and Moving-Average (MA)

10.3.1 Backshift Operator and Difference Equation

□ **Backshift Operator $\mathscr{B}$**

For clearer notation of ARMA and induce the solution, we first introduce backshift operator $\mathscr{B}$ of time series: given time series $\{X_t\}$[3]

$$\mathscr{B}X_t = X_{t-1}, \quad \forall t \tag{10.58}$$

further it can be used as variable of function by Laurant function series expansion:

$$\psi(z) = \sum_{j=-\infty}^{\infty} \psi_j z^j \tag{10.59}$$

$$\psi(\mathscr{B}) = \sum_{j=-\infty}^{\infty} \psi_j \mathscr{B}^j \tag{10.60}$$

$$\psi(\mathscr{B})X_t = \sum_{j=-\infty}^{\infty} \psi_j \mathscr{B}^j X_t = \sum_{j=-\infty}^{\infty} \psi_j X_{t-j} \tag{10.61}$$

Linearity: for time series $\{X_t\}$, $\{Y_t\}$. r.v. U, V, W:

$$\phi(\mathscr{B})(UX_t + VY_t + W) = U\psi(\mathscr{B})X_t + V\psi(\mathscr{B})Y_t + W\psi(1) \tag{10.62}$$

[3]Backshift operator could be used to construct difference operator $\Delta = (1 - \mathscr{B})$, e.g.

$$\Delta X_t = (1 - \mathscr{B})X_t = X_t - X_{t-1} \tag{10.54}$$

$$\Delta^2 X_t = (1 - \mathscr{B})^2 X_t = X_t - 2X_{t-1} + X_{t-2} \tag{10.55}$$

$$\ldots \tag{10.56}$$

or seasonal difference operator $\Delta_k = (1 - \mathscr{B}^k)$, e.g.

$$\Delta_4 X_t = (1 - \mathscr{B}^4) = X_t - X_{t-4} \tag{10.57}$$

□ **Difference Equation**

p^{th} order ordinary difference equation:

$$X_t - [a_1 X_{t-1} + a_2 X_{t-2} + \ldots + a_p X_{t-p}] = 0 \tag{10.63}$$

can be solved using backshift operator: define characteristic equation which would have p roots ζ_j

$$\begin{aligned} A(z) =& 1 - \left[a_1 z + a_2 z^2 + \ldots + a_p z^p\right] && (10.64)\\ =& 1 - \sum_{j=1}^{p} a_j z^j && (10.65)\\ =& \prod_{j=1}^{p} (1 - \zeta_j z) && (10.66)\\ A(\mathscr{B}) =& 1 - \sum_{j=1}^{p} a_j \mathscr{B}^j && (10.67)\\ =& \prod_{j=1}^{p} (1 - \zeta_j \mathscr{B}) && (10.68) \end{aligned}$$

similar to ODE, we can construct general solution from ζ_j, and particular solution.[4]

10.3.2 AR(p) Model

Auto-Regression model (of order p) contains (p^{th} order) backshift on X_t:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(\mu_\varepsilon, \sigma^2) \tag{10.69}$$

or expressed in backshift operator with $\phi(z) = 1 - \sum_{j=1}^{p} \phi_j z^j$, where the root of $\phi(z) = 0$ denoted α_j

$$\phi(\mathscr{B}) X_t = \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(\mu_\varepsilon, \sigma^2) \tag{10.70}$$

$$\phi(z) = 1 - \sum_{j=1}^{p} \phi_j z^j = \prod_{j=1}^{p} (1 - \alpha_j z) \tag{10.71}$$

□ **Properties and Solution: (here we consider stationary case $\mu_\varepsilon = 0$)**

- (Weak) Stationarity condition:

$$|\alpha_j| > 1, \forall j \tag{10.72}$$

- Solution of X_t: using the expansion of ϕ^{-1}

$$\begin{aligned} \phi(z) =& 1 - \sum_{j=1}^{p} \phi_j z^j && (10.73)\\ \phi^{-1}(z) =& \sum_{j=0}^{\infty} \psi_j z^j, \quad \psi_0 = 1 && (10.74) \end{aligned}$$

[4] Cases for multiple root see https://www.math.pku.edu.cn/teachers/lidf/course/atsa/atsanotes/html/_atsanotes/atsa-lagdiff.html

naturally expressed in the form of Wold Decomposition:

$$\phi(\mathscr{B})X_t = \varepsilon_t \Rightarrow X_t = \phi^{-1}(\mathscr{B})\varepsilon_t = \sum_{j=0}^{\infty}\psi_j\varepsilon_{t-j}, \quad \psi_0 = 1 \tag{10.75}$$

- ACF and ACVF:

$$\gamma_k = \sigma^2\sum_{j=0}^{\infty}\psi_j\psi_{j+k} \tag{10.76}$$

$$\rho_k = \frac{\sum_{j=0}^{\infty}\psi_j\psi_{j+k}}{\sum_{j=0}^{\infty}\psi_j^2} \tag{10.77}$$

- Spectrum density $\nu(\lambda)$:

$$\nu(\lambda) = \frac{\sigma^2}{2\pi}\left|\sum_{j=0}^{\infty}\psi_j e^{i\lambda j}\right|^2 \tag{10.78}$$

$$= \frac{\sigma^2}{2\pi}\left|\phi^{-1}(e^{i\lambda})\right|^2 \tag{10.79}$$

- Yule-Walker Equation: we have

$$\mathbb{E}\left(X_tX_{t-k}\right) = \phi_1\mathbb{E}\left(X_{t-1}X_{t-k}\right) + \ldots + \phi_p\mathbb{E}\left(X_{t-p}X_{t-k}\right) + \mathbb{E}\left(\varepsilon_tX_{t-k}\right), \quad \forall k = 1, 2, \ldots, p \tag{10.80}$$

$$\Rightarrow \gamma_k = \phi_1\gamma_{k-1} + \ldots + \phi_p\gamma_{k-p}, \quad \forall k = 1, 2, \ldots, p \tag{10.81}$$

and for $k = 0$:

$$\gamma_0 = \phi_1\gamma_1 + \ldots + \phi_p\gamma_p + \sigma^2 \tag{10.82}$$

written in matrix form to get Yule-Walker Equation:

$$\begin{bmatrix}\gamma_1\\ \gamma_2\\ \vdots\\ \gamma_p\end{bmatrix} = \begin{bmatrix}\gamma_0 & \gamma_1 & \cdots & \gamma_{p-1}\\ \gamma_1 & \gamma_0 & \cdots & \gamma_{p-2}\\ \vdots & \vdots & \ddots & \vdots\\ \gamma_{p-1} & \gamma_{p-2} & \cdots & \gamma_0\end{bmatrix}\begin{bmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_p\end{bmatrix} \tag{10.83}$$

$$\sigma^2 = \gamma_0 - \phi_1\gamma_1 - \ldots - \phi_p\gamma_p \tag{10.84}$$

or in dense matrix form (1):

$$\gamma = \Gamma\phi \tag{10.85}$$

$$\sigma^2 = \gamma_0 - \phi'\gamma \tag{10.86}$$

dense form (2):

$$\begin{bmatrix}-\sigma^2\\ 0\\ 0\\ \vdots\\ 0\end{bmatrix} = \begin{bmatrix}\gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_p\\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{p-1}\\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \gamma_{p-2}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \gamma_p & \gamma_{p-1} & \gamma_{p-2} & \cdots & \gamma_0\end{bmatrix}\begin{bmatrix}-1\\ \phi_1\\ \phi_2\\ \vdots\\ \phi_p\end{bmatrix} \tag{10.87}$$

- PACF: the coefficient of AR(p) has straight relation with $\phi_{k,j}$: for all given $k \geq p$

$$(\phi_1, \ldots, \phi_p, 0, \ldots, 0) = (\phi_{k,1}, \ldots, \phi_{k,p}, \phi_{k,p+1}, \ldots, \phi_{k,k}) \tag{10.88}$$

(Note that $\phi_{p,j} = \phi_{p+1,j} = \phi_{p+2,j} = \ldots$ using Levinson-Durbin' recursion at equation 10.32 ~ page 271).

□ **Estimation: Key focus is the estimation of ϕ_i, $i = 1, 2, \ldots, p$ and σ^2 (assume a TS of $\mu_\varepsilon = 0$)**

Y-W Estimation and OLS Estimation are moment methods, asymptotically the same. MLE Estimation is usually more precise, but hard to calculate.

- Yule-Walker Estimation: use $\gamma = \Gamma\phi$. First estimate $\hat{\gamma}$, as well as $\hat{\Gamma}$, and get estimation for ϕ, σ^2

$$\hat{\phi} = \hat{\Gamma}^{-1}\hat{\gamma} \tag{10.89}$$

$$\hat{\sigma}^2 = \hat{\gamma}_0 - \hat{\gamma}'\hat{\Gamma}^{-1}\hat{\gamma} \tag{10.90}$$

Asymptotic distribution:

$$\sqrt{n}(\hat{\phi} - \phi) \xrightarrow{\text{d}} N_p(0, \sigma^2\Gamma^{-1}) \tag{10.91}$$

- Levinson-Durbin's recursion for Yule-Walker Estimation: since PACF are the same as coefficients $\phi_{k,j} = \phi_j$, we can use Durbin's recursion to avoid calculation of $\hat{\Gamma}^{-1}$

$$\hat{\phi}_{11} = \hat{\rho}_1 \tag{10.92}$$

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k} \hat{\phi}_{k,j}\hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k} \hat{\phi}_{k,j}\hat{\rho}_j}, \quad k \geq 1 \tag{10.93}$$

$$\hat{\phi}_{k+1,j} = \hat{\phi}_{k,j} - \hat{\phi}_{k+1,k+1}\hat{\phi}_{k,k+1-j}, \quad j = 1, 2, \ldots, k \tag{10.94}$$

$$\hat{\sigma}_0^2 = \hat{\gamma}_0 \tag{10.95}$$

$$\hat{\sigma}_k^2 = \hat{\sigma}_{k-1}^2(1 - \hat{\phi}_{k,k}^2) \tag{10.96}$$

estimator:

$$\hat{\phi}_j = \hat{\phi}_{p,j} \tag{10.97}$$

- OLS Estimation: using the linear combination form of AR model:

$$\hat{\phi} = \arg\min_{\phi} \sum_{t=p+1}^{n} \left[x_t - \sum_{j=1}^{p} \phi_j x_{t-j} \right]^2 \tag{10.98}$$

the solution is in the form of OLS estimator $(X'X)^{-1}XY$, with X, Y properly defined

- MLE Estimation: under normal assumption

$$\phi(\mathscr{B})X_t = \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2) \tag{10.99}$$

Likelihood: define $\theta = \{\phi_1, \ldots, \phi_p, \sigma^2\}$

$$L(\theta; x_1, \ldots, x_n) = f(x_1, \ldots, x_p|\theta) \prod_{t=p+1}^{n} f(x_t|x_{t-1}, \ldots, x_1; \theta) \tag{10.100}$$

$$\approx\propto \prod_{t=p+1}^{n} f(x_t|x_{t-1}, \ldots, x_1; \theta) \tag{10.101}$$

$$= \prod_{t=p+1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x_t - \sum_{j=1}^{p} \phi_j x_{t-j})^2\right\} \tag{10.102}$$

$$=(2\pi\sigma^2)^{-(n-p)/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{t=p+1}^{n} (x_t - \sum_{j=1}^{p} \phi_j x_{t-j})^2\right\} \tag{10.103}$$

- Estimation to spectrum density:

$$\hat{\nu}(\lambda) = \frac{\hat{\sigma}^2}{2\pi} \left|1 - \sum_{j=1}^{\hat{p}} \hat{\phi}_j e^{i\lambda j}\right|^{-2} \tag{10.104}$$

10.3.3 MA(q) Model

Moving-Average model (of order$q$) contains (q^{th} order) backshift on ε_t:

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}, \quad \varepsilon_t \sim \text{WN}(\mu_\varepsilon, \sigma^2) \tag{10.105}$$

or expressed in backshift operator with $\theta(z) = 1 + \sum_{j=1}^{q} \theta_j z^j$, where the root of $\theta(z) = 0$ denoted κ_j

$$X_t = \theta(\mathscr{B})\varepsilon_t, \quad \varepsilon_t \sim \text{WN}(\mu_\varepsilon, \sigma^2) \tag{10.106}$$

$$\theta(z) = 1 + \sum_{j=1}^{q} \theta_j z^j = \prod_{j=1}^{q} (1 - \kappa_j z) \tag{10.107}$$

$$= \sum_{j=0}^{q} \theta_j z^j, \quad \theta_j = 1 \tag{10.108}$$

here we could note that AR(p) model has solution in the form of MA(∞):

$$\text{AR}(p) : X_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}, \quad \psi_0 = 1 \tag{10.109}$$

□ **Properties and Solution: (here we consider stationary case $\mu_\varepsilon = 0$)**

- Invertibility: if and only if

$$|\kappa_j| > 1, \forall j \tag{10.110}$$

- ACF and ACVF:

$$\gamma_k = \begin{cases} \sigma^2 \sum_{j=0}^{q-k} \theta_j \theta_{j+k}, & 0 \le k \le q \\ 0, & k > q \end{cases} \tag{10.111}$$

$$\rho_k = \begin{cases} \frac{\sum_{j=0}^{q-k} \theta_j \theta_{j+k}}{\sum_{j=0}^{q} \theta_j^2}, & 0 \le k \le q \\ 0, & k > q \end{cases} \tag{10.112}$$

- Solution: $\hat{\theta}_j$ could solved from $\{\gamma_k\}$

10.3.4 ARMA(p,q) Model

Auto-Regerssion-Moving-Average model ARMA(p,q) in the form of

$$\phi(\mathscr{B})X_t = \theta(\mathscr{B})\varepsilon_t \tag{10.113}$$

$$\phi(z) = 1 - \sum_{j=1}^{p}\phi_j z^j = \prod_{j=1}^{p}(1-\alpha_j z) \tag{10.114}$$

$$\theta(z) = 1 + \sum_{j=1}^{q}\theta_j z^j = \prod_{j=1}^{q}(1-\kappa_j z) \tag{10.115}$$

□ **Properties and Solution: (here we consider stationary case $\mu_\varepsilon = 0$)**

- Solution:

$$X_t = \phi^{-1}(\mathscr{B})\theta(\mathscr{B})\varepsilon_t =\equiv \Psi(\mathscr{B})\varepsilon_t \tag{10.116}$$

- Weak Stationarity: if and only if AR part is WS, i.e.

$$|\alpha_j| > 1,\ \forall j \tag{10.117}$$

- Invertibility: if and only if MA part is invertible, i.e.

$$|\kappa_j| > 1,\ \forall j \tag{10.118}$$

10.3.5 ARIMA(p,d,q) Model

ARIMA(p,d,q) model adds an difference term $\Delta^d = (1-\mathscr{B})^d$ in ARMA(p,q):

$$\phi(\mathscr{B})(1-\mathscr{B})^d X_t = \theta(\mathscr{B}) \tag{10.119}$$

Section 10.4 Seasonal Model for Time Series

This part includes some ideas for modelling seasonal term (usually as well as trend term) in $Y_t = \mathbf{T}_t + \mathbf{S}_t + X_t$.

Usually we describe the trend term as the 'mean' of time series over time, and sensonal term with zero-mean and period $P > 1$.

表 10.1: Buys-Ballot Table of seasonal period s

	Season (j)					
Period (i)	1	2	...	s	$\bar{y}_{i\cdot}$	$\hat{\sigma}^2_{i\cdot}$
1	y_1	y_2	...	y_s	$\bar{y}_{1\cdot}$	$\hat{\sigma}_{1\cdot}$
2	y_{s+1}	y_{s+2}	...	y_{2s}	$\bar{y}_{2\cdot}$	$\hat{\sigma}_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
m	$y_{(m-1)s+1}$	$y_{(m-1)s+2}$	...	y_{ms}	$\bar{y}_{m\cdot}$	$\hat{\sigma}^2_{m\cdot}$
$\bar{y}_{\cdot j}$	$\bar{y}_{\cdot 1}$	$\bar{y}_{\cdot 2}$	...	$\bar{y}_{\cdot s}$	$\bar{y}_{\cdot\cdot}$	-
$\hat{\sigma}^2_{\cdot i}$	$\hat{\sigma}^2_{\cdot 1}$	$\hat{\sigma}^2_{\cdot 2}$	...	$\hat{\sigma}^2_{\cdot s}$	-	$\hat{\sigma}^2_{\cdot\cdot}$

10.4.1 Regression Model

A common functional description is polynomial trend + Fourier expansion senson, i.e.

$$Y_t = T_t + S_t + X_t \tag{10.120}$$

$$= \alpha_0 + \sum_{j=1}^{m} \alpha_j t^j + \sum_{j=1}^{[s/2]} \left[\beta_j \sin(\frac{2\pi}{s} jt) + \gamma_j \cos(\frac{2\pi}{s} jt) \right] \tag{10.121}$$

Note: for regression model, T_t and S_t are treated as invariant term.

Estimation of paramters $\{\alpha_0, \alpha_j, \beta_j, \gamma_j\}$ use e.g. MSE estimator:

$$\{\hat{\alpha}_0, \hat{\alpha}_j, \hat{\beta}_j, \hat{\gamma}_j\} = \underset{\{\alpha_0, \alpha_j, \beta_j, \gamma_j\}}{\arg\min} \sum_{t \in T} \left[y_t - (T_t + S_t) \right]^2 \tag{10.122}$$

10.4.2 Moving Average Model

First estimate Trend term, then Seasonal term

Trend term is estimated by a symmetric moving average window $\{\omega_j\}_{j=-w}^{w}$ with band width w

$$\hat{T}_t = \sum_{j=-w}^{w} \omega_j y_{t-j} \tag{10.123}$$

$$\omega_j = \omega_{-j}, \quad j = -w, -w+1, \ldots, w-1, w \tag{10.124}$$

$$\sum_{j=-w}^{w} \omega_j = 1 \tag{10.125}$$

then seasonal term is naturally estimated by

$$\hat{S}_t = y_t - \hat{T}_t \tag{10.126}$$

10.4.3 Seasonal ARIMA Model

Multiplicative seasonal ARIMA model with period s of Y_t: ARIMA$(p, d, q) \times (P, D, Q)_s$

$$\Phi_P(\mathscr{B}^s)\phi_p(\mathscr{B})(1-\mathscr{B})^d(1-\mathscr{B}^s)^D Y_t = \Theta_Q(\mathscr{B}^s)\theta_q(\mathscr{B}^s)\varepsilon_t, \quad \varepsilon_t \sim \text{WN}(0, \sigma^2) \tag{10.127}$$

On the ACF plot of SARIMA, you should see peak at $t_{\text{lag}} \propto s$

Section 10.5 Model Selection and Diagnostics

10.5.1 Model Building of ARIMA

□ **Box-Jenkins Approach for ARIMA Model:**

1. Data Transformation: Note that in the general model $Y_t = T_t + S_t + X_t$ we would expect a 'stationary' random term, thus a transform for stable variance is needed, see section 3.5.1 ~ page 102 for detailed methods. Then we could preliminarily detect the Stationarity of sequence, e.g. by plotting.
2. Seaonal Term Detection: usually by plotting ACF plot & ACVF plot, further we could also use spectrum plot, seasonal subseries plot.
3. Stationarity Detection: Detect stationarity e.g. by unit-root test.
4.

10.5.2 Order Determination of ARIMA Model

□ **Order Determination of AR(p)**

- PACF test: use the proper of $\phi_{k,k}$ for $k \geq p$ where

$$\phi_{kk} = \begin{cases} \phi_p, & k \leq p \\ 0, & k > p \end{cases} \tag{10.128}$$

for all given $k > p$: Asymptotic distribution:

$$\sqrt{n}(\hat{\phi}_{k,1} - \phi_{k,1}, \ldots, \hat{\phi}_{k,k} - \phi_{k,k}) \xrightarrow{\text{d}} N(0, \sigma^2 \Gamma_k^{-1}) \tag{10.129}$$

specially it could be proved that $\left(\sigma^2 \Gamma_k^{-1}\right)_{k,k} = \left(\sigma^2 \Gamma_k^{-1}\right)_{1,1} = 1, \quad k > p.$

i.e. test statistics for AR(p):

$$\hat{\phi}_{k,k} \xrightarrow{\text{d}} N(0,1), \quad w.r.t.\, H_0 : \phi_{k,k} = 0, \quad k > p \tag{10.130}$$

Plot $\hat{\phi}_{k,k}$-k to determine the proper k as $\hat{p}$.

- AIC/BIC method: use $\hat{p} = \arg\min \text{AIC}(k)$ or $\arg\min \text{BIC}(k)$:

$$\text{AIC}(k) = \ln \hat{\sigma}_k^2 + \frac{2k}{n} \tag{10.131}$$

$$\text{BIC}(k) = \ln \hat{\sigma}_k^2 + \frac{k \ln n}{n} \tag{10.132}$$

□ **Order Determination of MA(q)**

- ACF test: use the cut off property of ρ_k of MA(q):

$$\rho_k = \begin{cases} \frac{\sum_{j=0}^{q-k} \theta_j \theta_{j+k}}{\sum_{j=0}^{q} \theta_j^2}, & 0 \le k \le q \\ 0, & k > q \end{cases} \tag{10.133}$$

use the asymptotic distribution of $\hat{\rho}_m$ in equation 10.52 ~ page 273, for $m > q$:

$$\sqrt{n}\hat{\rho}_m \xrightarrow{\text{d}} R_m \tag{10.134}$$

$$= \sum_{t=1}^{\infty} (\rho_{t+m} + \rho_{t-m} - \rho_t \rho_m) W_t \tag{10.135}$$

$$= \sum_{l=-q}^{q} \rho_l W_{l+m}, \quad m > q \tag{10.136}$$

$$\sim N(0, 1 + 2\sum_{j=1}^{q} \rho_j^2) \tag{10.137}$$

i.e. test statistics for MA(q):

$$T_q(m) = \frac{\sqrt{n}\hat{\rho}_m}{\sqrt{1 + 2\sum_{j=1}^{q} \hat{\rho}_j^2}} \xrightarrow{\text{d}} N(0,1), \quad H_0 : \rho_m = 0, \quad m > q \tag{10.138}$$

- AIC/BIC method: use $\hat{q} = \arg\min \text{AIC}(m)$ or $\arg\min \text{BIC}(m)$:

$$\text{AIC}(m) = \ln \hat{\sigma}_m^2 + \frac{2m}{n} \tag{10.139}$$

$$\text{BIC}(m) = \ln \hat{\sigma}_m^2 + \frac{m \ln n}{n} \tag{10.140}$$

□ **Order Determination of ARMA(p, q)**

- AIC/BIC method:

$$\hat{p}, \hat{q} = \arg\min_{k,m} \text{AIC}(k, m) = \arg\min_{k,m} \ln \hat{\sigma}_{k,m}^2 + \frac{2(k+m)}{n} \tag{10.141}$$

$$\hat{p}, \hat{q} = \arg\min_{k,m} \text{BIC}(k, m) = \arg\min_{k,m} \ln \hat{\sigma}_{k,m}^2 + \frac{(k+m)\ln n}{n} \tag{10.142}$$

- EACF for ARIMA(p, d, q): Extended ACF forms a matrix for determining (p, d, q) using extended Yule-Walker Equation

10.5.3 Outlier Detection

Here we introduce two kinds of outlier in time series: Additive Outlier (AO) and Innovative Outlier (IO).

□ **Notation for Outlier**

- Step function in time series: a rise of value 1 at time τ:

$$S_t^{(\tau)} = \begin{cases} 0, & t < \tau \\ 1, & t \ge \tau \end{cases} \tag{10.143}$$

- Pulse function in time series: a pulse of value 1 at time τ:

$$P_t^{(\tau)} = (1-\mathscr{B})S_t^{(\tau)} = \begin{cases} 0, & t \neq \tau \\ 1, & t = \tau \end{cases} \tag{10.144}$$

□ **Additive Outlier**

A pulse outlier of y at τ:

$$\tilde{y}_t = y_t + \omega_A P_t^{(\tau)} \tag{10.145}$$

the outlier would not influence $t \neq \tau$, thus is additive.

□ **Innovative Outlier**

A pulse outlier of ε at τ:

$$\varepsilon_\tau = \varepsilon_t + \omega_I \tag{10.146}$$

$t \neq \tau$ would also be influenced by this outlier.

Section 10.6 Forecast of Time Series

10.6.1 MSE Forecast Criterion

The criterion for forecasting is to minimizing some loss function, usually taken as MSE loss:

$$\hat{X}_{\tau|t} = \underset{X_\tau}{\arg\min}\, \mathbb{E}[\left(X_\tau - X_{\tau|t}\right)^2] = \mathbb{E}\left(X_{\tau|t}\right) \tag{10.147}$$

our mission is to construct a function $g(\,\cdot\,)$ so that $\hat{X}_{\tau|t} = g(\mathscr{F}_t)$ can act as the estimator. $\mathscr{F}_t = \{X_t, X_{t-1}, X_{t-2}, \ldots\}$ denotes the history until t.

10.6.2 Best Linear Estimator

A simple and straightforward method is a linear combination form of $\mathscr{F}_t$:

$$\hat{X}_{\tau|t} = \sum_{j=0}^{\infty} \beta_j X_{t-j} \tag{10.148}$$

$$\beta_j = \underset{\{\beta_j\}}{\arg\min}\, \mathbb{E}\left[\left(X_\tau - \sum_{j=0}^{\infty}\beta_j X_{t-j}\right)^2\right] \tag{10.149}$$

i.e.

$$\hat{X}_{\tau|t} = L(X_\tau|\mathscr{F}_t) \tag{10.150}$$

Solution was given in equation 10.28 ~ page 270:

$$\beta = \Sigma_{X_{\mathscr{F}_t}}^{-1}\Sigma_{\mathscr{F}_t, X_\tau} \tag{10.151}$$

- e.g. ont-step forecast for zero-mean weakly stationary sequence:

$$\hat{X}_{t+1|t} = \sum_{j=0}^{\infty} \beta_j X_{t-j} \tag{10.152}$$

$$\vec{\beta} = \Gamma^{-1}\gamma \tag{10.153}$$

(For actual case, calculate for a proper truncation p for $\hat{X}_{t+1|t} = \sum_{j=0}^{p} \beta_j X_{t-j}$ would be fine)

Best linear estimator is the best estimator for ARMA(p, q) with WN noise.

10.6.3 Forecast of AR(p)

AR(p):

$$X_{t+1} = \sum_{j=1}^{p} \phi_j X_{t+1-j} + \varepsilon_{t+1} \tag{10.154}$$

1. First estimate coefficients, e.g. using Yule-Walker estimator $\hat{\phi}_j$, $j = 1, 2, \ldots, p$
2. Esitmate of $X_{t+1|t}$

$$\hat{X}_{t+1|t} = \sum_{j=1}^{p} \hat{\phi}_1 X_{t+1-j} \tag{10.155}$$

$$\hat{\sigma}^2_{t+1|t} = \hat{\sigma}^2 = \hat{\gamma}_0 \tag{10.156}$$

3. Estimate of $X_{t+h|t}$: estimation conduct sequentially for $h = 1, 2, \ldots$:

$$\hat{X}_{t+1|t} = \hat{\phi}_1 X_t + \hat{\phi}_2 X_{t-1} + \ldots + \hat{\phi}_p X_{t+1-p} \tag{10.157}$$

$$\hat{X}_{t+2|t} = \hat{\phi}_1 \hat{X}_{t+1|t} + \hat{\phi}_2 X_t + \hat{\phi}_3 X_{t-1} + \ldots + \hat{\phi}_p X_{t+2-p} \tag{10.158}$$

$$\hat{X}_{t+3|t} = \hat{\phi}_1 \hat{X}_{t+2|t} + \hat{\phi}_2 \hat{X}_{t+1|t} + \hat{\phi}_3 X_t + \hat{\phi}_4 X_{t-1} + \ldots + \hat{\phi}_p X_{t+3-p} \tag{10.159}$$

$$\ldots \tag{10.160}$$

10.6.4 Forecast of MA(q)

MA(q):

$$X_{t+1} = \varepsilon_{t+1} + \sum_{j=1}^{q} \theta_j \varepsilon_{t+1-j} \tag{10.161}$$

1. First estimate coefficients $\hat{\theta}_j$, $j = 1, 2, \ldots, q$
2. Estimate of $X_{t+h|t}$: first for each $k = 1, 2, \ldots, t$, calculate residual estimator:

$$\hat{\varepsilon}_k = X_k - L(X_k|\mathscr{F}_{k-1}) = X_k - L(X_k|X_{k-1}, \ldots, X_1) \tag{10.162}$$

then calculate forecast:

$$\hat{X}_{t+h|t} = \begin{cases} \sum_{j=h}^{q} \hat{\theta}_j \hat{\varepsilon}_{t+1-j}, & h = 1, 2, \ldots, q \\ 0, & h > q \end{cases} \tag{10.163}$$

10.6.5 Forecast of ARMA(p, q)

ARMA(p, q):

$$\phi(\mathscr{B})X_t = \theta(\mathscr{B})\varepsilon_t \Rightarrow X_t = \phi^{-1}(\mathscr{B})\theta(\mathscr{B})\varepsilon_t \equiv \psi(\mathscr{B})\varepsilon_t \tag{10.164}$$

similarly estimate ψ_j and ε_j and forecast as MA(∞)

10.6.6 Forecast of ARIMA(p, d, q)

Chapter. XI　因果推断导论部分

Instructor: Wanlu Deng

Section 11.1　Neyman-Rubin Potential Outcome Framework

Neyman-Rubin Framework (Donald B.Rubin, 1978), also called Potential Outcome Framework is based on **counter-factual outcome** inference to judge causal effect.

11.1.1　Description of Causal Effect and Challenge

Causality concerns 'what would happen when **an action** is applied to **a unit**'. Here the 'unit' is how causality is different from correlation.

- A unit is the physical object at that specific time, which is similar to the event in Einstein's relativity.[1]
- An action is the treatment/intervention that could be **potentially** applied to the unit.

In this section we mainly focus on cases with binary intervention, i.e.[2]

$$\{\text{treatment, control}\} = \{1, 0\} \tag{11.1}$$

□ **Potential Outcome**

With this notation, the causal effect could be expressed by the **estimand** as follows by comparing the **potential outcomes**, here's a commonly used form:

$$\tau := Y_{\text{treatment}} - Y_{\text{control}} := Y(1) - Y(0) \tag{11.2}$$

To estimate the causal effect (on a unit), we need to obtain both potential outcomes of $Y(1)$ and $Y(0)$, but in the real world we can only observe one of them, say, the patient took the medicine, and we got $Y(1)$, while $Y(0)$ is missing.

Relevant Notation:

- **Unit**: The atomic object in causal inference. $i = 1, 2, \ldots, N$
- **Treatment** W_i: (possible) assignment.

[1] Which means that one object at different time ((x, t) & (x, t')) is not the same unit (event). However if the assumption of time independency is valid, then object in different time could be the same unit (usually less resonable for human subjects).

[2] Habitually we denote the more 'active' intervention as treatment, but in mathematical form they are symmetric.

  - Treatment Group: Set of $\{\text{Unit}_i | W_i = 1\}$;
  - Controlled Group: Set of $\{\text{Unit}_i | W_i = 0\}$.

- **Potential Outcome** (PO) Y_i: For each unit with action treatment(or control), the potential outcome $Y(W = w)$, $w = 0, 1$ is the 'Eigen Outcome' of the model, despite of what really happens. It can be seen as what would happen when the operation had not been done.
- **Observed Outcome** Y_i^{obs}: The actually happened outcome, $Y_i^{\text{obs}} = Y_i(W = w_{\text{REAL\_CASE}}) := Y_i(W = w_i^{\text{obs}})$.
- **Missing Outcome** Y_i^{mis}: The potential outcome when the $w_i^{\text{mis}} :=!w_i^{\text{obs}}$ would have been operated (it does exist but we cannot observe the 'world-line' where w_i^{mis} was operated, thus is unknown to us), $Y_i^{\text{mis}} = Y_i(W_i = 1 - w_{\text{REAL\_CASE}}) := Y_i(W_i = w_i^{\text{mis}})$

$$Y_i^{\text{obs}} = Y_i(W_i^{\text{obs}}) = \begin{cases} Y_i(1) & W_i = 1 \\ Y_i(0) & W_i = 0 \end{cases} \tag{11.3}$$

$$Y_i^{\text{mis}} = Y_i(1 - W_i^{\text{obs}}) = \begin{cases} Y_i(0) & W_i = 1 \\ Y_i(1) & W_i = 0 \end{cases} \tag{11.4}$$

or in condensed notation

$$\begin{bmatrix} Y_i^{\text{obs}} \\ Y_i^{\text{mis}} \end{bmatrix} = \begin{bmatrix} W_i & 1 - W_i \\ 1 - W_i & W_i \end{bmatrix} \begin{bmatrix} Y_i(1) \\ Y_i(0) \end{bmatrix} \Leftrightarrow \begin{bmatrix} Y_i(1) \\ Y_i(0) \end{bmatrix} = \begin{bmatrix} W_i & 1 - W_i \\ 1 - W_i & W_i \end{bmatrix} \begin{bmatrix} Y_i^{\text{obs}} \\ Y_i^{\text{mis}} \end{bmatrix} \tag{11.5}$$

- **Causal Effect** τ_i (defined by difference of PO): Difference between potential outcome, $\tau_i = Y_i(W_i = 1) - Y_i(W_i = 0) = Y_i(1) - Y_i(0)$
- **Pre-Treatment Variables / Covariates** X_i: Some background elements that might attribute to treatment selection/potential outcome. Anyway they may cause confusion to causal inference. For example, the gender of patients $X_i \in \{\text{female}, \text{male}\} := \{1, 0\}$.
- **Subgroup**: Treatment/Contorl group could be further divided in subgroup according to covariates, e.g. categorical covariates $X_i \in \mathcal{X}$

$$\{(X_i, Y_i, W_i)\} = \bigotimes_{\xi \in \mathcal{X}} \{(Y_i, W_i)\}_{i:X_i=\xi} \tag{11.6}$$

With the above basic notation, a dataset / sample can be expressed as

$$\mathcal{D} = \{(X_i, Y_i, W_i)\}_{i=1}^N \tag{11.7}$$

□ **Assignment Mechanism and Super Population**

- Our observation sample is a **finite sample** $\{X_i, Y_i\}_{i=1}^N$ in which Y_i is perceived fixed as potential outcome. And the above notation are studying the causal information within the finite sample. The randomness of the causal effect in the sample is the **assignment mechanism** $W_i \sim f_{W|X,Y}$. i.e. in finite sample, POs

are fixed and actually different assignment mechanisms give randomized data (in a finite sample). So if we can control the assignment mechanism $W|Y, X$, which is the case for randomized experiment, then the assignment mechanism can help estimate the missing values. Some widely used mechanism includes Completely Randomized Experiment, Stratified Randomized Experiment, Pairwise Randomized Experiment, etc. Proper assignment can help avoid the influence of covariants (recall Simpson's Paradox).

- Before that, the finite sample of $\{X_i, Y_i\}_{i=1}^N$ is drawn from a **super population** with some distribution.

To summarize, The whole model has 2 stages of randomness: sampling from super population, and assign treatment to the finite sample.

$$\text{Super Population} \xrightarrow[\text{sample } N]{f_X, f_{Y|X}} \text{Finite Sample } \{X, Y\} \xrightarrow[\text{assignment}]{f_{W|X,Y}} \text{Observation } \mathcal{D} = \{X, Y, W\} \tag{11.8}$$

11.1.2 Assumptions

The null model is complicated, say, there could be multiple PO levels / interference between assignments / complex assignment mechanism, etc. There are some basic assumptions to help simplified the model.

Note: In actual usage of causal model, the assumptions should be checked.

- **SUTVA**: To solve the problem of omitted treatment (e.g. $Y_i \in \{Y_i(0), Y_i(1), Y_i(2)\}$), and the intervention between units (e.g. $Y_i(W_{j=1:N})$) to simplify the model, we usually put the assumption of SUTVA, which has two components:
  - No Interference

$$Y_i(W_{j=1:N}) = Y_i(W_i) \tag{11.9}$$

  - No Hidden Variation of Treatment:

$$Y_i(W_{j=1:N}) = Y_i(W_i) \in \{Y_i(1), Y_i(0)\}, \quad W_i \in \{1, 0\} := \mathbb{T}_i = \mathbb{T} \tag{11.10}$$

- **Regular Assignment Mechanisms (RAM)**
  - **Individualistic Assignment**: Assignment probability of each unit does **not** depends on the covariants and PO of other units:

$$\begin{aligned}\mathbb{P}(W_i = w_i|X, Y(1), Y(0)) =& \mathbb{P}\left(W_i = w_i|X_i, Y_i(1), Y_i(0)\right) &(11.11)\\ =& \mathbb{P}(W_i|X_i, Y_i(1), Y_i(0))^{w_i}(1 - \mathbb{P}(W|X_i, Y_i(1), Y_i(0)))^{1-w_i}, \forall i = 1, 2, \ldots, N &(11.12)\end{aligned}$$

    Sometimes for simplification, denoted as

$$\mathbb{P}_i(W = 1|X, Y(1), Y(0)) := q(X, Y(1), Y(0)) \tag{11.13}$$

  - **Probabilistic Assignment**: Probility for both $W_i = 1$ and $W_i = 0$ are non-zero (to ensure a reasonable causal model)

$$0 < \mathbb{P}(W|X, Y(1)Y(0)) < 1, \forall X, Y(1), Y(0) \tag{11.14}$$

– **Unconfounded Assignment**: Assignment mechanism is independent of PO

$$\mathbb{P}(W|X,Y(1),Y(0)) = \mathbb{P}(W|X) \tag{11.15}$$

when $q(X,Y)$ mentioned above does *not* involve Y, i.e. with unconfoundedness, it is denoted as *propensity score*.

$$q(X,Y) := e(X), \quad \text{case } W \perp\!\!\!\perp Y \big| X \tag{11.16}$$

Note: Unconfoundedness is *not* testable (always invloves the missing value Y^{mis}). We can only pre-design it (randomized experiment) or make it an appropriate assumption (RAM).

□ **With all the above assumptions, assignment mechanism can be simplified in the following form:**

$$\text{Assignment Mechanism: } \mathbb{P}(W|X,Y(1),Y(0)) = \frac{1}{Z}\prod_{i=1}^{N} e(X_i)^{W_i}(1-e(X_i))^{1-W_i} \tag{11.17}$$

$$\tag{11.18}$$

□ **Data Example**

表 11.1: Illustration of Causal Data

	Potential Outcomes		Assignment	Observation	Causal Estimand
Unit i	$Y_i(1)$	$Y_i(0)$	W_i	Y_i^{obs}	$Y_i(1)-Y_i(0)$
# 1	$Y_1(1)$	$Y_1(0)$	$W_1=1$	$Y_1^{\text{obs}}=Y_1(1)$	$Y_1(1)-Y_1(0)$
# 2	$Y_2(1)$	$Y_2(0)$	$W_2=0$	$Y_2^{\text{obs}}=Y_2(0)$	$Y_2(1)-Y_2(0)$
# 3	$Y_3(1)$	$Y_3(0)$	$W_3=0$	$Y_3^{\text{obs}}=Y_3(0)$	$Y_3(1)-Y_3(0)$
# 4	$Y_4(1)$	$Y_4(0)$	$W_4=1$	$Y_4^{\text{obs}}=Y_4(1)$	$Y_4(1)-Y_4(0)$
⋮	⋮	⋮	⋮	⋮	⋮

Section 11.2 Inference to Causal Effect in Completely Randomized Experiment

First we focus on the randomness in finite sample, i.e. randomness of assignment mechanism. Specifically we usually consider the case of Completely Randomized Experiment (CRE): N_t in N items are given treatment and $N_c = N - N_t$ in N are given control, and the assignment is given **randomly**.

$$\mathbb{P}(W|X,Y) = 1 \Big/ \binom{N}{N_{\text{t}}}, \quad W \in \mathbb{W}^{\text{CRE}} := \left\{ W : \sum_{i=1}^{N} W_i = N_{\text{t}} \right\} \tag{11.19}$$

The assumption of CRP is important in causal inference because it fixes the gap between Y^{obs} and Y^{mis} by randomly assign treatment/control.

11.2.1 Fisher's Exact p-value

Test of Fisher's Sharp Null Hypothesis:

$$H_0 : \tau_i = 0, \forall i = 1, 2, \ldots, N \leftrightsquigarrow H_a : \exists j \; s.t. \tau_j \neq 0 \tag{11.20}$$

With the hypothesis, we could fill in all the Y^{mis} by $Y_i^{\text{mis}} = Y_i^{\text{obs}} \forall i$. And by traversing all possible $\tilde{W}$ assignments and calculate corresponding $\tau_{\tilde{W}}$, we could calculate the Fisher's exact p-value

$$\hat{p} = \#(|\tau_{\tilde{W}}| \geq |\tau_W|) \Big/ \binom{N}{N_t} \tag{11.21}$$

Comments:

- Since the basic idea is traversing all W, so it could be applied to differenet deigns of τ, say

$$\hat{\tau}^{\text{diff}} = |\bar{Y}_{\text{t}}^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}}| \tag{11.22}$$

$$\hat{\tau}^{\text{median}} = |\text{med}_{\text{t}}(Y^{\text{obs}}) - \text{med}_{\text{c}}(Y^{\text{obs}})| \tag{11.23}$$

$$\hat{\tau}^{t\text{-stat}} = \left| \frac{\bar{Y}_{\text{t}}^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}}}{\sqrt{s_{\text{t}}^2/N_t + s_{\text{c}}^2/N_c}} \right| \tag{11.24}$$

$$\hat{\tau}^{\text{rank}} = |\bar{R}_{\text{t}} - \bar{R}_{\text{c}}|, \quad R_i = \sum_{j=1}^{N} \left(\mathbb{I}_{Y_j < Y_i} + \frac{1}{2}\mathbb{I}_{Y_j = Y_i} \right) - \frac{N}{2} \tag{11.25}$$

$$\hat{\tau}^{\text{reg}} = \underset{\tau:(\beta_0, \beta_X, \tau)}{\arg\min} \sum_{i=1}^{N} \left(Y_i^{\text{obs}} - \beta_0 - \tau W_i - X_i' \beta_X \right)^2 \tag{11.26}$$

  Or even some other specially designed statistics on e.g. difference in variance

$$\hat{\tau}^{var} = \hat{var}_{\text{t}}^{\text{obs}} \big/ \hat{var}_{\text{c}}^{\text{obs}} \tag{11.27}$$

- High computation complexity for large N. e.g. for $N_t \approx \dfrac{N}{2}$

$$\text{flops} \sim \binom{N}{N_t} \sim 2^N \tag{11.28}$$

- Random simulation for large N: the p-value is actually

$$\hat{\mathbb{P}}\left(\text{more extreme } \hat{\tau}\right) = \hat{\mathbb{E}}\left[\mathbf{1}(\text{more extreme } \hat{\tau})\right] \tag{11.29}$$

  which can be approached by random sampling

$$\hat{p} = \#(|\tau_{\tilde{W}}| \geq |\tau_W|) \Big/ \#(\text{sample}) \tag{11.30}$$

- A fiducial interval can be constructed. But generally speaking the hypothesis testing just help reject the sharp hypothesis, but cannot help determine the casual effect strength.

11.2.2 Neyman's Repeated Sampling Approach

Neyman's method uses the distribution of W for completely randomized experiment to obtain the property of the finite sample estimator

$$\hat{\tau}_{\text{fs}} = \bar{Y}_{\text{t}}^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}} = \frac{1}{N_{\text{t}}} \sum_{i=1}^{N} W_i Y_i(1) - \frac{1}{N_{\text{c}}} \sum_{i=1}^{N} (1 - W_i) Y_i(0) \tag{11.31}$$

- Property

$$\mathbb{E}_W \left[\hat{\tau}_{\text{fs}}\right] = \tau_{\text{fs}} = \frac{1}{N} \sum_{i=1}^{N} Y_(1) - Y_i(0) \tag{11.32}$$

$$var_W(\hat{\tau}_{\text{fs}}) = \frac{S_{\text{t}}^2}{N_{\text{t}}} + \frac{S_{\text{c}}^2}{N_{\text{c}}} - \frac{S_{\text{tc}}^2}{N} \tag{11.33}$$

$$= \frac{N_{\text{c}}}{N N_{\text{t}}} S_{\text{t}}^2 + \frac{N_{\text{t}}}{N N_{\text{c}}} S_{\text{c}}^2 + \frac{2}{N} \rho_{\text{tc}} S_{\text{t}} S_{\text{c}} \tag{11.34}$$

$$\begin{cases} S_{\text{t}}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (Y_i(1) - \bar{Y}(1))^2 \\ S_{\text{c}}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (Y_i(0) - \bar{Y}(0))^2 \\ S_{\text{tc}}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left([Y_i(1) - Y_i(0)] - [\bar{Y}(1) - \bar{Y}(0)]\right)^2 = S_{\text{t}}^2 + S_{\text{c}}^2 - 2\rho_{\text{tc}} S_{\text{t}} S_{\text{c}} \\ \rho_{\text{tc}} = \frac{1}{(N-1) S_{\text{t}} S_{\text{c}}} \sum_{i=1}^{N} \left(Y_i(1) - \bar{Y}(1)\right) \left(Y_i(0) - \bar{Y}(0)\right) \end{cases} \tag{11.35}$$

- Estimator

$$\hat{\tau}_{\text{fs}} = \bar{Y}_{\text{t}}^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}} \tag{11.36}$$

$$\hat{var}(\hat{\tau}_{\text{fs}}) = \frac{s_{\text{t}}^2}{N_t} + \frac{s_{\text{c}}^2}{N_c} \tag{11.37}$$

$$\hat{var}_\rho(\hat{\tau}_{\text{fs}}) = \frac{N_{\text{c}}}{N N_{\text{t}}} s_{\text{t}}^2 + \frac{N_{\text{t}}}{N N_{\text{c}}} s_{\text{c}}^2 + \frac{2}{N} \rho s_{\text{t}} s_{\text{c}}, \ -1 \le \rho \le 1 \tag{11.38}$$

$$\text{e.g.} \hat{var}_{\rho=1}(\hat{\tau}_{\text{fs}}) = \frac{s_{\text{t}}^2}{N_t} + \frac{s_{\text{c}}^2}{N_c} - \frac{(s_{\text{t}} - s_{\text{c}})^2}{N} \le \hat{var}(\hat{\tau}_{\text{fs}}) \tag{11.39}$$

$$\begin{cases} s_{\text{t}}^2 = \frac{1}{N_t - 1} \sum_{i:W_i=1} \left(Y_i^{\text{obs}} - \bar{Y}_{\text{t}}^{\text{obs}}\right)^2 \\ s_{\text{c}}^2 = \frac{1}{N_c - 1} \sum_{i:W_i=0} \left(Y_i^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}}\right)^2 \end{cases} \tag{11.40}$$

  i.e. $\hat{var}(\hat{\tau}_{\text{fs}})$ provides an upper bound of $\hat{var}_\rho(\hat{\tau}_{\text{fs}})$ (equal when $\rho = 1$). And $\hat{var}(\hat{\tau}_{\text{fs}})$ also acts as the estimator at $\tau_i = \text{const}, \forall i$.[3]

- Confidence Interval

$$\text{CI} = \hat{\tau}_{\text{fs}} \pm N_{\alpha/2} \sqrt{\hat{var}(\hat{\tau}_{\text{fs}})} \tag{11.41}$$

$$\text{CI}_\rho = \hat{\tau}_{\text{fs}} \pm N_{\alpha/2} \sqrt{\hat{var}_\rho(\hat{\tau}_{\text{fs}})} \tag{11.42}$$

[3] Actually in this case we should have $s_{\text{t}} = s_{\text{c}} := s$ and the estimator reduces to $\hat{var}(\hat{\tau}_{\text{fs}}) = s^2(1/N_{\text{t}} + 1/N_{\text{c}})$

where the version with pre-specified ρ is applied to improve accuracy, if we have prior knowledge about ρ_{tc}.

- Hypothesis Testing

$$H_0 : \bar{Y}(1) - \bar{Y}(0) = 0 \nleftrightarrow H_a : \bar{Y}(1) - \bar{Y}(0) \neq 0 \tag{11.43}$$

and t-test

$$T = \frac{\hat{\tau}_{\text{fs}}}{\sqrt{\hat{var}(\hat{\tau}_{\text{fs}})}} \sim t_1 \tag{11.44}$$

- Comment on three components S_{t}^2 / S_{c}^2 / S_{tc}^2: they each corresponds to the natural distribution of treatment / natrual distribution of control / variation arises from assigning on finite sample.

  So when dealing with the estimator under distribution of super population, in which we need to add the randomness of $f_{X,Y}$ back, the S_{tc}^2 term eliminates (which can be proven).

$$\mathbb{E}_{\text{sp}}\left[\hat{\tau}_{\text{fs}}\right] = \mathbb{E}_{\text{sp}}\left[\mathbb{E}_W\left[\hat{\tau}_{\text{fs}}\right]\right] = \tau_{\text{sp}} \tag{11.45}$$

$$var_{\text{sp}}(\hat{\tau}_{\text{fs}}) = \mathbb{E}_{\text{sp}}\left[(\bar{Y}_{\text{t}}^{\text{obs}} - \bar{Y}_{\text{c}}^{\text{obs}} - \mathbb{E}_{\text{sp}}\left[\bar{Y}(1) - \bar{Y}(0)\right])^2\right] = \frac{\sigma_{\text{t}}^2}{N_t} + \frac{\sigma_{\text{c}}^2}{N_c} \tag{11.46}$$

$$\hat{var}_{\text{sp}}(\hat{\tau}_{\text{fs}}) = \frac{s_{\text{t}}^2}{N_t} + \frac{s_{\text{c}}^2}{N_c} \tag{11.47}$$

where $\sigma_{\cdot}^2$ is the variance under the distribution of super population $Y|X$, X.

$$\sigma_{\text{t}}^2(x) = var_{\text{sp: }Y|X}\left(Y(1)|X = x\right), \qquad \sigma_{\text{t}}^2 = var_{\text{sp}}\left(Y(1)\right) \tag{11.48}$$

$$\sigma_{\text{c}}^2(x) = var_{\text{sp: }Y|X}\left(Y(0)|X = x\right), \qquad \sigma_{\text{c}}^2 = var_{\text{sp}}\left(Y(0)\right) \tag{11.49}$$

$$\sigma_{\text{ct}}^2(x) = var_{\text{sp: }Y|X}\left(Y(1) - Y(0)|X = x\right), \qquad \sigma_{\text{ct}}^2 = var_{\text{sp}}\left(Y(1) - Y(0)\right) \tag{11.50}$$

11.2.3 Regression Methods

Regression methods in Potential Outcome Framework is used to introduce covariates and lower the variance estimation, the idea is similar to variance decomposition in ANOVA.

□ **Requisite Knowledge: M-Estimator**

With data $\mathcal{D}_n$ given, parameter estimation problem can usually be expressed in a **M**aximization problem with linear combination target function $Q_n(\theta; \mathcal{D}_n)$

$$\hat{\theta}_n = \arg\max_{\theta} Q_n(\theta; \mathcal{D}_n) \tag{11.51}$$

e.g. for regression estimation $Y = X\beta + \varepsilon$, $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n = (X, Y)$

- OLS quadratic form $\theta = \beta$

$$Q_n(\theta) := -\frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i'\beta\right)^2 = -\frac{1}{n}(Y - X\beta)'(Y - X\beta) \tag{11.52}$$

- MLE form with $\varepsilon \sim f(\varepsilon;\phi)$, and $\theta = (\beta, \phi)$

$$Q_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} f\left(y_i - x_i'\beta;\phi\right) \tag{11.53}$$

Denote the ground truth θ^*, and the M-Estimator $\hat{\theta}_n$ that maximizes Q_n. Then

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{\mathcal{D}}\left[Q(\theta;\mathcal{D})\right] \tag{11.54}$$

$$\hat{\theta}_n = \arg\max_{\theta} Q_n \tag{11.55}$$

$$\text{with } Q_n \to \mathbb{E}\left[Q\right] \Rightarrow \hat{\theta}_n \to \theta^* \tag{11.56}$$

The solution $\hat{\theta}_n$ is obtained at $\dfrac{\partial Q_n(\theta)}{\partial\theta} = 0$, so we first focus on first order derivative

$$\psi_n(\theta;\mathcal{D}_n) := \frac{\partial Q_n(\theta;\mathcal{D}_n)}{\partial\theta}, \quad \hat{\theta}_n = \arg_{\theta}(\psi_n(\theta;\mathcal{D}_n) = 0) \tag{11.57}$$

Note: a more important reason we study the property of $\psi_n(\theta;\mathcal{D}_n)$ is that: we do **not** have an explicit expression of $\hat{\theta}_n$ because it's just a maximizer. $\psi_n(\theta;\mathcal{D}_n)$ together with taylor expansion provide us with an approach to (asymptotically) express $\hat{\theta}_n$ explicitly.

with LLN, we have

$$\psi_n(\theta^*;\mathcal{D}_n) \xrightarrow{\text{d}} \mathbb{E}\left[\psi(\theta^*)\right] = 0 \tag{11.58}$$

with CLT, $\psi_n(\theta^*,\mathcal{D}_n)$ is a statistic asymptotically distributed normally:

$$\sqrt{n}\left(\psi_n(\theta^*;\mathcal{D}_n) - \mathbb{E}\left[\psi(\theta^*;\mathcal{D})\right]\right) = \sqrt{n}\psi_n(\theta^*;\mathcal{D}_n) \xrightarrow{d} N(0,\Sigma_\psi) \tag{11.59}$$

Taylor series of $\psi_n(\,\cdot\,;\mathcal{D}_n)$ at $\hat{\theta}_n$:

$$\psi_n(\theta^*;\mathcal{D}_n) = 0 + \frac{\partial\psi_n(\theta=\hat{\theta}_n;\mathcal{D}_n)}{\partial\theta}\left(\theta^* - \hat{\theta}_n\right) + O((\theta^* - \hat{\theta}_n)^2) \tag{11.60}$$

$$\Rightarrow \left(\hat{\theta}_n - \theta^*\right) \approx \left(\frac{\partial\psi_n(\theta=\hat{\theta}_n;\mathcal{D}_n)}{\partial\theta}\right)^{-1} \psi_n(\theta^*;\mathcal{D}_n) \tag{11.61}$$

$$\Rightarrow \hat{\theta}_n \xrightarrow{d} N(\theta^*, \Gamma^{-1}\Sigma_\psi\Gamma^{-1}/n), \qquad \Gamma := \frac{\partial\psi_n(\theta=\hat{\theta}_n;\mathcal{D}_n)}{\partial\theta} \tag{11.62}$$

and specifically if $Q_n(\theta;\mathcal{D})$ is a log-likelihood, then $\psi(\theta;\mathcal{D})$ here is Score function in equation 2.78 ~ page 48. And $\Sigma_\psi = I(\theta)$ is Fisher Information in equation 2.89 ~ page 48. With the nice property of Fisher Information $I(\theta) = \Sigma_\psi = -\mathbb{E}\left[\Gamma\right]$, M-Estimator reduces to the asymptotic distribution of MLE in equation 2.68 ~ page 46

$$\hat{\theta}_n \xrightarrow{d} (\theta^*, \frac{I(\theta)^{-1}}{n}) \tag{11.63}$$

□ **Regression Model on Super Population**

Motivation: regression model

$$Y_i^{\text{obs}} = \alpha + \tau\cdot W_i + f(X_i;\beta) + \varepsilon_i \tag{11.64}$$

concerns a quadratic loss function of the following form:

$$(\hat{\alpha},\hat{\tau},\hat{\beta})_{\text{ols},X}=\underset{(\alpha,\tau,\beta)}{\arg\min}\frac{1}{N}\sum_{i=1}^{N}\left(Y_i^{\text{obs}}-\alpha-\tau\cdot W_i-f(X_i;\beta)\right)^2:=\underset{(\alpha,\tau,\beta)}{\arg\min}\,Q_N\left((\alpha,\tau,\beta);\{X_i,Y_i,W_i\}_{i=1}^N\right) \tag{11.65}$$

Note :

- Covariate dependency function $f(\,\cdot\,;\beta)$ is a properly selected prior, e.g. linear regression $X'\beta$
- In functional form it's the same as regression (reflects correlation), the causality comes from CRE of W_i.

Solution:

- Model without covariates

$$Y_i^{\text{obs}}=\alpha+\tau\cdot W_i+\varepsilon_i \tag{11.66}$$

OLS solution:

$$\hat{\tau}_{\text{ols}}=\frac{\sum_{i=1}^N(W_i-\bar{W})(Y_i^{\text{obs}}-barY^{\text{obs}})}{\sum_{i=1}^N(W_i-\bar{W})^2}=\bar{Y}_{\text{t}}^{\text{obs}}-\bar{Y}_{\text{c}}^{\text{obs}} \tag{11.67}$$

$$\hat{\alpha}_{\text{ols}}=\bar{Y}^{\text{obs}}-\hat{\tau}_{\text{ols}}\cdot\bar{W} \tag{11.68}$$

$$var(\hat{\tau})=\frac{\sigma_{\text{t}}^2}{N_{\text{t}}}+\frac{\sigma_{\text{c}}^2}{N_{\text{c}}} \tag{11.69}$$

$$s_{\text{t}}^2=\hat{\sigma}_{\text{t}}^2=\frac{1}{N-1}\sum_{i=1}^{N}W_i\left(Y_i^{\text{obs}}-\hat{Y}_i^{\text{obs}}\right)^2=\frac{1}{N-1}\sum_{i=1}^{N}W_i\left(Y_i^{\text{obs}}-\hat{\tau}-\hat{\alpha}\right)^2 \tag{11.70}$$

$$s_{\text{c}}^2=\hat{\sigma}_{\text{c}}^2=\frac{1}{N-1}\sum_{i=1}^{N}(1-W_i)\left(Y_i^{\text{obs}}-\hat{Y}_i^{\text{obs}}\right)^2=\frac{1}{N-1}\sum_{i=1}^{N}(1-W_i)\left(Y_i^{\text{obs}}-\hat{\alpha}\right)^2 \tag{11.71}$$

$$\hat{var}(\hat{\tau}_{\text{ols}})=\frac{s_{\text{t}}^2}{N_t}+\frac{s_{\text{c}}^2}{N_c}=\hat{var}(\hat{\tau}_{\text{fs}}) \tag{11.72}$$

- Model with Covariates and Asymptotic Property:

$$Y_i^{\text{obs}}=\alpha+\tau\cdot W_i+f(X_i;\beta)+\varepsilon_i \tag{11.73}$$

The quadratic loss $Q_N(\,\cdot\,)$ regression model gives a M-Estimator with ground truth as the parameters in super population

$$(\alpha,\tau,\beta)^*=\mathbb{E}_{\text{sp}}\left[(\alpha,\tau,\beta)\right]:=(\alpha_{\text{sp}},\tau_{\text{sp}},\beta_{\text{sp}}) \tag{11.74}$$

OLS with covariates gives the same optimization solution to τ as OLS without covariate: $\hat{\tau}_{\text{ols},X}=\hat{\tau}_{\text{ols}}\to\tau^*=\tau_{\text{sp}}$. And appending covariate dependency term can help **improve variance estimation** , with unbiasness property kept.

e.g. for linear dependency $f(X;\beta)=X'\beta$

$$\hat{\tau}_{\text{ols},X}=\hat{\tau}_{\text{ols}}\to\tau_{\text{sp}} \tag{11.75}$$

$$\sqrt{N}(\hat{\tau}_{\text{ols},X}-\tau_{\text{sp}})\xrightarrow{d}N\left(0,(\Gamma^{-1}\Sigma_{\psi}\Gamma^{-1})_{22}\right)=N\left(0,\frac{\Sigma_{\psi,22}}{p^2(1-p)^2}\right) \tag{11.76}$$

$$\begin{cases}\Sigma_{\psi,22}=\mathbb{E}_{\text{sp}}\left[\dfrac{\partial Q_N(\alpha,\tau,\beta)}{\partial(\alpha,\tau,\beta)}\dfrac{\partial Q_N(\alpha,\tau,\beta)}{\partial(\alpha,\tau,\beta)'}\right]_{22}\\ \qquad=\mathbb{E}\left[(W_i-p)^2\left(Y^{\text{obs}}-\alpha^*-\tau^*W_i-X'\beta^*\right)^2\right]\\ p=\bar{W}_{N\to\infty}\end{cases} \tag{11.77}$$

using the asymptotic normality, we can construct variance estimation $\hat{var}_{\text{hetero}}(\hat{\tau}_{\text{ols},X})=\hat{\Sigma}_{\psi,22}/\hat{p}^2(1-\hat{p})^2$ to $\hat{\tau}_{\text{ols},X}$ (with heteroskedasticity)

$$\hat{var}_{\text{hetero}}(\hat{\tau}_{\text{ols},X})=\frac{1}{N(N-\dim(X_i)-1)}\cdot\frac{\sum_{i=1}^N(W_i-\bar{W})^2\left(Y_i^{\text{obs}}-\hat{\alpha}_{\text{ols},X}-\hat{\tau}_{\text{ols},X}W_i-X_i'\hat{\beta}_{\text{ols},X}\right)^2}{\bar{W}^2\left(1-\bar{W}\right)^2} \tag{11.78}$$

11.2.4 Model Based Inference using Bayesian Statistics

Motivation: how to use prior information about distribution? Basically with $\mathbb{P}(W|Y,\theta)$, $f(Y|\theta)$, $\pi(\theta)$ we can construct any posterior distribution from

$$f(W,Y,X)=\mathbb{P}(W|Y,\theta)\,f(Y|\theta)pi(\theta) \tag{11.79}$$

□ **Bayesian Statistics Precap**

Estimation target: $f(Y^{\text{mis}}|Y^{\text{obs}},W)$, with assumptions

$$\text{CRE: }\mathbb{P}(W|Y,\theta)=\binom{N}{N_t}^{-1} \tag{11.80}$$

$$\text{Distribution: }\begin{pmatrix}Y(1)\\Y(0)\end{pmatrix}\Big|\,\theta\sim f(Y|\theta),\ \text{say } N\left(\begin{pmatrix}\mu_{\text{t}}\\\mu_{\text{c}}\end{pmatrix},\begin{pmatrix}\sigma_{\text{t}}^2&\rho\sigma_{\text{t}}\sigma_{\text{c}}\\\rho\sigma_{\text{t}}\sigma_{\text{c}}&\sigma_{\text{c}}^2\end{pmatrix}\right),\quad\theta=[\mu_{\text{t}},\mu_{\text{c}},\sigma_{\text{t}}^2,\sigma_{\text{c}}^2,\rho] \tag{11.81}$$

$$\text{Prior: }\theta\sim\pi(\theta) \tag{11.82}$$

$$\text{Transformation: }(Y^{\text{obs}},Y^{\text{mis}})=g(Y(1),Y(0),W) \tag{11.83}$$

- Transformation between $Y=[Y(1),Y(0)]\mapsto[Y^{\text{obs}},Y^{\text{mis}}]$:

$$f\left(Y^{\text{obs}},Y^{\text{mis}}|W,\theta\right)=f(Y|W,\theta)\left|\frac{\partial Y(1),Y(0)}{\partial g(Y(1),Y(0),W)}\right|=\frac{f(Y,W|\theta)}{\int_y f(Y,W|\theta)\,\mathrm{d}y}\left|\frac{\partial Y(1),Y(0)}{\partial g(Y(1),Y(0),W)}\right| \tag{11.84}$$

$$=\frac{f(W|Y,\theta)\,f(Y|\theta)}{\int_y f(W|Y,\theta)\,f(Y|\theta)\,\mathrm{d}y}\left|\frac{\partial g(Y(1),Y(0),W)}{\partial Y(1),Y(0)}\right|^{-1} \tag{11.85}$$

$$\Rightarrow f\left(Y^{\text{mis}}|Y^{\text{obs}},W,\theta\right)=\frac{f\left(Y^{\text{obs}},Y^{\text{mis}}|W,\theta\right)}{\int_{y^{\text{mis}}}f\left(Y^{\text{obs}},Y^{\text{mis}}|W,\theta\right)\mathrm{d}y^{\text{mis}}} \tag{11.86}$$

- Calculating posterior of parameter

$$p(\theta|Y^{\text{obs}},W)=\frac{\pi(\theta)\cdot f(Y^{\text{obs}},W|\theta)}{f(Y^{\text{obs}},W)}=\frac{\pi(\theta)\cdot\int_{y^{\text{mis}}}f(W|Y,\theta)f(Y^{\text{obs}},Y^{\text{mis}}|\theta)\,\mathrm{d}y^{\text{mis}}}{\int_\theta\pi(\theta)\cdot\int_{y^{\text{mis}}}f(W|Y,\theta)f(Y^{\text{obs}},Y^{\text{mis}}|\theta)\,\mathrm{d}y^{\text{mis}}\,\mathrm{d}\theta} \tag{11.87}$$

- Marginal Integration

$$f\left(Y^{\text{mis}}|Y^{\text{obs}},W\right)=\int_\theta f\left(Y^{\text{mis}},\theta|Y^{\text{obs}},W\right)\,\mathrm{d}\theta \tag{11.88}$$

$$=\int_\theta f\left(Y^{\text{mis}}|Y^{\text{obs}},W,\theta\right)p(\theta|Y^{\text{obs}},W)\,\mathrm{d}\theta \tag{11.89}$$

With the above (a little bit complex) steps we could estimate Y^{mis}, and also give the (bayesian posterior) distribution of $\hat{\tau}$

$$f(\tau|Y^{\text{obs}},W)=f(Y^{\text{obs}}-Y^{\text{mis}}|Y^{\text{obs}},W) \tag{11.90}$$

Model with covariate involved need modification with assumptions as

$$f\left(Y(1),Y(0),X|\theta_{Y|X},\theta_X\right)=f\left(Y(1),Y(0)|X,\theta_{Y|X}\right)\cdot f\left(X|\theta_X\right) \tag{11.91}$$

$$\pi(\theta_{Y|X},\theta_X)=\pi(\theta_{Y|X})\pi(\theta_X) \tag{11.92}$$

and corresponding intergrations need to consider intergral on X.

Usually computation of integral terms is complex, simulation methods like random integration can be used, see section 5.6 ~ page 185 and section 13.3 ~ page 343 for a brief introduction.

Section 11.3 More Assignment Mechanism and Observational Study

Some other classical randomized experiment are also used in causal experiments. This section includes Stratified Randomized Experiment (SRE) and Pairwise Randomized Experiment (PRE).

In more cases we can only deal with observational data, which means the assignment mechanism is beyond our control, thus some estimation is needed.

11.3.1 Other Classical Randomized Experiment

□ Stratified Randomized Experiment

Usually when we notice that some covariate X can have significant influence on $\hat{\tau}$, we consider a SRE by dividing data into stratum according to X

$$\mathcal{S}:\ (N,N_{\text{t}},N_{\text{c}})\to\{(N(j),N_{\text{t}}(j),N_{\text{c}}(j))\}_{j=1}^{J},\qquad S_i:=\mathcal{S}(X_i)=\text{Strata of }\mathcal{D}_i\ \in\{1,2,\ldots,J\} \tag{11.93}$$

with *proportion of strata* $q(j)$ and *propensity score* $e(j)$

$$q(j):=\frac{N(j)}{N}\qquad e(j)=\frac{N_{\text{t}}(j)}{N(j)} \tag{11.94}$$

SRE Assignment Mechanism:

$$\mathbb{P}(W|S,Y) = \prod_{j=1}^{J} \binom{N(j)}{N_{\mathrm{t}}(j)}^{-1}, \quad W \in \mathbb{W}^{\mathrm{SRE}} := \left\{ W : \sum_{i=1}^{N} W_i \mathbb{I}_{S_i=j} = N(j),\ \forall j = 1,2,\ldots,J \right\} \tag{11.95}$$

With the notation above, the within-strata ACE $\tau(j)$ estimation follows exactly the same estimation as in CRE, the key step is to *aggregate* $\{\tau(j)\}_{j=1}^{J} \mapsto \tau$.

- **Fisher's Exact p-Value**: with the same sharp null hypothesis

$$H_0 : \tau_i = 0,\ \forall i = 1,2,\ldots,N \leftrightsquigarrow H_a : \exists j\ s.t. \tau_j \neq 0 \tag{11.96}$$

we can conduct similar testing by traversing $W \in \mathbb{W}^{\mathrm{SRE}}$, with a slight modification on test statistics, e.g. using $\hat{\tau}^{\mathrm{diff}} \rightsquigarrow \hat{\tau}^{\mathrm{diff},\lambda}$ as example. Some other statistics used see equation 11.22 ~ page 289

$$\hat{\tau}^{\mathrm{diff}} = \left| \bar{Y}_{\mathrm{t}}^{\mathrm{obs}} - \bar{Y}_{\mathrm{c}}^{\mathrm{obs}} \right| \tag{11.97}$$

$$\hat{\tau}^{\mathrm{diff},\lambda} = \left| \sum_{j=1}^{J} \lambda(j) \left(\bar{Y}_{\mathrm{t}}^{\mathrm{obs}}(j) - \bar{Y}_{\mathrm{c}}^{\mathrm{obs}}(j) \right) \right|, \quad w.r.t. \sum_{i=1}^{J} \lambda(j) = 1 \tag{11.98}$$

Note: $\hat{\tau}^{\mathrm{diff},\mu}$ returns to $\hat{\tau}^{\mathrm{diff}}$ if λ is chosen as proportion of strata $\lambda(j) = q(j)$.

- Neyman's Repeated Sampling Approach: use similar aggregation method of weighting strata to form unbiased estimator

$$\hat{\tau}_{\mathrm{fs}} = \sum_{i=1}^{J} q(j)\hat{\tau}(j), \quad \hat{\tau}(j) = \frac{1}{N_{\mathrm{t}}(j)} \sum_{i:S_i=j} W_i Y_i^{\mathrm{obs}} - \frac{1}{N_{\mathrm{c}}(j)} \sum_{i:S_i=j} (1-W_i) Y_i^{\mathrm{obs}} \tag{11.99}$$

$$var(\hat{\tau}_{\mathrm{fs}}) = \sum_{j=1}^{J} q(j)^2 var\left(\hat{\tau}(j)\right) = \sum_{j=1}^{J} q(j)^2 \left(\frac{S_{\mathrm{t}}^2(j)}{N_{\mathrm{t}}(j)} + \frac{S_{\mathrm{c}}^2(j)}{N_{\mathrm{c}}(j)} - \frac{S_{\mathrm{tc}}^2(j)}{N(j)} \right) \tag{11.100}$$

$$\hat{var}(\hat{\tau}_{\mathrm{fs}}) = \sum_{j=1}^{J} q(j)^2 \hat{var}\left(\hat{\tau}(j)\right) = \sum_{j=1}^{J} q(j)^2 \left(\frac{s_{\mathrm{t}}^2(j)}{N_{\mathrm{t}}(j)} + \frac{s_{\mathrm{c}}^2(j)}{N_{\mathrm{c}}(j)} \right) \tag{11.101}$$

$$\begin{cases} s_{\mathrm{t}}^2(j) = \dfrac{1}{N_{\mathrm{t}}(j)-1} \sum_{i:S_i=j,W_i=1} (Y_i^{\mathrm{obs}} - \bar{Y}_{\mathrm{t}}^{\mathrm{obs}}(j))^2 \\ s_{\mathrm{c}}^2(j) = \dfrac{1}{N_{\mathrm{c}}(j)-1} \sum_{i:S_i=j,W_i=0} (Y_i^{\mathrm{obs}} - \bar{Y}_{\mathrm{c}}^{\mathrm{obs}}(j))^2 \end{cases} \tag{11.102}$$

- Regression Method: basic stratified regression model:

$$Y_i^{\mathrm{obs}} = \tau \cdot W_i + \sum_{j=1}^{J} \beta_j \mathbb{I}_{S_i=j} + \varepsilon_i \tag{11.103}$$

MMSE limit:

$$\hat{\tau}_{\mathrm{ols}} \to \tau^* = \frac{1}{\sum_{k=1}^{J} q(k)e(k)(1-e(k))} \sum_{j=1}^{J} q(j)e(j)(1-e(j))\tau_{\mathrm{sp}}(j) \tag{11.104}$$

- Model Based Infernece: Similar process as in CRE. We could further assess population average by setting *hyper parameter* ϕ

$$Y(j)|\theta(j) \sim f(Y|\theta(j)), \qquad \theta_j|\phi \sim \pi_\theta(\theta_j|\phi), \quad \phi \sim \pi_\phi(\phi) \tag{11.105}$$

□ **Pairwise Randomized Experiment**

Pairwise Randomized Experiment (PRE) can (in some sense) be vies as a special case that $J = \frac{N}{2}$, which can deal with continuous covariate cases. But a main difficult arises in variance estimation in Neyman's mathod.

To estimate the variance, we put assumption of constant causal effect within group, which gives

$$S_{\mathrm{t}}^2(j) = S_{\mathrm{c}}^2(j) \equiv S^2, \quad S_{\mathrm{tc}}^2(j) = 0 \tag{11.106}$$

and we can access $\hat{var}(\tau_{\mathrm{fs}})$ as

$$var(\tau_{\mathrm{fs}}) = \frac{4}{N} S^2 \tag{11.107}$$

$$\hat{var}(\tau_{\mathrm{fs}}) = \frac{4}{N(N-2)} \sum_{i=1}^{N/2} \left(\hat{\tau}(j) - \bar{\hat{\tau}}\right)^2 \tag{11.108}$$

11.3.2 Observational Study with Regular Assignment Mechanisms

Recap RAM:

$$\begin{cases} \text{Individualistic:} & \mathbb{P}(W_i|X,Y) = \mathbb{P}(W_i|X_i,Y_i) := q(X,Y) \\ \text{Probabilistic:} & 0 < \mathbb{P}(W_i|X,Y) < 1 \\ \text{Unconfounded:} & \mathbb{P}(W|X,Y) = \mathbb{P}(W|X) \end{cases} \Rightarrow \mathbb{P}(W|X,Y) = \frac{1}{Z} \prod_{i=1}^{N} e(X_i)^{W_i} (1 - e(X_i))^{1-W_i} \tag{11.109}$$

With the above assumptions and notations, propensity score $e(x)$ can help fix the problem of Simpson's Paradox by **Covariate Balance**

$$W_i \perp\!\!\!\perp X_i \big| e(X_i) \tag{11.110}$$

Note: there could be some other selection of balancing variable $\epsilon(x)$, in which e_i is the coarsest, i.e. $e(x) = e(\epsilon(x))$

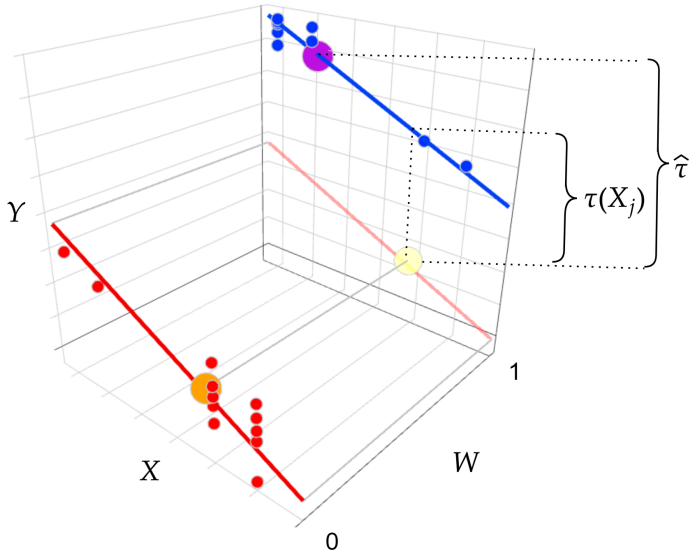


图 11.1: Illustration of covariate balance of propensity score (An example with linear dependence)

□ Statistical Inference to Propensity Score

Property of propensity score:

$$\Delta_{\text{tc}} := \mathbb{E}\left[e(X)|W=1\right] - \mathbb{E}\left[e(X)|W=0\right] = \frac{var(e(X))}{p(1-p)} \tag{11.111}$$

$$var(e(X)) = \mathbb{E}\left[\left(\frac{f_{\text{t}}(X) - f_{\text{c}}(X)}{pf_{\text{t}}(X) + (1-p)f_{\text{c}}(X)}\right)^2\right] \cdot p^2(1-p)^2 \tag{11.112}$$

- Propensity Score test can be accessed by

$$\hat{\Delta}^{\ell}_{\text{tc}} = \frac{\bar{\ell}_{\text{t}} - \bar{\ell}_{\text{c}}}{\sqrt{(s^2_{\ell,\text{t}} + s^2_{\ell,\text{c}})/2}} \sim t_{N-2}, \quad \ell(x) = \ln\left(\frac{e(x)}{1-e(x)}\right) = \text{logistic}(x) \tag{11.113}$$

$$\hat{\Delta}^{\ell}_{\text{tc}} = 0 \leftrightsquigarrow \Delta_{\text{tc}} = 0 \leftrightsquigarrow var(e(X)) = 0 \leftrightsquigarrow f_{\text{t}}(x) = f_{\text{c}}(x) \tag{11.114}$$

- Estimate $\hat{e}(X_i)$
  - For categorical X with small $|\mathcal{X}|$, estimation

$$\hat{e}(x) = \frac{N(X_j)}{N} \tag{11.115}$$

  - (Kernel) logistic regression is sometimes useful[4]

$$\hat{e}(x) = \hat{\mathbb{P}}\left(W_i = 1 | X_i = x; \beta\right) = \frac{e^{x'\beta}}{1 + e^{x'\beta}} \tag{11.116}$$

□ Useful Methods to Induce Propensity Score in Estimation

- Weighting: using the modulation of $e(x)$ on $\mathbb{P}\left(W|X\right)$

$$\begin{cases} \mathbb{E}\left[\dfrac{Y_i^{\text{obs}} \cdot W_i}{e(X_i)}\right] = \mathbb{E}\left[\dfrac{\mathbb{E}\left[Y_i(1)|X_i\right]\mathbb{E}\left[W_i|X_i\right]}{e(X_i)}\right] = \mathbb{E}\left[Y_i(1)\right] \\ \mathbb{E}\left[\dfrac{Y_i^{\text{obs}} \cdot (1-W_i)}{1-e(X_i)}\right] = \mathbb{E}\left[\dfrac{\mathbb{E}\left[Y_i(0)|X_i\right]\mathbb{E}\left[1-W_i|X_i\right]}{1-e(X_i)}\right] = \mathbb{E}\left[Y_i(0)\right] \end{cases} \tag{11.117}$$

  to *weight* estimators through X: Horvitz-Thompson Estimator

$$\hat{\tau}^{\text{HT}} = \frac{1}{N}\sum_{i=1}^{N}\frac{W_i Y_i^{\text{obs}}}{\hat{e}(X_i)} - \frac{1}{N}\sum_{i=1}^{N}\frac{(1-W_i)Y_i^{\text{obs}}}{1-\hat{e}(X_i)} = \frac{1}{N}\sum_{i=1}^{N}\frac{(W_i - e(X_i))\cdot Y_i^{\text{obs}}}{e(X_i)\cdot(1-e(X_i))} \tag{11.118}$$

$$\hat{\tau}^{\text{HT,mod}} = \sum_{i=1}^{N}\lambda_i W_i Y_i^{\text{obs}} - \sum_{i=1}^{N}\lambda_i(1-W_i)Y_i^{\text{obs}}, \quad \lambda_i = \begin{cases} \dfrac{1/\hat{e}(X_i)}{\sum_{k=1}^{N} W_i/\hat{e}(X_j)}, & W_i = 1 \\ \dfrac{1/(1-\hat{e}(X_i))}{\sum_{k=1}^{N}(1-W_i)/(1-\hat{e}(X_k))}, & W_i = 0 \end{cases} \tag{11.119}$$

  where the modification version is used to avoid extreme $\hat{e}$ value.

  The Horvitz-Thompson estimator is linked to stratified Neyman estimator equation 11.99 ~ page 296 as

$$\hat{\tau}^{\text{strata}} = \sum_{j=1}^{J} q(j)\hat{\tau}(j) = \frac{1}{N}\sum_{i=1}^{N}\tilde{e}_i W_i Y_i^{\text{obs}} - \sum_{i=1}^{N}\tilde{e}_i(1-W_i)Y_i^{\text{obs}}, \quad \tilde{e}_i = \begin{cases} \mathbb{I}_{S_i=j}\dfrac{1}{N_{\text{t}}(j)/N(j)}, & W_i = 1 \\ \mathbb{I}_{S_i=j}\dfrac{1}{N_{\text{c}}(j/N(j))}, & W_i = 0 \end{cases} \tag{11.120}$$

  where $\tilde{e}_i$ is the propensity score for each strata.

[4] Instruction of Kernel logistic regression see section 9.4.5 ~ page 258.

- Blocking / Stratifying according to X, and then follows similar idea as SRE. (Because $S(X_i)$ is still a covariate.)
- Matching 'similar' data points. e.g. for each data point $(W_i = 1, Y_i, X_i)$, select in $\mathcal{D}_{W=1-W_i=0}$ for units with small distance $d(X_i, X)$ as $\mathcal{M}_i$, and have a matching data

$$\{(W_i = 1, Y_i^{\text{obs}}, X_i, \mathcal{M}_i)\}, \quad \mathcal{M}_i = \{(W_j = 0, Y_j, X_j)\}_{d(X_i,X_j) \text{ small}} \tag{11.121}$$

and then

$$\hat{\tau} = \frac{1}{N_{\text{t}}} \sum_{i:W_i=1} \left(Y_i^{\text{obs}} - \bar{Y}_{\mathcal{M}_i} \right) \tag{11.122}$$

Section 11.4 Pearl Causal Bayesian Framework

Pearl Bayesian Framework[5] (Judea Pearl, 1995) uses causal information on a graph to construct inference.

11.4.1 Causal Bayesian Network

The language of *Graph* is used to describe the causal relations.

□ **Directed Acyclic Graph**

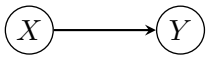
In Pearl's causal network we focus on **Directed Acyclic Graphs** (DAGs) . Here are some key notions:

▷ DAG is a graph in which all edges are directed, and no path is a loop (acyclic).

▷ **Graph** $\mathcal{G}$ is composed of a set of **Vertices** / Nodes $\mathcal{V}$ and the **Edge**s $\mathcal{E}$ connecting them; $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$.

- Adjacency: Two vertices v_i, v_j are adjacent if they are linked by an edge e_{ij}.
- **Path**: A (non-intersecting) routine tracing through edges to connect two vertices.

▷ **Direction** of edges: two vertices are connected by directed edge, pointing from the first to the second, say the following meta $X \to Y$.



in which X is a **parent** of Y and Y is a children of X. Parent of note v_i is denoted pa_i

- **Skeleton** : The graph with all direction removed (looks like a graph with only nodes and line, without arrow).

▷ **Acyclic**: a graph without loop is acyclic. The structure is naturally required to make the causal structure healthy by clearly distinguish cause from effect.

□ **Bayesian Network**

[5] Also called Bayesian Network / Belief Network / Directed Acyclic Graphical (DAG) Model.

A probability distribution $\mathbb{P}(X_1, X_2, \ldots, X_n)$ on vertices of graph has factorization given by conditional probability:

$$\mathbb{P}(X_1, \ldots, X_n) = \mathbb{P}(X_{i_1}|X_{i_2}, \ldots)\,\mathbb{P}(X_{i_2}|X_{i_3}, \ldots)\ldots\mathbb{P}\left(X_{i_{n-1}}|X_{i_n}\right)\mathbb{P}(X_n) \tag{11.123}$$

in which indices $\{i_1, i_2, \ldots, i_n\}$ can be any reshuffle of $\{1, 2, \ldots, n\}$. But if we attach a graph on the probability to guide the factorization, the shuffle has to following some order, and form of conditional probability follows the Markov parents on DAG:

$$\mathbb{P}(X_1, \ldots, X_n) = \prod_i \mathbb{P}(X_i|pa_i) \tag{11.124}$$

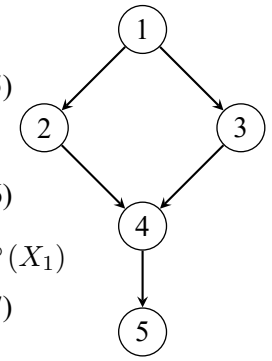
the r.v. sequence is **causal ordering** if X_i only dependent on $X_{j:j<i}$, i.e. $pa_i \subset \{X_1, \ldots, X_{i-1}\}$.

Here's an example of Markov factorization on a DAG graph:

$$\mathbb{P}(X_1, X_2, X_3, X_4, X_5) = \mathbb{P}(X_5|X_4)\,\mathbb{P}(X_1, X_2, X_3, X_4) \tag{11.125}$$

$$= \mathbb{P}(X_5|X_4)\,\mathbb{P}(X_4|X_2, X_3)\,\mathbb{P}(X_1, X_2, X_3) \tag{11.126}$$

$$= \mathbb{P}(X_5|X_4)\,\mathbb{P}(X_4|X_2, X_3)\,\mathbb{P}(X_3|X_1)\,\mathbb{P}(X_2|X_1)\,\mathbb{P}(X_1) \tag{11.127}$$



□ **Basic structures in a DAG**

Starting from triplets in DAG as the key elements in a graph.

- Chain $X \to Y \to Z$, in which Y is the *mediator*. We have

$$\mathbb{P}(X, Z|Y) = \frac{\mathbb{P}(X)\,\mathbb{P}(Y|X)\,\mathbb{P}(Z|Y)}{\mathbb{P}(Y)} = \mathbb{P}(X|Y)\,\mathbb{P}(Z|Y) \tag{11.128}$$

  i.e. we have a conditional independency in chain $X \perp\!\!\!\perp Z|Y$

- Fork $X \leftarrow Y \to Z$. We have

$$\mathbb{P}(X, Z|Y) = \frac{\mathbb{P}(Y)\,\mathbb{P}(X|Y)\,\mathbb{P}(Z|Y)}{\mathbb{P}(Y)} = \mathbb{P}(X|Y)\,\mathbb{P}(Z|Y) \tag{11.129}$$

  i.e. we have a conditional independency in chain $X \perp\!\!\!\perp Z|Y$

- Collider $X \to Y \leftarrow Z$. In the collider, $X \perp\!\!\!\perp Z$ marginally, but given Y are conditionally dependent. If there is no edge between X and Z, it's also called v-structure.

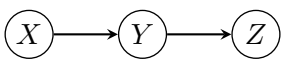


图 11.2: Chain

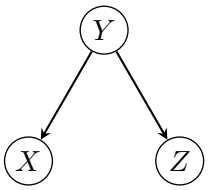


图 11.3: Fork

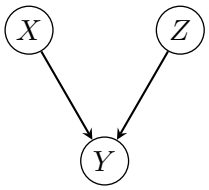


图 11.4: Collider

□ **d-Separation**

d-separation in a graph is directional-separation of two vertices.

- Blocked path: a path p from X to Y is blocked by $\{Z\}$ if for all triplets along the path:
  - **In** Z: the middle point of all chains and forks; and
  - **Not In** Z: the middle point itself of collider, and its descendants
- d-separation: X is d-deparated from Y given Z if all paths $p_{X \rightsquigarrow Y}$ are blocked by Z.

□ **Markov Compatibility**

Markov compatibility is a match between DAG and probability distribution, a description of how $\mathcal{G}$ represents $\mathbb{P}(\cdot)$.

A textbook definition is given as:

> If a probability distribution $\mathbb{P}(\cdot)$ admits a factorization $\mathbb{P}(X) = \prod_i \mathbb{P}(X_i|pa_i)$ relative to DAG $\mathcal{G}$, then $\mathbb{P}$ is *Markov Compatible* relative to $\mathcal{G}$.

Here 'admits' means that d-separation on graph finds its corresponding conditional probability.

$$X \perp\!\!\!\perp_{\mathcal{G}} Y|Z \Rightarrow X \perp\!\!\!\perp_{\mathbb{P}} Y|Z \tag{11.130}$$

Markov compatibility means that we can generate data following $\mathbb{P}$ using $\mathcal{G}$ as 'Blueprint'.

Related notions and comments:

- I-Map: is a set of conditional inpendence statements read out from $\mathcal{G}$. If X and Y are d-separated by Z in $\mathcal{G}$ (denoted $X \perp\!\!\!\perp_{\mathcal{G}} Y|Z$), then we should have $X \perp\!\!\!\perp_{mathbbP} Y|Z$ in **every** $\mathbb{P}$ distribution compatible with $\mathcal{G}$.

$$I(\mathcal{G}) = \{(X \perp\!\!\!\perp_{\mathcal{G}} Y|Z) : (X \perp\!\!\!\perp_{\mathbb{P}} Y|Z)\, \forall \mathbb{P} \text{ compatible with } \mathcal{G}\} \tag{11.131}$$

- From I-Map we can have definition of I-equivalence: i.e. if $\mathcal{G}_1$ and $\mathcal{G}_2$ yield the same I-map $I(\mathcal{G}_1) = I(\mathcal{G}_2)$.
- Note that Markov compatible states that $X \perp\!\!\!\perp_{\mathcal{G}} Y|Z \Rightarrow X \perp\!\!\!\perp_{\mathbb{P}} Y|Z$ but not reversely, which means that $I(\mathcal{G}) \subset I(\mathbb{P})$
- An concrete example: two r.v. are independently generated $\mathbb{P}(X, Y) = \mathbb{P}(X)\,\mathbb{P}(Y)$, i.e. $I(\mathbb{P}) = X \perp\!\!\!\perp_{\mathbb{P}} Y$. All the following graphs are markov compatible:
  - $\mathcal{G}_0 : X \ \ Y$, in which $I(\mathcal{G}_0) = X \perp\!\!\!\perp_{\mathcal{G}} Y$
  - $\mathcal{G}_1 : X \rightarrow Y$, in which $I(\mathcal{G}_1) = \emptyset$
  - $\mathcal{G}_2 : X \leftarrow Y$, in which $I(\mathcal{G}_2) = \emptyset$

  (because they all have $I(\mathcal{G}_i) \subset I(\mathbb{P})$)
- Perfect I-map: if $I(\mathcal{G}) = I(\mathbb{P})$.
- **Observational Equivalence**: A set of graphs are observational equivalent / belong to the same equivalent class if they encode the same conditional independencies.

Note that the key causal structures in DAGs are chain, fork, and collider; in which chain and fork imply the same conditional independence while collider is different. i.e.

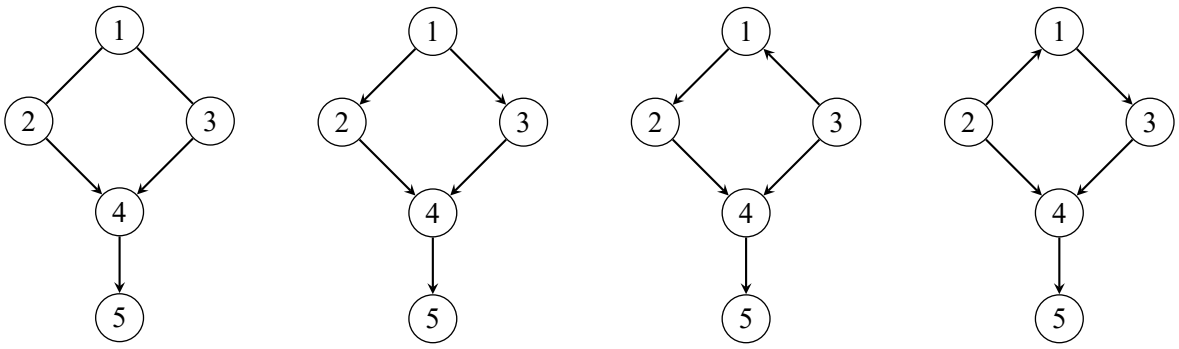
$$X \to Y \to Z, \qquad X \leftarrow Y \leftarrow Z, \qquad X \leftarrow Y \to Z \tag{11.132}$$

are observational equivalent by encoding $X \perp\!\!\!\perp Z|Y$.

The above argument gives the hint for identifying observational equivalent graphs:

- Having the same skeleton
- Having the same set of colliders.

Here's an example of observational equivalent graphs:



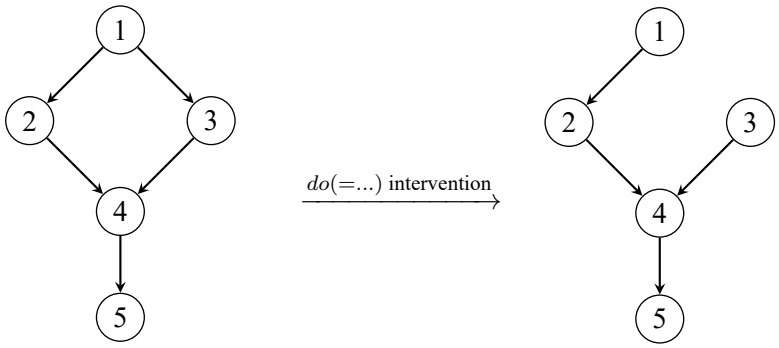
(Partial) Skeleton $\mathcal{G}_1$ $\mathcal{G}_2$ $\mathcal{G}_3$

□ **Causality on Bayesian Network**

Recall that in Rubin's Potential Outcome Framework, causality was induced by counterfactual side of potential outcomes $Y^{\text{mis}} = Y(1-W)$. In Bayesian Network framework, causality is induced by **intervention**, in formulas expressed by $do(\cdot)$ operator, e.g.

$$\mathbb{P}\left(X|do(Z=z)\right) \tag{11.133}$$

where $Z \subset X$ is the set to conduct intervention on. Intervention would remove all 'incoming' edges to Z, as illustrated below an example of $do(3=...)$:



Original $\mathcal{G}$ Interventional $\mathcal{G}$

Since intervention produces different subgraphs, we can obtain causality by comparing the probability distribution. e.g. in the simplest instance $X \to Y$ v.s. $X \leftarrow Y$, intervention $do(X = x)$ yields $x \to Y$ and $X \quad Y$, respectively, which have different observational outcome.

Causal Bayesian Network (CBN) is a DAG $\mathcal{G}$ compatible with $\mathcal{P}$, if

- Notation: here $\mathcal{P} = \{\mathbb{P}(X|do(Z = z)) : \forall Z \subset X\}$ is the set of all interventional probability distribution.
- $\forall \mathcal{P} \ni \mathbb{P}(X|do(Z = z))$ is compatible with $\mathcal{G}$
- $\mathbb{P}(x_i|do(Z = z)) = 1$ if $X_i \in Z$ and $x_i = z_{\text{corresponding value}}$ ($X_i = x_i$ is consistent with $Z = z$)
- $\mathbb{P}(X_i|pa_i)$ is invariant to interventions not involving X_i itself.

Comments:

- Note that intervention $do(Z = z)$ cancels some edges, so it would only add newindependencies, which holds $I(\mathcal{G}) \subset I(\mathbb{P})$ (still compatible).
- With some intervention $do(Z = z)$, the *truncated* factorization of $\mathbb{P}(\cdot)$ is

$$\mathbb{P}(X|do(Z = z)) = \prod_{i:X_i \notin Z} \mathbb{P}(X_i|pa_i) \tag{11.134}$$

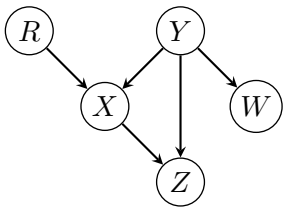
11.4.2 Network Structure Learning

□ IC/PC Algorithm

IC/PC Algorithm (Inductive Causation Algorithm with Peter & Clark Algorithm Refinement) is a constraint-based method. DAG is constructed through identifying conditional independencies.

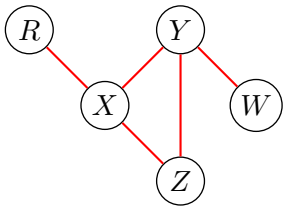
Here illustrated with the following example with conditional independencies. Ground truth is shown on the right

$$X \not\perp\!\!\!\perp R|\mathcal{S}, \forall \mathcal{S} \subset \{Y, Z, W\} \tag{11.135}$$
$$X \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{Y, R, W\} \tag{11.136}$$
$$X \not\perp\!\!\!\perp Y|\mathcal{S}, \forall \mathcal{S} \subset \{Z, R, W\} \tag{11.137}$$
$$Y \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{X, R, W\} \tag{11.138}$$
$$Y \not\perp\!\!\!\perp W|\mathcal{S}, \forall \mathcal{S} \subset \{X, Z, R\} \tag{11.139}$$
$$X \perp\!\!\!\perp W|Y \tag{11.140}$$
$$Y \perp\!\!\!\perp R \tag{11.141}$$
$$Z \perp\!\!\!\perp W|Y \tag{11.142}$$
$$Z \perp\!\!\!\perp R|\{X, Y\} \tag{11.143}$$
$$W \perp\!\!\!\perp R \tag{11.144}$$



1. Learning Skeleton: For all paris $(a, b) \in \mathcal{V} \times \mathcal{V}$:
   - Connect a, b iff no $\mathcal{S}_{ab}$ such that $a \perp\!\!\!\perp b|S_{ab}$ can be found. i.e. $a,\ b$ have an edge if $a \not\perp\!\!\!\perp b|$any set of other nodes.

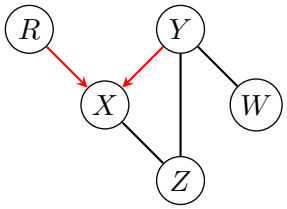
$$X \not\perp\!\!\!\perp R|\mathcal{S}, \forall \mathcal{S} \subset \{Y, Z, W\} \quad (11.145)$$
$$X \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{Y, R, W\} \quad (11.146)$$
$$X \not\perp\!\!\!\perp Y|\mathcal{S}, \forall \mathcal{S} \subset \{Z, R, W\} \quad (11.147)$$
$$Y \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{X, R, W\} \quad (11.148)$$
$$Y \not\perp\!\!\!\perp W|\mathcal{S}, \forall \mathcal{S} \subset \{X, Z, R\} \quad (11.149)$$
$$X \perp\!\!\!\perp W|Y \quad (11.150)$$
$$Y \perp\!\!\!\perp R \quad (11.151)$$
$$Z \perp\!\!\!\perp W|Y \quad (11.152)$$
$$Z \perp\!\!\!\perp R|\{X, Y\} \quad (11.153)$$
$$W \perp\!\!\!\perp R \quad (11.154)$$



2. v-structure Orientation: For all (a, b) with common neighbour c but not adjacent, i.e. have a-c-b

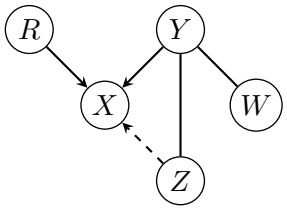
   - If $c \notin \mathcal{S}_{ab}$, then there is a v-structure $a \to c \leftarrow b$

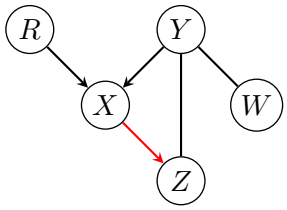
$$X \not\perp\!\!\!\perp R|\mathcal{S}, \forall \mathcal{S} \subset \{Y, Z, W\} \quad (11.155)$$
$$X \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{Y, R, W\} \quad (11.156)$$
$$X \not\perp\!\!\!\perp Y|\mathcal{S}, \forall \mathcal{S} \subset \{Z, R, W\} \quad (11.157)$$
$$Y \not\perp\!\!\!\perp Z|\mathcal{S}, \forall \mathcal{S} \subset \{X, R, W\} \quad (11.158)$$
$$Y \not\perp\!\!\!\perp W|\mathcal{S}, \forall \mathcal{S} \subset \{X, Z, R\} \quad (11.159)$$
$$X \perp\!\!\!\perp W|Y \quad (11.160)$$
$$Y \perp\!\!\!\perp R \quad (11.161)$$
$$Z \perp\!\!\!\perp W|Y \quad (11.162)$$
$$Z \perp\!\!\!\perp R|\{X, Y\} \quad (11.163)$$
$$W \perp\!\!\!\perp R \quad (11.164)$$



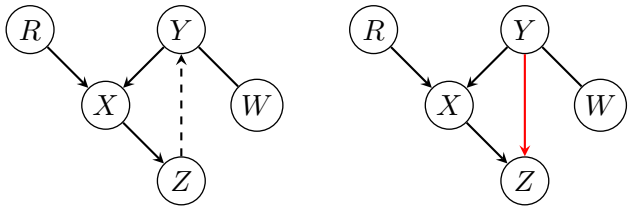
3. Meek's rule Orientation: orient as many edges as possible subject to:

   - Alternative direction yields new v-structure





   - Alternative direction yields cycle (acyclic rule is of more priority)



The above procedure can produce identify the DAG up to its observational equivalent class.

□ **Search and Score Methods**

We could also simply search in the space of all possible networks and select the one scoring the highest. Some frequently used metrics include AIC and BIC

$$\text{AIC} = -2\log\mathbb{P}\left(\mathcal{G}|\hat{\theta}\right) + 2\text{dof}_{\mathcal{G}} \tag{11.165}$$

$$\text{BIC} = -2\log\mathbb{P}\left(\mathcal{G}|\hat{\theta}\right) + \log n \cdot \text{dof}_{\mathcal{G}} \tag{11.166}$$

11.4.3 Network Parameter Learning

Basically, parameter learning (given BN strucure) is simply estimating edge weights, denoted Θ. Two basic methods are

- Bayesian approach

$$\arg\max_{\Theta} \mathbb{P}(\Theta|\mathcal{D}) \tag{11.167}$$

- Frequentist approach, e.g. with MLE loss+penalty form

$$\arg\min_{\Theta} -\log\mathbb{P}\left(\mathcal{D}|\Theta\right) + \lambda P(\Theta) \tag{11.168}$$

A trivial solution for categorical variables is

$$\hat{\theta}_{ij\vec{k}} = \hat{\mathbb{P}}(X_i = j|pa_i = \vec{k}) = \frac{N_{ijk}}{N_{i\cdot k}},\ \forall j = 1,\ldots,J_i,\ \forall i \in \mathcal{V} \tag{11.169}$$

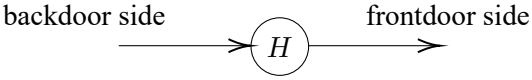
11.4.4 Average Causal Effect Estimation

Average Causal Effects on BN are defined in terms of $do(\cdot)$ operator,

$$\text{ACE}(Y|H) := \mathbb{E}\left[Y|do(H = h_1)\right] - \mathbb{E}\left[Y|do(H = h_2)\right] \tag{11.170}$$

Calculation of ACE given known BN relies on do-calculus. A $do(\cdot)$ operator would cancal all edges pointing to the vertex, i.e. produce a modified graph $\mathcal{G}_{do(\cdot)}$, what we need to estimate is the probability in the modified graph.

$$\mathbb{E}_{\mathcal{G}}\left[Y|do(H)\right] \twoheadleftarrow \mathbb{P}_{\mathcal{G}}\left(Y|do(H)\right) \twoheadleftarrow \mathbb{P}_{\mathcal{G}_{do(H)}}\left(Y\right) \xleftarrow{do\text{-calculus}} \mathbb{P}_{\mathcal{G}}\left(\mathbf{X}\right) \twoheadleftarrow \mathcal{D}ata \tag{11.171}$$



□ *do*-**Calculus**

- Module invariant

$$\mathbb{P}\left(X_i = x_i \middle| do(PA_i = pa_i)\right) = \mathbb{P}\left(X_i = x_i \middle| PA_i = pa_i\right) \tag{11.172}$$

▷ **The Adjustment Formula**

$$\mathbb{P}\left(Y = y \middle| do(X = x)\right) = \sum_{z \in \{pa_y\}} \mathbb{P}\left(Y = y \middle| X = x, PA_y = z\right) \mathbb{P}\left(PA_y = z\right) \tag{11.173}$$

$$= \sum_{z \in \{pa_y\}} \frac{\mathbb{P}\left(X = x, Y = y, PA_y = z\right)}{\mathbb{P}\left(X = x \middle| PA_y = z\right)} \tag{11.174}$$

in which we use the Markovian factorization on $\mathcal{G}$

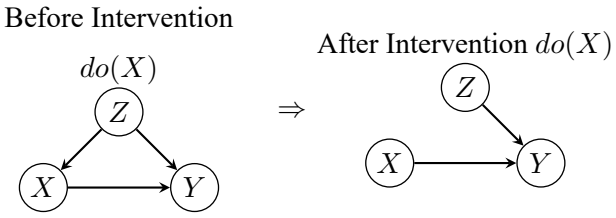
$$\mathbb{P}\left(X, Y, PA_y\right) = \mathbb{P}\left(Y|X, PA_y\right) \mathbb{P}\left(X|PA_y\right) \mathbb{P}\left(PA_y\right) \tag{11.175}$$

with X considered as assignment mechanism, Z considered as covariates, the formula shares the same idea as equation 11.117 ~ page 298.

Through adjustment formula, we could obtain ACE from observed data (without intervention $\rightsquigarrow$ with intervention).

Example: assessing $Y|do(X = x)$, in which $PA_y = \{X, Z\}$ with X being fixed by $do(X = x)$.

$$\mathbb{P}\left(Y = y \middle| do(X = x)\right) = \sum_{z \in \{z\}} \mathbb{P}\left(Y \middle| X = x, Z = z\right) \mathbb{P}\left(Z = z\right) \tag{11.176}$$



▷ **Backdoor criterion**

Given (X, Y) in BN, a 'backdoor set' $\boldsymbol{Z}$ is one such that $\boldsymbol{Z}$:

- Blocks **all** paths with arrow onto X (i.e. backdoor side of X is blocked by $\boldsymbol{Z}$)
- $\boldsymbol{Z}$ contains **no** descendants of X

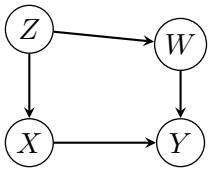
then we could use the backdoor variable set $\boldsymbol{Z}$ to have the **backdoor adjustment** of $Y|do(X)$ as

$$\mathbb{P}\left(Y = y|do(X = x)\right) = \sum_{\boldsymbol{z}} \mathbb{P}\left(Y = y|X = x, \boldsymbol{Z} = \boldsymbol{z}\right) \mathbb{P}\left(\boldsymbol{Z} = \boldsymbol{z}\right) \tag{11.177}$$

The selection of backdoor set $\boldsymbol{Z}$ is not unique. e.g. sometimes due to observablility problem we could only obtain Partial DAG / have multiple methods to block the path, then we could pick proper nodes to form the backdoor set.

Example: assessing $Y|do(X=x)$, where Z is an observable while W is a hidden unobservable.

$$\mathbb{P}\left(Y=y\middle|do(X=x)\right)=\sum_{z\in\{z\}}\mathbb{P}\left(Y\middle|X=x,Z=z\right)\mathbb{P}\left(Z=z\right) \tag{11.178}$$



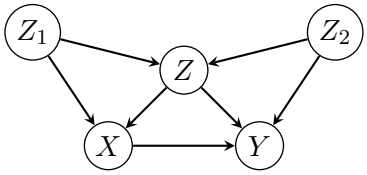
Example: assessing $Y|do(X=x)$.

$$\mathbb{P}\left(Y=y\middle|do(X=x)\right)=\sum_{(z,z_1)\in\{(z,z_1)\}}\mathbb{P}\left(Y\middle|X=x,Z=z,Z_1=z_1\right)\mathbb{P}\left(Z=z,Z_1=z_1\right) \tag{11.179}$$

$$=\sum_{(z,z_2)\in\{(z,z_2)\}}\mathbb{P}\left(Y\middle|X=x,Z=z,Z_2=z_2\right)\mathbb{P}\left(Z=z,Z_2=z_2\right) \tag{11.180}$$

$$=\sum_{(z,z_1,z_2)\in\{(z,z_1,z_2)\}}\mathbb{P}\left(Y\middle|X=x,Z=z,Z_1=z_1,Z_2=z_2\right) \tag{11.181}$$

$$\cdot\,\mathbb{P}\left(Z=z,Z_1=z_1,Z_2=z_2\right) \tag{11.182}$$



i.e. we could adjust for either (Z,Z_1) or (Z,Z_2) or (Z,Z_1,Z_2) as the backdoor set.

▷ **Frontdoor criterion**

Given (X,Y) in BN, a 'frontdoor set' $\boldsymbol{Z}$ is one such that $\boldsymbol{Z}$:

- Intercepts **all** paths from X to Y (i.e. frontdoor side of X is intercepted $\boldsymbol{Z}$)
- **No** unblocked backdoor path from X to Z[6]
- **All** backdoor paths from Z to Y blocked by X

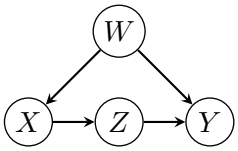
then we could use the frontdoor variable set $\boldsymbol{Z}$ to have the frontkdoor adjustment of $Y|do(X)$ as

$$\mathbb{P}\left(Y=y\middle|do(X=x)\right)=\sum_{z}\mathbb{P}\left(Z=z|X=x\right)\sum_{x'}\mathbb{P}\left(Y=y|X'=x',Z=z\right)\mathbb{P}\left(X'=X'\right) \tag{11.183}$$

[6]backdoor path from X to Z means containing a backdoor arrow of X, e.g. in the example, $X\leftarrow W\rightarrow Y\leftarrow Z$, is blocked.

Example: assessing $Y|do(X=x)$.

$$\mathbb{P}\left(Y=y\middle|do(X=x)\right)=\sum_{z}\mathbb{P}\left(Z=z|X=x\right)\sum_{x'}\mathbb{P}\left(Y=y|X'=x',Z=z\right)\mathbb{P}\left(X'=X'\right) \tag{11.184}$$



The derivation is using backdoor adjustment twice

$$\begin{cases}\mathbb{P}\left(Y=y|do(X=x)\right)=\sum_{z}\mathbb{P}\left(Y=y|do(Z=z)\right)\mathbb{P}\left(Z=z|do(X=x)\right)\\ \mathbb{P}\left(Y=y|do(Z=z)\right)=\sum_{x}\mathbb{P}\left(Y=y|Z=z,X=x\right)\mathbb{P}\left(X=x\right)\\ \mathbb{P}\left(Z=z|do(X=x)\right)=\mathbb{P}\left(Z=z|X=x\right)\end{cases} \tag{11.185}$$

▷ General Rules of do-calculus*

11.4.5 Instrumental Variable Method*

Chapter. XII　应用随机过程部分

Instructor: Pengkun Yang

Section 12.1　Properties of Stochastic Process

12.1.1　Basic Concepts

Some basic concepts about stochastic process / random process are introduced in section 10.2 ~ page 268. Here's a brief recap.

A stochastic process is a mapping

$$\{X_t : t \in \mathcal{T}\} : \Omega \mapsto \mathcal{T} \times \mathbb{R} \tag{12.1}$$

- For given $t \in \mathcal{T}$, $X_t(\cdot)$ is a r.v. defined on Ω.
- For given $\omega \in \Omega$, $X_\cdot(\omega)$ is a function on $\mathcal{T}$, which is called sample path.

According to the continuity of index Fourier Transformset $\mathcal{T}$ and sample path values, Stochastic process can be categorized in discrete / continuous Time + discrete / continuous State processes.

Some functions of stochastic processes include

- Mean function:

$$\mu_X(t) = \mathbb{E}[X_t] \tag{12.2}$$

- AutoCovariance function (ACVF):

$$\gamma_{s,t} := cov(X_s, X_t) \tag{12.3}$$

- AutoCorrelation function (ACF):

$$\rho_{s,t} := corr(X_s, X_t) = \frac{\gamma_{s,t}}{\sqrt{\gamma_{s,s}\gamma_{t,t}}} \tag{12.4}$$

- n^{th} order CDF:

$$F_{X,n}(x_1, t_1; x_2, t_2; \ldots; x_n, t_n) = \mathbb{P}(X_{t_1} \le x_1,\, X_{t_2} \le x_2, \ldots,\, X_{t_n} \le x_n) \tag{12.5}$$

12.1.2 Properties of Discrete Time Markov Chain

A basic case for Markov Chain is Discrete Time Markov Chain (DTMC)

□ **Notations and Properties of DTMC**

- State: denote the state space / phase space of DTMC as

$$X_n \in \mathcal{S} \tag{12.6}$$

- Conditional Independency:

$$\mathbb{P}\left(X_{n+1}\middle|X_0, X_1, \ldots, X_n\right) = \mathbb{P}\left(X_{n+1}\middle|X_n\right) \tag{12.7}$$

- State transition and transition probability matrix:

$$P^{(k)} = \left\{P_{ij}^{(k)}\right\} = \left\{\mathbb{P}\left(X_{k+1} = j\middle|X_k = i\right)\right\}, \quad i, j \in \mathcal{S} \tag{12.8}$$

transition pr matrix P is called a (row) stochastic matrix, with

$$0 \le P_{ij}^{(k)} \le 1, \quad \sum_j P_{ij}^{(k)} = 1 \tag{12.9}$$

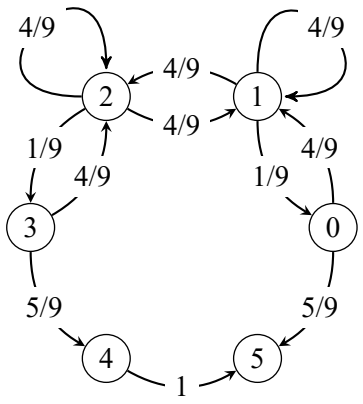
- Time homogeneity: transition probability is independent of step / time

$$P^{(k)} = P, \quad \forall k \tag{12.10}$$

we usually focus on time-homogeneous DTMC.

- State diagram: a useful way to visualize DTMC, in which vertices / nodes for states and edges / arrows for transition. Here's an example of 'Mickey Mouse' diagram with six states:

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} & 4/9 & & & & 5/9 \\ 1/9 & 4/9 & 4/9 & & & \\ & 4/9 & 4/9 & 1/9 & & \\ & & 4/9 & & 5/9 & \\ & & & & & 1 \\ & & & & & 1 \end{pmatrix} \end{array} \quad \leftrightharpoons$$



□ **Stationary Distribution**

State transition between steps are like jumping in state diagram. Denote $\pi(k)$ the probability distribution at step k, then a transition is

$$\pi(k+1) = \pi(k)P^{(k)} = \pi(k)P \tag{12.11}$$

A stationary distribution / equilibrium of DTMC is the eigen distribution of transition matrix

$$\pi^* = \pi^* P = \pi^* P^i, \forall i \tag{12.12}$$

A sufficient condition for stationary state is the detailed balance condition

$$\pi_i^* = \sum_j \pi_j^* P_{ji} \tag{12.13}$$

$$\Leftrightarrow \pi^* \sum_{j \neq i} P_{ij} = \pi_i^* (1 - P_{ii}) = \sum_{j \neq i} \pi^* j P_{ji} \tag{12.14}$$

$$\Leftarrow \pi_i P_{ij} = \pi_j P_{ji} \tag{12.15}$$

Some concepts related to stationary distribution

- Reachable: we can arrive at j starting from i, denoted $i \rightsquigarrow j$

$$\exists n < \infty s.t. \mathbb{P}\left(X_n = j | X_0 = i\right) > 0 \tag{12.16}$$

  Sometimes I use the notation $i \xrightarrow{k} j$ for 'reaching j in k steps from i'

- Irreducible: every state is reachable from any other states

$$i \rightsquigarrow j, \forall i, j \in \mathcal{S} \tag{12.17}$$

- Periodic: the period d_i for state i is the greatest common divisor (GCD) of step-to-come-back.

$$d_i := \gcd\left\{n : \mathbb{P}\left(X_n = i | X_0 = i\right) > 0\right\} \tag{12.18}$$

  Irreducible DTMC has the same period for all states.

  For any two states i, j, with periods d_i, d_j. Then d_i contains the following process:

$$\{i \xrightarrow{k_1} j \xrightarrow{m \times d_j} j \xrightarrow{k_n} i\}, \quad m \in \mathbb{N} \tag{12.19}$$

  there are infinite elements. then

$$d_i = \text{cd}\left\{k_1 + k_2 + m d_j; m = 0, 1, 2, \ldots\right\} \Rightarrow d_j = \text{multiple of } d_i \tag{12.20}$$

  With the argument applied to all state pairs $(i, j) \in \mathcal{S} \times \mathcal{S}$, obviously $d_i = d, \forall i \in \mathcal{S}$

- Aperiodic: is the case that $d_i = 1$, i.e. possible to come back anytime. For irreducible DTMC, if one state is aperiodic, then all are.

  Naturally if a node is self looped $P_{ii} > 0$ (e.g. node 1 or 2 in 'Mickey Mouse' loops back with pr $4/9$), then all the states are aperiodic.

- Sojourn Time T_i: is the time to stay at the state

$$T_i \sim \text{Geo}(1 - P_{ii}) \tag{12.21}$$

- Classification of States.

  Denote Hitting Time (without itself include) τ_i^+ and its mean

$$\tau_i^+ := \min\{k \geq 1 : X_k = i\} \tag{12.22}$$

$$\mu_i := \mathbb{E}\left[\tau_i^+ | X_0 = i\right] \tag{12.23}$$

  – Recurrent State

$$\mathbb{P}\left(\tau_i^+ < \infty | X_0 = i\right) = 1 \tag{12.24}$$

  in which

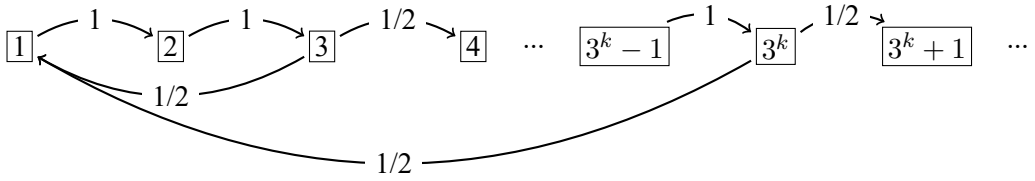
  ∗ Positive Recurrent

$$\mu_i < \infty \tag{12.25}$$

  ∗ Null Recurrent

$$\mu_i = \infty \tag{12.26}$$

  An example:



  where

$$\mu_1 = \mathbb{E}\left[\tau_1^+ | X_0 = 1\right] = \sum_{i=1}^{\infty} \left(\frac{3}{2}\right)^i \to \infty \tag{12.27}$$

  – Transient State

$$\mathbb{P}\left(\tau_i^+ < \infty | X_0 = i\right) < 1 \tag{12.28}$$

□ **DTMC: Irreducible & Aperiodic & Positive Recurrent ⇒ Unique Stationary Distribution π^* Exists**

Given irreducible & aperiodic DTMC, we have

- All states have the same state classification: null recurrent / positive recurrent / transient.
- if all states are positive recurrent $\mu_i < \infty$, then stationary distribution exists.

$$\lim_{n\to\infty} \frac{1}{n} \sum_{l=1}^{n} (P^l)_{ij} = \frac{1}{\mu_i} \Rightarrow \pi_i(\infty) = (\pi(0)P^\infty)_i = \frac{1}{\mu_i}, \forall \pi(0) \tag{12.29}$$

- Further if states are positive recurrent $\mu_i < \infty$, then stationary distribution.

$$\pi^* = \frac{1}{\mu_i} \tag{12.30}$$

The proof is a little bit complicated, but an intuition is direct. For a realization of Markov Process $\{X_t\}_{t=1}^{\infty}$, in which $\{X_{i_1}, X_{i_2}, \ldots\}$ is the set that $X_t = i$ for any given i, and $\{u_{i_1}, u_{i_2}, \ldots\} = \{i_1 - i_2, i_3 - i_2, \ldots\}$ is the time-btw-event, i.e. $u_{i_j} \sim_{i.i.d.} \sim \tau_i^+ | X_0 = i, \forall j$. Then

$$\lim_{n\to\infty} \sum_{t=1}^{n} \mathbb{I}_{X_t=i} = \lim_{k\to\infty} \frac{k}{u_{i_1} + u_{i_2} + \ldots + u_{i_k}} = \frac{1}{\mathbb{E}\left[\tau_i^+ | X_0 = i\right]} = \frac{1}{\mu_i} \tag{12.31}$$

Comment: Ergodicity = irreducible & aperiodic condition. It *creates link between phase structure and time structure*, which makes $\bar{u}$ (time-average) converge in an appropriate sense to μ_i (phase-average).

Some algorithm about Markov Chain see section 5.6.2 ~ page 188.

□ **Concrete examples of DTMC**

- Random Walk
- Gambler's Model
- Branching Process

12.1.3 Properties of Continuous Time Markov Chain

Another case of Markov Chain is Continuous Time Markov Chain (CTMC)

□ **Notations and Properties of CTMC**

- Concepts of state and conditional independency are similar to DTMC

$$\mathbb{P}\left(X_{t_{n+1}} \middle| X_{t_0}, X_{t_1}, \ldots, X_{t_n}\right) = \mathbb{P}\left(X_{t_{n+1}} \middle| X_{t_n}\right) \tag{12.32}$$

- Transition probability matrix

$$H(s,t) := \{H_{ij}(s,t)\} = \{\mathbb{P}\left(X_t = j | X_s = i\right)\}, \quad s < t \tag{12.33}$$

with a trivial case that $H(t,t) = I$. State transition could be expressed by matrix $H(s,t)$ as

$$p(t) = p(s)H(s,t) \tag{12.34}$$

- Chapman-Kolmogorov Equation

$$H(r,t) = H(r,s)H(s,t), \quad r < s < t \tag{12.35}$$

- Time homogeneity: transition probability is independent of time interval:

$$H(s,t) = H(0, t-s) \tag{12.36}$$

- Generator of time homogeneous CTMC: The **Transition Rate Matrix** is

$$Q := \lim_{\delta\to 0} \frac{H(\delta) - H(0)}{\delta}, \quad H(\delta) = I + \delta Q + o(\delta) \tag{12.37}$$

with Chapman-Kolmogorov Equation we could see that Q is the generator of the transition matrix (group)

$$H(t) \underset{t=n\delta}{=} \lim_{n\to\infty} H(\delta)^n = \lim_{n\to\infty} \left(I + \frac{t}{n}Q\right)^n = e^{Qt} \tag{12.38}$$

And note that $H(t)$ has 1 row-sum, $\sum_j \left(e^{Qt}\right)_{ij} = 1$:

$$0 = \frac{\mathrm{d}\sum_j \left(e^{Qt}\right)_{ij}}{\mathrm{d}t} = \sum_{j,k} Q_{ik}\left(e^{Qt}\right)_{kj} = \sum_k Q_{ik} = 0 \tag{12.39}$$

$$\Rightarrow Q_{ii} = -\sum_{k\neq i} Q_{ik}, \quad \forall i \tag{12.40}$$

i.e. generator Q has 0 row-sum.

Comment: with Gershgorin Circle Theorem [1], Q as a diagonal dominant matrix, is negative definite, which guarentee the convergence of $H(t) = e^{Qt} < \infty$

- Kolmogorov Forward Equation:[2]

$$\dot{p}(t) = \frac{\mathrm{d}p(0)e^{Qt}}{\mathrm{d}t} = p(0)e^{Qt}Q = p(t)Q \tag{12.42}$$

Kolmogorov forward could also be deduced for some other specifically defined event / probability.

- Stationary Distribution: with $\dot{\pi}^* = 0$ in Kolmogorov forward, stationary distribution of CTMC:

$$\pi^* = \pi^* H(t),\ \forall t \Leftrightarrow \pi^* Q = 0 \tag{12.43}$$

thus yield the detailed balance in CTMC version:

$$\pi^* Q = 0 \Leftarrow \pi_i^* q_{ij} = 0,\ \forall i, j \tag{12.44}$$

- Dynamics of CTMC: Each step (say, $0 \rightsquigarrow t \rightsquigarrow t+\delta$) in state transitions in CTMC could be decomposed in two sub-steps:

$$\begin{cases} \text{Sojourn}: & T_i \sim \mathbb{P}\left(t : X_\tau = i\, \forall 0 \le \tau \le t \middle| X_0 = i\right) \\ \text{Jump}: & p_{ij}^{\mathrm{J}} \sim \mathbb{P}\left(X_{t+\delta} = j \middle| X_t = i, X_{t+\delta} \neq i\right) \end{cases} \tag{12.45}$$

which has the following dynamics

$$\begin{cases} T_i \sim \varepsilon(-q_{ii}) \\ p_{ij}^{\mathrm{J}} = (\delta_{ij} - 1)\dfrac{q_{ij}}{q_{ii}} \end{cases} \tag{12.46}$$

Where sojourn time T_i is a continuous correspondance of 12.21. In both versions it is memoryless.

[1] Detail see https://v1ncent19.github.io/texts/DiagonalDominant/.

[2] Note that Q and e^{Qt} are commutable

$$Qe^{Qt} = Q\sum_{i=0}^{\infty} \frac{Q^i t^i}{i!} = e^{Qt}Q \tag{12.41}$$

□ **CTMC: Irreducible & Non-explosive & Positive Recurrent ⇒ Unique Stationary Distribution π^***

Given irreducible & non-explosive CTMC, we have

- All states have the same state classification: null recurrent / positive recurrent / transient
- Stationary distribution exists ⇔ all states are positive recurrent

$$\lim_{t\to\infty} p_i(t) = \frac{1}{-q_{ii}\mu_i} = \pi_i^* \tag{12.47}$$

□ **Concrete examples of CTMC**

- Brownian Process: CTMC with continuous states;
- Poisson Process: CTMC with discrete states;

12.1.4 Independent Increment Process and Martingale

Motivation: Sometimes a process is a 'summation of all past events'.

- Independent Increment: Def. $\{X_t\}$ a **independent increment process** if $\forall t_0 < t_1 < \ldots < t_n, \forall n$

$$X_{t_n} - X_{t_{n-1}} \perp\!\!\!\perp X_{t_{n-1}} - X_{t_{n-2}} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X_{t_1} - X_{t_0} \tag{12.48}$$

- Martingale: Def. $\{X_t\}$ a **Martingale** if $\forall t_0 < t_1 < \ldots < t_n, \forall n$

$$\mathbb{E}\left[X_{t_n} | X_{t_{n-1}}, \ldots, X_{t_0}\right] = X_{t_{n-1}} \tag{12.49}$$

with a technical condition of bounded expectation $\mathbb{E}\left[|X_t|\right] < \infty$.

- Martingale: Def. $\{X_t\}$ being a Martingale w.r.t. $\{Y_t\}$ if

$$\mathbb{E}\left[X_{t_n} | Y_{t_{n-1}}, \ldots, Y_{t_0}\right] = X_{t_{n-1}} \tag{12.50}$$

with bounded expectation $\mathbb{E}\left[|X_t|\right] < \infty$.

□ **Concrete examples of independent increment processes**

- Brownian Process: homogeneous events, probabilistic increment.
- Poisson Process: probabilistic events, homogeneous increment.

12.1.5 Ergodicity*

Section 12.2 Useful Instances of Stochastic Processes

12.2.1 Random Walk

Random walk is a renewal process X_n with each step W_i takes value ± 1

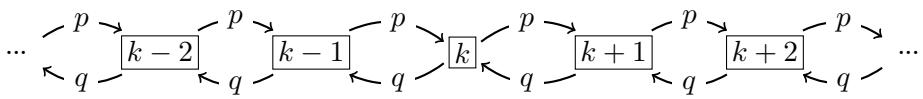
$$X_n := X_0 + \sum_{i=1}^{n} W_i \quad W_i = \begin{cases} +1 & \text{w.p. } p \\ -1 & \text{w.p. } q := 1-p \end{cases} \tag{12.51}$$

where $X_0 = k$ is the initial position.

□ **Simple Random Walk**

Simple random walk is the case with no ends, i.e. $X_n \in \mathbb{Z}$

- State Diagram for Simple Random Walk



- Parameters

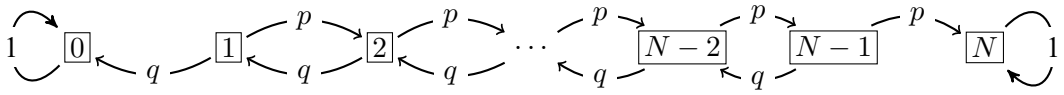
$$\begin{cases} \text{Mean Function : } \mu_n = k + n(2p-1) \\ \text{Covariance : } \gamma_{m,n} = 4pq\min\{m,n\} \\ \text{CLT : } \dfrac{X_n - k - n(2p-1)}{\sqrt{4npq}} \xrightarrow{d} N(0,1) \end{cases} \tag{12.52}$$

12.2.2 Gambler's Model

Gambler's model is the case with one/two ends, usually one of the ends is denoted 0, as Gambler's ruin, and the other denoted N as Gambler's success.

Reaching 0 or N stops the chain, so are called 'absorbing state'.

- State Diagram of Gambler's model with two ends



- Gambler's Ruin / Success: Denote Hitting Time (allowing itself included) $\tau_i = \min\{n \geq 0 : X_n = i\}$, and probability of ruin r_i and probability of success s_i respectively

$$r_i := \mathbb{P}(X_{\tau_0} = 0 | X_0 = i) \tag{12.53}$$

$$s_i := \mathbb{P}(X_{\tau_N} = N | X_0 = i) \tag{12.54}$$

with itertion relation

$$s_i = p \cdot s_{i+1} + q \cdot s_{i-1}, \quad s_0 = 0,\ s_N = 1 \tag{12.55}$$

$$r_i = q \cdot r_{i+1} + p \cdot r_{i-1}, \quad r_0 = 1,\ r_N = 0 \tag{12.56}$$

we could get[3]

$$s_i = \frac{1-(q/p)^i}{1-(q/p)^N} \tag{12.59}$$

$$r_i = \frac{(q/p)^i-(q/p)^N}{1-(q/p)^N} = 1 - s_i \tag{12.60}$$

- Mean Hitting Time $T_{i\rightsquigarrow\{0,N\}}$ for $i \rightsquigarrow \{0,N\}$: $T_{i\rightsquigarrow\{0,N\}} = \mathbb{E}\left[\min\{\tau_0,\tau_N\}|X_0=i\right]$:

$$T_{i\rightsquigarrow\{0,N\}} = p\left(1+T_{i+1\rightsquigarrow\{0,N\}}\right) + q\left(1+T_{i-1\rightsquigarrow\{0,N\}}\right), \quad T_{N\rightsquigarrow\{0,N\}} = T_{0\rightsquigarrow\{0,N\}} = 0 \tag{12.61}$$

solution

$$T_{i\rightsquigarrow\{0,N\}} = \frac{\left(1-(q/p)^i\right)(N-i)}{\left(1-(q/p)^N\right)(p-q)} \tag{12.62}$$

- One-end case (greedy gambler) is just having $N \to \infty$

$$r_i = \begin{cases} 1, & p \le \frac{1}{2} \\ \left(\frac{q}{p}\right)^i, & p > \frac{1}{2} \end{cases} \tag{12.63}$$

Note: i.e. there is a phase transition at $p = \frac{1}{2}$.

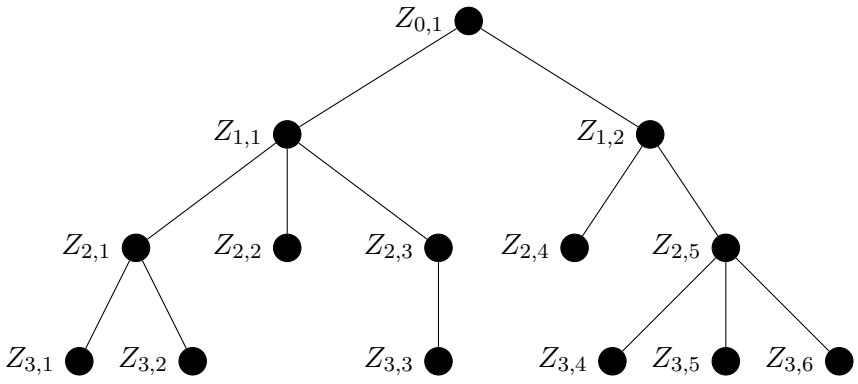
12.2.3 Branching Process

Branching process / Galton-Watson Tree focuses on the case of population growth / epidemic infection / nuclear fission chain reaction, etc. Each steps the state X_n denotes the number of individuals, update of state is given as

$$X_{t+1} = \sum_{j=1}^{X_t} Z_{t,j}, \quad Z_{t,j} \text{ i.i.d. } \sim Z_t, \quad X_0 = 1 \tag{12.64}$$

and we usually assume the simple case of Z_t i.i.d. $\sim Z$.

- State Diagram



[3]For the case $q = p = 1/2$, take the natural limit to get corresponding solution

$$s_i = \frac{k}{N} \tag{12.57}$$

$$r_i = 1 - \frac{k}{N} \tag{12.58}$$

- z-transform for distribution of X_t:

$$\Pi_t(s) = \mathbb{E}\left[s_t^X\right] = \sum_{j=0}^{\infty} s^j \mathbb{P}\left(X_t = j\right) \quad L(s) = \mathbb{E}\left[s^Z\right] = \sum_{j=0}^{\infty} s^j \mathbb{P}\left(Z = j\right) \tag{12.65}$$

and

$$\Pi_t(s) = \sum_{j=0}^{\infty} \mathbb{E}\left[s^{X_t} | X_{t-1} = h\right] \mathbb{P}\left(X_{t-1} = j\right) \tag{12.66}$$

$$= \sum_{j=0}^{\infty} \left(L(s)\right)^j \mathbb{P}\left(X_{t-1} = j\right) \tag{12.67}$$

$$= \Pi_{t-1}\left(L(s)\right) \tag{12.68}$$

$$\left(\Pi_1(s) = L(s)\right) = L^{(t)}(s) \tag{12.69}$$

- Mean and Variance:

$$\text{Mean} : \mu(t) = \Pi_t'(1) = \mu(0)^t \tag{12.70}$$

$$\text{Variance} : var(t) = \Pi_t''(1) + \Pi_t'(1) - [\Pi_t'(1)]^2 \tag{12.71}$$

- Extinction Probability

$$\theta_t = \mathbb{P}\left(X_t = 0\right) = \sum_{j=0}^{\infty} \theta_{t-1}^j \mathbb{P}\left(Z = j\right) \tag{12.72}$$

$$= L(\theta_{t-1}) \tag{12.73}$$

The eventual extinction is $\theta^* = L(\theta^*)$, the fixed point of $L(\,\cdot\,)$. There is a phase transformation at $\mu = 1$

$$\mathbb{P}\left(\theta^* = 1\right) = \begin{cases} 1, & \mu \le 1 \\ \text{the first root of } L(\theta) = \theta, & \mu > 1 \end{cases} \tag{12.74}$$

Convergence order at phase transition point:

$$\mathbb{P}\left(X_T > n\right) \sim \begin{cases} c_1 \mu^n, & \mu < 1 \\ \dfrac{c_2}{n}, & \mu = 1 \end{cases} \tag{12.75}$$

12.2.4 Brownian Motion

Motivation: Brownian motion / Weiner Process W_t[4] is similar to a random walk model with $p = q = 1/2$, but with initial state $X_0 = 0$, and 'steps' defined as 'a short enough time segmentation'.

$$W_{t=\frac{k}{N}} := \frac{1}{\sqrt{N}} \sum_{i=1}^{k} \varpi_i, \quad \varpi_i \sim_{\text{i.i.d.}} \text{Unif}\{+1, -1\} \tag{12.76}$$

and have $N \to \infty$ as a Brownian Motion (Donsker Theorem)

Rigorous definition of **Brownian / Wiener Process**: $\{W_t : T \ge 0\}$ with $0 < \sigma^2 < \infty$ is Brownian if

[4]Symbol W_t for 'Wiener', sometimes uses B_t for 'Brown'.

1. Starts from 0: $\mathbb{P}(W_0) = 1$
2. Independent increment: $W_{t_1} - W_{s_1} \perp\!\!\!\perp W_{t_2} - W_{s_2}, \forall [t_1, s_1] \cap [t_2, s_2] = \emptyset$
3. Zero mean Normal: $W_t - W_s \sim N(0, \sigma^2|t-s|)$
4. continuity: $\mathbb{P}(W_t \text{ continuous}) = 1$

Properties:

- Parameters

$$\begin{cases} \text{Mean Function} : \mu(t) = 0 \\ \text{Covariance} : \gamma(t,s) = \sigma^2 \min\{s,t\} \end{cases} \tag{12.77}$$

- m.s. indifferentiable

$$\mathbb{E}\left[\left(\frac{\partial W_t}{\partial t}\right)^2\right] \to \infty \tag{12.78}$$

which is the reason why the plots for Brownian Motion always looks rugged.

- Conditional distribution / Brownian Bridge B_t:

$$B_t := W_t | W_T = 0 \sim N(0, \sigma^2 \frac{t(T-t)}{T}) \tag{12.79}$$

  - Dependent increment: non-zero covariance

$$\gamma_{\text{Bridge}}(t,s) = \sigma^2 \left(\min\{t,s\} - \frac{ts}{T}\right) \tag{12.80}$$

  - Cross definition between Wiener Process and Brownian Bridge:

$$\begin{cases} B_t := W_t - \frac{t}{T} W_T \\ W_t := B_t + t\sigma^2 N(0,1) \end{cases} \tag{12.81}$$

i.e. Brownian Bridge is independent of the terminal of its corresponding Wiener Process $B_t \perp\!\!\!\perp W_T$.

12.2.5 Poisson Process

Motivation: The accumulate events happens at random, with 'happening rate' of events as λ

$$N_{t=\frac{k}{N}} := \sum_{i=1}^{k} \nu_i, \quad \nu_i \sim_{\text{i.i.d.}} \text{Bern}(\frac{\lambda}{n}) \tag{12.82}$$

Rigorous Definition of **Poisson Process**: $\{N_t : t \geq 0\}$ with rate $\lambda > 0$ is Poisson if

- Counting Process N_t: $N_0 = 0$, $N_t \in \mathbb{N}$
- Independent Increment: $N_{t_1} - N_{s_1} \perp\!\!\!\perp N_{t_2} - N_{s_2}, \forall [t_1, s_1 \cap [t_2, s_2] = \emptyset$
- Poisson increment: $N_t - N_s \sim P(\lambda(t-s)), t \geq s$[5]

[5] A proof & another kind of definition concerning the intuition of 'rate λ' is here: https://v1ncent19.github.io//texts/Poisson/.

Properties:

- Parameters

$$\begin{cases} \text{Mean Function} : \mu(t) = \lambda t \\ \text{Covariance} : \gamma(t,s) = \lambda \min\{s,t\} \end{cases} \tag{12.83}$$

- Arrival time: $N_{t_n} = n$ means there are n events before (and including) t_n, denoted $\{t_1, t_2, \dots, t_n\}$. PDF

$$f_{T_1,T_2,\dots,T_n}(t_1,t_2,\dots,t_n) = \lambda^n e^{-\lambda t_n} \mathbb{I}_{0<t_1<t_2<\dots<t_n} \tag{12.84}$$

- Inter-event time: PDF of time-between-events $\{u_1,u_2,\dots,u_n\} := \{t_1, t_2-t_1, \dots, t_n - t_{n-1}\}$

$$f_{U_1,U_2,\dots,U_n}(u_1,u_2,\dots,u_n) = \prod_{i=1}^{n} \lambda e^{-\lambda u_i} \mathbb{I}_{u_i \geq 0} =\sim \otimes i = 1^n \varepsilon_i(\lambda) \tag{12.85}$$

i.e. time-between-events satisfies exponential distribution

$$U_i \sim_{\text{i.i.d.}} \varepsilon(\lambda) \tag{12.86}$$

- Conditional distribution

$$f_{T_1,T_2,\dots,T_n|N_t=n}(t_1,t_2,\dots,t_n) = \frac{n!}{t^n}\mathbb{I}_{0<t_1<t_2<\dots<t_n} \sim \text{Unif}\left(\mathbb{I}_{0<t_1<t_2<\dots<t_n\leq t}\right) \tag{12.87}$$

is the PDF of order statistics[6] of i.i.d. Unif$(0,t)$.

- Poisson Process and Martingale:

$$\tilde{N}_t := N_t - \lambda t \sim \text{Martingale} \tag{12.88}$$

12.2.6 Birth-Death Process

Birth-death process looks like a one-end random-walk with 'step' as poisson r.v.(i.e. exponential time-interval) The transition rate & diagram are:

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & \cdots \end{array} \\ \begin{pmatrix} -\lambda_0 & \lambda_0 & & & \\ \mu_1 & -\mu_1-\lambda_1 & \lambda_1 & & \\ & \mu_2 & -\mu_2-\lambda_2 & \lambda_2 & \\ & & \mu_3 & \ddots & \ddots \\ & & & \ddots & \ddots \end{pmatrix} \end{array} \quad \leftrightharpoons \quad \begin{array}{ccccccccc} \boxed{0} & \overset{\lambda_0}{\rightleftarrows} & \boxed{1} & \overset{\lambda_1}{\rightleftarrows} & \boxed{2} & \overset{\lambda_2}{\rightleftarrows} & \cdots \\ & \underset{\mu_1}{} & & \underset{\mu_2}{} & & \underset{\mu_3}{} & \end{array}$$

- Kolmogorov forward: with a trivial notation that $\lambda_{-1} = \mu_0 = 0$, we have

$$\dot{p}_i(t) = \lambda_{i-1}p_{i-1}(t) + \mu_{i+1}p_{i+1}(t) - (\lambda_i + \mu_i)p_i(t) \tag{12.89}$$

[6] See equation 1.47 ~ page 24.

- Stationary Distribution: $\dot{\pi}^* = 0$ yields

$$(\lambda_i + \mu_i)\pi_i^* = \lambda_{i-1}\pi_{i-1}^* + \mu_{i+1}\pi_{i+1}^* \tag{12.90}$$

Solution:

$$\pi_i^* = \begin{cases} \frac{1}{Z}\frac{\lambda_0\lambda_1\dots\lambda_{i-1}}{\mu_1\mu_2\dots\mu_i}, & i \neq 0 \\ \frac{1}{Z}, & i = 0 \end{cases}, \quad Z = 1 + \sum_{j=1}^{\infty}\frac{\lambda_0\lambda_1\dots\lambda_{j-1}}{\mu_1\mu_2\dots\mu_j} \tag{12.91}$$

Section 12.3 Applications

12.3.1 Innovation Sequence

Motivation of Innovation Sequence (新息序列): construction of linear MMSE $L(X|Y_1, Y_2, \dots, Y_n) = L(X|Y)$. Assume that $\mathbb{E}\left[\vec{Y}\right] = 0$, the prediction is

$$L(X|\vec{Y}) = \mathbb{E}[X] + cov(X, \vec{Y})var(\vec{Y})^{-1}\vec{Y} \tag{12.92}$$

which causes the problem of computation complexity when dimension n is large.

Innovation sequence fixed this problem by: instead of projecting on the whole linear combination $\vec{Y}$ space of size $(n+1)$, we project on space of each Y_i sequentially. i.e. define an **innovation sequence**

$$\tilde{Y}_1 = Y_1 - \mathbb{E}[Y_1] = Y_1 - \mathbb{E}[Y_1] \tag{12.93}$$

$$\tilde{Y}_2 = Y_2 - L(Y_2|Y_1) = Y_2 - L(Y_2|\tilde{Y}_1) \tag{12.94}$$

$$\tilde{Y}_3 = Y_3 - L(Y_3|Y_2Y_1) = Y_3 - L(Y_3|\tilde{Y}_2\tilde{Y}_1) \tag{12.95}$$

$$\dots \tag{12.96}$$

$$\tilde{Y}_n = Y_n - L(Y_n|Y_{n-1}\dots Y_2Y_1) = Y_n - L(Y_n|\tilde{Y}_{n-1}\dots\tilde{Y}_2\tilde{Y}_1) \tag{12.97}$$

where 'innovation' means each $\tilde{Y}_i$ contains the 'new information without correlation with previous sequence': $\mathbb{E}\left[\tilde{Y}_i\tilde{Y}_j\right] = 0 \,\forall i \neq j$. Computation of innovation sequence:

$$\tilde{Y}_k = Y_k - L(Y_k|\tilde{Y}_{k-1}\dots\tilde{Y}_1) = Y_k - \mathbb{E}[Y_k] - \sum_{j=1}^{k-1}\frac{cov(Y_k, \tilde{Y}_j)}{var(\tilde{Y}_j)}\tilde{Y}_j, \quad k = 1, 2, \dots, n \tag{12.98}$$

with a trivial notation that $Y_0 = 1$

In this way a linear MMSE $L(X|\vec{Y})$ could be written as

$$L(X|\vec{Y}) = L(X|\vec{\tilde{Y}}) = \mathbb{E}[X] + \sum_{i=1}^{n}\frac{cov(X, \tilde{Y}_i)}{var(\tilde{Y}_i)}\tilde{Y}_i = \mathbb{E}[X] + \sum_{i=1}^{n}L\left(X - \mathbb{E}[X]\,|\tilde{Y}_i\right) \tag{12.99}$$

I think the idea here is similar to Gram-Schmidt orthogonalization (section 5.2.4 ~ page 153), in which we also construct new components by eliminating projection on previous parts. As a result we have a set of orthogonal elements (here orthogonal means $\mathbb{E}\left[\tilde{Y}_i\tilde{Y}_j\right] = 0$ and in Gram-Schmidt means $q_i'q_j = 0, i \neq j$). And the result is a 'change of basis' of space.

12.3.2 Markov Decision Processes

In decision process/episode, say $\{(s_t, a_t)\}_{t=0}^{T}$, we need to determine a **policy** π_t to take **action** a_t given **state** s_t as

$$a_t \sim \pi_t(\cdot|s_t) \text{ or simply } a_t = \pi_t(s_t) \tag{12.100}$$

then (conditional) **transition** probability is a model pre-assumed, say

$$s_{t+1} \sim p_t(\cdot|s_t, a_t) \tag{12.101}$$

□ **Optimization Target**

The optimization target (in each step) is **reward function**

$$r_t(s_t, s_{t+1}|a_t) \tag{12.102}$$

The 'cumulative reward' from step t is denoted $\mathcal{V}_{t\rightsquigarrow T}$[7]

$$\mathcal{V}_{t\rightsquigarrow T}^{\pi_{t:T}}(s_t) = \mathbb{E}_{s_{t+1}\sim p(\cdot|s_t, a_t=\pi(s_t))}\left[r_t(s_t, s_{t+1}|a_t=\pi_t(s_t)) + \gamma \mathcal{V}_{(t+1)\rightsquigarrow T}^{\pi_{(t+1):T}}(s_{t+1})\middle| s_t\right] \tag{12.103}$$

where **discount factor** $\gamma < 1$ is induced to focus on recent rewards. By expanding all iteration terms we have

$$\mathcal{V}_{t\rightsquigarrow T}^{\pi_{t:T}}(s_t) = \mathbb{E}_{s_{(t+1):(T+1)}}\left[\sum_{\tau=t}^{T}\gamma^{\tau-t} r_\tau(s_\tau, s_{\tau+1}|a_\tau=\pi_\tau(s_\tau))\middle| s_t\right] \tag{12.104}$$

and the final optimize goal is maximize total reward $\mathcal{V}$

$$\pi_{0:T}^* = \arg\max_{\pi_{0:T}} \mathbb{E}_{s_0\sim p_0(\cdot)}\left[\mathcal{V}_{0\rightsquigarrow T}^{\pi_{0:T}}(s_0)\right] \tag{12.105}$$

$$= \arg\max_{\pi_{0:T}} \mathbb{E}_{s_{0:(T+1)}}\left[\sum_{\tau=0}^{T}\gamma^{\tau} r_\tau(s_\tau, s_{\tau+1}|a_\tau=\pi_\tau(s_\tau))\right] \tag{12.106}$$

Comments:

- The joint distribution of $s_{t+1,T+1}$ has a complicated dependence on $p_\tau(\cdot|s_\tau, a_\tau)$, making the optimization hard to solve directly.
- Actually when making decision we should consider a complete process, i.e. $T \to \infty$, but note that with $\gamma < 1$, reward at far future is dispensable if rewards are upper-bounded $r_\tau(s_\tau, s_{\tau+1}|a_\tau) \le \tilde{r}$, then

$$\sum_{\tau=T}^{\infty}\gamma^{\tau} r_\tau(s_\tau, s_{\tau+1}|a_\tau=\pi_\tau(s_\tau)) \le \tilde{r}\frac{\gamma^T}{1-\gamma} \tag{12.107}$$

which can be bounded below $\varepsilon\tilde{r}$ for a large enough **Effective Length** T_ε

$$\tilde{r}\frac{\gamma^T}{1-\gamma} < \tilde{r}\varepsilon \Rightarrow T_\varepsilon \approx \frac{\log[(1-\gamma)\varepsilon]}{\log\gamma} \sim \mathcal{O}\left(\frac{1}{1-\gamma}\log\frac{1}{\varepsilon(1-\gamma)}\right) \sim \mathcal{O}(\frac{1}{1-\gamma}) \tag{12.108}$$

[7] In this subsection I usually use the superscript $\cdot^{\pi_{t:T}}$ to specify the optimize target.

□ **Algorithm**

Solving all $\pi_{0:T}$ jointly in equation 12.105 ~ page 322 is complex. It would be wiser to use the iteration form equation 12.103 ~ page 322 and *separate decision making* a_t *and processing* $p(\cdot|s_t,a_t)$. With expected rewards denoted

$$R_t(s_t,a_t)=\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[r_t(s_t,s_{t+1}|a_t)\middle|s_t,a_t\right] \tag{12.109}$$

total reward $\mathcal{V}_{t\rightsquigarrow T}$ could be written as[8]

$$\mathcal{V}^{\pi_{t:T}}_{t\rightsquigarrow T}(s_t)=\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t\sim\pi(s_t))}\left[r_t\left(s_t,s_{t+1}|a_t\sim\pi_t(s_t)\right)+\gamma\mathcal{V}^{\pi_{(t+1):T}}_{(t+1)\rightsquigarrow T}(s_{t+1})\middle|s_t\right] \tag{12.110}$$

$$=\mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[R_t\left(s_t,a_t\right)+\gamma\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[\mathcal{V}^{\pi_{(t+1):T}}_{(t+1)\rightsquigarrow T}(s_{t+1})\middle|s_t,a_t\right]\middle|s_t\right] \tag{12.111}$$

with the red part as **State-Value Function**, or V-value; the blue part as **Action-Value Function**, or Q-value

$$V^{\pi_{t:T}}_{t\rightsquigarrow T}(s_t)=\mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[Q^{\pi_{(t+1):T}}_{t\rightsquigarrow T}(s_t,a_t)\middle|s_t\right] \tag{12.112}$$

$$Q^{\pi_{(t+1):T}}_{t\rightsquigarrow T}(s_t,a_t)=R_t\left(s_t,a_t\right)+\gamma\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[V^{\pi_{(t+1):T}}_{(t+1)\rightsquigarrow T}(s_{t+1})\middle|s_t,a_t\right] \tag{12.113}$$

Comments:

- The decision process $(s_0,a_0)\rightsquigarrow(s_1,a_1)\rightsquigarrow\ldots\rightsquigarrow(s_T,a_T)$ is Markovian in $t=0\rightsquigarrow T$ sense, while the reward propagation $V_{T\rightsquigarrow T}\rightsquigarrow Q_{(T-1)\rightsquigarrow T}\rightsquigarrow V_{(T-1)\rightsquigarrow T}\rightsquigarrow\ldots\rightsquigarrow Q_{0\rightsquigarrow T}\rightsquigarrow V_{0\rightsquigarrow T}$ is 'Markovian' in $t=T\rightsquigarrow 0$ sense. i.e. solution to optimal π^* obtained by maximizing total reward should go backward.
- Duality of optimal $\{V^{\pi_{t:T}}_{t\rightsquigarrow T}\}_{t=0}^T$ (V-learning) and optimal $\{Q^{\pi_{t:T}}_{t\rightsquigarrow T}\}_{t=0}^T$ (Q-learning): With $R_t(s_t,a_t)$ actually a given function (for given model $p(s_{\tau+1}|s_\tau,a_\tau)$),

$$\begin{cases}V^{\pi_{t:T}}_{t\rightsquigarrow T}(s_t)=\mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[R_t\left(s_t,a_t\right)+\gamma\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[V^{\pi_{(t+1):T}}_{(t+1)\rightsquigarrow T}(s_{t+1})\middle|s_t,a_t\right]\middle|s_t\right]\\ Q^{\pi_{(t+1):T}}_{t\rightsquigarrow T}(s_t,a_t)=R_t(s_t,a_t)+\gamma\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[\mathbb{E}_{a_{t+1}\sim\pi(\cdot|s_{t+1})}\left[Q^{\pi_{(t+2):T}}_{(t+1)\rightsquigarrow T}(s_{t+1},a_{t+1})\middle|s_{t+1}\right]\middle|s_{t+1},a_{t+1}\right]\end{cases} \tag{12.114}$$

  are equivalent, with the same optimization core $\mathbb{E}_{a_\tau\sim\pi(\cdot|s_\tau)}[\cdot|s_\tau]$.

Δ Value function iteration for optimal policy π^*:

$$\pi^*_t(s)=\arg\max_a Q^*_t(s,a),\quad t=T,T-1,\ldots,0 \tag{12.115}$$

Algorithm *Value Iteration*

1. $V^*_{T+1}\equiv 0$
2. *for* $t=T,T-1,\ldots,1$

[8] Here the ugly symbol means, e.g.

$$\mathcal{V}^{\text{is influenced by which policies }\pi.}_{\text{describes the process from when to when}}(\text{as a function of which state})$$

(a) Q-expectation step:

$$Q_t^*(s,a) = R_t(s,a) + \gamma \mathbb{E}_{\tilde{s}\sim p(\,\cdot\,|s,a)}\left[V_{t+1}^*(\tilde{s})\middle|s,a\right] \tag{12.116}$$

(b) V-Optimal step:

$$\begin{cases} \pi_t^*(s) = & \arg\max\limits_a Q_t^*(s,a) \\ V_t^*(s) = & \max\limits_a Q_t^*(s,a) = Q_t^*(s,\pi_t^*(s)) \end{cases} \tag{12.117}$$

i.e. a (Q_t, V_t) 'backward propagation'.

□ **Q-Learning**

Motivation: for some more complex cases, e.g.

- The functional form of reward $r_t(s_t, s_{t+1}|a_t)$ or $R_t(s_t, a_t)$ is unknown
- The transition probability $s_{t+1} \sim p(\,\cdot\,|s_t, a_t)$ is unknown
- The phase space is too large to compute point wise

Note that the above optimize process equation 12.117 ~ page 324 is an optimization w.r.t. $Q_t(\,\cdot\,,\,\cdot\,)$, we can first learn the functional form of $Q(\,\cdot\,,\,\cdot\,)$ (or its function approximation), and thus get the policy π^*. The Q-learning process can have the following form:

$$\hat{Q}^{(\tau+1)}(s_t,a_t) \leftarrow \overbrace{\hat{Q}^{(\tau)}(s_t,a_t)}^{\text{current value}} + \alpha \cdot \overbrace{\Bigg(\underbrace{R_t(s_t,a_t) + \gamma \cdot \underbrace{\max_a \hat{Q}^{(\tau)}(s_{t+1},a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{\hat{Q}^{(\tau)}(s_t,a_t)}_{\text{current value}} \Bigg)}^{\text{temporal difference}} \tag{12.118}$$

with some known *final/terminal* state $\{s_{\text{final}}\}$, where $Q(s_{\text{final}}, a) \equiv 0,\ \forall a$

Algorithm *Q-Learning*

1. Initialize a tentative $Q_t^{(0)}(\,\cdot\,,\,\cdot\,)$, say $Q \equiv 0$
2. *for* $\tau = 0, 1, 2, \ldots$ *until* $Q(\,\cdot\,,\,\cdot\,)$ converge:
   (a) Initialize some s_1
   (b) *for* $t = 1, 2, \ldots$ *until* $s_t \in \{s_{\text{final}}\}$: optimize the function form (approximation) $Q(\,\cdot\,,\,\cdot\,)$

$$\hat{Q}^{(\tau+1)}(s_t,a_t) \leftarrow \hat{Q}^{(\tau)}(s_t,a_t) + \alpha\left(R_t(s_t,a_t) + \gamma \max_a \hat{Q}_{t+1}^{(\tau)}(s_{t+1},a) - \hat{Q}_t^{(\tau)}(s_t,a_t)\right) \tag{12.119}$$

$$s_{t+1} \leftarrow p_t(s_t,a_t) \tag{12.120}$$

12.3.3 Karhunen-Loève Expansion

Karhunen-Loève Expansion (KL Expansion) is a coutinuous version of PCA section 4.3 ~ page 129. The idea is a decomposition

$$X(t) = \sum_i X_i \phi_i(t) \tag{12.121}$$

i.e. we add an extra step in mapping

$$X(\cdot) : \Omega \mapsto \{X_i\} \mapsto \mathcal{T} \times \mathbb{R} \tag{12.122}$$

a special set of $\{X_i, \phi_i\}$ is given by KL expansion.

□ **Derivations**

First note that $R(s,t) := \mathbb{E}[X(s)X(t)]$ is a Kernel (see equation 9.100 ~ page 254), with positive semi-definition and symmetry. Then by Mercer's Theorem, it has eigen-function decomposition

$$R(s,t) = \sum_i \lambda_i \phi_i(s)\phi_i(t) \Leftrightarrow \langle R(s,\cdot), \phi_i \rangle = \lambda_i \phi_i(s) \tag{12.123}$$

where eigen functions are orthonormal

$$\langle \phi_i, \phi_j \rangle := \int_\tau \phi_i(\tau)\phi_j(\tau)\,\mathrm{d}\tau = \delta_{ij} \tag{12.124}$$

using $\{\phi_i\}$ as function basis, KL coefficients are r.v.

$$X_i = \langle X_t, \phi_i \rangle \tag{12.125}$$

with

$$\mathbb{E}[X_i X_j] = \langle \phi_i | X_t \rangle \langle X_t | \phi_j \rangle = \langle \phi_i | R(s,t) | \phi_j \rangle = \delta_{ij}\lambda_i \tag{12.126}$$

□ **Other Concepts**

- Total energy:

$$E = \mathbb{E}[\langle X_t, X_t \rangle] = \sum_i \lambda_i \tag{12.127}$$

- Rank: $\operatorname{rank}(\{\mathbb{E}[X_i X_j]\}) = \#(\lambda_i \neq 0)$ is also the rank of the process.

12.3.4 Kalman Filter

□ **Model**

Kalman Filter is an auto-regressive / iterative filter for estimating the **state** x_t from **observable**[9] z_t. The model structure, as in figure 12.1 ~ page 326, is a Hidden Markov Model (HMM) with linear operator.

$$\text{State: } x_k = F_k x_{k-1} + w_k \tag{12.128}$$

$$\text{Observable: } z_k = H_k x_k + v_k \tag{12.129}$$

[9] Here I prefer the name as in Quantum mechanics 'Observable'.

where w_k, v_k is noise / random error, usually with (multivariate) Normal distribution

$$w_k \sim N(0, Q_k), \qquad v_k \sim N(0, R_k) \tag{12.130}$$

the initial state denoted

$$x_0 \sim N(\hat{x}_{0|0}, P_{0|0}) \tag{12.131}$$

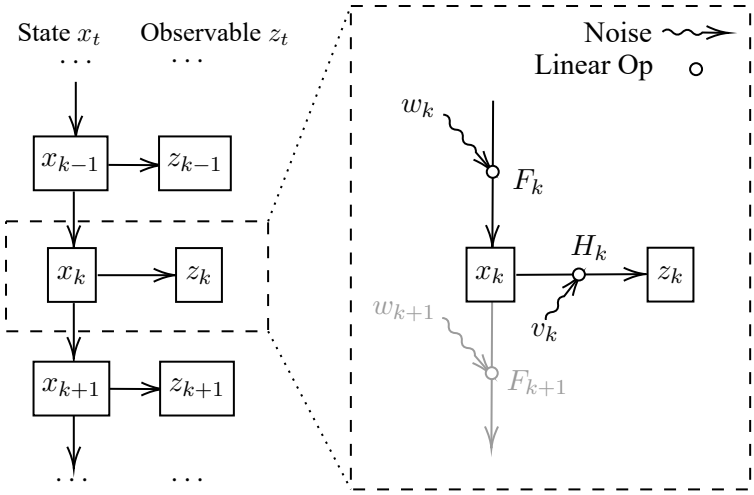


图 12.1: HMM structure of Kalman Filter

□ Algorithm

Motivation: what we could observe is $\{z_k\}$ sequence, with pre-specified $\{F_k, H_k, Q_k, R_k\}$, which are part of the model. We hope to (linearly) estimate the value and variance of the hidden state x_k

$$\text{value: } \hat{x}_{k|k-1} := L(x_k | z_1 \dots z_{k-1}) \tag{12.132}$$

$$\text{variance: } P_{k|k-1} := var\left(x_k - \hat{x}_{k|k-1}\right) \tag{12.133}$$

Algorithm *Kalman Filter*

1. Initial State: $x_0 \sim N(\hat{x}_{0|0}, P_{0|0})$; Model given $\{F_k, H_k, Q_k, R_k\}$;
2. *for* $k = 1, 2, \dots$
   (a) State Predict: $\cdot_{k-1|k-1} \mapsto \cdot_{k|k-1}$

$$\text{prior state: } \hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \tag{12.134}$$

$$\text{prior cov: } P_{k|k-1} = F_k P_{k-1|k-1} F_k' + Q_k \tag{12.135}$$

   (b) Information Update: weighting btw. $\cdot_{k|k-1}$ and $\cdot_k$

$$\text{innovation seq: } \tilde{z}_k = z_k - H_k \hat{x}_{k|k-1} \tag{12.136}$$

$$\text{innovation cov: } S_k = H_k P_{k|k-1} H_k' + R_k \tag{12.137}$$

$$\text{(Optimal) Kalman gain: } K_k = P_{k|k-1} H_k' S_k^{-1} \tag{12.138}$$

$$= P_{k|k-1} H_k' \left(H_k P_{k|k-1} H_k' + R_k\right)^{-1} \tag{12.139}$$

(c) State Update: $\cdot_{k|k-1} \mapsto \cdot_{k|k}$

$$\text{posterior state: } \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{z}_k \tag{12.140}$$
$$= \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1}\right) \tag{12.141}$$
$$= \left(I - K_k H_k\right) \hat{x}_{k|k-1} + K_k z_k \tag{12.142}$$
$$\text{posterior cov: } P_{k|k} = \left(I - K_k H_k\right) P_{k|k-1} \left(I - K_k H_k\right)' + K_k R_k K_k' \tag{12.143}$$
$$= \left(I - K_k H_k\right) P_{k|k-1} \tag{12.144}$$

□ **Derivation Details**

- Key concepts in Kalman Filter:

$$\text{prior state: } \hat{x}_{k|k-1} = L(x_k | z_1 \dots z_{k-1}) \tag{12.145}$$
$$\text{prior covariance: } P_{k|k-1} = var\left(x_k - \hat{x}_{k|k-1}\right) \tag{12.146}$$
$$\text{posterior state: } \hat{x}_{k|k} = L(x_k | z_1 \dots z_k) \tag{12.147}$$
$$\text{posterior covariance: } P_{k|k} = var\left(x_k - \hat{x}_{k|k}\right) \tag{12.148}$$
$$\text{Kalman gain: } K_k \tag{12.149}$$

(a1) prior state prediction

$$\hat{x}_{k|k-1} = L(x_k | z_1 \dots z_{k-1}) = L(F_k x_{k-1} + w_k | z_1 \dots z_{k-1}) = F_k \hat{x}_{k-1|k-1} \tag{12.150}$$

(a2) prior covariance prediction

$$P_{k|k-1} = var\left(x_k - \hat{x}_{k|k-1}\right) = var\left(F_k(x_{k-1} - \hat{x}_{k-1|k-1}) + w_k\right) = F_k P_{k-1|k-1} F_k' + Q_k \tag{12.151}$$

(b1) innovation sequence of z_k

$$\tilde{z}_k = z_k - L(z_k | z_1 \dots z_{k-1}) = z_k - L(H_k x_k + v_k | z_1 \dots z_{k-1}) = z_k - H_k \hat{x}_{k|k-1} \tag{12.152}$$

(b2) innovation sequence variance

$$S_k := var(\tilde{z}_k) = var\left(z_k - H_k \hat{x}_{k|k-1}\right) = var\left(H_k(x_k - \hat{x}_{k|k-1}) + v_k\right) = H_k P_{k|k-1} H_k' + R_k \tag{12.153}$$

(b3) Optimal Kalman gain is obtained by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L(x_k - \mathbb{E}[x] | \tilde{z}_k) = \hat{x}_{k|k-1} + cov(x_k, \tilde{z}_k) var(\tilde{z}_k)^{-1} \tilde{z}_k := \hat{x}_{k|k-1} + K_k \tilde{z}_k \tag{12.154}$$

i.e. Optimal Kalman gain in the combination coefficient in MMSE.

$$K_k = cov(x_k, \tilde{z}_k) var(\tilde{z}_k)^{-1} = cov(x_k, H_k(x_k - \hat{x}_{k|k-1}) + v_k) S_k^{-1} \tag{12.155}$$
$$= cov(x_k - \hat{x}_{k|k-1}, x_k - \hat{x}_{k|k-1}) H_k' S_k^{-1} \tag{12.156}$$
$$= P_{k|k-1} H_k' S_k^{-1} \tag{12.157}$$

here we use the property of MMSE

$$cov(\hat{x}_{k|k-1}, x_k - \hat{x}_{k|k-1}) = 0 \tag{12.158}$$

(c1) posterior state update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{z}_k = (I - K_kH_k)\,\hat{x}_{k|k-1} + K_kz_k \tag{12.159}$$

(c2) posterior variance update

$$\begin{aligned} P_{k|k} =& var(x_k - \hat{x}_{k|k}) = var\left(x_k - \hat{x}_{k|k-1} - K_k(z_k - H_k\hat{x}_{k|k-1})\right) && (12.160)\\ =& var\left(x_k - \hat{x}_{k|k-1} - K_k(H_kx_k + v_k - H_k\hat{x}_{k|k-1})\right) && (12.161)\\ =& var\left((I - K_kH_k)(x_k - \hat{x}_{k|k-1}) - K_kv_k\right) && (12.162)\\ =& (I - K_kH_k)P_{k|k-1}(I - K_kH_k)' + K_kR_kK_k' && (12.163) \end{aligned}$$

further if K_k takes optimal Kalman gain,

$$K_kS_kK_k' = P_{k|k-1}H_k'K_k' \tag{12.164}$$

we have a simplification

$$\begin{aligned} P_{k|k} =& (I - K_kH_k)P_{k|k-1}(I - K_kH_k)' + K_kR_kK_k' && (12.165)\\ =& P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}H_k'K_k' + K_k\left(H_kP_{k|k-1}H_k' + R_k\right)K_k' && (12.166)\\ =& P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}H_k'K_k' + K_kS_kK_k' && (12.167)\\ =& (I - K_kH_k)\,P_{k|k-1} && (12.168) \end{aligned}$$

□ **Comments**

- Optimality of Kalman Filter as a MMSE: in equation 12.160 ~ page 328, posterior variance does **not** depend on a concrete form of Kalman gain, thus in which Kalman filter can be selected as some other ones $\tilde{K}_k$ (e.g. to avoid numerical instability). The optimal Kalman gain is the one that minimizes $tr\left(P_{k|k}\right)$

$$K_k = \arg\min_K tr\left((I - KH_k)P_{k|k-1}(I - KH_k)' + KR_kK'\right) \tag{12.169}$$

obtained by[10]

$$\frac{\partial tr\left(P_{k|k}\right)}{\partial K} = -2\left(H_kP_{k|k-1}\right)' + 2K_kS_k = 0 \Rightarrow K_k = P_{k|k-1}H_k'S_k^{-1} \tag{12.170}$$

- Role of Kalman gain K_k: in posterior update equation 12.159 ~ page 328 we can see that K_k looks like a weighting factor btw. history information $\hat{x}_{k|k-1}$ and new observation z_k.

$$\hat{x}_{k|k} = (I - K_kH_k)\,\hat{x}_{k|k-1} + K_kz_k \tag{12.171}$$

and note that the Kalman gain update

$$\begin{aligned} P_{k|k-1} =& F_kP_{k-1|k-1}F_k' + Q_k && (12.172)\\ S_k =& H_kP_{k|k-1}H_k' + R_k && (12.173)\\ K_k =& P_{k|k-1}H_k'S_k^{-1} && (12.174)\\ P_{k|k} =& (I - K_kH_k)P_{k|k-1} && (12.175) \end{aligned}$$

[10] Matrix differentiation see section 4.1.2 ~ page 118

only involve $\{F_k, H_k, Q_k, R_k\}$ and initial $P_{0|0}$. It means Kalman gain K_k could be computed offline. In actual application scenario we can just compute state iteratively

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \tag{12.176}$$

$$\hat{x}_{k|k} = (I - K_k H_k)\, \hat{x}_{k|k-1} + K_k z_k \tag{12.177}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k' + Q_k \tag{12.178}$$

$$P_{k|k} = (I - K_k H_k)\, P_{k|k-1} \tag{12.179}$$

- Asymptotic form: when step $k \to \infty$, we may have limit

$$F_k \to F, \quad H_k \to H, \quad Q_k \to Q, \quad R_k \to R \tag{12.180}$$

then Kalman filter and variance estimation have asymptotic form by solving

$$P_\infty = F\left(P_\infty - P_\infty H' \left(H P_\infty H + R\right)^{-1} H P_\infty\right) F' + Q \tag{12.181}$$

$$K_\infty = P_\infty H' \left(H P_\infty H' + R\right)^{-1} \tag{12.182}$$

and the asymptotic update

$$\hat{x}_{k+1} = F\left(I - K_\infty H\right) \hat{x}_k + F K_\infty z_k \tag{12.183}$$

- Extended Kalman Filter (EKF): Kalman filter assumes a linear model with noise. Usually it's a good-enough approximator to the real case. For non-linear case, i.e. Extended Kalman filter, has model

$$\text{State: } x_k = f_k(x_{k-1}) + w_k \tag{12.184}$$

$$\text{Observable: } z_k = h_k(x_k) + v_k \tag{12.185}$$

the update could be obtained by replacement

$$F_k = \frac{\partial}{\partial x} f_k\left(\hat{x}_{k-1|k-1}\right), \qquad H_k = \frac{\partial}{\partial x} h_k\left(\hat{x}_{k|k-1}\right) \tag{12.186}$$

- Kalman-Bucy Filter is the continuous time version of Kalman filter, with model

$$\text{State: } \frac{\mathrm{d}x(t)}{\mathrm{d}t} = F(t)x(t) + w(t) \tag{12.187}$$

$$\text{Observable: } z(t) = H(t)x(t) + v(t) \tag{12.188}$$

where $w(t)$, $v(t)$ are white noise.

Kalman update:

$$\frac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} = (F(t) - K(t)H(t)\hat{x}(t)) + K(t)z(t) \tag{12.189}$$

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = F(t)P(t) + P(t)F(t)' + Q(t) - K(t)R(t)K(t)' \tag{12.190}$$

with Kalman gain

$$K(t) = P(t)H(t)'R(t)^{-1} \tag{12.191}$$

12.3.5 Linear Time Invariant Systems

Linear Time Invariant Systems (LTI Systems) models data generation process as a convolution

$$x(t) = \int_{\mathbb{R}} z(\tau)h(t-\tau)\,\mathrm{d}\tau = (z * g)(t) \tag{12.192}$$

where $\int$ for linear, and $h(t-\tau)$ for time-invariant.

LTI systems could be conveniently parsed with Fourier Transform, introduced in section 12.4.3 ~ page 334.

□ **Cross Correlation Structure**

Usually we consider weak stationary case, with notation:

$$\mu_X, \quad \mu_Z, \quad R_Z(t) = \mathbb{E}\left[z(s)z(s+t)\right], \forall s, \quad R_{XZ}(t) = \mathbb{E}\left[x(s)z(s+t)\right], \forall s \tag{12.193}$$

corresponding Fourier transform:

$$R_Z(t) \fallingdotseq S_Z(\omega), \quad R_{XZ}(t) \fallingdotseq S_{XZ}(\omega), \quad h(t) \fallingdotseq H(\omega) \tag{12.194}$$

Relations:

$$\mu_X = \sqrt{2\pi}\mu_Z H(0) \tag{12.195}$$

$$R_{XZ}(t) = (R_Z * h)(t) \tag{12.196}$$

$$R_X(t) = (h * R_Z * \tilde{h})(t), \quad \tilde{h}(\tau) = h(-\tau) \tag{12.197}$$

$$S_{XZ}(\omega) = \sqrt{2\pi}S_Z(\omega)H(\omega) \tag{12.198}$$

$$S_X(\omega) = 2\pi S_Z(\omega)|H(\omega)|^2 \tag{12.199}$$

12.3.6 Wiener Filter

Goal of Wiener Filter is to estimate some x_u from $z_t : t \in [a,b]$ with a linear function in MMSE sense $\hat{x}_u = L(x_u | z_t : t \in [a,b])$:

$$\hat{x}_u = \int_a^b z_\tau h(\tau, u)\,\mathrm{d}\tau, \quad w.r.t.h(\,\cdot\,) = \arg\min_h \mathbb{E}\left[(x_u - \hat{x}_u)^2\right] \tag{12.200}$$

the solution, as explained in section 12.4.1 ~ page 331, satisfies $\mathbb{E}\left[(x_u - \hat{x}_u)z_t\right] = 0, \forall t \in [a,b]$, which yields

$$R_{XZ}(u,t) = \int_a^b R_Z(t,\tau)h(u-\tau)\,\mathrm{d}\tau, \quad \forall t \in [a,b] \tag{12.201}$$

usually we also consider weak stationary case, with $[a,b] = \mathbb{R}$

$$R_{XZ}(u-t) = \int_{\mathbb{R}} R_Z(\tau - t)h(u-\tau)\,\mathrm{d}\tau, \quad \forall t \in [a,b] \tag{12.202}$$

□ **Non-Causal Solution** A general solution $L(x_u | z_t : t \in \mathbb{R})$ is easily obtained by Fourier transform, with the convolution expression of estimator

$$S_{XZ}(\omega) = \sqrt{2\pi}S_Z(\omega)H(\omega) \Rightarrow H(\omega) = \frac{S_{XZ}(\omega)}{\sqrt{2\pi}S_Z(\omega)} \tag{12.203}$$

with MSE[11]

$$\text{MSE} = \int_{\mathbb{R}} S_X(\omega) - \frac{|S_{XZ}(\omega)|^2}{S_Z(\omega)}\,\mathrm{d}\omega \tag{12.204}$$

□ **Causal Solution**

Causal solution demands that estimation cannot use future information, modelled as

$$\hat{x}_T = L(x_T | z_t : t \in (-\infty, 0]), \quad T > 0 \tag{12.205}$$

i.e.

$$\hat{x}_T = \int_{\mathbb{R}} z_\tau h(-\tau)\,\mathrm{d}\tau, \quad w.r.t.\, h(\varsigma) = h(\varsigma)\eta(\varsigma) \tag{12.206}$$

$$R_{XZ}(T+t) = \int_{\mathbb{R}} R_Z(\tau + t)h(-\tau)\,\mathrm{d}\tau, \quad \forall t \geq 0 \tag{12.207}$$

MMSE condition

$$\left[e^{i\omega T} S_{XZ}\right]_+ = \left[S_Z(\omega)H(\omega)\right]_+ \tag{12.208}$$

where $[\,\cdot\,]_+$ corresponds to the causal component of FT

$$f(t) = \eta(t)f(t) + (1-\eta(t))f(t) \coloneqq [F(\omega)]_+ + [F(\omega)]_- \tag{12.209}$$

$$[F(\omega)]_+ = \frac{1}{\sqrt{2\pi}} \int_0^\infty f(t)e^{-i\omega t}\,\mathrm{d}t \tag{12.210}$$

with factor decomposition $S_Z(\omega) = S_Z^+(\omega)S_Z^-(\omega)$, where S_Z^+ is a causal function[12], we have solution

$$H(\omega) = \frac{1}{S_Z^+}\left[\frac{e^{i\omega T}S_{XZ}}{S_Z^-}\right]_+ \tag{12.213}$$

Notes on causal function:

- Convolution is causal invariant:

$$(\eta f * \eta g)(t) = \int_0^\infty f(\tau)g(t-\tau)\,\mathrm{d}\tau = 0 \text{ if } t < 0 \tag{12.214}$$

Section 12.4 Miscellanea

12.4.1 Minimum Mean Squared Estimator

Motivation: Here's a signal transmission process in which source is $X \sim f_X$ and observation is $\vec{Z} \sim f_Z$, we need to find a (theoretically best) information process function $g(\,\cdot\,)$ such that we can reproduce X with $g(\vec{Z}) \in \mathscr{F}$

[11] Derivation uses Parseval's Theorem equation 12.243 ~ page 334.

[12] An illustration: since convolution function is causal invariant, then

$$e^H = \sum_{i=0}^\infty \frac{H^i}{i!} \coloneqq \sum_{i=0}^\infty \frac{(*h)^i}{i!} \tag{12.211}$$

is also causal invariant, i.e. $H = [H]_+ \Rightarrow e^H = [e^H]_+$, then we could have

$$S = S^+S^- = e^{\varsigma^+ + \varsigma^-} = e^{[s]_+ + [s]_-} \tag{12.212}$$

with minimum 'error' (Note that X and $\vec{Z}$ can be dependent)., i.e.

$$\hat{g} = \underset{g(\cdot)\in\mathscr{F}}{\arg\min}\, \mathbb{E}\left[(X - g(\vec{Z}))^2\right] \tag{12.215}$$

which is the **Minimum Mean Squared Error Estimator** (MMSE). [13]

□ **General Solution to MMSE**

The solution to MMSE is that

$$\hat{g}(\cdot)\, s.t. \begin{cases} \hat{g}(\vec{Z}) \in \mathscr{F}(Z) \\ e := X - \hat{g}(\vec{Z}) \perp h(\vec{Z}), \quad \forall h(\vec{Z}) \in \mathscr{F}(Z) \end{cases} \tag{12.217}$$

here $\perp$ in the sense that $\imath \perp \jmath \Leftrightarrow \mathbb{E}[\imath\jmath] = 0$

Denote $\mathscr{F}(Z) \ni g(Z) = \hat{g}(Z) + ch(Z),\ h(Z) \in \mathscr{F}(Z)$, then

$$\mathbb{E}\left[(X - g(Z))^2\right] = \mathbb{E}\left[(X - \hat{g}(Z) - ch(Z))^2\right] \tag{12.218}$$

$$= \mathbb{E}\left[(X - \hat{g}(Z))^2\right] - 2c\mathbb{E}\left[(X - \hat{g}(Z))h(Z)\right] + c^2\mathbb{E}\left[h(Z)^2\right] \tag{12.219}$$

- If $X - \hat{g}(\vec{Z}) \perp h(\vec{Z})$: $\mathbb{E}\left[(X - g(Z))^2\right] = \mathbb{E}\left[(X - \hat{g}(Z))^2\right] + c^2\mathbb{E}\left[h(Z)^2\right] \geq \mathbb{E}\left[(X - \hat{g}(Z))^2\right]$
- If $X - \hat{g}(\vec{Z}) \not\perp h(\vec{Z})$, then for $|c|$ small enough we could have $\mathbb{E}\left[(X - g(Z))^2\right] < \mathbb{E}\left[(X - \hat{g}(Z))^2\right]$.

which gives that the above condition is necessary and sufficient.

The above expression is similar to the projection operator onto space $\mathscr{F}$, i.e.

$$\hat{g}(\cdot) = \Pi_{\mathscr{F}(\cdot)}(X), \quad \begin{cases} \Pi_{\mathscr{F}(\cdot)}(X) \in \mathscr{F} \\ X - \Pi_{\mathscr{F}(\cdot)}(X) \perp \mathscr{F} \end{cases} \tag{12.220}$$

□ **Properties of Projection Operator $\Pi_{\mathcal{V}}$ (where function space $\mathscr{F}$ is a kind of linear space $\mathcal{V}$)**

- Linearity

$$\Pi_{\mathcal{V}}(aX + bY) = a\Pi_{\mathcal{V}}(X) + b\Pi_{\mathcal{V}}(Y) \tag{12.221}$$

- Project within subspace: for $\mathcal{V}_2 \subset \mathcal{V}_1$

$$\Pi_{\mathcal{V}_2}(X) = \Pi_{\mathcal{V}_2}\left(\Pi_{\mathcal{V}_1}(X)\right) \tag{12.222}$$

- Projection onto orthogonal space: for $\mathcal{V}_1 \perp \mathcal{V}_2$

$$\Pi_{\mathcal{V}_1\oplus\mathcal{V}_2}(X) = \Pi_{\mathcal{V}_1}(X) + \Pi_{\mathcal{V}_2}(X) \tag{12.223}$$

[13] **Note**: the function space $\mathscr{F}(\vec{Z})$ (by default) is the arbitrary measurable function space $:= \mathscr{V}(\vec{Z})$, but you can specifically select a proper one, e.g. linear combination of some power function $\mathbb{V}(1, \vec{Z}, \vec{Z}^2) := \{a + bZ + cZ^2\}_{a,b,c\in\mathbb{R}} \subset \mathscr{F}(\vec{Z})$.

I am not quite sure (actually I believe it's wrong lol) but maybe for some commonly used function form, we could view that

$$\mathscr{V}(\vec{Z}) \approx \mathbb{V}(\{\vec{Z}^p\}_{p=-\infty}^{\infty}) \tag{12.216}$$

□ **Important Cases**

- $\mathscr{F}(Z)=\mathscr{V}(Z)$: Solution is

$$\mathbb{E}[X|Z] \tag{12.224}$$

in which

$$\begin{cases}\mathbb{E}[X|Z]\in\mathscr{F}(Z)\\ \mathbb{E}[(X-\mathbb{E}[X|Z])g(Z)]=\mathbb{E}[Xg(Z)]-\mathbb{E}[\mathbb{E}[g(Z)X|Z]]=0\end{cases} \tag{12.225}$$

- $\mathscr{F}(Z)=$ const: Solution is

$$\mathbb{E}[X] \tag{12.226}$$

in which

$$\begin{cases}\mathbb{E}[X]\in\mathcal{R}\\ \mathbb{E}\left[(X-\mathbb{E}[X])\middle|\text{const}\right]=0\end{cases} \tag{12.227}$$

which is also a kind of variance definition:

$$var(X):=\min_{c\in\mathbb{R}}\mathbb{E}\left[(X-c)^2\right] \tag{12.228}$$

- $\mathscr{F}(Z)=\mathbb{V}(1,\vec{Z})$ i.e. linear conbination of $\vec{Z}$ as $a+\vec{Z}'b$. Solution is

$$L(X|\vec{Z}):=\mathbb{E}[X]+cov(X,\vec{Z})var(\vec{Z})^{-1}\left(\vec{Z}-\mathbb{E}\left[\vec{Z}\right]\right) \tag{12.229}$$

in which

$$\begin{cases}\mathbb{E}[X]+cov(X,\vec{Z})var(\vec{Z})^{-1}\left(\vec{Z}-\mathbb{E}\left[\vec{Z}\right]\right)\in\mathbb{V}(1,\vec{Z})\\ \mathbb{E}\left[(X-L(X|\vec{Z}))(a+\vec{Z}'b)\right]=0\end{cases} \tag{12.230}$$

12.4.2 Conditional Independence

Conditional independence : say X and Z are conditionally independent given Y, i.e. X-Y-Z

$$f_{X|YZ}=f_{X|Y}\Leftrightarrow f_{XZ|Y}=f_{X|Y}f_{Z|Y} \tag{12.231}$$

Further if $(X,Y,Z)\sim N(\mu,\Sigma)$ (a joint Gaussian Dist.). Then

$$cov(X,Z)=cov(X,Y)var(Y)^{-1}cov(Y,Z) \tag{12.232}$$

it could be deduced using linar MMSE + innovation sequence of jointly Gaussian

$$cov(Z,X-L(X|Y))=0\Rightarrow cov(X,Z)=cov(X,Y)var(Y)^{-1}cov(Y,Z) \tag{12.233}$$

or use equation 4.71 ~ page 124, in which $X_1=(X,Z)$, $X_2=Y$

$$\Sigma_{X,Z|Y}=\begin{bmatrix}\Sigma_X-\Sigma_{XY}-\Sigma_Y^{-1}\Sigma_{YX} & \Sigma_{XZ}-\Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YZ}\\ \Sigma_{ZX}-\Sigma_{ZY}\Sigma_Y^{-1}\Sigma_{YX} & \Sigma_Z-\Sigma_{ZY}\Sigma_Y^{-1}\Sigma_{YZ}\end{bmatrix}\Rightarrow \Sigma_{XZ}=\Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YZ} \tag{12.234}$$

12.4.3 Fourier Transform and Convolution

□ **Fourier Transform**

Fourier Transform (FT) $g(t) \coloneqq G(\omega)$ is a link between time domain and frequency domain[14]

$$g(t) \coloneqq G(\omega) : \begin{cases} g(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} G(\omega) e^{i\omega t} \, \mathrm{d}\omega \\ G(\omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} g(t) e^{-i\omega t} \, \mathrm{d}t \end{cases} \tag{12.235}$$

Fourier operator is denoted $\mathscr{F}[\cdot]$

$$G = \mathscr{F}[g] \leftrightsquigarrow g = \mathscr{F}^{-1}[G] \tag{12.236}$$

Properties

- Linearity

$$\mathscr{F}[\alpha f + \beta g] = \alpha \mathscr{F}[f] + \beta \mathscr{F}[g] \tag{12.237}$$

- Time shifting / Frequency shifting

$$g(t - \tilde{t}) \coloneqq G(\omega) e^{-i\omega\tilde{t}} \qquad G(\omega - \tilde{\omega}) \coloneqq g(t) e^{i\tilde{\omega}t} \tag{12.238}$$

- Convolution Theorem

$$\mathscr{F}[f * g] = \sqrt{2\pi} FG \tag{12.239}$$

where convolution operator is

$$(f * g)(t) = \int_{\tau} f(\tau) g(t - \tau) \, \mathrm{d}\tau \tag{12.240}$$

- Differentiation

$$\frac{\mathrm{d}^k}{\mathrm{d}t^k} g(t) \coloneqq (i\omega)^k G(\omega) \tag{12.241}$$

- Duality

$$\mathscr{F}[\mathscr{F}[g(t)]] = \frac{1}{2\pi} g(-t) \tag{12.242}$$

- Parseval's Theorem:

$$\int_{\mathbb{R}} f(t) g^{\dagger}(t) \, \mathrm{d}t = \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} F(\omega_1) e^{i\omega_1 t} \, \mathrm{d}\omega_1 \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} G^{\dagger}(\omega_2) e^{-i\omega_2 t} \, \mathrm{d}\omega_2 \, \mathrm{d}t \tag{12.243}$$

$$= \int_{\omega_1} \int_{\omega_2} F(\omega_1) G^{\dagger}(\omega_2) \int_t \frac{1}{2\pi} e^{i(\omega_1 - \omega_2)t} \, \mathrm{d}t \, \mathrm{d}\omega_1 \, \mathrm{d}\omega \tag{12.244}$$

$$= \int_{\omega} F(\omega) G^{\dagger}(\omega) \, \mathrm{d}\omega \tag{12.245}$$

[14] For symmetry consideration, I usually use $\frac{1}{\sqrt{2\pi}}$ in both transform and inversed.

(if the integration above can be properly defined.)

A physical intuition is the energy convervation in both time domain and spetrum domain (which is also a reason I prefer the $\frac{1}{\sqrt{2\pi}}$ transform —- no extra coefficient in this energy conservation)

$$\int_{\mathbb{R}} |f(t)|^2 \, \mathrm{d}t = \int_{\omega} |F(\omega)|^2 \, \mathrm{d}\omega \tag{12.246}$$

Instances

- Dirac δ function for unit impulse at t_0

$$\int_{-\infty}^{s} \delta(t-t_0) \, \mathrm{d}t = \eta(s-t_0) = \begin{cases} 0, & s < t_0 \\ 1, & s > t_0 \end{cases} \tag{12.247}$$

some commonly used definition of δ function:

$$\delta(t) = \lim_{\Delta \to 0} \frac{1}{\Delta} \mathbb{I}_{-\Delta/2<t<\Delta/2} \tag{12.248}$$

$$\delta(t) = \lim_{\Delta \to 0} \frac{1}{\pi\Delta} \mathrm{sinc}(\Delta t) \tag{12.249}$$

Integration of Dirac δ yields

$$\int_{\mathbb{R}} \delta(t-t_0) f(t) \, \mathrm{d}t = f(t_0) \tag{12.250}$$

FT of Dirac δ is harmonic wave

$$\delta(t-t_0) \fallingdotseq \frac{1}{\sqrt{2\pi}} e^{-i\omega t_0}, \quad e^{i\omega_0 t} \fallingdotseq \sqrt{2\pi}\delta(\omega-\omega_0) \tag{12.251}$$

- FT for periodic function $g(t) = g(t+T)$ is Fourier series

$$\begin{cases} g(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{i\frac{2\pi n}{T}t} \\ c_n = \dfrac{1}{T} \int_{\text{one period}} f(t) e^{-i\frac{2\pi n}{T}t} \, \mathrm{d}t \end{cases} \tag{12.252}$$

where c_0 is the DC component of the function.

- Discrete Time FT: discrete time case can be viewed as a sample of frequency T from continuous case

$$\begin{cases} g_T(t) = \sum_{n=-\infty}^{\infty} g(t)\delta(t-nT) \\ \mathscr{F}\left[g_T\right](\omega) = \dfrac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{\infty} g(nT) e^{-i\omega nT} \end{cases} \tag{12.253}$$

which dual with FT for periodic function.

Chapter. XIII　贝叶斯统计导论部分

Instructor: Wanlu Deng

Section 13.1　Calculation Preparation

Some useful calculation results / tricks are listed in this part, including r.v. distribution / integration, etc.

13.1.1　Calculation

- Gamma integral

$$\Gamma(z) \equiv \int_0^\infty t^{z-1} e^{-t}\,\mathrm{d}t, \qquad \mathrm{Re}\, t > 0 \tag{13.1}$$

  – scaling λ form

$$\int_0^\infty t^{z-1} e^{-\lambda t}\,\mathrm{d}t = \frac{\Gamma(z)}{\lambda^z} \tag{13.2}$$

  – Gaussian integral

$$\int_0^\infty t^p e^{-\alpha t^2}\,\mathrm{d}t = \frac{\alpha^{-\frac{p+1}{2}}}{2}\Gamma(\frac{p+1}{2}) \tag{13.3}$$

  with $\alpha = \dfrac{1}{2\sigma^2}$ gives the normalization const of Gaussian distribution

$$\begin{cases} \int_{\mathbb{R}} e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t = 2 \times \dfrac{\sqrt{2}\sigma}{2}\Gamma(\dfrac{1}{2}) = \sqrt{2\pi}\sigma \\ \int_{\mathbb{R}} t^2 \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t = \dfrac{2}{\sqrt{2\pi}\sigma} \times \dfrac{(2\sigma^2)^{3/2}}{2}\Gamma(\dfrac{3}{2}) = \sigma^2 \end{cases} \tag{13.4}$$

  – complementary formula

$$\Gamma(z)\Gamma(1-z) = \frac{\pi}{\sin \pi z} \tag{13.5}$$

  – special values

$$\Gamma(1) = 1, \qquad \Gamma(\frac{1}{2}) = \sqrt{\pi} \tag{13.6}$$

  – recursion

$$\Gamma(z+1) = z\Gamma(z) \Rightarrow \begin{cases} \Gamma(n) = (n-1)!, & n \in \mathbb{N}^+ \\ \Gamma(n+\dfrac{1}{2}) = \sqrt{\pi}\dfrac{(2n-1)!!}{2^n}, & n \in \mathbb{N}^+ \end{cases} \tag{13.7}$$

  (factorial expression)

- Beta Integral

$$B(p,q) \equiv \int_0^1 t^{p-1}(1-t)^{q-1}\,\mathrm{d}t, \qquad \mathrm{Re}\,p,\ \mathrm{Re}q\ > 0 \tag{13.8}$$

  – symmetry

$$B(p,q) = B(q,p) \tag{13.9}$$

  – Gamma function expression

$$B(p,q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)},\ \text{for integer}\ p,q\ = \frac{(p-1)!(q-1)!}{(p+q-1)!} \tag{13.10}$$

13.1.2 Useful Distribution Recap

- (p-dimensional) Normal distribution $Z \sim N(\mu, \Sigma)$

$$f_Z(z) = (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\left[-\frac{1}{2}(z-\mu)'\Sigma^{-1}(z-\mu)\right], \qquad \text{with}\ \begin{cases}\mathbb{E}\left[Z\right] = \mu \\ var(z) = \Sigma\end{cases} \tag{13.11}$$

- Gamma distribution $X \sim \Gamma(\alpha, \lambda)$

$$f_X(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\lambda x}, \qquad \text{with}\ \begin{cases}\mathbb{E}\left[X\right] = \dfrac{\alpha}{\lambda} \\ var(X) = \dfrac{\alpha}{\lambda^2}\end{cases} \tag{13.12}$$

  – summation

$$\Gamma(\sum_i \alpha_i, \lambda) = \sum_i \Gamma(\alpha_i, \lambda), \quad \Gamma(1,\lambda) = \varepsilon(\lambda) \tag{13.13}$$

- χ^2-distribution $X \sim \chi^2_n$

$$f_X(x) = \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}x^{\frac{n}{2}-1}e^{-\frac{x}{2}}, \qquad \text{with}\ \begin{cases}\mathbb{E}\left[\chi^2_n\right] = n \\ var(\chi^2_n) = 2n\end{cases} \tag{13.14}$$

  – relation to Γ

$$\Gamma(\frac{n}{2}, \frac{1}{2}) = \chi^2_n \tag{13.15}$$

- Beta distribution $X \sim \text{Beta}(\alpha, \beta)$

$$f_X(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\text{Beta}(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}, \qquad \text{with}\ \begin{cases}\mathbb{E}\left[X\right] = \dfrac{\alpha}{\alpha+\beta} \\ var(X) = \dfrac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}\end{cases} \tag{13.16}$$

- t-distribution $T \sim t_\nu$

$$f_T(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} = \frac{1}{\sqrt{\nu}\text{Beta}(\frac{\nu}{2},\frac{1}{2})}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \qquad \text{with } \begin{cases} \mathbb{E}[X]=0 \\ var(X)=\dfrac{\nu}{\nu-2} \end{cases} \tag{13.17}$$

- Wishart distribution: a multi-dim version of χ^2. If $Z_1,\dots,Z_m$ i.i.d. $\sim N_p(0,\Lambda)$, then

$$W_p = \sum_{i=1}^{m} Z_i Z_i' \sim \text{Wishart}_m(\Lambda) \tag{13.18}$$

expression see section 4.2.3 ~ page 125. Kernel term

$$f_W(w;p,m,\Lambda) \propto |w|^{\frac{m-p-1}{2}} \exp\left[-\frac{1}{2}tr(\Lambda^{-1}w)\right], \qquad w \in \mathbb{R}^{p\times p} \tag{13.19}$$

- Dirichlet distribution : A multi-parameter version of Beta distribution $(x_1,x_2,\dots,x_J) \sim \text{Dirichlet}(\alpha_1,\alpha_2,\dots,\alpha_J)$, w.r.t. $\sum_{j=1}^J x_j = 1$

$$f_X(x_1,x_2,\dots,x_J) = \frac{\Gamma\left(\sum_{j=1}^J \alpha_j\right)}{\prod_{j=1}^J \Gamma(\alpha_j)} \prod_{j=1}^J x_i^{\alpha_j-1}, \qquad \sum_{j=1}^J x_j = 1 \tag{13.20}$$

Beta distribution is the case of $J=2$.

Δ Inverse distribution. General formula for Inv-f_X

$$X \sim f_X(x), \quad Z=\frac{1}{X}, \quad f_Z(z) = \frac{1}{z^2} f_X(\frac{1}{z}) \tag{13.21}$$

Instances:

– Inv-$\Gamma(\alpha,\lambda) = \dfrac{1}{\Gamma(\alpha,\lambda)}$

$$f_Z(z) = \frac{\lambda^\alpha}{\Gamma(\alpha)} z^{-\alpha-1} e^{-\frac{\lambda}{z}}, \qquad \text{with } \begin{cases} \mathbb{E}[Z] = \dfrac{\lambda}{\alpha-1} \\ var(Z) = \dfrac{\lambda^2}{(\alpha-1)^2(\alpha-2)} \end{cases} \tag{13.22}$$

– (scaled) Inv-$\chi^2(n,s^2) = \dfrac{ns^2}{\chi_n^2} = \text{Inv-}\Gamma(\dfrac{n}{2},\dfrac{ns^2}{2})$

$$f_Z(z) = \frac{n^{\frac{n}{2}}}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} z^{-\frac{n}{2}-1} e^{-\frac{ns^2}{2z}}, \qquad \text{with } \begin{cases} \mathbb{E}[Z] = \dfrac{n}{n-2}s^2 \\ var(Z) = \dfrac{2n^2s^4}{(n-2)^2(n-4)} \end{cases} \tag{13.23}$$

– $Z \sim \text{Inv-Wishart}(\Lambda) \Leftrightarrow Z^{-1} \sim \text{Wishart}(\Lambda)$[1]

$$f_Z(z) = f_W(z^{-1};p,m,\Lambda)\,|z|^{-(p+1)} \propto |z|^{-\frac{m+p+1}{2}} \exp\left[-\frac{1}{2}tr(\Lambda^{-1}z^{-1})\right] \tag{13.24}$$

[2]

[1] In R. and Python., functional input form is Inv-Wishart(Λ^{-1}) = (Wishart(Λ))$^{-1}$

[2] Proof note for Jacobian $\left|\dfrac{\partial A^{-1}}{\partial A}\right|$: For an *arbitrary* matrix $\left|\dfrac{\partial A^{-1}}{\partial A}\right| = |A|^{-2\dim_A}$

Section 13.2 Elements in Bayesian Model

Key idea: Bayesian rule

$$\mathbb{P}(X|Y) = \frac{\mathbb{P}(Y|X)\,\mathbb{P}(X)}{\mathbb{P}(Y)} = \frac{\mathbb{P}(Y|X)\,\mathbb{P}(X)}{\int_{\Omega_X}\mathbb{P}(Y|X)\,\mathbb{P}(X)\,\mathrm{d}X} \tag{13.32}$$

In both Bayesian & Frequentist statistics, we care about updating our 'belief' on **parameter**.

$$\underbrace{\mathbb{P}(\theta|y)}_{\text{posterior}} = \frac{\mathbb{P}(\theta)\,\mathbb{P}(y|\theta)}{\mathbb{P}(y)} \propto \underbrace{\mathbb{P}(\theta)}_{\text{prior}}\ \underbrace{\mathbb{P}(y|\theta)}_{\text{data likelihood}} \tag{13.33}$$

13.2.1 Prior Selection

Selection of prior distribution $p(\theta)$ could greatly influence posterior because it provides prior information about the parameter. The selection could be flexible, here are some frequently-used approaches

- Conjugate Prior: defined for the case that (conjugate) prior and posterior belong to the same distribution

1. First construct mapping $\mathbb{R}^{p\times p}\mapsto\mathbb{R}^{p^2}$ e.g. by $\vec{a}_{I\equiv ip+j}=A_{ij}$
2. Differentiation: where e_i is the unit vector on the i^{th} coord.

$$\frac{\partial A^{-1}_{ij}}{\partial A} = \frac{\partial q_i' A^{-1} q_j}{\partial A} = \frac{\partial tr(A^{-1}q_j q_i')}{\partial A} \tag{13.25}$$

$$= -A^{-1}q_i q_j' A^{-1} = A^{-1}_{:i}A^{-1}_{j:} \tag{13.26}$$

$$\Rightarrow \frac{\partial A^{-1}_{ij}}{\partial A_{kl}} = -A^{-1}_{ki}A^{-1}_{jl} \tag{13.27}$$

$$\Rightarrow \frac{\partial \vec{a}^{-1}_I}{\partial \vec{a}_J} = -\left((A')^{-1}\otimes A^{-1}\right)_{IJ} \tag{13.28}$$

where $\otimes$ is Kronecker product $\mathbb{R}^{u\times u}\times\mathbb{R}^{v\times v}\mapsto\mathbb{R}^{uv\times uv}$ for

$$(\underset{u\times u}{U}\otimes\underset{v\times v}{V})_{iu+j,kv+l} = U_{ik}V_{jl} \tag{13.29}$$

which has property

$$|U\otimes V| = |U|^v\,|V|^u \tag{13.30}$$

3. Determinant for Kronecker product

$$\left\|\frac{\partial A^{-1}}{\partial A}\right\| \equiv \left\|\frac{\partial \vec{a}^{-1}}{\partial \vec{a}}\right\| = \left\|-(A')^{-1}\otimes A^{-1}\right\| = |A|^{-2p} \tag{13.31}$$

Further here for Wishart distribution, we have a constraint for *positive definition*. The constraint causes a 'degree of freedom reduction' so $\left|\frac{\partial A^{-1}}{\partial A}\right| = |A|^{-(\dim_A+1)}$.

family. [3]

$$p(\theta|y) \propto p(y|\theta)p(\theta) \in \mathscr{F}(\Theta),\ \forall p(y|\theta) \in \mathscr{F}(Y|\theta)\ \&\ p(\theta) \in \mathscr{F}(\Theta) \tag{13.36}$$

Instances see

- Binomial Model
- Poisson Model
- Exponential Model
- UniNormal with known variance Model
- UniNormal with known mean Model
- Multinomial Model
- UniNormal Model
- MultiNormal Model

- Non-informative Prior: Jeffrey's Prior. Idea is to choose a distribution 'covariant' with parameterization [4]. i.e. under different parameterization, say $\theta \leftrightharpoons \phi$, we should follow the same deduction method to get corresponding prior $p_\theta \leftrightharpoons p_\phi$ that could covariant with parameter transform

$$p_\theta(\theta) = p_\phi(\phi(\theta)) \left|\frac{\partial \phi(\theta)}{\partial \theta}\right| \tag{13.37}$$

Notice that (sqrt) Fisher Information $|I(\theta)|^{1/2}$ meets such requirement, which gives Jeffrey's Prior.

$$L(y|\theta) = L(y|\phi) \Rightarrow |I(\theta)| = \left|\mathbb{E}_y\left[\frac{\partial \log L(y|\theta)}{\partial \theta}\frac{\partial \log L(y|\theta)}{\partial \theta'}\right]\right| \tag{13.38}$$

$$= \left|\mathbb{E}_y\left[\frac{\partial L(y|\phi)}{\partial \phi}\frac{\partial L(y|\phi)}{\partial \phi'}\right]\right| \left|\frac{\partial \phi}{\partial \theta}\right|^2 \tag{13.39}$$

$$= |I(\phi)| \left|\frac{\partial \phi}{\partial \theta}\right|^2 \tag{13.40}$$

So Jeffrey's prior is expressed

$$p_{\text{Jeffrey}}(\theta) \propto |I(\theta)|^{1/2}$$

Note: Usually Jeffrey's is an improper prior (diverge).

- Other suitable prior that reflects our knowledge.

[3] Concept of distribution family see section 2.1 ~ page 37. In this section we use notation

$$f(x;\theta) \in \mathscr{F}(\Theta) \tag{13.34}$$

to express the distribution family generated on parameter space $(\alpha, \lambda) \in A \times \Lambda$, e.g. family of Γ distribution

$$\mathscr{F}_\Gamma(A, \Lambda) \tag{13.35}$$

[4] 此处 covariant 一词类似于广相中的"协变"义。

13.2.2 Posterior Distribution

After wisely select the prior, we can have it combined with data and get formula for posterior

$$p(\theta|y) \propto p(\theta)p(y|\theta) \tag{13.41}$$

which is a function of θ, so we just need to take care of θ-terms and normalization condition would help fixed the constant.

□ **Calculation Trick**

- Identify the distribution with the variable related term.

  Example: Obtain the predictive distribution of Poisson model with conjugate Gamma distribution

$$p(y|\theta) = \prod_{i=1}^{N} \frac{\theta^{y_i}}{y_i!} e^{-\theta} \tag{13.42}$$

$$p(\theta) \sim \Gamma(\alpha, \beta) \tag{13.43}$$

$$p(\tilde{y}|y) \propto \int \frac{\theta^{\tilde{y}}}{\tilde{y}!} e^{-\theta} \theta^{\alpha-1} e^{-\beta\theta} \theta^{N\bar{y}} e^{-N\theta} \, \mathrm{d}\theta \tag{13.44}$$

$$= \frac{1}{\tilde{y}!} \int \theta^{\alpha+N\bar{y}+\tilde{y}-1} e^{-(\beta+N+1)\theta} \, \mathrm{d}\theta \tag{13.45}$$

$$= \frac{1}{\tilde{y}!} \frac{\Gamma(\alpha + N\bar{y} + \tilde{y})}{(\beta+N+1)^{\alpha+N\bar{y}+\tilde{y}}} \tag{13.46}$$

$$\propto \binom{\alpha + N\bar{y} + \tilde{y} - 1}{\tilde{y}} \left(\frac{\beta+N}{\beta+N+1}\right)^{\alpha+N\bar{y}} \left(\frac{1}{\beta+N+1}\right)^{\tilde{y}} \tag{13.47}$$

$$\sim \text{Neg-Binom}(\alpha + N\bar{y}, \beta + N) \tag{13.48}$$

- Get marginal posterior with Conditional probability formula

$$p(\beta|y) = \frac{p(\alpha, \beta|y)}{p(\alpha|\beta, y)} \tag{13.49}$$

  in which L.H.S. is free from α, so R.H.S. should be invariant of α, i.e. take the same value for any α value. We can simplify the calculation by taking some convenient values.

  Example: Marginal posterior distribution in Normal model.

$$p(\mu, \sigma^2|y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \sim N\text{-Inv-}\chi^2(\mu_n, \sigma_n^2/\kappa_n; \nu_n, \sigma_n^2) \tag{13.50}$$

$$p(\sigma^2|\mu; y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \sim \text{Inv-}\chi^2(\nu_n + 1, \frac{\nu_0^2\sigma_0^2 + \kappa_0(\mu-\mu_0)^2 + N\text{MSE}}{\nu_n + 1}) \tag{13.51}$$

$$p(\mu|y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) = \frac{p(\mu, \sigma^2|y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2)}{p(\sigma^2|\mu; y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2)} \tag{13.52}$$

$$(\text{take } \sigma^2 = 1) \quad \propto (\nu_0^2\sigma_0^2 + \kappa_0(\mu-\mu_0)^2 + N\text{MSE})^{-(\nu_n+1)/2} \tag{13.53}$$

$$= \left(\nu_0^2\sigma_0^2 + (N-1)s^2 + \frac{\kappa_0 N}{\kappa_0 + N}(\bar{y}-\mu_0)^2 + \kappa_n(\mu-\mu_n)^2\right)^{-(\nu_n+1)/2} \tag{13.54}$$

$$\propto \left(1 + \frac{\kappa_n(\mu-\mu_n)^2}{\nu_n\sigma_n^2}\right)^{-(\nu_n+1)/2} \sim t_{\nu_n}(\mu_n, \sigma_n^2/\kappa_n) \tag{13.55}$$

13.2.3 Asymptotics

The Maximum A Posteriori (MAP) describes a point estimation by maximize posterior distribution of parameter, say

$$\hat{\theta} = \arg\max_{\theta} p(\theta|y) = \arg\max_{\theta} p(\theta)p(y|\theta) \tag{13.56}$$

The maximizer has consistency at large sample

$$p(\theta|y) \to \delta(\theta - \theta^*), \quad \text{as } n \to \infty \tag{13.57}$$

The maximizer would further give the asymptotic normal distribution centered around it. Use the taylor series at $\hat{\theta}$:

$$\log p(\theta|y) = \log p(\hat{\theta}|y) + \frac{1}{2}(\theta - \hat{\theta})' \left[\frac{\partial \log p(\theta|y)}{\partial\theta\partial\theta'}\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + o(\theta^2) \tag{13.58}$$

$$\Rightarrow p(\theta|y) \to N(\hat{\theta}, \mathcal{I}(\hat{\theta})^{-1}/n) \tag{13.59}$$

Note: Here $\hat{\theta}$, as is calculated from the data, is considered fixed, while θ is the random one. It's just in contrast to frequentist's version where $\hat{\theta}$ is random while θ is fixed. (The estimator is also different, here maximizes posterior $p(\theta|y)$, frequentists maximize likelihood $p(y|\theta)$)

13.2.4 Predictive Distribution

Generally speaking we are studying the posteior predictive distribution

$$p_{\text{post}}(\tilde{y}) = \mathbb{E}_{\theta|y}\left[p(\tilde{y}|\theta)\right] = \int p(\tilde{y}|\theta)p(\theta|y)\,\mathrm{d}\theta \tag{13.60}$$

Related concept:

- Expected log Prediction Distribution for New Data (elpd). Where the ground truth distribution of $\tilde{y}$ is denoted $\tilde{f}(\tilde{y})$.

$$\text{elpd} \equiv \mathbb{E}_{\tilde{f}}\left[\log \int p(\tilde{y}|\theta)p(\theta|y)\,\mathrm{d}\theta\right] = \int_{\tilde{y}} \log \int_{\theta} p(\tilde{y}|\theta)p(\theta|y)\,\mathrm{d}\theta\ \mathrm{d}\tilde{y} \tag{13.61}$$

At large sample, the $\mathrm{d}\theta$ integration is dominated by $\hat{\theta} = \arg\max_{\theta} p(\theta|y)$, yielding[5]

$$\text{elpd} = \int_{\tilde{y}} \log \int_{\theta} p(\tilde{y}|\theta)p(\theta|y)\,\mathrm{d}\theta\ \tilde{f}(\tilde{y})\,\mathrm{d}\tilde{y} \underset{N\to\infty}{\approx} \int_{\tilde{y}} \tilde{f}(\tilde{y}) \log p(\tilde{y}|\hat{\theta})\,\mathrm{d}\tilde{y} \equiv \text{elpd}_{\hat{\theta}} \tag{13.62}$$

13.2.5 Model Checking and Comparison

□ Posterior Predictive Checking

The idea is similar to construct p-value in hypothesis testing. But now we are using the posterior predictive distribution to check the model fit.

[5] The integration method is called 'steepest descent'.

Posterior distribution given by probability model $\mathcal{M}$ denoted $p_{\mathcal{M}}(\theta|y)$. We could further properly define a test statistic $T(y)$, and the posterior predictive distribution of $T(y)$ is

$$p_{\mathcal{M}}(T(\tilde{y})|y) = \int p(T(\tilde{y})|\theta)p_{\mathcal{M}}(\theta|y)\,\mathrm{d}\theta \tag{13.63}$$

which could be easily simulated by

1. Generate $\theta^{(i)} \sim p_{\mathcal{M}}(\theta|y), i = 1, 2, \ldots, S$;
2. Generate $\tilde{y}^{(i)} \sim p(\tilde{y}|\theta^{(i)})$;
3. Compute $T(\tilde{y}^{(i)})$ to form $\hat{p}_{\mathcal{M}}(T|y)$;
4. By calculating the p-value corresponding to our data $T_0 = T(y)$, i.e.

$$p = \mathbb{P}_{\mathcal{M}}\left[T(\tilde{y}) \geqslant T_0\right] = \int \mathbb{I}_{\{T(\tilde{y})\geqslant T_0\}} p(\tilde{y}|\theta)p_{\mathcal{M}}(\theta|y)\,\mathrm{d}\theta \tag{13.64}$$

$$\hat{p} = \frac{\sum_{i=1}^{S} \mathbb{I}_{T(\tilde{y}^{(i)})\geq T_0}}{S} \tag{13.65}$$

and check, say $\hat{p} \leq 0.05$ to rejct the model.

Note: the test statistics $T(y)$ should be chosen carefully.

□ **Model Performance Measure**

Section 13.3 Simulation

In Bayesian inference, the key target is posterior distribution

$$p(\theta|y) \propto p(\theta)p(y|\theta) \tag{13.66}$$

which can be

- Intractable: ugly kernel term.
- High-dimentional: multi-dim parameter sapce Θ.
- Unnormalized: with an unknown normalize constant $\dfrac{1}{p(y)}$

so usually a closed form is not accessable. Simulation is needed to carry out further inference.

13.3.1 Random Number Generation and Simulation

Basic knowledge about simulation methods were covered in section 5.6 ~ page 185. Here are some topic contents:

- Linear Congruential Method for $U(0, 1)$. Other complicated distribution starts from uniform distributed r.v.
- Quantile Method/Inverse Transform Method: Use inverse CDF to obtain r.v.

$$X_i = F_X^{-1}(U_i) \sim f_X \tag{13.67}$$

- Acceptance-Rejection Sampling: Suitable for intractable CDF or high-dimensional cases.
- MCMC: Deal with high-dimentional case or unnormalized distribution.
- Importance Sampling Estimator: Quick way to obtain some observable.

In the following part I would briefly recap their basic idea and give some examples. Some improved / modified version of these algorithms would be introduced, too.

The sampling target in this section is usually some posterior $p(\theta|y)$, or more specifically usually an unnormalized one $\tilde{p}(\theta|y) \propto p(\theta|y)$. **For simplicity, I would just use $p(\cdot)$ and $\tilde{p}(\cdot)$ for normalized and unnormalized distribution, respectively.**

13.3.2 Inverse Transform Method

If a closed form of (inversed) CDF could be obtained, then we could use the inverse transform method to generate r.v. from the target distribution as

$$X_i = F^{-1}(U_i) \sim F, \quad U_i \sim U(0,1) \tag{13.68}$$

Algorithm *Inverse Transform Method*

1. Generate $U_i \sim U(0,1), i = 1, 2, \dots, n$;
2. Compute $X_i = F^{-1}(U_i), i = 1, 2, \dots, n$.

13.3.3 Acceptance-Rejection Sampling

Usually the distribution might be high-dim $\theta \in \Theta$ / unnormalized $\tilde{p}$ / intractable CDF F, so we could not use the inverse transform method.

The idea of acceptance-rejection sampling is to find a **proposal distribution** $g(\theta)$, which is easy to sample from, and a **constant** $\tilde{c}$, such that we have

$$\tilde{p}(\theta) \leqslant \tilde{c} g(\theta) \tag{13.69}$$

then we could generate $\theta_i \sim g(\theta)$, and accept it with probability **acceptance ratio** $\alpha_i = \dfrac{\tilde{p}(\theta_i)}{\tilde{c} g(\theta_i)}$.

$$p(\theta_i|\text{accept}) = \frac{p(\text{accept}|\theta_i)p(\theta_i)}{p(\text{accept})} = \frac{\frac{\tilde{p}(\theta_i)}{\tilde{c}g(\theta_i)} \cdot g(\theta_i)}{\int \frac{\tilde{p}(\vartheta)}{\tilde{c}g(\vartheta)} \cdot g(\vartheta)\, \mathrm{d}\vartheta} \tag{13.70}$$

$$= \frac{\tilde{p}(\theta_i)}{\int \tilde{p}(\vartheta)\, \mathrm{d}\vartheta} = p(\theta_i|y) \tag{13.71}$$

Algorithm *Acceptance-Rejection Sampling*

1. Set the proposal distribution $g(\theta)$ and constant $\tilde{c}$;

2. Generate $\theta_i \sim g(\theta)$, $i = 1, 2, \ldots, n$;
3. Compute $\alpha_i = \dfrac{\tilde{p}(\theta_i)}{\tilde{c}g(\theta_i)}$, $i = 1, 2, \ldots, n$;
4. Accept θ_i with probability α_i, $i = 1, 2, \ldots, n$. This step is done by:
   (a) Generate $U_i \sim U(0, 1)$, $i = 1, 2, \ldots, n$;
   (b) Accept θ_i if $U_i \leqslant \alpha_i$, $i = 1, 2, \ldots, n$.
5. Use the accepted sequence $\{\theta_i\}_{i:\text{accepted}}$ as the r.v. sequence $\sim p(\theta)$

13.3.4 Importance Sampling Estimator and Importance Resampling

Motivation: Through posteriori sampling we finally still care about some statistics, say $h(\theta)$. So directly obtaining an estimator of, say $\mathbb{E}_{\theta\sim p(\theta|y)}[h(\theta)]$ would also be acceptable.

The idea is borrowed from mean value method of numerical integration. Suppose we want to compute $\mathbb{E}_{\theta\sim p(\theta|y)}[h(\theta)]$, but we could only obtain an easily sampled $g(\theta)$, then we could write it as

$$\mathbb{E}_{\theta\sim p(\theta|y)}[h(\theta)] = \int h(\theta)p(\theta|y)\,\mathrm{d}\theta \tag{13.72}$$

$$= \int h(\theta)\frac{p(\theta|y)}{g(\theta)}g(\theta)\,\mathrm{d}\theta \tag{13.73}$$

$$= \mathbb{E}_{\theta\sim g(\theta)}\left[h(\theta)\frac{p(\theta|y)}{g(\theta)}\right] := \mathbb{E}_{\theta\sim g(\theta)}[h(\theta)w(\theta)] \tag{13.74}$$

or in the case of unnormalized distribution $\tilde{p}(\theta|y)$:

$$\mathbb{E}_{\theta\sim p(\theta|y)}[h(\theta)] = \int h(\theta)p(\theta|y)\,\mathrm{d}\theta \tag{13.75}$$

$$= \frac{\int h(\theta)\dfrac{\tilde{p}(\theta|y)}{g(\theta)}g(\theta)\,\mathrm{d}\theta}{\int \dfrac{\tilde{p}(\theta|y)}{g(\theta)}g(\theta)\,\mathrm{d}\theta} \tag{13.76}$$

$$= \mathbb{E}_{\theta\sim g(\theta)}[h(\theta)\tilde{w}(\theta)] \Big/ \mathbb{E}_{\theta\sim g(\theta)}[\tilde{w}(\theta)] \tag{13.77}$$

Estimator:

$$\begin{cases} \hat{h} = \sum_{i=1}^n h(\theta_i)w(\theta_i) & \text{normalized} \\ \hat{h} = \dfrac{\sum_{i=1}^n h(\theta_i)\tilde{w}(\theta_i)}{\sum_{i=1}^n \tilde{w}(\theta_i)}, & \text{unnormalized} \end{cases} \tag{13.78}$$

Effective sample size (Number of independent sample unit to get equivalent precision):

$$n_{\text{effect}} = n\frac{var(\text{estimator with perfect importance})}{var(\hat{h})} \approx \frac{n}{\mathbb{E}[w(\theta)^2]} \approx \begin{cases} \dfrac{n^2}{\sum_{i=1}^n w(\theta_i)^2}, & \text{normalized} \\ \dfrac{(\sum_{i=1}^n \tilde{w}(\theta_i))^2}{\sum_{i=1}^n \tilde{w}(\theta_i)^2}, & \text{unnormalized} \end{cases} \tag{13.79}$$

Algorithm *Importance Sampling*

1. Set the proposal distribution $g(\theta)$;
2. Generate $\theta_i \sim g(\theta)$, $i = 1, 2, \dots, n$;
3. Compute importance $w(\theta_i) = \dfrac{p(\theta_i|y)}{g(\theta_i)}$, $i = 1, 2, \dots, n$;
4. Compute $\hat{h} = \sum_{i=1}^{n} h(\theta_i) w(\theta_i)$.

□ **Importance Resampling**

Importance Sampling could also be used to obtain random sample by 'resampling' from the proposal distribution with weight $w(\cdot)$.

Algorithm *Importance Resampling*

1. Set the proposal distribution $g(\theta)$;
2. Generate $\theta_i \sim g(\theta)$, $i = 1, 2, \dots, N$;
3. Compute $w(\theta_i) = \dfrac{p(\theta_i|y)}{g(\theta_i)}$, $i = 1, 2, \dots, N$;
4. Resample a subset of size $n \ll N$: θ_i with probability $\dfrac{w(\theta_i)}{\sum_{i=1}^{N} w(\theta_i)}$, as the r.v. sequence $\sim p(\theta|y)$.

13.3.5 MCMC

Theory of MCMC see section 12.1.2 ~ page 310. Markov Chain Monte Carlo (MCMC) is useful in sampling high-dim, unnormalized distribution, using the stationary distribution of DTMC.

□ **Metropolis-Hastings Algorithm**

M-H is the basic version of MCMC by inducing a acceptance ratio, together with the proposal distribution $g(\tilde{\theta}|\theta)$ to obtain the transition kernel

$$p_{\theta,\tilde{\theta}} = \underbrace{g(\tilde{\theta}|\theta)}_{\text{propose}} \underbrace{\alpha(\tilde{\theta}|\theta)}_{\text{accept}} = g(\tilde{\theta}, \theta) \min\left\{1, \frac{\tilde{p}(\tilde{\theta}|y) g(\theta|\tilde{\theta})}{\tilde{p}(\theta|y) g(\tilde{\theta}|\theta)}\right\} \tag{13.80}$$

which satisfies the detalied balance condition

$$p(\theta) p_{\theta,\tilde{\theta}} = p(\tilde{\theta}) p_{\tilde{\theta},\theta} \tag{13.81}$$

and with some regular condition we have a stationary distribution

$$p^* = p(\theta|y) \tag{13.82}$$

Algorithm *Metropolis-Hastings Sampling*

1. Set the proposal distribution $g(\tilde{\theta}|\theta)$ and a starting value $\theta^{(0)}$;
2. For $i = 1, 2, \dots, n$:

(a) Generate $\tilde{\theta}$ from $g(\tilde{\theta}|\theta^{(i-1)})$;

(b) Compute the acceptance ratio $\alpha(\tilde{\theta}|\theta^{(i-1)})$;

(c) Generate $u \sim U(0,1)$;

(d) If $u < \alpha(\tilde{\theta}|\theta^{(i-1)})$ (accept), set $\theta^{(i)} = \tilde{\theta}$, otherwise repeat the proposal-accept $p_{\theta^{(i-1)},\tilde{\theta}}$ until accept;

3. Discard the first $\tilde{n}$ samples as burn-in period (i.e. wait until the chain converges to the stationary distribution);
4. Keep the following samples, i.e. $\{\theta^{(j)}\}_{j=\tilde{n}}^{n}$, as the r.v. sequence $\sim p(\theta|y)$.

Note on 'When to converge':

- A test for a good MCMC setting is **Gelman-Rubin** potential scale reduction factor. With the same proposal setting, we run M independent MCMC from various initial value to form $\{\{\theta^{j}\}_{j=\tilde{n}}^{n}\}_{m=1}^{M}$. Then run a 'ANOVA' test to confirm fast convergence

$$\text{PSR Factor} = \sqrt{F} = \sqrt{\frac{\text{MST}}{\text{MSE}}} \tag{13.83}$$

which should be close to 1 if the chain converges well.

□ **Hamiltonian MC / Hybrid MC**

Hamiltonian MC views the sampling variable as 'position', and by introducing a 'momentum' variable, the sampling process is viewed as a physical system, and the revolution of the system helps construct better state transition between states.

Recap of Hamiltonian Dynamics. For a physical system with Hamiltonian $H(q,p)$ in which q for coordinate and p for momentum, the dynamic is

$$\begin{cases} \dfrac{\mathrm{d}q}{\mathrm{d}t} = \dfrac{\partial H}{\partial p} \\ \dfrac{\mathrm{d}p}{\mathrm{d}t} = -\dfrac{\partial H}{\partial q} \end{cases} \tag{13.84}$$

where holds $H(q,p) \equiv \text{const}$.

HMC contains three steps: at current state $(\theta^{(i-1)}, \phi^{(i-1)})$ (for coordinate and momentum, respectively)

1. Proposal: propose a new momentum $\tilde{\phi}$ from $g(\tilde{\phi}|\phi)$;
2. Revolution: Hamiltonian dynamics guided by $H(\theta,\phi) = -\log p(\theta) - \log p(\phi)$ (for some given time T / steps) to obtain a new state $(\theta^{(i-1)}, \tilde{\phi}) \mapsto (\tilde{\theta}_T, \tilde{\phi}_T)$
3. MCMC: to accept / reject the new state

$$\alpha(\tilde{\theta}_T, \tilde{\phi}_T|\theta^{(i-1)}, \phi^{(i-1)}) = \min\left\{1, \frac{p(\tilde{\theta}_T)p(\tilde{\phi}_T)g(\theta^{(i-1)}|\tilde{\theta}_T)g(\phi^{(i-1)}|\tilde{\phi}_T)}{p(\theta^{(i-1)})p(\phi^{(i-1)})g(\tilde{\theta}_T|\theta^{(i-1)})g(\tilde{\phi}_T|\phi^{(i-1)})}\right\} \tag{13.85}$$

$$\text{(Hamiltonian invariant)} = \min\left\{1, \frac{p(\theta^{(i-1)})p(\tilde{\phi})g(\phi^{(i-1)}|\tilde{\phi}_T)}{p(\theta^{(i-1)})p(\phi^{(i-1)})g(\tilde{\phi}_T|\phi^{(i-1)})}\right\} \tag{13.86}$$

$$= \min\left\{1, \frac{p(\tilde{\phi})g(\phi^{(i-1)}|\tilde{\phi}_T)}{p(\phi^{(i-1)})g(\tilde{\phi}_T|\phi^{(i-1)})}\right\} \tag{13.87}$$

---

which gives stationary distribution $p(\theta,\phi) = p(\theta)p(\phi)$, keep the θ component to obtain the target r.v. sequence.

Note: the acceptance ratio α depends on momentum proposal $g(\phi)$ and the 'Kinetic Energy' $\log p(\phi)$, so by wisely choose these parameter we could construct a well-behaved MC.

Algorithm *Hamiltonian MC*

1. Construct the Hamiltonian $H(\theta,\phi) = -\log p(\theta) - \log p(\phi)$ (could both be unnormalized). Set the momentum proposal distribution $g(\tilde{\phi}|\phi)$ and a starting value $(\theta^{(0)},\phi^{(0)})$;
2. For $i = 1,2,\dots,n$:
   (a) Generate $\tilde{\phi}$ from $g(\tilde{\phi}|\phi^{(i-1)})$;
   (b) Run Hamiltonian dynamics for T steps, usually by leapfrog process, to obtain $(\tilde{\theta}_T,\tilde{\phi}_T)$. For $t = 1,2,\dots,T$:
   $$\begin{cases} \phi \leftarrow \phi + \dfrac{\varepsilon}{2}\dfrac{\mathrm{d}\log p(\theta)}{\mathrm{d}\theta} \\ \theta \leftarrow \theta - \varepsilon\dfrac{\mathrm{d}\log p(\phi)}{\mathrm{d}\phi} \\ \phi \leftarrow \phi + \dfrac{\varepsilon}{2}\dfrac{\mathrm{d}\log p(\theta)}{\mathrm{d}\theta} \end{cases} \tag{13.88}$$
   (c) Compute the acceptance ratio
   $$\alpha(\tilde{\theta}_T,\tilde{\phi}_T|\theta^{(i-1)},\phi^{(i-1)}) = \min\left\{1, \frac{p(\tilde{\phi})g(\phi^{(i-1)}|\tilde{\phi}_T)}{p(\phi^{(i-1)})g(\tilde{\phi}_T|\phi^{(i-1)})}\right\} \tag{13.89}$$
   (d) Generate $u \sim U(0,1)$; If $u < \alpha(\tilde{\theta}_T,\tilde{\phi}_T|\theta^{(i-1)},\phi^{(i-1)})$ (accept), set $(\theta^{(i)},\phi^{(i)}) = (\tilde{\theta}_T,\tilde{\phi}_T)$; otherwise (reject), repeat the proposal-accept until accepted;
3. Discard the first $\tilde{n}$ samples as burn-in period (i.e. wait until the chain converges to the stationary distribution);
4. Keep the θ component in the following samples, i.e. $\{\theta^j\}_{j=\tilde{n}}^n$, as the r.v. sequence $\sim p(\theta)$.

13.3.6 Gibbs Sampling

Gibbs sampling is a variant for high-dim case, by sampling from each dimensions based on the marginal distribution (on other dims). Say the sampling target is $\theta = \vec{\theta} = [\theta_1,\dots,\theta_p]$, then Gibbs sampling is

Algorithm *Gibbs Sampling*

1. Set a starting value $\vec{\theta}^{(0)}_{j=1,\dots,p}$;
2. For $i = 1,2,\dots,n$:
   (a) Generate $\theta_1^{(i)}$ from $p(\theta_1|\theta_2^{(i-1)},\theta_3^{(i-1)},\dots,\theta_p^{(i-1)})$;

(b) Generate $\theta_2^{(i)}$ from $p(\theta_2|\theta_1^{(i)}, \theta_3^{(i-1)}, \dots, \theta_p^{(i-1)})$;

(c) $\cdots$

(d) Generate $\theta_p^{(i)}$ from $p(\theta_p|\theta_1^{(i)}, \dots, \theta_{p-1}^{(i)})$;

3. Discard the first $\tilde{n}$ samples as burn-in period (i.e. wait until the chain converges to the stationary distribution);

4. Keep the following samples, i.e. $\{\vec{\theta}^{(j)}\}_{j=\tilde{n}}^{n}$, as the r.v. sequence $\sim p(\theta|y)$.

Comment:

- Decomposition into conditional distribution is ensured by Hammersley Clifford Theorem .[6]

$$p(\theta_1, \theta_2, \dots, \theta_p) = p(\phi_1, \dots, \phi_p) \prod_{j=1}^{p} \frac{p_{\theta_j|\theta_{\wedge j}}(\theta_j|\theta_1, \dots, \theta_{j-1}, \phi_{j+1}, \dots, \phi_p)}{p_{\theta_j|\theta_{\wedge j}}(\phi_j|\theta_1, \dots, \theta_{j-1}, \phi_{j+1}, \dots, \phi_p)} \tag{13.91}$$

which is the hint for a MCMC kernel $K(\cdot, \cdot) = p_{\cdot,\cdot}$ of the Gibbs process, with the target distribution as stationary distribution

$$K(\theta^{(i-1)}, \theta^{(i)}) = \prod_{j=1}^{p} p_{\theta_j|\theta_{\wedge j}}(\theta_j^{(i)}|\theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \dots, \theta_p^{(i-1)}) \tag{13.92}$$

Stationary Distribution:

$$\int p_\theta(\theta^{(i-1)}) K(\theta^{(i-1)}, \theta^{(i)}) \, \mathrm{d}^p\theta^{(i-1)} \tag{13.93}$$

$$= \int_{\theta_p} \cdots \int_{\theta_1} p_\theta(\theta^{(i-1)}) \prod_{j=1}^{p} p_{\theta_j|\theta_{\wedge j}}(\theta_j^{(i)}|\theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \dots, \theta_p^{(i-1)}) \, \mathrm{d}\theta_1^{(i-1)} \cdots \mathrm{d}\theta_p^{(i-1)} \tag{13.94}$$

$$= \int_{\theta_p} \cdots \int_{\theta_2} p_{\theta_{\wedge 1}}(\theta^{(i-1)}) \prod_{j=1}^{p} p_{\theta_j|\theta_{\wedge j}}(\theta_j^{(i)}|\theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \dots, \theta_p^{(i-1)}) \, \mathrm{d}\theta_2^{(i-1)} \cdots \mathrm{d}\theta_p^{(i-1)} \tag{13.95}$$

$$= \int_{\theta_p} \cdots \int_{\theta_2} p_\theta(\theta_1^{(i)}, \theta_{2:p}^{(i-1)}) \prod_{j=2}^{p} p_{\theta_j|\theta_{\wedge j}}(\theta_j^{(i)}|\theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \dots, \theta_p^{(i-1)}) \, \mathrm{d}\theta_2^{(i-1)} \cdots \mathrm{d}\theta_p^{(i-1)} \tag{13.96}$$

$$= \int_{\theta_p} \cdots \int_{\theta_3} p_\theta(\theta_{1:2}^{(i)}, \theta_{3:p}^{(i-1)}) \prod_{j=3}^{p} p_{\theta_j|\theta_{\wedge j}}(\theta_j^{(i)}|\theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \dots, \theta_p^{(i-1)}) \, \mathrm{d}\theta_3^{(i-1)} \cdots \mathrm{d}\theta_p^{(i-1)} \tag{13.97}$$

$$= \dots \tag{13.98}$$

$$= p_\theta(\theta^{(i)}) \tag{13.99}$$

---

[6] □ *Proof:* Use the following iteratively:

$$p(\theta_1, \theta_2, \dots, \theta_p) = p_\theta(\theta_p|\theta_1, \dots, \theta_{p-1}, \phi_p) \frac{p_{\theta_p|\theta_{\wedge p}}(\theta_p|\theta_1, \dots, \theta_{p-1})}{p_{\theta_p|\theta_{\wedge p}}(\phi_p|\theta_1, \dots, \theta_{p-1})} \tag{13.90}$$

□

- The Gibbs sampling could also be considered a special case of M-H, with the proposal distribution, with acceptance ratio $\alpha \equiv 1$.
- Metropolis-within-Gibbs: sometimes we cannot get all normalized marginal distribution $p_{\theta_j|\theta_{\wedge j}}$. We could simply use Gibbs on the known conditional distribution, and use Metropolis-Hastings on the unknown conditional distributions to solve this problem.

  e.g. for a two-dim sampling with $p(\alpha|\beta), \tilde{p}(\beta|\alpha)$, we could

$$\text{Gibbs: } \alpha^{(i)} \sim p(\alpha|\beta^{(i-1)}) \tag{13.100}$$

$$\text{M-H: } \beta^{(i)} \sim \text{MCMC}\beta^{(i-1)} \mapsto \beta^{(i)} \tag{13.101}$$

13.3.7 Mean Field Approximation and Variation Bayesian Inference

Section 13.4 Exactly Sovable Models

Note : In this section for a known/given parameter (i.e. we do **not** consider it an r.v., just a given param), we attach an fixed to label it, e.g. $N(\mu, \sigma^2)$ for the case σ^2 is given, and we only study the distribution of μ.

13.4.1 Binomial Model

Generating process $y_1, \dots, y_N$ i.i.d. $\sim$ Binom(n, p)

$$\text{Distribution:} f(y|p) = \binom{n}{y_i} p^y (1-p)^{n-y} \propto p^y (1-p)^{n-y} \tag{13.102}$$

$$\text{Likelihood:} L(y|p) \propto p^{\sum y_i} (1-p)^{Nn - \sum y_i} = p^{N\bar{y}} (1-p)^{N(n-\bar{y})} \tag{13.103}$$

$$\text{Score:} S(y|p) = \frac{N\bar{y}}{p} - \frac{N(n-\bar{y})}{1-p} \tag{13.104}$$

$$\text{Observed Info:} J(y|p) = \frac{N\bar{y}}{p^2} + \frac{N(n-\bar{y})}{(1-p)^2} \tag{13.105}$$

$$\text{Fisher Info:} I(p) = \frac{N}{p(1-p)} \tag{13.106}$$

- Conjugate prior: Beta distribution $B(\alpha, \beta)$

$$p(p|\alpha, \beta) = \frac{1}{B(\alpha\,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \sim B(\alpha, \beta) \tag{13.107}$$

$$p(p|y, \alpha, \beta) \propto p^{\alpha-1}(1-p)^{\beta-1} p^{N\bar{y}} (1-p)^{N(n-\bar{y})} \sim B(\alpha + N\bar{y}, \beta + N(n-\bar{y})) \tag{13.108}$$

  which suggests that prior distribution $B(\alpha, \beta)$ looks like some 'pre-drawn' data.

- Jeffrey Prior: $p(p) \sim B(\frac{1}{2}, \frac{1}{2})$.

13.4.2 Poisson Model

Generating process $y_1, \dots, y_N$ i.i.d. $\sim P(\lambda)$

$$\text{Distribution:} f(y|\lambda) = \frac{y!}{\lambda^y} e^{-\lambda} \propto \lambda^y e^{-\lambda} \tag{13.109}$$

$$\text{Likelihood:} L(y|\lambda) \propto \lambda^{N\bar{y}} e^{-N\lambda} \tag{13.110}$$

$$\text{Score:} S(y|\lambda) = N(\frac{\bar{y}}{\lambda} - 1) \tag{13.111}$$

$$\text{Observed Info:} J(y|\lambda) = \frac{N\bar{y}}{\lambda^2} \tag{13.112}$$

$$\text{Fisher Info:} I(\lambda) = \frac{N}{\lambda} \tag{13.113}$$

- Conjugate Prior: Gamma Distribution $\Gamma(\alpha, \beta)$

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \sim \Gamma(\alpha, \beta) \tag{13.114}$$

$$p(\lambda|y, \alpha, \beta) \propto \lambda^{\alpha-1} e^{-\beta\lambda} \lambda^{N\bar{y}} e^{-N\lambda} \sim \Gamma(\alpha + N\bar{y}, \beta + N) \tag{13.115}$$

- Jeffrey Prior: $p(\lambda) \sim \Gamma(\frac{1}{2}, 0)$. (actually $\beta \to 0^+$, similar for followings)

13.4.3 Exponential Model

Generating process $y_1, \dots, y_N$ i.i.d. $\sim \varepsilon(\lambda)$

$$\text{Distribution:} f(y|\lambda) = \lambda e^{-\lambda y} \propto \lambda^y e^{-\lambda y} \tag{13.116}$$

$$\text{Likelihood:} L(y|\lambda) \propto \lambda^N e^{-\lambda N\bar{y}} \tag{13.117}$$

$$\text{Score:} S(y|\lambda) = \frac{N}{\lambda} - N\bar{y} \tag{13.118}$$

$$\text{Observed Info:} J(y|\lambda) = \frac{N}{\lambda^2} \tag{13.119}$$

$$\text{Fisher Info:} I(\lambda) = \frac{N}{\lambda^2} \tag{13.120}$$

- Conjugate Prior: Gamma Distribution $\Gamma(\alpha, \beta)$

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \sim \Gamma(\alpha, \beta) \tag{13.121}$$

$$p(\lambda|y, \alpha, \beta) \propto \lambda^{\alpha-1} e^{-\beta\lambda} \lambda^N e^{-N\bar{y}\lambda} \sim \Gamma(\alpha + N, \beta + N\bar{y}) \tag{13.122}$$

- Jeffrey Prior: $p(\lambda) \sim \Gamma(0, 0)$.

13.4.4 Normal Model

□ **Model with known variance** σ^2

Generating process $y_1, \dots, y_N$ i.i.d. $\sim N(\mu, \sigma^2)$

$$\text{Distribution:} f(y|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right] \propto \exp\left[-\frac{\mu^2 - 2y\mu}{2\sigma^2}\right] \tag{13.123}$$

$$\text{Likelihood:} L(y|\mu) \propto \exp\left[-\frac{N(\mu^2 - 2\bar{y}\mu)}{2\sigma^2}\right] \tag{13.124}$$

$$\text{Score:} S(y|\mu) = -\frac{N(\mu - \bar{y})}{\sigma^2} \tag{13.125}$$

$$\text{Observed Info:} J(y|\mu) = \frac{N}{\sigma^2} \tag{13.126}$$

$$\text{Fisher Info:} I(\mu) = \frac{N}{\sigma^2} \tag{13.127}$$

- Conjugate Prior: Normal Distribution $N(\mu_0, \tau_0^2)$

$$p(\mu|\mu_0, \tau_0^2) = \frac{1}{\sqrt{2\pi\tau_0}} \exp\left[-\frac{(\mu-\mu_0)^2}{2\tau_0^2}\right] \sim N(\mu_0, \tau_0^2) \tag{13.128}$$

$$p(\mu|y, \mu_0, \tau_0^2) \propto \exp\left[-\frac{\mu^2 - 2\mu_0\mu}{2\tau_0^2} - \frac{N(\mu^2 - 2\bar{y}\mu)}{2\sigma^2}\right] \sim N\left(\frac{\frac{\mu_0}{\tau_0^2} + \frac{\bar{y}}{\sigma^2/N}}{\frac{1}{\tau_0^2} + \frac{1}{\sigma^2/N}}, \frac{1}{\frac{1}{\tau_0^2} + \frac{1}{\sigma^2/N}}\right) \tag{13.129}$$

- Jeffrey Prior: $p(\mu) \propto 1 \sim N(\wedge, \infty)$.

□ **Model with known mean** μ

Generating process $y_1, \dots, y_N$ i.i.d. $\sim N(\mu, \sigma^2)$

$$\text{Distribution:} f(y|\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right] \tag{13.130}$$

$$\text{Likelihood:} L(y|\sigma^2) \propto \sigma^{-N} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mu)^2\right] \equiv \sigma^{-N} \exp\left[-\frac{N\text{MSE}}{2\sigma^2}\right] \tag{13.131}$$

$$\text{Score:} S(y|\sigma^2) = -\frac{N}{\sigma} + \frac{N\text{MSE}}{\sigma^3} \tag{13.132}$$

$$\text{Observed Info:} J(y|\sigma^2) = -\frac{N}{\sigma^2} + \frac{3N\text{MSE}}{\sigma^4} \tag{13.133}$$

$$\text{Fisher Info:} I(\sigma^2) = \frac{2N}{\sigma^2} \tag{13.134}$$

- Conjugate Prior: Inv-$\chi^2(\nu_0, \sigma_0^2)$

$$p(\sigma^2|\nu_0, \sigma_0^2) = \frac{\nu_0^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0}{2}}\Gamma(\frac{\nu_0}{2})}(\sigma^2)^{-\frac{\nu_0}{2}-1}e^{-\frac{\nu_0\sigma_0^2}{2\sigma^2}} \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \tag{13.135}$$

$$p(\sigma^2|y, \nu_0, \sigma_0^2) \propto (\sigma^2)^{-\frac{\nu_0}{2}-1}e^{-\frac{\nu_0\sigma_0^2}{2\sigma^2}}(\sigma^2)^{-N/2} \exp\left[-\frac{N\text{MSE}}{2\sigma^2}\right] \sim \text{Inv-}\chi^2\left(\nu_0 + N, \frac{\nu_0\sigma_0^2 + N\text{MSE}}{\nu_0 + N}\right) \tag{13.136}$$

- Jeffrey Prior: $p(\sigma^2) \propto \frac{1}{\sigma} \sim \text{Inv-}\chi^2(1, 0)$.

□ **Full model**

Generating process $y_1, \ldots, y_N$ i.i.d. $\sim N(\mu, \sigma^2)$

$$\text{Distribution:} f(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right] \tag{13.137}$$

$$\text{Likelihood:} L(y|\mu,\sigma^2) \propto \sigma^{-N} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i-\mu)^2\right] \equiv \sigma^{-N} \exp\left[-\frac{N(\bar{y}-\mu)^2+(N-1)s^2}{2\sigma^2}\right] \tag{13.138}$$

$$\text{Score:} S(y|\mu,\sigma^2) = \begin{pmatrix} \frac{N}{\sigma^2}(\bar{y}-\mu) \\ -\frac{N}{2\sigma^2} + \frac{\sum(y_i-\mu)^2}{2(\sigma^2)^2} \end{pmatrix} \tag{13.139}$$

$$\text{Observed Info:} J(y|\mu,\sigma^2) = \begin{pmatrix} \frac{N}{\sigma^2} & \frac{N(\bar{y}-\mu)}{(\sigma^2)^2} \\ \frac{N(\bar{y}-\mu)}{(\sigma^2)^2} & -\frac{N}{2(\sigma^2)^2} + \frac{\sum(y_i-\mu)^2}{(\sigma^2)^3} \end{pmatrix} \tag{13.140}$$

$$\text{Fisher Info:} I(\mu,\sigma^2) = \begin{pmatrix} \frac{N}{\sigma^2} & \\ 0 & \frac{N}{2(\sigma^2)^2} \end{pmatrix} \tag{13.141}$$

Another parameterization $(\mu, \sigma^2) \mapsto (\mu, \log\sigma)$:

$$\text{Score:} S(y|\mu,\log\sigma) = \begin{pmatrix} \frac{N}{\sigma^2}(\bar{y}-\mu) \\ -N + \frac{\sum(y_i-\mu)^2}{\sigma^2} \end{pmatrix} \tag{13.142}$$

$$\text{Observed Info:} J(y|\mu,\log\sigma) = \begin{pmatrix} \frac{N}{\sigma^2} & \frac{N(\bar{y}-\mu)}{(\sigma^2)^2} \\ \frac{N(\bar{y}-\mu)}{(\sigma^2)^2} & \frac{2\sum(y_i-\mu)^2}{\sigma^2} \end{pmatrix} \tag{13.143}$$

$$\text{Fisher Info:} I(\mu,\log\sigma) = \begin{pmatrix} \frac{N}{\sigma^2} & 0 \\ 0 & 2N \end{pmatrix} \tag{13.144}$$

- Conjugate Prior for (μ, σ^2) parameterization: Normal-Inv-χ^2 Distribution N-Inv-$\chi^2(\mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2)$,

defined as $p(\sigma^2) \times p(\mu|\sigma^2) = \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \times N(\mu_0, \sigma^2/\kappa_0)$

$$p(\mu, \sigma^2|\mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \propto (\sigma^2)^{-(\nu_0/2+1)} (\sigma^2)^{-1/2} \exp\left[-\frac{1}{2\sigma^2}(\nu_0\sigma_0^2 + \kappa_0(\mu-\mu_0)^2)\right] \tag{13.145}$$

$$\sim N\text{-Inv-}\chi^2(\mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \tag{13.146}$$

$$p(\mu, \sigma^2|y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \propto (\sigma^2)^{-(\nu_0/2+1)} (\sigma^2)^{-1/2} \exp\left[-\frac{1}{2\sigma^2}(\nu_0\sigma_0^2 + \kappa_0(\mu-\mu_0)^2)\right] \tag{13.147}$$

$$\times (\sigma^2)^{-N/2} \exp\left[-\frac{N(\bar{y}-\mu)^2 + (N-1)s^2}{2\sigma^2}\right] \tag{13.148}$$

$$\sim N\text{-Inv-}\chi^2(\mu_n, \sigma_n^2/\kappa_n; \nu_n, \sigma_n^2) \tag{13.149}$$

$$\begin{cases} \mu_n &= \dfrac{\kappa_0}{\kappa_0+N}\mu_0 + \dfrac{N}{\kappa_0+N}\bar{y} \\ \kappa_n &= \kappa_0 + N \\ \nu_n &= \nu_0 + N \\ \nu_n\sigma_n^2 &= \nu_0\sigma_0^2 + (N-1)s^2 + \dfrac{\kappa_0 N}{\kappa_0+N}(\bar{y}-\mu_0)^2 \end{cases} \tag{13.150}$$

$$p(\mu|y, \sigma^2) \sim N\left(\frac{\dfrac{\kappa_0\mu_0}{\sigma^2} + \dfrac{\bar{y}}{\sigma^2/N}}{\dfrac{\kappa_0}{\sigma^2} + \dfrac{1}{\sigma^2/N}}, \frac{1}{\dfrac{\kappa_0}{\sigma^2} + \dfrac{1}{\sigma^2/N}}\right) = N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right) \tag{13.151}$$

$$p(\mu|y, \mu_0, \sigma_0^2/\kappa_0; \nu_0, \sigma_0^2) \propto \left(1 + \frac{\kappa_n(\mu-\mu_n)^2}{\nu_n\sigma_n^2}\right)^{-(\nu_n+1)/2} \sim t_{\nu_n}(\mu_n, \sigma_n^2/\kappa_n) \tag{13.152}$$

- Jeffrey Prior
  - for (μ, σ^2) parameterization *with independency assumption of* (μ, σ):

$$p(\mu, \sigma^2) \propto 1 \times (\sigma^2)^{-1} = \sigma^{-2} \tag{13.153}$$

  - for (μ, σ^2) parameterization *without independency assumption of* (μ, σ):

$$p(\mu, \sigma^2) \propto \sigma^{-3} \tag{13.154}$$

  - for $(\mu, \log\sigma)$ parameterization *with independency assumption of* (μ, σ):

$$p(\mu, \log\sigma) \propto 1 \times 1 = 1 \tag{13.155}$$

  - for $(\mu, \log\sigma)$ parameterization *without independency assumption of* (μ, σ):

$$p(\mu, \log\sigma) \propto \sigma^{-1} \tag{13.156}$$

13.4.5 Multinomial Model

(One sample item here) Generating process $(y_1, y_2, \ldots, y_J) \sim \text{Multino}(n; \theta_1, \theta_2, \ldots, \theta_J)$,, $w.r.t.\ \sum_{j=1}^J \theta_j = 1$, $\sum_{j=1}^J y_j = n$

$$\text{Distribution:} f(y|\theta) = \binom{n}{y_1 \ \ldots \ y_J} \prod_{j=1}^J \theta_j^{y_j}, \quad \sum_{j=1}^J \theta_j = 1, \sum_{j=1}^J y_j = n \tag{13.157}$$

$$\text{Likelihood:} L(y|\theta) \propto \prod_{j=1}^J \theta_j^{y_j}, \quad \sum_{j=1}^J \theta_j = 1 \tag{13.158}$$

the score function and Fisher information are slightly different because of the constraint $\sum_j \theta_j = 1$, i.e. $\vec{\theta} \in \mathbb{R}^{J-1} \subset \mathbb{R}^J$. Fortunately **for multinomial** the transformation function **happens to** reserve the $\det(I(\theta))$, i.e. we could simply 'pretend' their independence to get

$$\text{Fisher Info: } \det\left[I(\theta)\right] = \frac{1}{\theta_1 \theta_2 \ldots \theta_J}, \quad \sum_{j=1}^J \theta_j = 1 \tag{13.159}$$

- Conjugate Prior:Dirichlet Distribution $\text{Dirichlet}(\alpha_1, \ldots, \alpha_J)$

$$p(\theta|\alpha) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_J)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_J)} \prod_{j=1}^J \theta_j^{\alpha_j - 1} \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_J) \tag{13.160}$$

$$p(\theta|y, \alpha) \propto \prod_{j=1}^J \theta_j^{\alpha_j - 1} \prod_{j=1}^J \theta_j^{y_j} \sim \text{Dirichlet}(\alpha_1 + y_1, \ldots, \alpha_J + y_J) \tag{13.161}$$

- Jeffrey Prior: $p(\theta) \propto \prod_{j=1}^J \theta_j^{-1/2} \sim \text{Dirichlet}(\frac{1}{2}, \ldots, \frac{1}{2})$.

13.4.6 Multi-Normal Model

Generating process $y_1, \ldots, y_N$ i.i.d. $\sim N_d(\underset{d\times 1}{\mu}, \underset{d\times d}{\Sigma})$

$$\text{Distribution:} f(y|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)\right] \tag{13.162}$$

$$\text{Likelihood:} L(y|\mu, \Sigma) \propto |\Sigma|^{-N/2} \exp\left[-\frac{1}{2}\sum_{i=1}^N (y_i-\mu)'\Sigma^{-1}(y_i-\mu)\right] \tag{13.163}$$

$$= |\Sigma|^{-N/2} \exp\left[-\frac{1}{2} tr\left(\Sigma^{-1} S_0\right)\right] \tag{13.164}$$

$$\text{where } S_0 \equiv \sum_{i=1}^N (y_i-\mu)(y_i-\mu)' = N(\bar{y}-\mu)(\bar{y}-\mu)' + \sum_{i=1}^N (y_i-\bar{y})(y_i-\bar{y})' \tag{13.165}$$

$$\equiv N(\bar{y}-\mu)(\bar{y}-\mu)' + S \tag{13.166}$$

- Conjugate Prior: Normal-Inv-Wishart Distribution N-Inv-Wishart$(\mu_0, \Lambda_0/\kappa_0; \nu_0, \Lambda_0)$, defined as $p(\Sigma) \times$

$$p(\mu|\Sigma) = \text{Inv-Wishart}(\nu_0, \Lambda_0) \times N(\mu_0, \Sigma/\kappa_0)$$

$$p(\mu, \Sigma|\mu_0, \Lambda^2/\kappa_0; \nu_0, \Lambda_0) \propto (\Sigma)^{-(\nu_0/2+1)} (\Sigma)^{-p/2} \exp\left[-\frac{1}{2} tr(\Lambda_0 \Sigma^{-1}) - \frac{\kappa_0}{2}(\mu - \mu_0)\Sigma^{-1}(\mu - \mu_0)\right] \tag{13.167}$$

$$\sim N\text{-Inv-Wishart}(\mu_0, \Lambda_0/\kappa_0; \nu_0, \Lambda_0) \tag{13.168}$$

$$p(\mu, \Sigma|y, \mu_0, \Lambda_0/\kappa_0; \nu_0, \Lambda_0) \propto (\Sigma)^{-(\nu_0/2+1)} (\Sigma)^{-p/2} \exp\left[-\frac{1}{2} tr(\Lambda_0 \Sigma^{-1}) - \frac{\kappa_0}{2}(\mu - \mu_0)\Sigma^{-1}(\mu - \mu_0)\right] \tag{13.169}$$

$$\times (\Sigma)^{-N/2} \exp\left[-\frac{1}{2} tr\left([N(\bar{y} - \mu)(\bar{y} - \mu)' + S]\Sigma^{-1}\right)\right] \tag{13.170}$$

$$\sim N\text{-Inv-Wishart}(\mu_n, \Lambda_n/\kappa_n; \nu_n, \Lambda_n) \tag{13.171}$$

$$\begin{cases} \mu_n &= \dfrac{\kappa_0}{\kappa_0 + N}\mu_0 + \dfrac{N}{\kappa_0 + N}\bar{y} \\ \kappa_n &= \kappa_0 + N \\ \nu_n &= \nu_0 + N \\ \sigma_n^2 &= \Lambda_0 + S + \dfrac{\kappa_0 N}{\kappa_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)' \end{cases} \tag{13.172}$$

$$p(\mu|y, \Sigma) \sim N(\mu_n, \frac{\Sigma}{\kappa_n}) \tag{13.173}$$

$$p(\mu|y) \sim t_{\nu_n - d + 1}(\mu_n, \frac{\Lambda_n}{\kappa_n(\nu_n - d + 1)}) \tag{13.174}$$

Note: When generalizing from Inv-χ^2 to Inv-Wishart, there's a slight change $\nu_0\sigma_0^2 \mapsto \Lambda_0$.

- Jeffrey Prior: $p(\mu, \Sigma) \propto |\Sigma|^{-(d+1)/2} \sim N\text{-Inv-Wishart}(\wedge, \infty; -1, 0)$

13.4.7 Hierarchical Binomial Model

Generating process: $y_j \sim \text{Binom}(n_j, \theta_j)$, $\theta_j \sim B(\alpha, \beta)$, $j = 1, 2, \ldots, J$.

$$p(\theta, \alpha, \beta|y) \propto p(\alpha, \beta) \prod_{j=1}^{J} \frac{1}{B(\alpha, \beta)} \theta_j^{\alpha-1}(1 - \theta_j)^{\beta-1} \prod_{j=1}^{J} \theta_j^{y_j}(1 - \theta_j)^{n_j - y_j} \tag{13.175}$$

$$p(\theta|\alpha, \beta, y) \propto \prod_{j=1}^{J} \theta_j^{\alpha+y_j-1}(1 - \theta_j)^{\beta_j + n_j - y_j - 1} \sim B(\alpha + y_j, \beta_j + n_j - y_j) \tag{13.176}$$

$$p(\alpha, \beta|y) \propto p(\alpha, \beta) \prod_{j=1}^{J} \frac{B(\alpha + y_j, \beta + n_j - y_j)}{B(\alpha, \beta)} \tag{13.177}$$

Note: $p(\alpha, \beta)$ should be thin-tailed to avoid divergence at $\alpha, \beta \to \infty$.

13.4.8 Hierarchical Normal Model

Generating process: $y_{ij} \sim N(\theta_j, \sigma^2)$, $\theta_j \sim N(\mu, \tau^2)$, $i = 1, 2, \ldots, n_j, \quad j = 1, 2, \ldots, J$

$$p(\theta,\mu,\tau|y) \propto p(\mu,\tau)\prod_{j=1}^{J} N(\theta_j|\mu,\tau^2)\prod_{j=1}^{J} N(\bar{y}_j|\theta_j,\sigma^2/n_j) \tag{13.178}$$

$$p(\theta|\mu,\tau,y) \sim \prod_{j=1}^{J} N\left(\frac{\frac{\bar{y}_j}{\sigma^2/n_j}+\frac{\mu}{\tau^2}}{\frac{1}{\sigma^2/n_j}+\frac{1}{\tau^2}}, \frac{1}{\frac{1}{\sigma^2/n_j}+\frac{1}{\tau^2}}\right) \tag{13.179}$$

$$p(\mu,\tau|y) \propto p(\mu,\tau)\prod_{j=1}^{J} N(\bar{y}_j|\mu,\frac{\sigma^2}{n_j}+\tau^2) \tag{13.180}$$

$$p(\mu|\tau,y) \sim N\left(\tilde{\mu},\tilde{V}\right), \qquad \text{with } p(\mu|\tau)\propto 1 \tag{13.181}$$

$$\begin{cases} \tilde{\mu} \equiv \dfrac{\sum_{j=1}^{J}\frac{\bar{y}_j}{\sigma^2/n_j+\tau^2}}{\sum_{j=1}^{J}\frac{1}{\sigma^2/n_j+\tau^2}} \\ \tilde{V} \equiv \dfrac{1}{\sum_{j=1}^{J}\frac{1}{\sigma^2/n_j+\tau^2}} \end{cases} \tag{13.182}$$

$$p(\tau|y) \propto p(\tau)\left(\tilde{V}\right)^{1/2}\prod_{j=1}^{J}(\sigma^2/n_j+\tau^2)^{-1/2}\exp\left[-\frac{(\bar{y}_j-\tilde{\mu})^2}{2(\sigma^2/n_j+\tau^2)}\right] \tag{13.183}$$

13.4.9 Linear Model

Here we directly use multivatiate version of linear model:

$$Y = X\beta+\varepsilon, \quad \varepsilon\sim N(0,\sigma^2 I) \tag{13.184}$$

(13.185)

with assumption $X\sim p(X|\psi)$, $Y|X\sim p(y|X,\theta)$ where independent priori assumption of ψ and θ:

$$p(\psi,\theta) = p(\psi)p(\theta) \tag{13.186}$$

in this way we can conveniently only consider distribution of $\theta=(\theta,\sigma^2)$ in the posterior

$$p(\psi,\theta|X,Y) = \frac{p(\psi)p(\theta)p(Y|X,\theta)p(X|\psi)}{p(Y|X)p(X)} \tag{13.187}$$

$$= p(\psi|X)p(\theta|X,Y) \tag{13.188}$$

Likelihood under normal assumption:

$$p(Y|X,\theta) = \frac{1}{(2\pi)^{n/2}\sigma^n}\exp\left[-\frac{1}{2\sigma^2}(Y-X\beta)^T(Y-X\beta)\right] \tag{13.189}$$

Solution of posterior with prior $p(\beta,\sigma^2)\propto\sigma^{-2}$, i.e. **Default Bayesian Regression**[7]

$$\text{prior: } p(\beta,\sigma^2)\propto\sigma^{-2} \tag{13.191}$$

$$\text{posterior: } \beta|\sigma^2,X,Y\sim N\big((X'X)^{-1}X'Y,\sigma^2(X'X)^{-1}\big) \tag{13.192}$$

$$\sigma^2|X,Y\sim\text{Inv-}\chi^2(n-p,\hat{\sigma}^2) \tag{13.193}$$

$$\beta|X,Y\sim t_{n-p}\big((X'X)X'Y,\hat{\sigma}^2(X'X)^{-1}\big) \tag{13.194}$$

$$\hat{\sigma}^2=\frac{1}{n-p}\big(Y-X(X'X)^{-1}XY\big)'\big(Y-X(X'X)^{-1}XY\big) \tag{13.195}$$

$$\tilde{Y}|\tilde{X};X,Y\sim t_{n-p}\big(\tilde{X}'(X'X)^{-1}XY,(I+\tilde{X}'(X'X)^{-1}\tilde{X})\hat{\sigma}^2\big) \tag{13.196}$$

Solution of posterior with conjugate prior N-Inv-χ^2 $(m_0,s_0^2C_0;\nu_0,s_0^2)$:

$$\text{prior: } \beta|\sigma^2\sim N\big(m_0,\sigma^2C_0\big) \tag{13.197}$$

$$\sigma^2\sim\text{Inv-}\chi^2(\nu_0,s_0^2) \tag{13.198}$$

$$\text{posterior: } \beta|\sigma^2,X,Y\sim N\big(m_n,\sigma^2C_n\big) \tag{13.199}$$

$$\sigma^2|X,Y\sim\text{Inv-}\chi^2(\nu_n,s_n^2) \tag{13.200}$$

$$\begin{cases} m_n=m_0+C_0X'(XC_0X'+I)^{-1}(Y-Xm_0) \\ C_n=(X'X+C_0^{-1})^{-1}=C_0-C_0X'(XC_0X'+I)^{-1}XC_0 \\ \nu_n=\nu_0+n \\ \nu_ns_n^2=\nu_0s_0^2+(Y-Xm_0)'(XC_0X'+I)^{-1}(Y-Xm_0) \end{cases} \tag{13.201}$$

Solution of posterior with Ridge regression prior, i.e. take $C_0=c_0I$ in the above conjugate prior

$$\text{prior: } \beta|\sigma^2\sim N\big(0,c_0\sigma^2I\big) \tag{13.202}$$

$$p(\sigma^2)\propto\sigma^{-2} \tag{13.203}$$

Solution of posterior with Zellner's g-prior, i.e. take $C_0=(X'X)^{-1}$ in the above conjugate prior

$$\text{prior: } \beta|\sigma^2\sim N\big(b_0,g\sigma^2(X'X)^{-1}\big) \tag{13.204}$$

$$p(\sigma^2)\propto\sigma^{-2} \tag{13.205}$$

$$\text{posterior: } \beta|\sigma^2,X,Y\sim N\big(\frac{1}{g+1}b_0+\frac{g}{g+1}(X'X)^{-1}X'Y,\ \frac{g}{g+1}\sigma^2(X'X)^{-1}\big) \tag{13.206}$$

$$\sigma^2|X,Y\sim\text{Inv-}\chi^2\big(n,\frac{1}{n}\big[Y'(I-X'(X'X)^{-1}X)Y+ \tag{13.207}$$

$$\frac{1}{g+1}(b_0-(X'X)^{-1}XY)'(X'X)(b_0-(X'X)^{-1}XY)\big]\big) \tag{13.208}$$

$$\beta|X,Y\sim t_{n-p}\big(\frac{1}{g+1}b_0+\frac{g}{g+1}(X'X)^{-1}X'Y,\ \frac{1}{n}\frac{g}{g+1}\big[Y'(I-X'(X'X)^{-1}X)Y+ \tag{13.209}$$

$$\frac{1}{g+1}(b_0-(X'X)^{-1}XY)'(X'X)(b_0-(X'X)^{-1}XY)\big](X'X)^{-1}\big) \tag{13.210}$$

[7]Here the result uses the Woodbury Matrix Identity, introduced in section 4.1.2 ~ page 118.

$$(A+UCV)^{-1}=A^{-1}-A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1} \tag{13.190}$$

detail see https://en.wikipedia.org/wiki/Woodbury_matrix_identity?wprov=sfti1

13.4.10 Hierarchical Linear Model

Here is the bayesian version of the Mix Effect regression model introduced in DOE, e.g.

$$\text{Random Effect: } Y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2), \quad \tau_i \sim N(0, \sigma_\tau^2) \tag{13.211}$$

$$\text{Random Intercept: } Y_{ij} = \mu + \beta_{0i} + x_{ij}\beta_1 + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2), \quad \beta_{0i} \sim N(0, \sigma_{\beta_0}^2) \tag{13.212}$$

$$\text{R Intercept R Slope: } Y_{ij} = \beta_{0i} + x_{ij}\beta_{1i} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2), \quad \beta \sim N(\mu_\beta, \Sigma_\beta) \tag{13.213}$$

Chapter. XIV　实验设计与分析部分

Instructor: Zaiying Zhou

Design of Experiment (DoE) aims at understanding the cause-and-effect relation in systems (thus shares lots of similar language as Causal Inference). DoE is one step beyond Linear Regression where Xs are passively drawn while in DoE we are *deliberately* designing them to be more precise / more efficient in studying Y-X relation.

$$\underbrace{\text{Experiment Designing} \longrightarrow \text{Execution} \longrightarrow \overbrace{\text{Analysis of Data}}^{\text{Regression / Causal Inference}}}_{\text{DoE}} \tag{14.1}$$

□ **Philosophy of DoE**

- **Randomize** :
- **Replicate** :
- **Blocking** :

Section 14.1　Statistical Inference Methods for Factor Models

Basic inference methods are introduced in section 2.3.3 ~ page 54 (interval estimation) and section 2.4.2 ~ page 60 (hypothesis testing). ANOVA in Regression is introduced in section 3.3.4 ~ page 84. Preliminary introductions to factor model include section 3.1.2 ~ page 74 and Chapter 8 ~ page 232. Listed here for review.

14.1.1　One Sample Inference

With $X_1, X_2, \ldots, X_n$ i.i.d. $\sim N(\mu, \sigma^2)$:

- at null hypothesis $H_0 : \mu = \mu_0$,with known variance:

$$Z_0 = \frac{\sqrt{n}(\bar{X} - \mu_0)}{\sigma} \sim N(0, 1) \tag{14.2}$$

- at null hypothesis $H_0 : \mu = \mu_0$, with unknown variance:

$$t_0 = \frac{\sqrt{n}(\bar{X} - \mu_0)}{s} \sim t_{n-1} \tag{14.3}$$

- at null hypothesis $H_0 : \sigma = \sigma_0$, with unknown mean

$$\chi_0^2 = \frac{(n-1)s^2}{\sigma_0^2} \sim \chi_{n-1}^2 \tag{14.4}$$

14.1.2 Two Sample Comparison

Two sample comparison with Normal assumption is just similar to one-sample mean comparison. Usually the key problem is to find a t-statistics and get the dof for denominator.

Two sample: $X_{11}, X_{12}, \ldots, X_{1n_1}$ i.i.d. $\sim N(\mu_1, \sigma_1^2)$; $X_{21}, X_{22}, \ldots, X_{2n_2}$ i.i.d. $\sim N(\mu_2, \sigma_2^2)$.

- at null hypothesis $H_0 : \mu_1 - \mu_2 = \Delta_0$, with known variance

$$z_0 = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{\sqrt{\frac{\sigma_1}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0,1) \tag{14.5}$$

- at null hypothesis $H_0 : \mu_1 - \mu_2 = \Delta_0$, with unknown but same variance

$$t_0 = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{s_{\text{pooled}}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t_{n_1+n_2-2}, \quad s_{\text{pooled}} = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2} \tag{14.6}$$

- at null hypothesis $H_0 : \mu_1 - \mu_2 = \Delta_0$, with unknown variance (Welch-Satterthwaite approximation for the Behrens-Fisher problem[1]).

$$t_0^{\text{Welch}} = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \approx\sim t_\nu \quad \nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1+1} + \frac{(s_2^2/n_2)^2}{n_2+1}} - 2 \tag{14.7}$$

- at null hypothesis $H_0 : \sigma_1 = \sigma_2$, with unknown mean

$$F_0 = \frac{s_1^2}{s_2^2} \sim F_{n_1-1, n_2-2} \tag{14.8}$$

▷ R. **Code**

```r
y1 <- rnorm(100)
y2 <- rnorm(100, 1)

t.test(y1, y2, var.equal = TRUE) # Use pooled variance
t.test(y1, y2, var.equal = FALSE) # Welch's t-test
t.test(y1, y2, paired = TRUE) # pairwise t
```

14.1.3 One Way ANOVA

Generalization from two-sample t-test to Factor ANOVA: Use the trick that $F \sim t^2$, e.g.

$$t_0^2 = \frac{((\bar{X}_1 - \bar{X}_2) - \Delta_0)^2}{s_{\text{pooled}}^2(\frac{1}{n_1} + \frac{1}{n_2})} \sim F_{1, n_1+n_2-2} \tag{14.9}$$

[1] This is the output in `t.test(x1,x2, paired = FALSE, var.equal = FALSE)`

in which the nominator is 'difference in mean', denominator is 'fluctuation', i.e. in ANOVA language, variation caused by group difference MSR and variation caused by random effect MSE.

Model: (Factor model with balanced design here. Cell mean model & unbalanced design see section 8.1.1 ~ page 232)

- Fixed Effect:

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \varepsilon_{ij} \sim N(0, \sigma^2), \quad i = 1, 2, \ldots, a, \quad j = 1, 2, \ldots, n, \quad w.r.t. \sum_{i=1}^{a} \alpha_i = 0 \tag{14.10}$$

Solution could be obtained by traditional way $[\mu, \alpha_1, \ldots, \alpha_{a-1}] = (X'X)^{-1}XY$ with notation equation 3.10 ~ page 74.

$$\hat{\mu} = \bar{Y}_{\cdot\cdot} = \frac{1}{na} \sum_{i=1}^{a} \sum_{j=1}^{n} Y_{ij} \tag{14.11}$$

$$\hat{\alpha}_i = \frac{1}{n} \sum_{j=1}^{n} Y_{ij} - \hat{\mu} \tag{14.12}$$

$$s_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} \left(Y_{ij} - \bar{Y}_{i\cdot}\right)^2 \tag{14.13}$$

$$s^2 = \frac{(n-1)\sum_{i=1}^{a} s_i^2}{n_T - a} \tag{14.14}$$

ANOVA Table:

Source of Var	SS	dof	MS	$\mathbb{E}$ (MS)
α_i	$\text{SS}\alpha = \sum_{i=1}^{a} \left(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}\right)^2$	$a-1$	$\frac{\text{SS}\alpha}{a-1}$	$\sigma^2 + \frac{n\sum_{i=1}^{a} \alpha_i^2}{a-1}$
σ^2	$\text{SSE} = \sum_{i=1}^{a} \sum_{j=1}^{n} \left(Y_{ij} - Y_{i\cdot}\right)^2$	$n_T - a$	$\frac{\text{SSE}}{n(a-1)}$	σ^2

F statistics for $H_0 : \alpha_1 = \ldots = \alpha_a = 0$:

$$F_0 = \frac{\text{MS}\alpha}{\text{MSE}} \sim F_{a-1, n_T - a} \tag{14.15}$$

- Random Effect:

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \varepsilon_{ij} \sim N(0, \sigma^2), \quad i = 1, 2, \ldots, a, \quad j = 1, 2, \ldots, n_i, \quad \alpha_i \sim N(0, \sigma_\alpha^2)$$

Estimation:

$$\hat{\mu} = \bar{Y}_{\cdot\cdot} = \frac{1}{na} \sum_{i=1}^{a} \sum_{j=1}^{n} Y_{ij} \tag{14.16}$$

$$s_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} (Y_{ij} - \bar{Y}_{i\cdot})^2 \tag{14.17}$$

$$s^2 = \frac{(n-1)\sum_{i=1}^{a} s_i^2}{n_T - a} \tag{14.18}$$

$$\hat{\sigma}_\alpha^2 = \frac{1}{a} \left(\frac{\text{SS}\alpha}{a-1} - \frac{\text{SSE}}{n_T - a} \right) \tag{14.19}$$

ANOVA table:

Source of Var	SS	dof	MS	$\mathbb{E}$ (MS)
σ_α^2	$\text{SS}\alpha = n\sum_{i=1}^{a}\left(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}\right)^2$	$a-1$	$\dfrac{\text{SS}\alpha}{a-1}$	$\sigma^2 + n\sigma_\alpha^2$
σ^2	$\text{SSE} = \sum_{i=1}^{r}\sum_{j=1}^{n}\left(Y_{ij} - Y_{i\cdot}\right)^2$	$n_T - a$	$\dfrac{\text{SSE}}{n_T - a}$	σ^2

F statistics for $H_0 : \sigma_\alpha^2 = 0$:

$$F = \frac{\text{MS}\alpha}{\text{MSE}} \sim F_{a-1,n_T-a} \tag{14.20}$$

▷ R. **Code**

```r
library(agricolae)
dat <- data.frame(y = ..., trt = ... %>% factor())

# fixed effect
fit_fixed <- aov(y ~ trt, data = dat)
summary(fit_fixed)

# random effect
library(lme4)
fit_random <- lmer(y ~ (1|trt), data = dat)
summary(fit_random)
```

☐ **General Linear Test Point of View**

General Linear Test in linear regression see section 3.4.6 ~ page 96. The idea is to compare a full model and a reduced model

$$\begin{cases} \text{Full Model}: & Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \\ \text{Reduced Model}: & Y_{ij} = \mu + \varepsilon_{ij} \end{cases} \tag{14.21}$$

in this case the General Linear Test F is

$$F^{\text{GLT}} = \frac{(\text{SSE}_\text{R} - \text{SSE}_\text{F})/(dof_\text{R} - dof_\text{F})}{\text{SSE}_\text{F}/dof_\text{F}} = \frac{\text{MS}\alpha}{\text{MSE}} = F_0 \sim F_{a-1,n_T-a} \tag{14.22}$$

☐ **Likelihood Ratio Test Point of View**

Detail theory of LRT see section 2.4.3 ~ page 60. the test statistics is

$$\Lambda = \frac{\sup\limits_{\mu;\alpha=0} L(Y;\mu,\alpha)}{\sup\limits_{\mu;\alpha} L(Y;\mu,\alpha)} = \left(\frac{\text{SSTotal}}{\text{SSE}}\right)^{n_T/2}, \quad -2\log\Lambda \xrightarrow{\text{d}} \chi^2_{a-1} \tag{14.23}$$

and we have a bijection between Λ and F_0.

☐ **Homoscedasticity Assumption Diagnostics**

- Bartlett's Test for $H_0 : \sigma_1 = \ldots = \sigma_a$

$$T = \frac{(n_T - a)\log\frac{\sum_{i=1}^{a}(n-1)\text{MS}_i}{n_T - a} - \sum_{i=1}^{a}(n-1)\log\text{MS}_i}{1 + \frac{1}{3(a-1)}\left(\sum_{i=1}^{a}\frac{1}{n-1} - \frac{1}{n_T - a}\right)} \approx\sim \chi^2_{a-1} \tag{14.24}$$

  the idea is GeomMean $=$ ArithMean when all are equal.

- Levene's Test

$$T = (\text{ANOVA of } |y_{ij} - \bar{y}_{i\cdot}|) \approx\sim F_{a-1,n_T-a} \tag{14.25}$$

- Welch's ANOVA

  A generalized version of Welch's Test in two-sample t-test.

▷ R. **Code**

```r
bartlett.test(y ~ trt, data = dat) # Bartlett's Test
leveneTest(fit) # Levene's Test
oneway.test(y ~ trt, data = dat, var.equal=FALSE) # Welch's ANOVA
```

□ **Multiple Comparison**

Target: When compare level pairs, say some (α_i, α_j) pairs $\subset \{\alpha_i\}_{i=1}^{a}$, i.e. there are multiple tests, we need to adjust the testing procedure to avoid multiple comparison hazard.[2]

- Fisher's Least Significant Difference (LSD) without correction

$$t_{ij} = \frac{\bar{Y}_{i\cdot} - \bar{Y}_{j\cdot}}{\sqrt{\text{MSE}\left(\frac{1}{n_i} + \frac{1}{n_j}\right)}} \sim t_{N-a} \tag{14.27}$$

  rejection region construction use $t_{N-a,1-\alpha/2}$

- Fisher's Least Significant Difference (LSD) with Bonferroni correction: rejection region use $t_{N-a,1-\alpha/2m}$

- Tukey's Honestly Significant Difference (HSD): Under $H_0 : \alpha_1 = \ldots = \alpha_a$, treat Y_{ij} as sample of μ_i. We could study the range of $\{\bar{Y}_{i\cdot}\}_{i=1}^{a}$

$$q = \frac{\max\{\bar{Y}_{i\cdot}\}_{i=1}^{a} - \min\{\bar{Y}_{i\cdot}\}_{i=1}^{a}}{s\sqrt{n}} \sim q_{a,n_T-a} \tag{14.28}$$

  where $q_{\cdot,\cdot}$ is Tukey's studentized range distribution, see equation 3.227 ~ page 109.

- Scheffè's Method by testing contrasts. A $\phi \equiv \sum_{i=1}^{a}\xi_i\alpha_i \; w.r.t. \; \sum_{i=1}^{a}\xi_i = 0$ is called a **contrast**[3].

$$F_0 = \left(\frac{\hat{\phi}}{\sqrt{\hat{var}(\phi)}}\right)^2 /(a-1) = \frac{(\sum \xi_i\bar{Y}_{i\cdot})^2}{(a-1)\text{MSE}\sum\xi_i^2/n_i} \sim F_{a-1,N-a} \tag{14.29}$$

[2] An intuition: If we simply test each of m tests at $\alpha_i = 0.05$, then the overall type-I error is

$$\alpha = 1 - \prod_{i=1}^{m}(1-\alpha_i) > 0.05 \tag{14.26}$$

[3] First introduced in section 3.6.1 ~ page 107.

- Benjamini-Hochberg Method for False Discovery Rate control.
- Dunnett's Test for Many-to-One problem: e.g. we have a control group α_0 and $a-1$ treatment groups $\alpha_1, \ldots, \alpha_{a-1}$, we want to test a one-sided null hypothesis

$$H_0 : \mu_0 \leq \mu_i, \quad \forall i = 1, 2, \ldots, a-1$$

Dunnett's statistics are

$$t_i = \frac{\bar{y}_{i\cdot} - \bar{y}_{0\cdot}}{\sqrt{\text{MSE}(\frac{1}{n_i} + \frac{1}{n_0})}}, \quad \begin{bmatrix} t_1 \\ \vdots \\ t_{a-1} \end{bmatrix} \sim t_{a-1}(\rho = \{\rho_{ij} = \sqrt{\frac{n_i n_j}{(n_i + n_0)(n_j + n_0)}}\}_{i,j=1}^{a-1})$$

▷ R. **Code**

```r
fit <- aov(y ~ trt, data = dat)



# Fisher LSD
LSD.test(fit, 'trt', group = FALSE, console = TRUE)
# Fisher LSD with bonferroni
pairwise.t.test(dat$y, dat$trt, p.adj = 'bonferroni')
LSD.test(fit, 'trt', group = F, console = T, p.adj = 'bonferroni')
# Tukey's HSD
TukeyHSD(fit)
glht(fit, linfct = mcp(trt = "Tukey")) %>% summary
# Scheffe's Method
scheffe.test(fit, 'trt', group = F, console = T)
# Dunnett's Test
glht(fit, linfct = mcp(trt = "Dunnett")) %>% summary

## plotting confidence interval
glht(fit, linfct = mcp(trt = "Tukey")) %>% confint %>% plot
```

Interval Construction follows similar method, see section 3.6 ~ page 107.

14.1.4 Multi Factor ANOVA

ANOVA inference for multifactor case was introduced in section 8.1.4 ~ page 235. Here are some recap. And more complex models and some insights for DoE are included.

Take two factor model with interaction as exmaple:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk} \tag{14.30}$$

Decomposition of SS and dof:

$$Y_{ijk} - \bar{Y}_{...} = \left(\bar{Y}_{i..} - \bar{Y}_{...}\right) + \left(\bar{Y}_{.j.} - \bar{Y}_{...}\right) \tag{14.31}$$

$$+ \left(\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}\right) + \left(Y_{ijk} - \bar{Y}_{ij.}\right) \tag{14.32}$$

$$\alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk} = ((\mu + \alpha_i) - \mu) + ((\mu + \beta_j) - \mu) \tag{14.33}$$

$$+ \left((\mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}) - (\mu + \alpha_i) - (\mu + \beta_j) + \mu\right) + (\varepsilon_{ijk}) \tag{14.34}$$

$$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(Y_{ijk} - \bar{Y}_{...}) = bn\sum_{i=1}^{a}\left(\bar{Y}_{i..} - \bar{Y}_{...}\right)^2 + an\sum_{j=1}^{b}\left(\bar{Y}_{.j.} - \bar{Y}_{...}\right)^2 \tag{14.35}$$

$$+ n\sum_{i=1}^{a}\sum_{j=1}^{b}\left(\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}\right)^2 + \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(Y_{ijk} - \bar{Y}_{ij.}\right)^2 \tag{14.36}$$

$$nab - 1 = (a-1) + (b-1) \tag{14.37}$$

$$+ \left((a-1)(b-1)\right) + (n-1)ab \tag{14.38}$$

and ANOVA table (e.g. with α, β both fixed effect factor):

ANOVA table:

Source of Var	SS	dof	$\mathbb{E}$ (MS)
α_i	$bn\sum_{i=1}^{a}\left(\bar{Y}_{i..} - \bar{Y}_{...}\right)^2$	$a-1$	$\sigma^2 + \frac{bn\sum_{i=1}^{a}\alpha_i^2}{a-1}$
β_j	$an\sum_{j=1}^{b}\left(\bar{Y}_{.j.} - \bar{Y}_{...}\right)^2$	$b-1$	$\sigma^2 + \frac{an\sum_{j=1}^{b}\beta_j^2}{b-1}$
$(\alpha\beta)_{ij}$	$n\sum_{i=1}^{a}\sum_{j=1}^{b}\left(\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}\right)^2$	$(a-1)(b-1)$	$\sigma^2 + \frac{n\sum_{i=1}^{a}\sum_{j=1}^{n}(\alpha\beta)_{ij}^2}{(a-1)(b-1)}$
σ^2	$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(Y_{ijk} - \bar{Y}_{ij.}\right)^2$	$ab(n-1)$	σ^2

表 14.1: ANOVA for two fixed effect model

Calculation of complicated factor design, especially for fixed&random combined or more factors, see Montgomery's Method at section 8.1.4 ~ page 235.

Some Key Problems to Consider in DoE:

- Effect of factors?
- Include interaction term, say $(\alpha\beta)_{ij}$, or not? Better functional form for interation?
- Cost of experiment in the case of multi factor.

□ F **Test For Factor or Interaction term:**

F test is simply $\text{MS}_\imath/\text{MS}_\jmath$, where $\mathbb{E}[\text{MS}_\imath] - \mathbb{E}[\text{MS}_\jmath]$ should correctly reflect the quantity to study. e.g. Still use the above table 14.1 ~ page 366 example, to study whether to include interaction term $(\alpha\beta)_{ij}$:

$$F_{(\alpha\beta)} = \frac{\text{MS}(\alpha\beta)}{\text{MSE}} = \frac{n\sum_{i=1}^{a}\sum_{j=1}^{b}\left(\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}\right)^2/(a-1)(b-1)}{\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(Y_{ijk} - \bar{Y}_{ij.}\right)^2/ab(n-1)} \sim F_{(a-1)(b-1),ab(n-1)} \tag{14.39}$$

Example Code for two factor model

▷ R. **Code**

```r
y <- c(...)
factor1 <- c(...) %>% factor()
factor2 <- c(...) %>% factor()
dat <- data.frame(y, factor1, factor2)
aovfit1 <- aov(y ~ factor1 + factor2, data = dat) # without
   interaction
aovfit2 <- aov(y ~ factor1 * factor2, data = dat) # with interaction

aovfit1 %>% summary()
aovfit2 %>% summary()
```

□ **Graphic Method for Interation**

e.g. for each α level, plot y-β_j relation and observe the parallel relation.

▷ R. **Code**

```r
interaction.plot(x.fac = factor2, trace.fac = factor1, response = y)
```

□ **Tukey's One dof Test for Additive Interaction**

Use the general interaction $(\alpha\beta)_{ij}$ uses $(a-1)(b-1)$ degree of freedom, thus cause less dof in estimating σ^2, and sometimes prevents us from conduct a valid DoE due to cost limit (e.g. can only conduct one test for each level $n=1$).

Tukey's method is to use an analogue to linear model $(\alpha\beta)_{ij} = \lambda\alpha_i\beta_j$:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \lambda\alpha_i\beta_j + \varepsilon_{ijk} \tag{14.40}$$

- Estimation:

$$\hat{\alpha}_i = \bar{Y}_{i\cdot\cdot} - \bar{Y}_{\cdots} \tag{14.41}$$

$$\hat{\beta}_j = \bar{Y}_{\cdot j\cdot} - \bar{Y}_{\cdots} \tag{14.42}$$

$$\hat{\lambda} \approx \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}\hat{\alpha}_i\hat{\beta}_j\bar{Y}_{ij\cdot}}{\left(\sum_{i=1}^{a}\hat{\alpha}_i^2\right)\left(\sum_{j=1}^{b}\hat{\beta}_j^2\right)} \tag{14.43}$$

- Motivation and Justification of product form interaction $\lambda\alpha_i\beta_j$: Consider $(\alpha\beta)_{ij}$ as a function of α_i, β_j, expand to second order:

$$(\alpha\beta)_{ij} \text{ as function of } \alpha,\ \beta = C_0 + C_1\alpha_i + C_2\beta_j + C_{11}\alpha_i^2 + C_{12}\alpha_i\beta_j + C_{22}\beta_j^2 + o(2^{\text{nd}}) \tag{14.44}$$

normalization condition $\sum_{i=1}^{a}(\alpha\beta)_{ij} = 0$, $\sum_{j=1}^{b}(\alpha\beta)_{ij} = 0$ yields

$$(\alpha\beta)_{ij} = C_{12}\alpha_i\beta_j + o(2^{\text{nd}}) \tag{14.45}$$

- Tukey additive term test:

$$F = \frac{\text{SS}\lambda/1}{\text{MSE}} = \frac{\left(\sum_{i=1}^{a}\sum_{j=1}^{b}(\hat{\lambda}\hat{\alpha}_i\hat{\beta}_j)^2\right)/\left(\sum_{i=1}^{a}\hat{\alpha}_i^2\sum_{j=1}^{b}\hat{\beta}_j^2\right)}{\text{MSE}} \sim F_{1,abn-a-b} \tag{14.46}$$

▷ R. **Code**

```r
library(additivityTests)
datmat <- matrix(dat, ...) # dim(factor1) * dim(factor2)
tukey.test(datmat)
```

Section 14.2 Blocking Methods

Blocking methods deal with nuisance factors, i.e. factors that we are not interested in but may affect the result, to help reduce the variation of the result.

A concrete example: we want to study the effect of fertilizer (α) on crop yield (y), but the soil quality (β) may affect the result. But finally we care about the effect of fertilizer on some arbitrary soil quality, so we block the soil quality factor, which is the nuisance factor in this case.

14.2.1 The Randomized Complete Block Design

e.g. when assessing the effect of α_i, we might try to induce some other blocking factor β_j. Then the model turns to a 1 fixed (α) + 1 random (β) factor model, with replicate size $n = 1$

$$Y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \tag{14.47}$$

which is the case for the randomized complete block design (RCBD).

Intuition about F_α:

$$F_\alpha = \frac{\text{MS}\alpha}{\text{SSE}/dof_{\text{SSE}}} \tag{14.48}$$

adding a blocking factor β results in both smaller SSE and dof_{SSE}. We are expecting more reduction in SSE so finally MSE decreases, to yield higher power.

Testing on β is usually not quite necessary (only when considering whether to include β in the model).[4]

▷ R. **Code**

```r
# test for RCBD \alpha effect
```

[4] Such comparison is achieved by assessing relative efficiency.

$$\text{rf} = \frac{\text{MSE}_{\text{CRD}}}{\text{MSE}_{\text{RCBD}}}$$

```
dat <- data.frame(y = ..., alpha = ... %>% factor, beta = ... %>%
    factor)
aov(y ~ alpha + Error(beta / alpha), data = dat) %>% summary
```

□ **Generalized RCBD**

for model with replicates $n \geq 2$ we fit the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk}, \quad k = 1, \ldots, n$$

or, in this case error terms are acceptable

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}, \quad k = 1, \ldots, n$$

14.2.2 Latin Square Design for Multi Factor ANOVA

To handle the case of more blocking factors (or simply there are too many factors), but faced with budget limit and can only conduct one test for each level. Latin Square Design is a method to reduce the cost of experiment while still keep the validity of DoE.

Latin square is used for # level equal for all factors (say 3 factors with 4 levels each).

□ **3 Factors Latin Square Design**

e.g. model[5]

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk} \tag{14.49}$$

denote α_i as row effect, β_j as column effect, $\gamma \in \{A, B, C, D, \ldots\}$ as layer effect (elements apprear in equation 14.56 ~ page 370 matrix). # levels for factors $:= m$. Replicates size $= 1$ so is ignored. **For blocking experiment, we usually put the blocking factors at row & column, and the factor of interest at layer γ.**

[5] We are already using Latin square to solve the problem of limited sample size, so adding interaction terms like $(\alpha\beta)_{ij}$ is unwise because it uses dof, thus cause less dof in estimating σ^2 or even make it impossible.

We could assign $m \times m$ runs according to the following Latin square (take $m = 4$ as example):[6]

$$\begin{array}{c} \text{factor}\,\gamma \searrow \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \begin{array}{c} \begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array} \\ \begin{pmatrix} A & B & C & D \\ B & C & D & A \\ C & D & A & B \\ D & A & B & C \end{pmatrix} \end{array} \tag{14.56}$$

with ANOVA table:

Source of Var	SS	dof
α_i (row)	$m\sum_{i=1}^{m}\left(\bar{Y}_{i\cdot\cdot}-\bar{Y}_{\cdots}\right)^2$	$m-1$
β_j (column)	$m\sum_{j=1}^{m}\left(\bar{Y}_{\cdot j\cdot}-\bar{Y}_{\cdots}\right)^2$	$m-1$
γ_k (layer)	$m\sum_{k=1}^{m}\left(\bar{Y}_{\cdot\cdot k}-\bar{Y}_{\cdots}\right)^2$	$m-1$
σ^2	$\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}\mathbb{I}_{(i,j,k)\in\text{Latin}}\left(Y_{ijk}-\bar{Y}_{i\cdot\cdot}-\bar{Y}_{\cdot j\cdot}-\bar{Y}_{\cdot\cdot k}+2\bar{Y}_{\cdots}\right)^2$	$(m-1)(m-2)$

表 14.2: ANOVA for 3 Effects Latin Square Design

- An illustration for sample size reduction: for 3 factor experiment with 4 levels for each factor, we need $4^3 = 64$ runs. If we use Latin Square Design, we only need $4 \times 4 = 16$ runs.

▷ R. **Code**

```r
# obtain latin square design
library(agricolae)
trt <- LETTERS[1:4]
design3 <- design.lsd(trt, seed = 42)
```

□ **Replicated Latin Square Design**

m^{th} order Latin square could be used with l replicates to generate lm^2 runs. There are 3 variants, corresponding to different experiment settings.

[6] i.e. $m \times m$ runs (arranged by column):

$$\text{run1}: \alpha_1, \beta_1, A \tag{14.50}$$
$$\text{run2}: \alpha_2, \beta_1, B \tag{14.51}$$
$$\vdots \tag{14.52}$$
$$\text{run5}: \alpha_1, \beta_2, B \tag{14.53}$$
$$\text{run6}: \alpha_2, \beta_2, C \tag{14.54}$$
$$\vdots \tag{14.55}$$

Here we take $r = 3$, $m = 4$ as example. Replicates size $= 1$. And κ_l, $l = 1, 2, \ldots, r$ is used to denote the Latin square replicate effect.

- Same column + Same row effect

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \kappa_l + \varepsilon_{ijkl}, \quad i, j, k = 1, 2, \ldots, m, \quad l = 1, 2, \ldots, r.$$

(Note that here different Latin squares are **replicates**)

factor $\gamma \searrow$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_1 & A & B & C & D \\ \alpha_2 & B & C & D & A \\ \alpha_3 & C & D & A & B \\ \alpha_4 & D & A & B & C \end{array} \quad \underbrace{\qquad}_{\kappa_1}$$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_1 & D & A & B & C \\ \alpha_2 & A & B & C & D \\ \alpha_3 & B & C & D & A \\ \alpha_4 & C & D & A & B \end{array} \quad \underbrace{\qquad}_{\kappa_2}$$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_1 & C & D & A & B \\ \alpha_2 & D & A & B & C \\ \alpha_3 & A & B & C & D \\ \alpha_4 & B & C & D & A \end{array} \quad \underbrace{\qquad}_{\kappa_3}$$

with ANOVA table:

Source of Var	dof
α_i (row)	$m - 1$
β_j (column)	$m - 1$
γ_k (layer)	$m - 1$
κ_l (square replicate)	$r - 1$
σ^2	$(m - 1)(r(m + 1) - 3)$

- Same column + Different row effect (Or equivalently Different column + Same row effect)

$$Y_{ijkl} = \mu + \alpha_{i(l)} + \beta_j + \gamma_k + \kappa_l + \varepsilon_{ijkl}, \quad i(l) = 1, \ldots, m \times r \quad j, k = 1, \ldots, m, \quad l = 1, \ldots, r.$$

factor $\gamma \searrow$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_1 & A & B & C & D \\ \alpha_2 & B & C & D & A \\ \alpha_3 & C & D & A & B \\ \alpha_4 & D & A & B & C \end{array} \quad \underbrace{\qquad}_{\kappa_1}$$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_5 & D & A & B & C \\ \alpha_6 & A & B & C & D \\ \alpha_7 & B & C & D & A \\ \alpha_8 & C & D & A & B \end{array} \quad \underbrace{\qquad}_{\kappa_2}$$

$$\begin{array}{c|cccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \alpha_9 & C & D & A & B \\ \alpha_{10} & D & A & B & C \\ \alpha_{11} & A & B & C & D \\ \alpha_{12} & B & C & D & A \end{array} \quad \underbrace{\qquad}_{\kappa_3}$$

with ANOVA table:

Source of Var	dof
$\alpha_{i(l)}$ (row)	$r(m - 1)$
β_j (column)	$m - 1$
γ_k (layer)	$m - 1$
κ_l (square replicate)	$r - 1$
σ^2	$(m - 1)(rp - 2)$

- Different column + Different row effect

$$Y_{ijkl} = \mu + \alpha_{i(l)} + \beta_{j(l)} + \gamma_k + \kappa_l + \varepsilon_{ijkl}, \quad i(l), j(l) = 1, \ldots, m \times r \quad k = 1, \ldots, m, \quad l = 1, \ldots, r.$$

$$\begin{array}{c} \text{factor } \gamma \searrow \\ \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \begin{array}{c} \begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array} \\ \underbrace{\begin{pmatrix} A & B & C & D \\ B & C & D & A \\ C & D & A & B \\ D & A & B & C \end{pmatrix}}_{\kappa_1} \end{array} \begin{array}{c} \begin{array}{cccc} \beta_5 & \beta_6 & \beta_7 & \beta_8 \end{array} \\ \underbrace{\begin{pmatrix} D & A & B & C \\ A & B & C & D \\ B & C & D & A \\ C & D & A & B \end{pmatrix}}_{\kappa_2} \end{array} \begin{array}{c} \begin{array}{cccc} \beta_9 & \beta_{10} & \beta_{11} & \beta_{12} \end{array} \\ \underbrace{\begin{pmatrix} C & D & A & B \\ D & A & B & C \\ A & B & C & D \\ B & C & D & A \end{pmatrix}}_{\kappa_3} \end{array} \begin{array}{l} \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{array} \begin{array}{l} \alpha_9 \\ \alpha_{10} \\ \alpha_{11} \\ \alpha_{12} \end{array}$$

with ANOVA table:

Source of Var	dof
$\alpha_{i(l)}$ (row)	$r(m-1)$
$\beta_{j(l)}$ (column)	$r(m-1)$
γ_k (layer)	$m-1$
κ_l (square replicate)	$r-1$
σ^2	$(m-1)(r(p-1)-1)$

□ 4 Factors Graeco-Latin Square Design

e.g. model

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \phi_l + \varepsilon_{ijklr} \tag{14.57}$$

denote α_i as row effect, β_j as column effect, $\gamma \in \{A, B, C, D, \ldots\}$ as layer row effect, $\phi_l \in \{\alpha, \beta, \gamma, \delta, \ldots\}$ as layer column effect. # levels for factors $:= m$. Usually $l = 1$

We could assign $m \times m$ runs according to the following Graeco-Latin square, in which each Graeco-Latin

alphabet pair only appear once (take $m = 4$ as example):[7]

$$\begin{array}{c} \text{factor } \gamma, \phi \searrow \\ \end{array} \begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array}$$
$$\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \begin{pmatrix} A\alpha & B\beta & C\gamma & D\delta \\ B\gamma & A\delta & D\alpha & C\beta \\ C\delta & D\gamma & A\beta & B\alpha \\ D\beta & C\alpha & B\delta & A\gamma \end{pmatrix} \tag{14.67}$$

with ANOVA table:

Source of Var	SS	dof
α_i (row)	$m\sum_{i=1}^{m}\left(\bar{Y}_{i\cdots}-\bar{Y}_{\cdots}\right)^2$	$m-1$
β_j (column)	$m\sum_{j=1}^{m}\left(\bar{Y}_{\cdot j\cdot\cdot}-\bar{Y}_{\cdots}\right)^2$	$m-1$
γ_k (layer row)	$m\sum_{k=1}^{m}\left(\bar{Y}_{\cdot\cdot k\cdot}-\bar{Y}_{\cdots}\right)^2$	$m-1$
ϕ_l (layer column)	$m\sum_{l=1}^{m}\left(\bar{Y}_{\cdots l}-\bar{Y}_{\cdots}\right)^2$	$m-1$
σ^2	$\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\mathbb{I}_{\substack{(i,j,k,l)\in \\ \text{GraecoLatin}}}\left(Y_{ijkl}-\bar{Y}_{i\cdots}-\bar{Y}_{\cdot j\cdot\cdot}-\bar{Y}_{\cdot\cdot k\cdot}-\bar{Y}_{\cdots l}+3\bar{Y}_{\cdots}\right)^2$	$(m-1)(m-3)$

表 14.3: ANOVA for 4 Effects Graeco-Latin Square Design

□ **Balanced Incomplete Block Design**

A further case is that in two factor model

$$Y_{ij} \sim \mu + \tau_i + \beta_j + \varepsilon_{ij}, \quad i = 1, 2, \ldots, t \quad j = 1, 2, \ldots, b.$$

we might have restrictions on the number of runs for each blocking level β_j, say we could only have $k < t$ runs. i.e. for ideal case we should have $t \times b$ runs but now only $k \times b$. This is called the balanced incomplete block design (BIBD).

[7] i.e. $m \times m$ runs (arranged by column):

$$\text{run1}: \alpha_1, \beta_1, A, \alpha \tag{14.58}$$
$$\text{run2}: \alpha_2, \beta_1, B, \gamma \tag{14.59}$$
$$\vdots \tag{14.60}$$
$$\text{run5}: \alpha_1, \beta_2, B, \beta \tag{14.61}$$
$$\text{run6}: \alpha_2, \beta_2, A, \delta \tag{14.62}$$
$$\vdots \tag{14.63}$$
$$\text{run9}: \alpha_1, \beta_3, C, \gamma \tag{14.64}$$
$$\text{run10}: \alpha_2, \beta_3, D, \alpha \tag{14.65}$$
$$\vdots \tag{14.66}$$

To handle this case, we hope to: for each β level, wisely choose which of τ_is should be included in the experiment runs. Here's an example:

$$\begin{array}{r} \text{assign run? } \searrow \\ \tau_1 := A \\ \tau_2 := B \\ \tau_3 := C \\ \tau_4 := D \end{array} \begin{array}{c} \begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array} \\ \begin{pmatrix} \surd & \surd & & \surd \\ & \surd & \surd & \surd \\ \surd & \surd & \surd & \\ \surd & & \surd & \surd \end{pmatrix} \end{array} = \begin{array}{r} \text{assign } \tau \searrow \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array} \\ \begin{pmatrix} A & B & C & D \\ C & A & D & B \\ D & C & B & A \end{pmatrix} \end{array}$$

Comment:

- Such design is possible iff

$$\lambda = \frac{bk(k-1)}{t(t-1)} \text{ is an integer}$$

- The design can be generated from Latin square, e.g. BIBD assignment in equation 14.68 ~ page 375 is just 3 rows from 4×4 Latin square.

▷ R. **Code**

BIBD assignment generation

```r
trt <- LETTERS[1:4]
k <- 3
design.bib(trt, k, seed = 42)
```

14.2.3 Regression with Blocking

Regression could be combined in factor model. An example is a factor component α + a numeric component $x'\beta$, say

$$Y_{ij} = \sum_{i=1}^{a} \left(\beta_{0i} + x'_{ij}\beta_{1i}\right) + \varepsilon_{ij}$$

Section 14.3 Factorial Design

An important application scenario of DoE is factorial design (析因试验), which is a method of designing experiments to study the effects of multiple factors simultaneously. The goal of factorial design is to efficiently explore the existance of effects and interactions.

14.3.1 2^k Factorial Design

A typical setting is that we have k factors, each with 2 levels (because we just want to examine the existance of effects) denoted $\{+, -\}$. This case is called 2^k screening designs. An example of 2^3 factorial design:

Run	Factor A	Factor B	Factor C	Response y
1	$-$	$-$	$-$	$y_1 = y_{---}$
2	$+$	$-$	$-$	$y_2 = y_{+--}$
3	$-$	$+$	$-$	$y_3 = y_{-+-}$
4	$-$	$-$	$+$	$y_4 = y_{--+}$
5	$+$	$+$	$-$	$y_5 = y_{++-}$
6	$+$	$-$	$+$	$y_6 = y_{+-+}$
7	$-$	$+$	$+$	$y_7 = y_{-++}$
8	$+$	$+$	$+$	$y_8 = y_{+++}$

The estimation of effects can be done by Least Square Estimation by $(X'X)^{-1}X'Y$, and further note that the above ± 1 encoding yields

$$\text{effect} = 2 \times \text{corresp regression coef} \tag{14.68}$$

e.g. a model with all interaction term for the above 2^3 design:

$$y_{A,B,C} = \underbrace{(1)}_{\text{intercept}} + A + B + C + (AB) + (AC) + (BC) + (ABC) + \varepsilon_{A,B,C}, \quad A, B, C \in \{+1, -1\} \tag{14.69}$$

should be estimated by

$$\hat{\text{effect}} = 2 \cdot \hat{\beta} \tag{14.70}$$

$$\hat{\beta} = (X'X)^{-1}X'Y = 2^k X'Y \tag{14.71}$$

$$var(\text{effect}_i) = \frac{\sigma^2}{2^{k-2} \times \text{replicate}} \tag{14.72}$$

$$X \equiv \begin{array}{c} \text{run} \\ 1: y_{---} \\ 2: y_{+--} \\ 3: y_{-+-} \\ 4: y_{--+} \\ 5: y_{++-} \\ 6: y_{+-+} \\ 7: y_{-++} \\ 8: y_{+++} \end{array} \begin{array}{c} (1) \\ \left[\begin{array}{c} + \\ + \\ + \\ + \\ + \\ + \\ + \\ + \end{array}\right. \end{array} \begin{array}{c} A \\ \begin{array}{c} - \\ + \\ - \\ - \\ + \\ + \\ - \\ + \end{array} \end{array} \begin{array}{c} B \\ \begin{array}{c} - \\ - \\ + \\ - \\ + \\ - \\ + \\ + \end{array} \end{array} \begin{array}{c} C \\ \begin{array}{c} - \\ - \\ - \\ + \\ - \\ + \\ + \\ + \end{array} \end{array} \begin{array}{c} (AB) \\ \begin{array}{c} + \\ - \\ + \\ + \\ - \\ + \\ - \\ - \end{array} \end{array} \begin{array}{c} (AC) \\ \begin{array}{c} + \\ - \\ - \\ + \\ - \\ - \\ + \\ - \end{array} \end{array} \begin{array}{c} (BC) \\ \begin{array}{c} + \\ + \\ - \\ - \\ - \\ - \\ + \\ - \end{array} \end{array} \begin{array}{c} (ABC) \\ \left.\begin{array}{c} - \\ + \\ + \\ + \\ - \\ - \\ - \\ + \end{array}\right] \end{array} \tag{14.73}$$

$$Y \equiv \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} y_{---} \\ y_{+--} \\ y_{-+-} \\ y_{--+} \\ y_{++-} \\ y_{+-+} \\ y_{-++} \\ y_{+++} \end{bmatrix} \tag{14.74}$$

Comments:

- ± 1 encoding could ensure the orthogonality of the design matrix X, i.e. $X'X = I$, which benefits the estimation and avoid collinearity.
- We can choose which effect terms to be included in X, simply by multiplying corresponding components columns, e.g. ABC interaction in X is

$$\begin{array}{c} \text{run} \\ y_{---} \\ y_{+--} \\ y_{-+-} \\ y_{--+} \\ y_{++-} \\ y_{+-+} \\ y_{-++} \\ y_{+++} \end{array} \begin{array}{c} (ABC) \\ \left[\begin{array}{c} - \\ + \\ + \\ + \\ - \\ - \\ - \\ + \end{array}\right. \end{array} = \begin{array}{c} A \\ \begin{bmatrix} - \\ + \\ - \\ - \\ + \\ + \\ - \\ + \end{bmatrix} \end{array} \cdot \begin{array}{c} B \\ \begin{bmatrix} - \\ - \\ + \\ - \\ + \\ - \\ + \\ + \end{bmatrix} \end{array} \cdot \begin{array}{c} C \\ \begin{bmatrix} - \\ - \\ - \\ + \\ + \\ + \\ + \\ - \end{bmatrix} \end{array} \tag{14.75}$$

i.e. only A, B, C are 'assigned', other interactions are induced by multiplication.

Section 14.4 Miscaellaneous Topics

14.4.1 Missing Values

Here we introduce a Minimizing-SSE method by

$$y_{\text{missing}} = \underset{\text{with } y \text{ fill in}}{\arg\min}\ \text{SSE} \tag{14.76}$$

e.g. two-factor model with y_{ij} missing, with current statistics denoted with prime $'$, i.e. obtained without missing values:

$$y_{ij}^{\text{fill in}} = \frac{ay'_{i\cdot} + by'_{\cdot j} - y'_{\cdot\cdot}}{(a-1)(b-1)} \tag{14.77}$$

14.4.2 D-Optimal Design

Motivation: What is a good design matrix X? An answer is to minimize the generalized variance of $\hat{\beta}$. Use equation 3.56 ~ page 82 we have

$$var(\hat{\beta}) = \sigma^2(X^TX)^{-1} \Rightarrow \arg\max |X'X| \tag{14.78}$$

Example: Use D-Optimal to explain balanced design in one-way factor ANOVA, use cell mean model:

$$Y_{ij} = \tau_i + \varepsilon_{ij}, \quad i = 1, 2, \ldots, r \quad j = 1, 2, \ldots, n_i \quad \sum_{i=1}^{a} n_j = n_T \tag{14.79}$$

with design matrix

$$n_{T\times r} = \begin{bmatrix} \mathbf{1}_{n_1} & 0 & \ldots & 0 \\ 0 & \mathbf{1}_{n_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{1}_{n_r} \end{bmatrix} \tag{14.80}$$

D-Optimal:

$$|X'X| = \left| \begin{bmatrix} n_1 & 0 & \ldots & 0 \\ 0 & n_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & n_r \end{bmatrix} \right| = \prod_{i=1}^{r} n_i \le \left(\frac{n_T}{r}\right)^r \tag{14.81}$$

equality taken at $n_1 = n_2 = \ldots = n_r = \dfrac{n_T}{r}$, i.e. balance design.

Note: In the case that we have some constraint on X, solving the D-Optimal problem could be complicated.

参考文献

[1] 清华大学统计学研究中心辅修课程课件与讲义. W. Deng, J. Wang, Z. Zhou, D. Li, T. Wang, S. Yu, P. Yang.

[2] Springer Series in Statistics (SSS). https://www.springer.com/series/692

[3] RStudio Cheatsheets https://www.rstudio.com/resources/cheatsheets

[4] 概率导论 (第二版 • 修订版). Dimitri P. Bertsekas, John N. Tsitsiklis. 人民邮电出版社.

[5] 北京大学《概率统计 A》课程讲义. 李东风. https://www.math.pku.edu.cn/teachers/lidf/course/probstathsy/probstathsy.pdf

[6] 数理统计（第二版）. 韦来生. 科学出版社.

[7] Statistical Inference(2nd Edition). George Casella, Roger L. Berger. Duxbury Press.

[8] Applied Linear Statistical Models(5th Edition). Michael H. Kutner, Christopher J. Nachtsheim, John Neter, William Li. McGraw-Hill Compaines, Inc.

[9] 线性模型引论. 王松桂 et. al. 科学出版社.

[10] Linear Models with R(2nd Edition). Julian J. Faraway. CRC Press.

[11] Generalized Linear Model Lecture Note. Germán Rodríguez. https://data.princeton.edu/wws509/notes

[12] 实用多元统计分析（第六版）. Richard A. Johnson, Dean W. Wichern. 清华大学出版社.

[13] R In Action: Data Analysis and Graphics with R(2nd Edition). Robert I. Kabacoff. Manning Publications Co.

[14] R Programming For Data Science. Roger D. Peng. Lean Publishing.

[15] Numerical Linear Algebra. I Lloyd N. Trefethen, David Bau Ill. Society for Industrial and Applied Mathematics

[16] Numerical Optimization(2nd Edition). J. Nocedal, Stephen J. Wright. Springer Science+Business Media, LLC.

[17] 北京大学《统计计算》课程讲义. 李东风. https://www.math.pku.edu.cn/teachers/lidf/docs/statcomp/html/_statcompbook/statcomp2ndv.pdf

[18] 生存分析与可靠性. 陈家鼎. 北京大学出版社.

[19] 机器学习. 周志华. 清华大学出版社.

[20] 机器学习公式详解. 谢文睿, 秦州. 人民邮电出版社.

[21] 神经网络与深度学习. 邱锡鹏. https://nndl.github.io/

[22] Time Series Analysis With Applications in R(2nd Edition). Jonathan D. Cryer, Kung-Sik Chan. Springer Science+Business Media, LLC.

[23] 北京大学《应用时间序列分析》课程讲义. 李东风. https://www.math.pku.edu.cn/teachers/lidf/course/atsa/atsanotes/html/_atsanotes/atsanotes.pdf

[24] Forecasting: Principles and Practice (2nd Edition). Hyndman, R.J., Athanasopoulos, G. https://otexts.com/fppcn

[25] Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction. Guido W. Imbens & Donald B. Rubin. Cambridge University Press.

[26] Causal Inference in Statistics - A Primer. Judea Pearl. Wiley.

[27] Random Processes for Engineers. Bruce Hajek. Cambridge University Press

[28] 应用随机过程. 林元烈. 清华大学出版社.

[29] Bayesian Data Analysis (3rd Edition). Andrew Gelman, John B. Carlin, Hal S. Stern etc. Chapman and Hall CRC Press.

[30] Design and Analysis of Experiments. Douglas C. Montgomery. Wiley.

[31] Design and Analysis of Experiments with R. J.Lawson. Chapman and Hall CRC Press

[32] Planning of Experiments. D.R. Cox. Wiley.

后记

Sometimes I asked myself: is it necessary, or even make sense, to put such a personal note online? After all the contents in this note contain tons of 'naïve' statements so it's actually far from being well-organized or worth used for reference. But later when I took part in the edition work of THU 2023 Feiyue Project, I realized that things don't have to be perfect to be shared. Or, to quote one of my friends:

我觉得能让后面的人意识到我们也曾经挣扎过就已经很有意义了。

So I decided to put this note online, hoping that it could be of some help to anyone like me, once confusing and struggling. You don't even have to read it, just knowing that someone else has been through the same thing as you do might be enough. And for the same reason, I've been try to set up some online project, or connect with statistics minor students like me, hoping to help them in some way. I hope that this note could be a small step towards this goal. And I hope that I could keep this spirit in the future.

If any of the contents in this note is helpful to you, please feel free to share it with others. And if you have any questions, suggestions, or just want to chat, please contact me via email: v1ncent19@outlook.com.

最后，祝愿看到这里的各位，未来前程锦绣。

Tuorui 'v1ncent19' Peng

2024 年 9 月 20 日

索引